The Journal of Open Source Software

spotter: Hardware-Accelerated Forward Models of Pixelated Stars

Lionel Garcia [1] **and Benjamin V. Rackham** [2]

1 Center for Computational Astrophysics, Flatiron Institute, New York, NY, USA **2** Department of Earth, Atmospheric and Planetary Science, Massachusetts Institute of Technology, MA, USA

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor: Axel Donath

Reviewers:

- @tedjohnson12
- @nenasedk

Submitted: 06 January 2025
Published: unpublished

Summary

The study of exoplanets predominantly relies on measurements of host stars' observables, such as flux and spectral time-series. However, stellar activity, including spots and faculae, has been increasingly recognized as a significant source of noise in signal analysis. For instance, photospheric active regions can mimic or hide exoplanetary signals (Rackham et al., 2023). These active regions also complicate exoplanet detection, whether through transit observations (Garcia et al., 2024) or radial velocity measurements (Collier Cameron et al., 2021). Addressing these challenges requires modeling the stellar surface and its temporal evolution. Although various methods exist for representing stellar surfaces, such as Luger et al. (2019), accurate inference of their properties necessitates forward models that are computationally efficient and compatible with widely used inference tools. Furthermore, the growing volume and resolution of datasets, combined with advancements in instrumentation, emphasize the need for scalable and high-performance models.

Statement of Need

spotter is a Python package designed for fast forward modeling of non-uniform stellar surfaces. Built with JAX (Bradbury et al., 2018), it leverages hardware acceleration to enable efficient simulations and inferences through a Pythonic interface. *spotter* can model flux and spectral time-series from pixelated stellar surfaces and define surface Gaussian processes conditioned on parameters such as spot size and contrast, inspired by Luger et al. (2021). Its minimal and flexible design makes it a versatile tool for stellar and exoplanet science applications, including ensemble analyses of stellar light curves, akin to approaches similar to Luger et al. (2021) and Morris (2020).

A key advantage of *spotter* is its native implementation in JAX, which provides seamless integration with probabilistic programming tools in the JAX ecosystem and enables hardware acceleration on GPUs and TPUs. Unlike Luger et al. (2021), *spotter* supports modeling of surfaces at much higher spatial resolutions. In contrast to Morris (2020), *spotter* allows for the representation of arbitrarily shaped active regions, offering greater flexibility in simulating complex stellar surface features.

Principle

If $\mathbf{y}$ represents a vector of pixel values corresponding to the surface intensity of a star, *spotter* models the observable f (e.g., flux) as

$$f = \mathbf{X}\mathbf{y}$$

where, at any given time, $\mathbf{X}$ is constructed by accounting for:

- The pixels belonging to the visible hemisphere of the stellar surface,
- The projected area of each pixel, and
- The limb darkening intensity of each pixel.

spotter's documentation demonstrates how this linear model is applied to simulate both flux and spectral time-series for rotating stellar surfaces. To facilitate this computation, the HEALPix subdivision scheme (Górski et al., 2005) is employed to decompose the stellar surface into pixels.

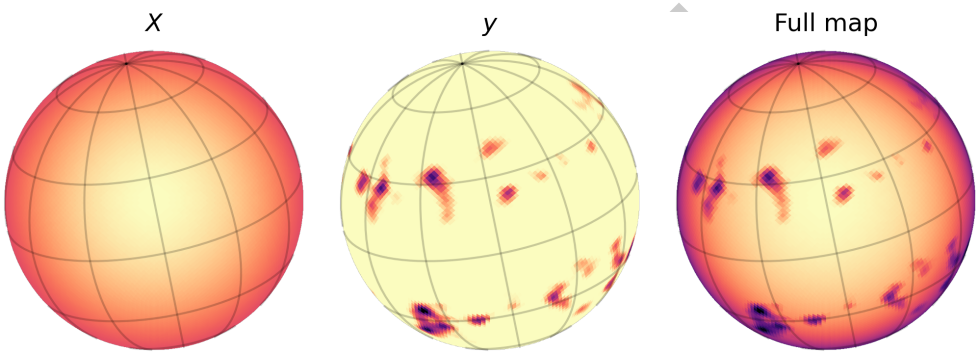

Figure 1: Example of a limb-darkened star whose surface has been drawn from a Gaussian Process representing Sun-like active latitudes. Left: design matrix of the map in orthographic projection. Center: Surface map of the star in the same projection. Right: Ortographic representation of the complete map.

