The Journal of Open Source Software

TidalPy: Software Suite for Solving Problems in Tidal Dynamics

Joe P. Renaud [1,2¶]

1 University of Maryland, College Park, Maryland USA **2** NASA Goddard Space Flight Center, Greenbelt, Maryland USA ¶ Corresponding author

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor: Open Journals

Reviewers:

- @openjournals

Submitted: 01 January 1970
Published: unpublished

Summary

`TidalPy` is an open-source Python package designed to model tidal heating, interior response, and long-term spin–orbit evolution of planets and moons across the Solar System and beyond. It combines Love number computation, advanced rheological models, and semi-analytical orbital dynamics into a single accessible framework with performance-critical components in C++ and Cython. The software enables forward modeling of interior structure, dissipation mechanisms, and tidal feedback on rotation and orbits, supporting applications ranging from Solar System geophysics to exoplanet characterization. TidalPy interfaces with the broader Python scientific ecosystem, offers extensive documentation and examples, and has already been applied in multiple planetary science studies.

Statement of need

`TidalPy` provides a flexible, accessible, and performant toolkit for solving problems in tides and tidal dynamics. The same tides that cause Earth's oceans to rise twice each day can churn the interiors of other planets and moons to the point that significant fractions of their bulk can melt, greatly altering their long-term thermal evolution. The energy that drives this heat originates in the orbits and rotations of the world and its host, which are also altered on geologic timescales. TidalPy provides functions and frameworks to apply the latest tidal modeling theories and methods to a wide variety of Solar System and exoplanetary worlds. Written for the Python ecosystem, TidalPy can easily interface with other popular packages used in planetary science and astronomy. This enables fast production of advanced simulations which can be used directly or as a benchmark against other models and tools.

Overview

TidalPy is written primarily in Python, with performance-critical components implemented in C++ and Cython (Behnel et al., 2011). Its API is designed to be intuitive and consistent with modern conventions, enabling both early career and experienced researchers to quickly learn its syntax and incorporate it in their scientific projects. TidalPy complements a robust community of other packages that perform similar or parallel calculations (Barnes et al., 2020; Lu et al., 2023; Martens et al., 2019; Melini et al., 2022; Neveu, 2025; Qin et al., 2014; Rovira-Navarro et al., 2024; Saikiran Tharimena, 2024) and expands on this prior work in three major areas described in the following sections.

TidalPy has been vetted and used in investigations of tides on Earth (Vidal & Cébron, 2025), in our Solar System (Cascioli et al., 2023; Goossens et al., 2024; Wagner & James, 2025), and beyond (Fauchez et al., 2025; Joe P. Renaud et al., 2021; Joe P. Renaud & Henning, 2018). Documentation and Jupyter Notebook (Kluyver et al., 2016) demonstrations are available on the GitHub repository, these are continuously added to and updated as TidalPy evolves. Future releases will focus on increasing performance, improving usability, and incorporating more physics. Get started using TidalPy by visiting https://tidalpy.info.

Love Number Solver (RadialSolver Module)

Learn more about TidalPy's RadialSolver Module here.

Love Numbers quantify a planet or moon's ability to respond to tidal or loading forces (Love, 1911; Shida, 1912). They are dynamic and depend on many physical factors such as a world's thermal state, physical structures (e.g., a presence of a solid or liquid core), past stress events, and orbit/spin state. These numbers can be measured, albeit with difficulty particularly if no spacecraft flyby or orbiter is available. Therefore, it is useful to perform forward modeling utilizing our best estimates of a world's structure and composition to provide a range of values to constrain a world's response efficiency.

TidalPy provides a Love number solver that uses information about a world's interior structure and thermal state to find these values. A user can turn on or off a variety of assumptions to determine their impact. This solver can be used in other modules to, for example, determine the effect of long-term heating on a world's tidal dissipation or in 3rd party packages such as Foreman-Mackey et al. (2013)'s Markov Chain Monte-Carlo code to predict a statistically likely interior.

