

¹ TidalPy: Software Suite for Solving Problems in Tidal Dynamics

³ Joe P. Renaud  

⁴ 1 University of Maryland, College Park, Maryland USA  ² NASA Goddard Space Flight Center, Greenbelt, Maryland USA  

DOI: [10.xxxxxx/draft](https://doi.org/10.xxxxxx/draft)

Software

- [Review](#) 
- [Repository](#) 
- [Archive](#) 

Editor: [Open Journals](#) 

Reviewers:

- [@openjournals](#)

Submitted: 01 January 1970

Published: unpublished

License

Authors of papers retain copyright and release the work under a

Creative Commons Attribution 4.0 International License ([CC BY 4.0](#))

⁶ Summary

⁷ TidalPy is an open-source Python package designed to model tidal heating, interior response,
⁸ and long-term spin–orbit evolution of planets and moons across the Solar System and beyond.
⁹ It combines Love number computation, advanced rheological models, and semi-analytical
¹⁰ orbital dynamics into a single accessible framework with performance-critical components in
¹¹ C++ and Cython. The software enables forward modeling of interior structure, dissipation
¹² mechanisms, and tidal feedback on rotation and orbits, supporting applications ranging from
¹³ Solar System geophysics to exoplanet characterization. TidalPy interfaces with the broader
¹⁴ Python scientific ecosystem, offers extensive documentation and examples, and has already
¹⁵ been applied in multiple planetary science studies.

Statement of need

¹⁹ TidalPy provides a flexible, accessible, and performant toolkit for solving problems in tides
²⁰ and tidal dynamics. The same tides that cause Earth's oceans to rise twice each day can
²¹ churn the interiors of other planets and moons to the point that significant fractions of their
²² bulk can melt, greatly altering their long-term thermal evolution. The energy that drives this
²³ heat originates in the orbits and rotations of the world and its host, which are also altered
²⁴ on geologic timescales. TidalPy provides functions and frameworks to apply the latest tidal
²⁵ modeling theories and methods to a wide variety of Solar System and exoplanetary worlds.
²⁶ Written for the Python ecosystem, TidalPy can easily interface with other popular packages
²⁷ used in planetary science and astronomy. This enables fast production of advanced simulations
²⁸ which can be used directly or as a benchmark against other models and tools.

²⁷ Overview

²⁸ TidalPy is written primarily in Python, with performance-critical components implemented in
²⁹ C++ and Cython ([Behnel et al., 2011](#)). Its API is designed to be intuitive and consistent with
³⁰ modern conventions, enabling both early career and experienced researchers to quickly learn its
³¹ syntax and incorporate it in their scientific projects. TidalPy complements a robust community
³² of other packages that perform similar or parallel calculations ([Barnes et al., 2020; Lu et al.,](#)
³³ [2023; Martens et al., 2019; Melini et al., 2022; Neveu, 2025; Qin et al., 2014; Rovira-Navarro](#)
³⁴ [et al., 2024; Saikiran Tharimena, 2024](#)) and expands on this prior work in three major areas
³⁵ described in the following sections.

³⁶ TidalPy has been vetted and used in investigations of tides on Earth ([Vidal & Cébron, 2025](#)),
³⁷ in our Solar System ([Cascioli et al., 2023; Goossens et al., 2024; Wagner & James, 2025](#)), and
³⁸ beyond ([Fauchez et al., 2025; Joe P. Renaud et al., 2021; Joe P. Renaud & Henning, 2018](#)).
³⁹ Documentation and Jupyter Notebook ([Kluyver et al., 2016](#)) demonstrations are available

40 on the [GitHub repository](#), these are continuously added to and updated as TidalPy evolves.
 41 Future releases will focus on increasing performance, improving usability, and incorporating
 42 more physics. Get started using TidalPy by visiting <https://tidalpy.info>.

43 Love Number Solver (RadialSolver Module)

44 Learn more about TidalPy's *RadialSolver Module* [here](#).

45 **Love Numbers** quantify a planet or moon's ability to respond to tidal or loading forces ([Love](#),
 46 [1911](#); [Shida, 1912](#)). They are dynamic and depend on many physical factors such as a world's
 47 thermal state, physical structures (e.g., a presence of a solid or liquid core), past stress events,
 48 and orbit/spin state. These numbers can be measured, albeit with difficulty particularly if no
 49 spacecraft flyby or orbiter is available. Therefore, it is useful to perform forward modeling
 50 utilizing our best estimates of a world's structure and composition to provide a range of values
 51 to constrain a world's response efficiency.