By adopting this formalism, *spotter* implements its model using simple linear algebra operations in JAX. It supports execution on hardware accelerators such as GPUs and TPUs, enabling fast and scalable simulations and inferences. This capability makes *spotter* well-suited for processing large datasets and high-resolution observations, whether spatial or temporal.

Performance

Speed

Figure 2 shows the performance of *spotter* in evaluating the flux forward model of a rotation surface with an increasing number of pixels, as well as for an increasing number of data points. This figure shows that running *spotter* on GPU provides two orders of magnitude speedups compared to running it on CPU.

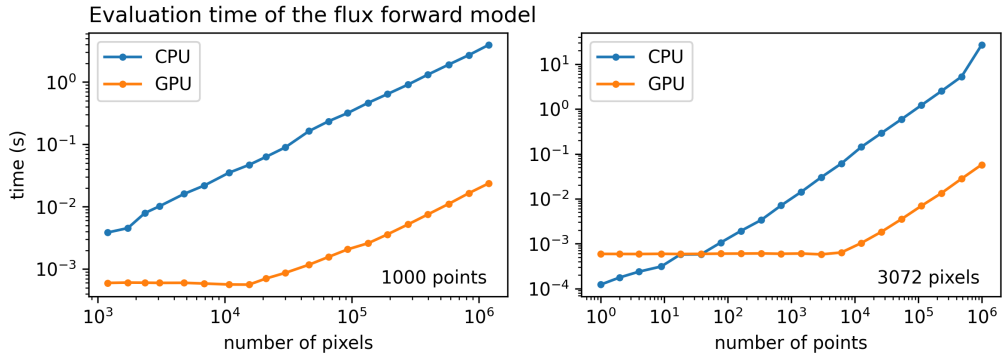

Figure 2: Flux forward model evaluation time on a single CPU and GPU. *Left*: evaluation time of a 1000-points rotation light curve versus the number of pixels used to represent the surface. *Right*: evaluation time of a 3072-pixels surface depending on the number of points in the time series.

Figure 3 shows the performance of *spotter* in evaluating the spectrum forward model of a star depending on the number of wavelength bins. In this case, using a GPU leads to an order of magnitude speedup compared to a CPU.

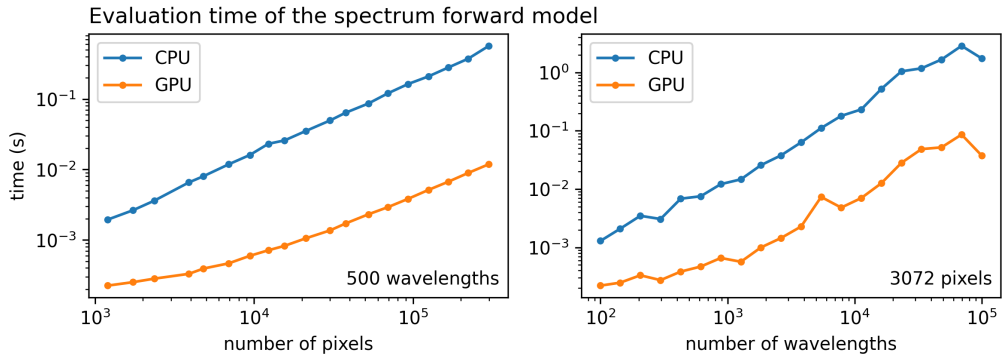

Figure 3: Spectrum forward model evaluation time on a single CPU and GPU. *Left*: evaluation time of the spectrum of a surface with an increasing number of pixels (for 500 wavelength bins). *Right*: evaluation time of the spectrum of a 3072-pixels surface versus the number of wavelength bins.