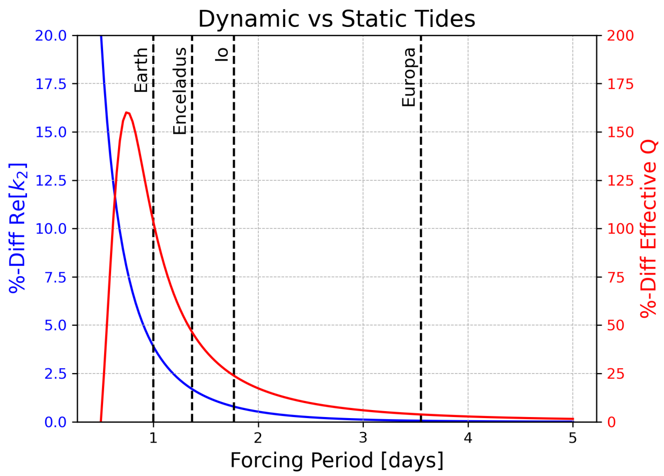

Figure 1: The percent difference between Love number using the dynamic and static assumption is shown for a icy moon with a significant ocean layer as a function of tidal forcing period. Several reference periods are shown to give a sense of when dynamic tides may be important to consider.

TidalPy's solver uses a shooting method (Takeuchi & Saito, 1972) to find tidal and loading Love numbers. This approach is advantageous as it enables more advanced physics, providing a more accurate description of a world. Specifically, TidalPy's solver allows for: liquid layers and oceans, dynamic tides (See Figure 1), and bulk compressibility (See Figure 2). This additional physics has been shown to be important for certain worlds during certain epochs. TidalPy's Love number solver has been benchmarked against others tools that provide some of the same functionality including `ALMA3` (Melini et al., 2022; Saikiran Tharimena, 2024) and `LoadDef` (Martens et al., 2019). Other tools exist that, unlike the current version of TidalPy, can calculate multidimensional Love numbers (Berne et al., 2023; Qin et al., 2014; Rovira-Navarro et al., 2024).

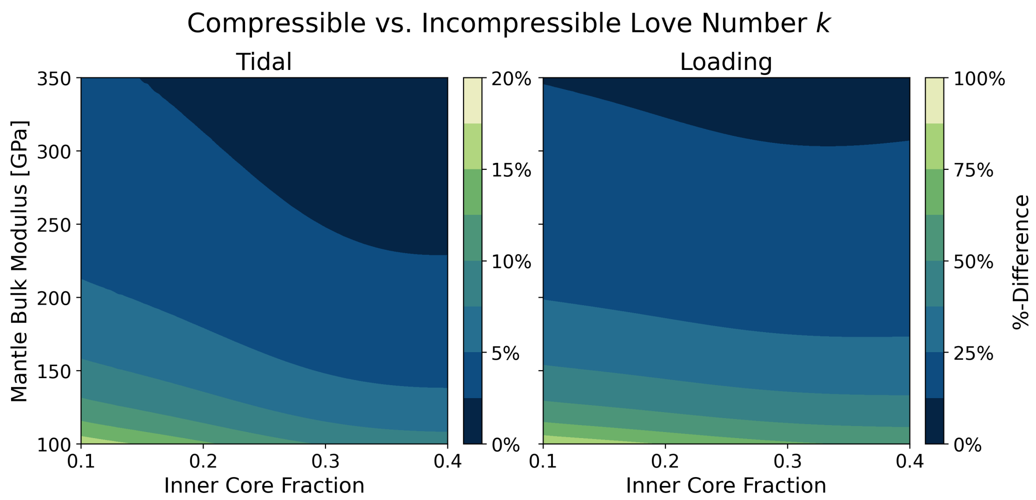

Figure 2: Bulk dissipation can lead to significant differences in both the Tidal (left) and Loading (right) Love numbers in this simplified Venus model.

Advanced Rheological Modeling (Rheology Module)

Learn more about TidalPy's Rheology Module here.

The calculation of tidal Love numbers requires knowing the viscoelastic state of a planet. This can be described through the shear and bulk modulus as well as the shear and bulk viscosity. The former describe how sound waves travel through a planet's bulk; the latter describe how material flows on long timescales. Linking these properties to tides requires making assumptions about the dominant mechanism driving dissipation in the rocks and ices (Bagheri et al., 2022; Joe P. Renaud & Henning, 2018). For example, microscopic grains of ice will tend to move more freely than larger solid crystalline chunks. Likewise, rock that has experienced significant fracturing or is porous tends to have more opportunity to create frictional heat then it would otherwise. The choice of **Rheology** determines which dissipation mechanism is dominant within a world.