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

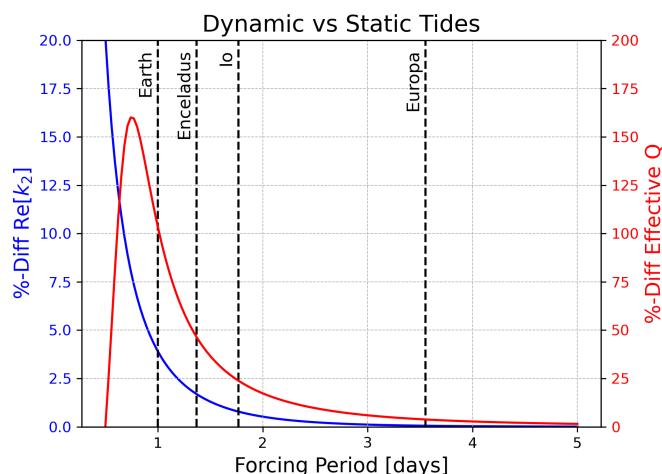


Figure 1: The percent difference between Love number using the dynamic and static assumption is shown for a icy moon with a significant ocean layer as a function of tidal forcing period. Several reference periods are shown to give a sense of when dynamic tides may be important to consider.

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

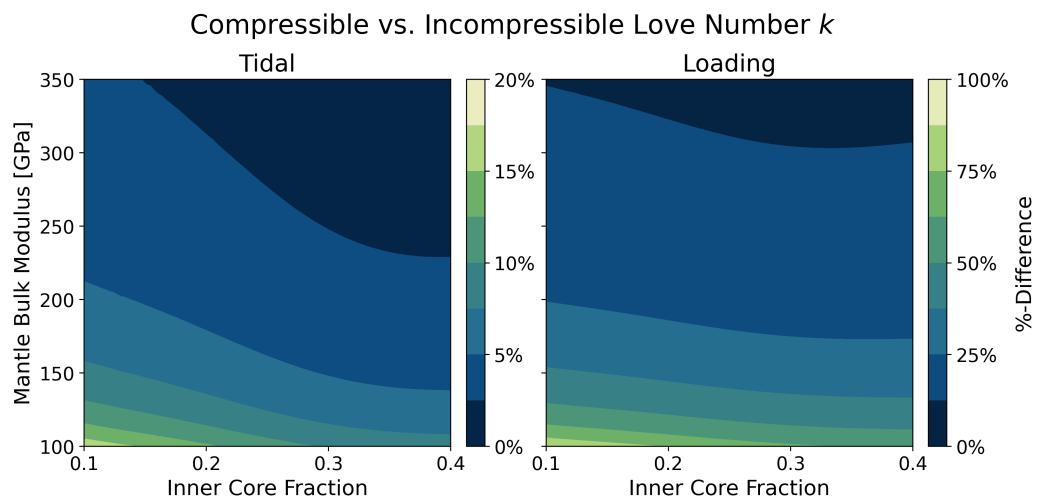


Figure 2: Bulk dissipation can lead to significant differences in both the Tidal (left) and Loading (right) Love numbers in this simplified Venus model.

⁶⁸ Advanced Rheological Modeling (Rheology Module)

⁶⁹ Learn more about TidalPy's Rheology Module [here](#).

⁷⁰ The calculation of tidal Love numbers requires knowing the viscoelastic state of a planet.
⁷¹ This can be described through the shear and bulk modulus as well as the shear and bulk
⁷² viscosity. The former describe how sound waves travel through a planet's bulk; the latter
⁷³ describe how material flows on long timescales. Linking these properties to tides requires
⁷⁴ making assumptions about the dominant mechanism driving dissipation in the rocks and ices
⁷⁵ ([Bagheri et al., 2022](#); [Joe P. Renaud & Henning, 2018](#)). For example, microscopic grains
⁷⁶ of ice will tend to move more freely than larger solid crystalline chunks. Likewise, rock that
⁷⁷ has experienced significant fracturing or is porous tends to have more opportunity to create
⁷⁸ frictional heat than it would otherwise. The choice of **Rheology** determines which dissipation
⁷⁹ mechanism is dominant within a world.

