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Published: unpublished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 for modeling tidal deformation, internal dissipation, and rotational–orbital evolution in planetary systems. The software combines a user-friendly Python interface with performance-critical routines implemented in C++ and Cython. Key capabilities include dynamic Love number computation, advanced rheological models, and coupled spin–orbit evolution. TidalPy addresses a gap in existing tools by providing these capabilities within a single accessible framework that easily interfaces with other popular Python packages used in the field. It has been extensively tested in a variety of studies examining tidal dissipation in Solar System planets and moons, as well as exoplanets.

Statement of need

TidalPy is a open-source Python package developed at NASA Goddard Space Flight Center and University of Maryland College Park to support tidal research in Planetary Science. The package provides a flexible, accessible, and performant toolkit for solving problems in tides and tidal dynamics. The same tides that cause Earth's ocean 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 the long-term thermal evolution of these worlds. The energy that drives this heat originates in the orbits and rotations of the planet and its hosts. TidalPy provides functions and frameworks to apply the latest tidal modeling theories and methods to a wide variety of Solar System and exoplanetary worlds.

Overview

TidalPy is written primarily in Python, with performance-critical components implemented in C++ and Cython [REF]. 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 joins a robust community of other packages that perform similar calculations [REF] and expands on this prior work in three major areas described in this section.

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 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 challenging if we

are unable to send a fly-by or orbiting satellite). Therefore, it is useful to perform forward modeling utilizing our best estimates of a world's structure and composition to provide a range of tidal efficiency.

TidalPy provides a Love number solver that uses information about a planet's interior structure and thermal state to estimate these numbers. A user can turn on or off a variety of assumptions to determine their impact. This solver can be used in other routines to, for example, determine the effect of long-term heating on a world's tidal dissipation or in a Markov Chain Monte-Carlo scheme to predict its most likely interior structure.

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.

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, bulk compressibility (See Figure 2), and dynamic tides (See Figure 1). 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 multi-dimensional Love numbers [Qin et al. (2014); Rovira-Navarro et al. (2024); Berne+2023nov].

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

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 later 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); RenaudHenning2018apr]. For example, microscopic grains of ice will tend to move more freely then larger solid crystalline chunks. Likewise, rock that has experienced significant fracturing or is porous tends to have more opportunity to create frictional heat. The choice of **Rheology** determines which dissipation mechanism is dominant within a world.

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

Figure 3: Tidal heating is shown for four different rheology models for a simplified 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. 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-Orbit Dynamics & Coupling - Tidal dissipation transforms the energy from the rotation and orbit of planets and moons into tidal heat. Complex dynamics can occur when a planet has a non-synchronous rotation, high eccentricity, and/or has a non-zero obliquity. TidalPy uses a sophisticated Spin-Orbit Resonance coupling system to account for potential resonance capture and other long-term dynamics [REF; REF].

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. [adapted from REF].

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. [adapted from REF].

TidalPy has proven to be a powerful tool in investigating tides within the Solar System [REFs] and beyond [REFs]. Future releases will focus on increasing performance, improving usability, and incorporating more physics. Get started using TidalPy by installing the package and checking out the documentation over at https://tidalpy.info

Citations

Citations to entries in paper.bib should be in rMarkdown format.

If you want to cite a software repository URL (e.g. something on GitHub without a preferred citation) then you can do it with the example BibTeX entry below for Smith et al. (2020).

For a quick reference, the following citation commands can be used: - `@author:2001` -> "Author et al. (2001)" - `[@author:2001]` -> "(Author et al., 2001)" - `[@author1:2001; @author2:2001]` -> "(Author1 et al., 2001; Author2 et al., 2002)"

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 PyPI and Conda-Forge. Major versions are also released with dedicated DOI's 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).

License

The Journal of Open Source Software

Acknowledgements

TidalPy was greatly improved by conversations, 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 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), Matplotlib (Hunter, 2007), 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

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

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

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*.

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

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

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., 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*.

Smith, A. M., Thaney, K., & Hahnel, M. (2020). Fidgit: An ungodly union of GitHub and figshare. In *GitHub repository*. GitHub. https://github.com/arfon/fidgit

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

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