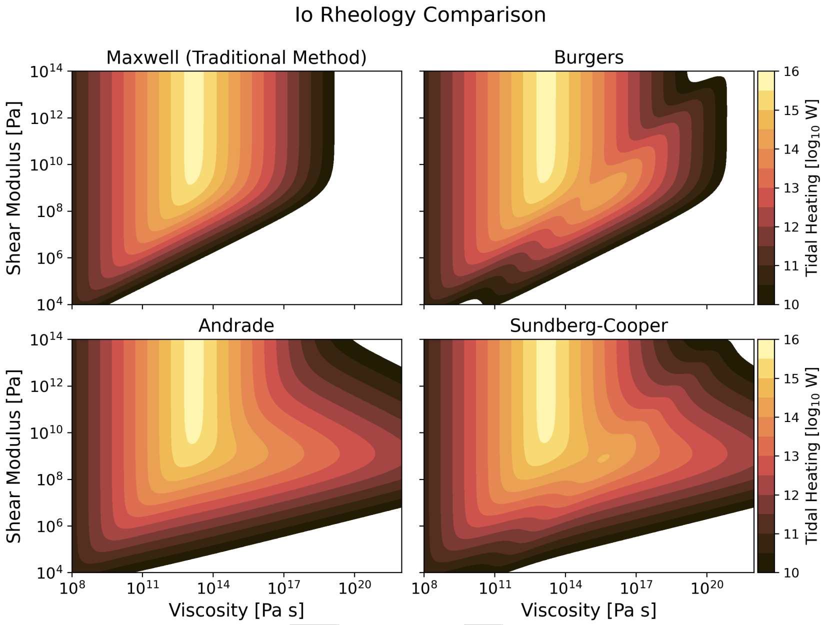

Figure 3: Tidal heating is shown for four different rheology models for a model of Jupiter's moon Io. Heating in certain viscoelastic phase spaces can be orders of magnitude different depending on your choice in rheology.

TidalPy provides several different rheological models in its Rheology Module (See a subset in Figure 3). Most rheologies have empirical parameters which are relatively unknown for rocks and ices at planetary temperatures and pressures. TidalPy suggests typical values used in the literature but allows you to vary them. These efficient rheological functions can be used with other TidalPy methods, like the Love number solver, or in your own scripts alongside other tools.

Spin and Orbital Evolution (Dynamics Module)

Learn more about TidalPy's Dynamics Module here.

The energy released as heat in a tidally active world originates in its orbit or in the rotation of it or its tidal host (which could be Jupiter in the case of Io, or a star in the case of a short-period exoplanet). Energy can be exchanged between adjacent planets and moons, further increasing the complexity of the problem (*e.g.*, Hussmann & Spohn, 2004). Different orbit and spin configurations will have a variety of forcing frequencies which can act like a tuning fork, dramatically increasing dissipation in narrow frequency bands (Bagheri et al., 2022; Joe P. Renaud & Henning, 2018). Likewise, troughs in dissipation can slow or stop the evolution of a world for millions of years. These "Spin-Orbit Resonances" are what drive Mercury's 3:2 Spin to Orbit ratio and what leads to systems like Pluto-Charon which are in a "dual-synchronous" configuration, the ultimate end state of tidal evolution.

The Journal of Open Source Software

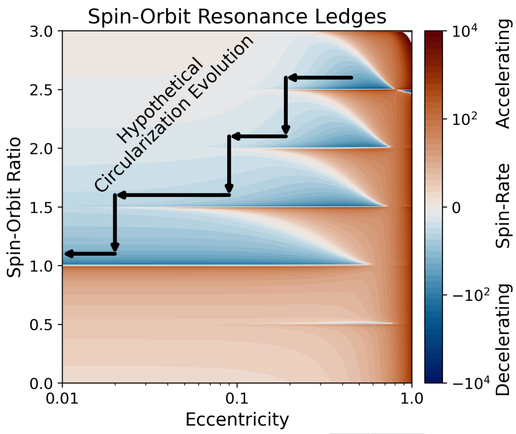

Figure 4: Spin-Orbit Resonance "ledges" calculated with TidalPy. A planet can become trapped on a ledge (stuck at a certain spin rate) for millions of years depending on its interior structure and thermal state.

To calculate these effects, TidalPy uses a semi-analytical Fourier decomposition model that has a long history of use in the field (*e.g.*, Kaula, 1964) and recently expanded by (Boué & Efroimsky, 2019). In this framework, TidalPy can track dissipation within both the target planet and host (dual body dissipation) and can be integrated with semi-analytical models to capture multi-body systems, like the Laplace Resonance found in the Galilean moons (Hussmann & Spohn, 2004). The rotation rate of all targets is tracked such that spin-orbit resonance capture potential can be determined (See Figure 4). TidalPy can also generate 3-dimensional stress and heating maps which can be used to determine locations where tidal heating is maximum (See Figure 5).