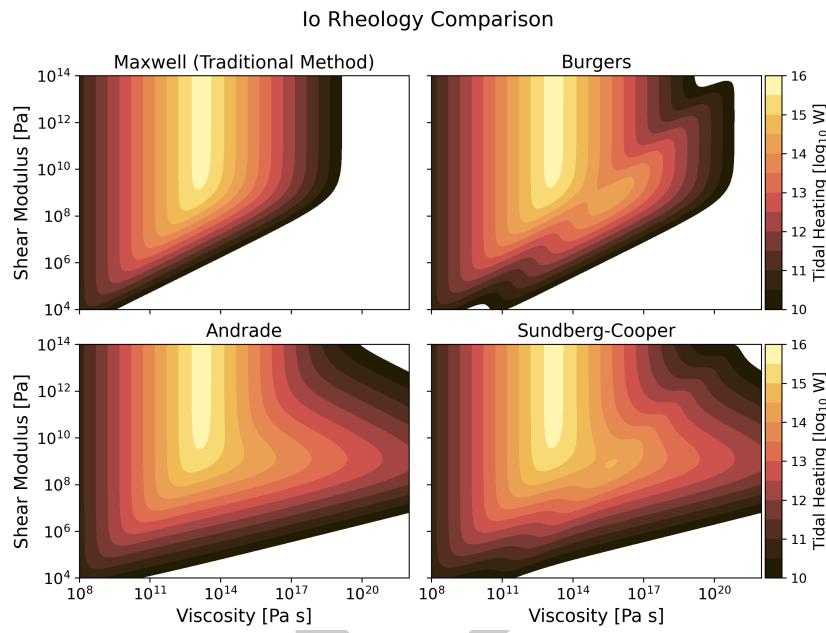


Figure 3: Tidal heating is shown for four different rheology models for a model of Jupiter's moon Io. Heating in certain viscoelastic phase spaces can be orders of magnitude different depending on your choice in rheology.

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

86 Spin and Orbital Evolution (Dynamics Module)

87 Learn more about TidalPy's Dynamics Module [here](#).

88 The energy released as heat in a tidally active world originates in its orbit or in the rotation
 89 of it or its tidal host (which could be Jupiter in the case of Io, or a star in the case of a
 90 short-period exoplanet). Energy can be exchanged between adjacent planets and moons,
 91 further increasing the complexity of the problem (e.g., [Hussmann & Spohn, 2004](#)). Different
 92 orbit and spin configurations will have a variety of forcing frequencies which can act like a
 93 tuning fork, dramatically increasing dissipation in narrow frequency bands ([Bagheri et al.,](#)
 94 [2022](#); [Joe P. Renaud & Henning, 2018](#)). Likewise, troughs in dissipation can slow or stop
 95 the evolution of a world for millions of years. These “Spin-Orbit Resonances” are what drive
 96 Mercury’s 3:2 Spin to Orbit ratio and what leads to systems like Pluto-Charon which are in a
 97 “dual-synchronous” configuration, the ultimate end state of tidal evolution.

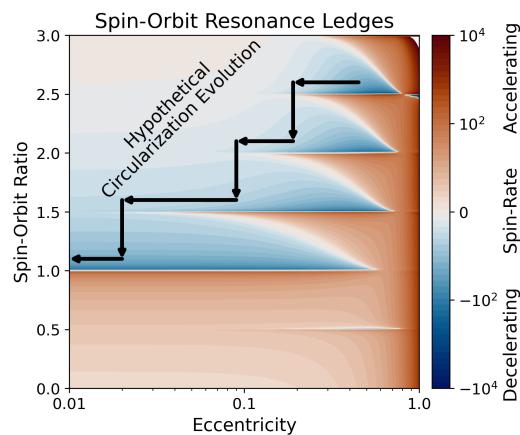


Figure 4: Spin-Orbit Resonance “ledges” calculated with TidalPy. A planet can become trapped on a ledge (stuck at a certain spin rate) for millions of years depending on its interior structure and thermal state.