On CPU, our benchmark is done on the single core of an Apple M2 max chip, while on GPU we use the single core of an NVIDIA Tesla V100 processor.

Precision

We validate the precision of *spotter*'s flux forward model against models evaluated with *starry* (Luger et al., 2019). We make these precision benchmarks part of the unit tests of *spotter*.

Acknowledgements

spotter makes use of the following dependencies: *numpy* (Harris et al., 2020), *healpy* (Zonca et al., 2019), *jax* (Bradbury et al., 2018), *equinox* (Kidger & Garcia, 2021) and *tinygp*.

References

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/jax-ml/jax

Collier Cameron, A., Ford, E. B., Shahaf, S., Aigrain, S., Dumusque, X., Haywood, R. D., Mortier, A., Phillips, D. F., Buchhave, L., Cecconi, M., Cegla, H., Cosentino, R., Crétignier, M., Ghedina, A., González, M., Latham, D. W., Lodi, M., López-Morales, M., Micela, G., … Watson, C. (2021). Separating planetary reflex Doppler shifts from stellar variability in the wavelength domain. *505*(2), 1699–1717. https://doi.org/10.1093/mnras/stab1323

Garcia, L. J., Foreman-Mackey, D., Murray, C. A., Aigrain, S., Feliz, D. L., & Pozuelos, F. J. (2024). nuance: Efficient Detection of Planets Transiting Active Stars. *167*(6), 284. https://doi.org/10.3847/1538-3881/ad3cd6

Górski, K. M., Hivon, E., Banday, A. J., Wandelt, B. D., Hansen, F. K., Reinecke, M., & Bartelmann, M. (2005). HEALPix: A Framework for High-Resolution Discretization and Fast Analysis of Data Distributed on the Sphere. *622*(2), 759–771. https://doi.org/10.1086/427976

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

Kidger, P., & Garcia, C. (2021). Equinox: Neural networks in JAX via callable PyTrees and filtered transformations. *Differentiable Programming Workshop at Neural Information Processing Systems 2021*.

Luger, R., Agol, E., Foreman-Mackey, D., Fleming, D. P., Lustig-Yaeger, J., & Deitrick, R. (2019). starry: Analytic Occultation Light Curves. *157*(2), 64. https://doi.org/10.3847/1538-3881/aae8e5

Luger, R., Foreman-Mackey, D., & Hedges, C. (2021). Mapping Stellar Surfaces. II. An Interpretable Gaussian Process Model for Light Curves. *162*(3), 124. https://doi.org/10.3847/1538-3881/abfdb9

Morris, B. (2020). fleck: Fast approximate light curves for starspot rotational modulation. *The Journal of Open Source Software*, *5*(47), 2103. https://doi.org/10.21105/joss.02103

Rackham, B. V., Espinoza, N., Berdyugina, S. V., Korhonen, H., MacDonald, R. J., Montet, B. T., Morris, B. M., Oshagh, M., Shapiro, A. I., Unruh, Y. C., Quintana, E. V., Zellem, R. T., Apai, D., Barclay, T., Barstow, J. K., Bruno, G., Carone, L., Casewell, S. L., Cegla, H. M., … Valenti, J. A. (2023). The effect of stellar contamination on low-resolution transmission spectroscopy: needs identified by NASA's Exoplanet Exploration Program Study Analysis Group 21. *RAS Techniques and Instruments*, *2*(1), 148–206. https://doi.org/10.1093/rasti/rzad009

Zonca, A., Singer, L., Lenz, D., Reinecke, M., Rosset, C., Hivon, E., & Gorski, K. (2019). Healpy: Equal area pixelization and spherical harmonics transforms for data on the sphere in python. *Journal of Open Source Software*, *4*(35), 1298. https://doi.org/10.21105/joss.01298