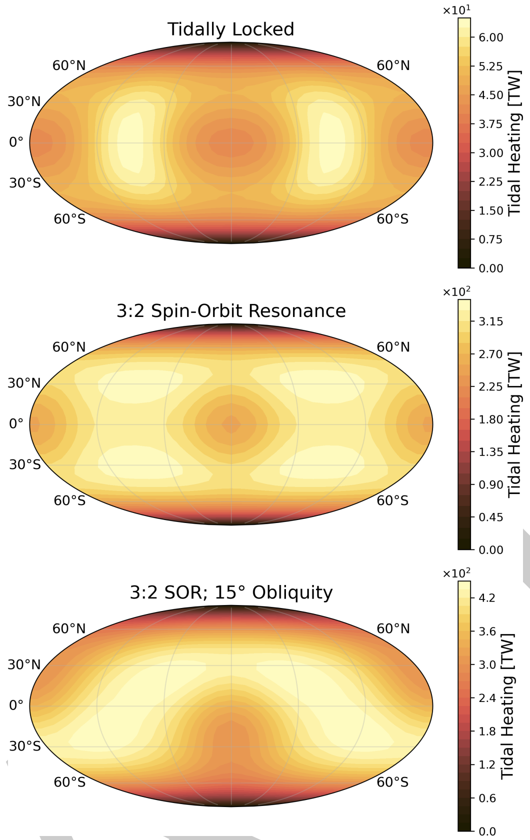

Figure 5: Tidal heating for a short-period exoplanet in three different spin configurations. Non-synchronous spin and a non-zero obliquity can lead to large differences in the magnitude and location of maximum heating.

Availability

TidalPy's source code is available and kept up to date on its GitHub Repository. All versions are released on GitHub as well as on PyPI and Conda-Forge. Major versions are also released with dedicated DOIs on TidalPy's Zenodo page. Anyone is welcome to open pull requests, create forks, or issue bug reports, suggestions, and questions. The latter can be made on the GitHub issue tracker. TidalPy can also be found on NASA's Exoplanet Modeling and Analysis Center (Joe P. Renaud et al., 2022).

All figures made for this paper were done using TidalPy and other packages listed in the acknowledgements. The scripts used to make these figures can be found on TidalPy's GitHub repository.

License

TidalPy is licensed under the Apache 2.0 License (Apache-2.0). This allows any user to share and reproduce TidalPy in whole or in part, as long as attribution is made back to the original repository and cites this paper. Full details about the license can be found in the repository's license and notice files.

Acknowledgements

TidalPy benefited greatly from conversations, code contributions, and testing performed by many in the community. We would like to specifically thank Wade G. Henning, Michael Efroimsky, Michaela Walterová, Sander Goossens, Marc Neveu, Nick Wagner, and Gael Cascioli. The development of TidalPy was supported by NASA Sellers' Exoplanet Environments Collaboration and Planetary Geodesy ISFMs. J. Renaud was additionally supported during its development by the CRESST-II cooperative agreement (NASA award 80GSFC24M0006). TidalPy makes extensive use of the following software: CyRK (J. P. Renaud, 2022), BurnMan (Myhill et al., 2023), NumPy (Harris et al., 2020), SciPy (Virtanen et al., 2020), Numba (Lam et al., 2015), Cython (Behnel et al., 2011), Matplotlib (Hunter, 2007), SymPy (Meurer et al., 2017), Jupyter (Kluyver et al., 2016), and cmcrameri (Crameri, 2023; Rollo, 2020).