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

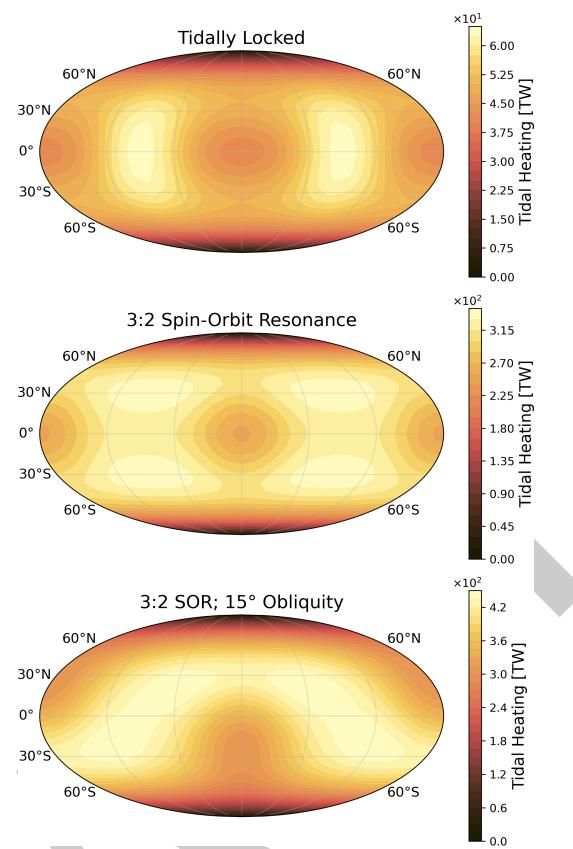


Figure 5: Tidal heating for a short-period exoplanet in three different spin configurations. Non-synchronous spin and a non-zero obliquity can lead to large differences in the magnitude and location of maximum heating.

Availability

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

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

License

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

122 Acknowledgements

123 TidalPy benefited greatly from conversations, code contributions, and testing performed by
124 many in the community. We would like to specifically thank Wade G. Henning, Michael
125 Efroimsky, Michaela Walterová, Sander Goossens, Marc Neveu, Nick Wagner, and Gael
126 Cascioli. The development of TidalPy was supported by NASA Sellers' Exoplanet Environments
127 Collaboration and Planetary Geodesy ISFMs. J. Renaud was additionally supported during
128 its development by the CRESST-II cooperative agreement (NASA award 80GSFC24M0006).
129 TidalPy makes extensive use of the following software: [CyRK](#) (J. P. Renaud, 2022), [BurnMan](#)
130 ([Myhill et al., 2023](#)), [NumPy](#) ([Harris et al., 2020](#)), [SciPy](#) ([Virtanen et al., 2020](#)), [Numba](#) ([Lam
131 et al., 2015](#)), [Cython](#) ([Behnel et al., 2011](#)), [Matplotlib](#) ([Hunter, 2007](#)), [SymPy](#) ([Meurer et al.,
132 2017](#)), [Jupyter](#) ([Kluyver et al., 2016](#)), and [cmcrameri](#) ([Crameri, 2023; Rollo, 2020](#)).

133 References

- 134 Bagheri, A., Efroimsky, M., Castillo-Rogez, J., Goossens, S., Plesa, A.-C., Rambaux, N.,
135 Rhoden, A., Walterová, M., Khan, A., & Giardini, D. (2022). Tidal insights into rocky
136 and icy bodies: An introduction and overview. *Advances in Geophysics*, 63, 231–320.
137 <https://doi.org/10.1016/bs.agph.2022.07.004>
- 138 Barnes, R., Luger, R., Deitrick, R., Driscoll, P., Quinn, T. R., Fleming, D. P., Smotherman, H.,
139 McDonald, D. V., Wilhelm, C., Garcia, R., Barth, P., Guyer, B., Meadows, V. S., Bitz, C.
140 M., Gupta, P., Domagal-Goldman, S. D., & Armstrong, J. (2020). VPPlanet: The Virtual
141 Planet Simulator. 132(1008), 024502. <https://doi.org/10.1088/1538-3873/ab3ce8>
- 142 Behnel, S., Bradshaw, R., Citro, C., Dalcin, L., Seljebotn, D. S., & Smith, K. (2011).
143 Cython: The Best of Both Worlds. *Computing in Science & Engineering*, 13(2), 31–39.
144 <https://doi.org/10.1109/MCSE.2010.118>
- 145 Berne, A., Simons, M., Keane, J. T., & Park, R. S. (2023). Using Tidally-Driven Elastic
146 Strains to Infer Regional Variations in Crustal Thickness at Enceladus. *Geophysical Research
147 Letters*, 50(22), e2023GL106656. <https://doi.org/10.1029/2023GL106656>
- 148 Boué, G., & Efroimsky, M. (2019). Tidal evolution of the Keplerian elements. *Celestial Mechanics and
149 Dynamical Astronomy*, 131(7), 30. <https://doi.org/10.1007/s10569-019-9908-2>
- 150 Cascioli, G., Renaud, J. P., Mazarico, E., Durante, D., less, L., Goossens, S., & Smrekar, S.
151 (2023). Constraining the Venus Interior Structure with Future VERITAS Measurements of
152 the Gravitational Atmospheric Loading. 4(4), 65. <https://doi.org/10.3847/PSJ/acc73c>
- 153 Crameri, F. (2023). Scientific colour maps. *Zenodo*. <https://doi.org/10.5281/zenodo.1243862>
- 154 Fauchez, T., Saxena, P., Wolf, E., Chen, H., Renaud, J., & Gonzalez Quiles, J. (2025).
155 Potential Effects of Exovolcanism on Transmission Spectroscopy of Earth-size Exoplanets.
156 *American Astronomical Society Meeting Abstracts #245*, 245, 314.04.
- 157 Foreman-Mackey, D., Conley, A., Meierjurgen Farr, W., Hogg, D. W., Lang, D., Marshall,
158 P., Price-Whelan, A., Sanders, J., & Zuntz, J. (2013). *emcee: The MCMC Hammer*.
159 Astrophysics Source Code Library, record ascl:1303.002.
- 160 Goossens, S., van Noort, B., Mateo, A., Mazarico, E., & van der Wal, W. (2024). A low-
161 density ocean inside Titan inferred from Cassini data. *Nature Astronomy*, 8, 846–855.
162 <https://doi.org/10.1038/s41550-024-02253-4>
- 163 Harris, C. R., Millman, K. J., Walt, S. J. van der, Gommers, R., Virtanen, P., Cournapeau, D.,
164 Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., Kerkwijk,
165 M. H. van, Brett, M., Haldane, A., Río, J. F. del, Wiebe, M., Peterson, P., ... Oliphant,
166 T. E. (2020). Array programming with NumPy. *Nature*, 585(7825), 357–362. <https://doi.org/10.1038/s41586-020-2649-2>

