Erebus: An exoplanet secondary-eclipse photometry data reduction pipeline

Nicholas Connors [1]

1 Department of Physics and Trottier Institute for Research on Exoplanets, Universite de Montreal, Montreal, QC, Canada

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor: Open Journals

Reviewers:

- @openjournals

Submitted: 01 January 1970
Published: unpublished

Summary

The James Webb Space Telescope (JWST) is uniquely suited to probing the mid-infra red wavelengths of light, enabling exoplanet research to be performed that was previously impossible. By observing rocky exoplanet photometry at 15 microns using its Mid Infrared Instrument (MIRI) astrophysicists can infer the presense or absense of an atmosphere by comparing the dip in brightness as an exoplanet passes behind its star (the secondary eclipse) against theoretical values determined using surface and atmospheric models.

`Erebus` is a pipeline for analyzing time-series secondary-eclipse photometry data from MIRI with a focus on detrending pixel-level detector systematic effects in order to determine the magnitude of these lightcurve dips (eclipse depths). `Erebus` performs aperture photometry to retrieve the brightness of the star system over time (the lightcurve) and uses a Markov Chain Monte Carlo to fit for the eclipse depth while detrending against inter-pixel systematics found using frame-normalized Principal Component Analysis (FN-PCA). By performing PCA on the normalized time-series of image frames, all astrophysical signals (seen as uniform increases/decreases in brightness across the entire detector) are removed leaving behind only inter-pixel systematic effects to be detrended against. `Erebus` is configured using simple yaml files describing the physical parameters of the planets and the specifications of each run, and outputs an h5 file containing all relevant information as well as creating informative plots. Alternatively, `Erebus` provides utility functions for analyzing MIRI detector systematics using FN-PCA which can then be used as a part of another data reduction pipeline. An example plot from using `Erebus` to analyze an eclipse of the planet TRAPPIST-1b can be seen in figure 1.

Statement of need

The existing literature on detrending and analyzing secondary-eclipse MIRI photometry varies in its methods and source code availability. `Erebus` provides a uniform method for analyzing this data that is specifically tailored around the MIRI instrument, while being easy to use and open source to further enable collaboration as more JWST 15 micron data is released.

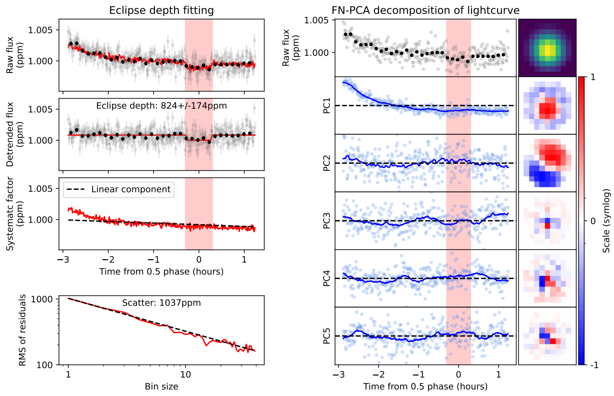

Figure 1: Example of frame normalized principal component analysis detrending fit for the planet TRAPPIST-1b using `Erebus`. The panels on the left show the raw and detrended lightcurves and the systematic model, as well as a goodness of fit metric comparing RMS of siduals to bin size. The panels on the right show the decomposition of the time-series image data into principal components used to detrend the lightcurve. This plot is automatically generated by the software.

Documentation

Erebus documentation can be found at https://nicholasconnors.github.io/erebus/latest/. It includes information on the API, the settings available in all yaml configuration files, and example Jupyter notebooks for how to use the software.

Uses in scientific literature

Erebus was used to perform a uniform reanalysis of existing MIRI 15 micron rocky exoplanet photometry in Connors et al (in prep).

Similar tools

Open source tools similar to `Erebus` include:

- `jwst` (Bushouse et al., 2022) and `Eureka!` (Bell et al., 2022): General-purpose end-to-end pipelines for reduction of JWST data including a broader range of instruments.
- `TransitSpectroscopy` (Nestor Espinoza, 2022): A Python package with routines for reduction of both exoplanet transit and secondary eclipse observations, with support for spectroscopy and photometry.
- `juliet` (Néstor Espinoza et al., 2019): A tool for fitting exoplanet transit photometry. Also supports fitting radial velocity data.

Unlike `jwst` or `Eureka!`, `Erebus` does not provide an end-to-end reduction of JWST MIRI data, and must be ran on data that was pre-processed with either software (up to Stage 2). `Erebus` differs from similar softwares in how it performs detrending of the exoplanet lightcurve using frame-normalized principal component analysis, and its specific focus on exoplanet secondary eclipse observations.

Acknowledgements

`Erebus` was built using `astropy` Astropy Collaboration et al. (2022), `emcee` (Foreman-Mackey et al., 2013), `NumPy` (Harris et al., 2020), `batman` (Kreidberg, 2015), `sklearn` (Pedregosa et al., 2011), `matplotlib` (Hunter, 2007), `h5py` (Collette, 2013), `SciPy` (Virtanen et al., 2020), `corner` (Foreman-Mackey, 2016), `uncertainties`, and `pydantic`.

NC acknowledges financial support from the University of Montreal. This work was made with the support of the Institut Trottier de Recherche sur les Exoplanetes (iREx).

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. *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. *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. *558*, A33. https://doi.org/10.1051/0004-6361/201322068

Bell, T. J., Ahrer, E.-M., Brande, J., Carter, A. L., Feinstein, A. D., Guzman Caloca, G., Mansfield, M., Zieba, S., Piaulet, C., Benneke, B., Filippazzo, J., May, E. M., Roy, P.-A., Kreidberg, L., & Stevenson, K. B. (2022). Eureka!: An end-to-end pipeline for JWST time-series observations. *Journal of Open Source Software*, *7*(79), 4503. https://doi.org/10.21105/joss.04503

Bushouse, H., Eisenhamer, J., Dencheva, N., Davies, J., Greenfield, P., Morrison, J., Hodge, P., Simon, B., Grumm, D., Droettboom, M., Slavich, E., Sosey, M., Pauly, T., Miller, T., Jedrzejewski, R., Hack, W., Davis, D., Crawford, S., Law, D., … Williams, T. (2022). *JWST calibration pipeline* (Version 1.8.2). Zenodo. https://doi.org/10.5281/zenodo.7229890

Collette, A. (2013). *Python and HDF5*. O'Reilly.

Espinoza, Nestor. (2022). *TransitSpectroscopy* (Version 0.3.11). Zenodo. https://doi.org/10.5281/zenodo.6960924

Espinoza, Néstor, Kossakowski, D., & Brahm, R. (2019). juliet: a versatile modelling tool for transiting and non-transiting exoplanetary systems. *490*(2), 2262–2283. https://doi.org/10.1093/mnras/stz2688

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

Foreman-Mackey, D., Hogg, D. W., Lang, D., & Goodman, J. (2013). <Tt>emcee</tt>: The MCMC hammer. *Publications of the Astronomical Society of the Pacific*, *125*(925), 306–312. https://doi.org/10.1086/670067

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

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

Kreidberg, L. (2015). batman: BAsic Transit Model cAlculatioN in Python. *127*(957), 1161. https://doi.org/10.1086/683602

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

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