References

Bagheri, A., Efroimsky, M., Castillo-Rogez, J., Goossens, S., Plesa, A.-C., Rambaux, N., Rhoden, A., Walterová, M., Khan, A., & Giardini, D. (2022). Tidal insights into rocky and icy bodies: An introduction and overview. *Advances in Geophysics*, *63*, 231–320. https://doi.org/10.1016/bs.agph.2022.07.004

Barnes, R., Luger, R., Deitrick, R., Driscoll, P., Quinn, T. R., Fleming, D. P., Smotherman, H., McDonald, D. V., Wilhelm, C., Garcia, R., Barth, P., Guyer, B., Meadows, V. S., Bitz, C. M., Gupta, P., Domagal-Goldman, S. D., & Armstrong, J. (2020). VPLanet: The Virtual Planet Simulator. *132*(1008), 024502. https://doi.org/10.1088/1538-3873/ab3ce8

Behnel, S., Bradshaw, R., Citro, C., Dalcin, L., Seljebotn, D. S., & Smith, K. (2011). Cython: The Best of Both Worlds. *Computing in Science & Engineering*, *13*(2), 31–39. https://doi.org/10.1109/MCSE.2010.118

Berne, A., Simons, M., Keane, J. T., & Park, R. S. (2023). Using Tidally-Driven Elastic Strains to Infer Regional Variations in Crustal Thickness at Enceladus. *Geophysical Research Letters*, *50*(22), e2023GL106656. https://doi.org/10.1029/2023GL106656

Boué, G., & Efroimsky, M. (2019). Tidal evolution of the Keplerian elements. *Celestial Mechanics and Dynamical Astronomy*, *131*(7), 30. https://doi.org/10.1007/s10569-019-9908-2

Cascioli, G., Renaud, J. P., Mazarico, E., Durante, D., Iess, L., Goossens, S., & Smrekar, S. (2023). Constraining the Venus Interior Structure with Future VERITAS Measurements of the Gravitational Atmospheric Loading. *4*(4), 65. https://doi.org/10.3847/PSJ/acc73c

Crameri, F. (2023). Scientific colour maps. *Zenodo*. https://doi.org/10.5281/zenodo.1243862

Fauchez, T., Saxena, P., Wolf, E., Chen, H., Renaud, J., & Gonzalez Quiles, J. (2025). Potential Effects of Exovolcanism on Transmission Spectroscopy of Earth-size Exoplanets. *American Astronomical Society Meeting Abstracts #245*, *245*, 314.04.

Foreman-Mackey, D., Conley, A., Meierjurgen Farr, W., Hogg, D. W., Lang, D., Marshall, P., Price-Whelan, A., Sanders, J., & Zuntz, J. (2013). *emcee: The MCMC Hammer*. Astrophysics Source Code Library, record ascl:1303.002.

Goossens, S., van Noort, B., Mateo, A., Mazarico, E., & van der Wal, W. (2024). A low-density ocean inside Titan inferred from Cassini data. *Nature Astronomy*, *8*, 846–855. https://doi.org/10.1038/s41550-024-02253-4

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

Hussmann, H., & Spohn, T. (2004). Thermal-orbital evolution of Io and Europa. *Icarus*, *171*(2), 391–410. https://doi.org/10.1016/j.icarus.2004.05.020

Kaula, W. M. (1964). Tidal dissipation by solid friction and the resulting orbital evolution. *Reviews of Geophysics*, *2*(4), 661. https://doi.org/10.1029/RG002i004p00661

Kluyver, T., Ragan-Kelley, B., Pérez, F., Granger, B., Bussonnier, M., Frederic, J., Kelley, K., Hamrick, J., Grout, J., Corlay, S., Ivanov, P., Avila, D., Abdalla, S., Willing, C., & team, J. development. (2016). Jupyter notebooks - a publishing format for reproducible computational workflows. In F. Loizides & B. Scmidt (Eds.), *Positioning and power in academic publishing: Players, agents and agendas* (pp. 87–90). IOS Press. https://eprints.soton.ac.uk/403913/

Lam, S. K., Pitrou, A., & Seibert, S. (2015). Numba: A LLVM-based python JIT compiler. *Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC*. https://doi.org/10.1145/2833157.2833162

Love, A. E. H. (1911). *Some Problems of Geodynamics*.

Lu, T., Rein, H., Tamayo, D., Hadden, S., Mardling, R., Millholland, S. C., & Laughlin, G. (2023). Self-consistent Spin, Tidal, and Dynamical Equations of Motion in the REBOUNDx Framework. *948*(1), 41. https://doi.org/10.3847/1538-4357/acc06d

Martens, H. R., Rivera, L., & Simons, M. (2019). LoadDef: A Python-Based Toolkit to Model Elastic Deformation Caused by Surface Mass Loading on Spherically Symmetric Bodies. *Earth and Space Science*, *6*(2), 311–323. https://doi.org/10.1029/2018EA000462

Melini, D., Saliby, C., & Spada, G. (2022). On computing viscoelastic Love numbers for general planetary models: The ALMA3 code. *Geophysical Journal International*, *231*(3), 1502–1517. https://doi.org/10.1093/gji/ggac263

Meurer, A., Smith, C. P., Paprocki, M., Čertík, O., Kirpichev, S. B., Rocklin, M., Kumar, A., Ivanov, S., Moore, J. K., Singh, S., Rathnayake, T., Vig, S., Granger, B. E., Muller, R. P., Bonazzi, F., Gupta, H., Vats, S., Johansson, F., Pedregosa, F., … Scopatz, A. (2017). SymPy: Symbolic computing in python. *PeerJ Computer Science*, *3*, e103. https://doi.org/10.7717/peerj-cs.103

Myhill, R., Cottaar, S., Heister, T., Rose, I., Unterborn, C., Dannberg, J., & Gassmoeller, R. (2023). BurnMan – a python toolkit for planetary geophysics, geochemistry and thermodynamics. *Journal of Open Source Software*, *8*(87), 5389. https://doi.org/10.21105/joss.05389

Neveu, M. (2025). IcyDwarf: A coupled physical-chemical simulator of icy dwarf planets or moons. In *GitHub repository*. GitHub. https://github.com/MarcNeveu/IcyDwarf

Qin, C., Zhong, S., & Wahr, J. (2014). A perturbation method and its application: Elastic tidal response of a laterally heterogeneous planet. *Geophysical Journal International*, *199*(2), 631–647. https://doi.org/10.1093/gji/ggu279

Renaud, J. P. (2022). CyRK - ODE Integrator Implemented in Cython and Numba. *Zenodo*. https://doi.org/10.5281/zenodo.7093266

Renaud, Joe P., & Henning, W. G. (2018). Increased Tidal Dissipation Using Advanced Rheological Models: Implications for Io and Tidally Active Exoplanets. *The Astrophysical Journal*, *857*(2), 98. https://doi.org/10.3847/1538-4357/aab784

Renaud, Joe P., Henning, W. G., Saxena, P., Neveu, M., Bagheri, A., Mandell, A., & Hurford, T. (2021). Tidal Dissipation in Dual-body, Highly Eccentric, and Nonsynchronously Rotating Systems: Applications to Pluto-Charon and the Exoplanet TRAPPIST-1e. *2*(1),

4. https://doi.org/10.3847/PSJ/abc0f3

Renaud, Joe P., Lopez, E., Brande, J., Cruz-Arce, C. E., Kelahan, C., Susemiehl, N., Cristy, D., Hostetter, C., Moore, M. D., Patel, A., & Mandell, A. M. (2022). The Exoplanet Modeling and Analysis Center at NASA Goddard. *Research Notes of the AAS*, *6*(9), 185. https://doi.org/10.3847/2515-5172/ac9060

Rollo, C. (2020). Cmcrameri: Python wrapper around fabio crameri's perceptually uniform colormaps. In *GitHub repository*. GitHub. https://github.com/callumrollo/cmcrameri

Rovira-Navarro, M., Matsuyama, I., & Berne, A. (2024). A Spectral Method to Compute the Tides of Laterally Heterogeneous Bodies. *The Planetary Science Journal*, *5*(5), 129. https://doi.org/10.3847/PSJ/ad381f

Saikiran Tharimena, D. M., Marshall J. Styczinski. (2024). PyALMA3: A pythonized version of ALMA 3. In *GitHub repository*. GitHub. https://github.com/drsaikirant88/PyALMA3

Shida, T. (1912). *On the elasticity of the Earth and the Earth's crust.*

Takeuchi, H., & Saito, M. (1972). Seismic Surface Waves. In *Methods in Computational Physics: Advances in Research and Applications* (Vol. 11, pp. 217–295). https://doi.org/10.1016/B978-0-12-460811-5.50010-6

Vidal, J., & Cébron, D. (2025). Did lunar tides sustain the early Earth's geodynamo? *arXiv e-Prints*, arXiv:2506.19039. https://doi.org/10.48550/arXiv.2506.19039

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

Wagner, N. L., & James, P. B. (2025). New Geophysical Constraints for Intrusive Magmatism at Large Martian Volcanoes: Implications for Crustal Thickness and Volatile Outgassing. *Journal of Geophysical Research (Planets)*, *130*(8), e2025JE008959. https://doi.org/10.1029/2025JE008959