- 168 Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. *Computing in Science &*
 169 *Engineering*, 9(3), 90–95. <https://doi.org/10.1109/MCSE.2007.55>
- 170 Hussmann, H., & Spohn, T. (2004). Thermal-orbital evolution of Io and Europa. *Icarus*,
 171 171(2), 391–410. <https://doi.org/10.1016/j.icarus.2004.05.020>
- 172 Kaula, W. M. (1964). Tidal dissipation by solid friction and the resulting orbital evolution.
 173 *Reviews of Geophysics*, 2(4), 661. <https://doi.org/10.1029/RG002i004p00661>
- 174 Kluyver, T., Ragan-Kelley, B., Pérez, F., Granger, B., Bussonnier, M., Frederic, J., Kelley,
 175 K., Hamrick, J., Grout, J., Corlay, S., Ivanov, P., Avila, D., Abdalla, S., Willing, C., &
 176 team, J. development. (2016). Jupyter notebooks - a publishing format for reproducible
 177 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/>
- 178 Lam, S. K., Pitrou, A., & Seibert, S. (2015). Numba: A LLVM-based python JIT compiler.
 179 *Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC*.
 180 <https://doi.org/10.1145/2833157.2833162>
- 181 Love, A. E. H. (1911). *Some Problems of Geodynamics*.
- 182 Lu, T., Rein, H., Tamayo, D., Hadden, S., Mardling, R., Millholland, S. C., & Laughlin, G.
 183 (2023). Self-consistent Spin, Tidal, and Dynamical Equations of Motion in the REBOUNDx
 184 Framework. 948(1), 41. <https://doi.org/10.3847/1538-4357/acc06d>
- 185 Martens, H. R., Rivera, L., & Simons, M. (2019). LoadDef: A Python-Based Toolkit to Model
 186 Elastic Deformation Caused by Surface Mass Loading on Spherically Symmetric Bodies.
 187 *Earth and Space Science*, 6(2), 311–323. <https://doi.org/10.1029/2018EA000462>
- 188 Melini, D., Salibi, C., & Spada, G. (2022). On computing viscoelastic Love numbers for
 189 general planetary models: The ALMA3 code. *Geophysical Journal International*, 231(3),
 190 1502–1517. <https://doi.org/10.1093/gji/ggac263>
- 191 Meurer, A., Smith, C. P., Paprocki, M., Čertík, O., Kirpichev, S. B., Rocklin, M., Kumar,
 192 A., Ivanov, S., Moore, J. K., Singh, S., Rathnayake, T., Vig, S., Granger, B. E., Muller,
 193 R. P., Bonazzi, F., Gupta, H., Vats, S., Johansson, F., Pedregosa, F., ... Scopatz, A.
 194 (2017). SymPy: Symbolic computing in python. *PeerJ Computer Science*, 3, e103.
 195 <https://doi.org/10.7717/peerj-cs.103>
- 196 Myhill, R., Cottaar, S., Heister, T., Rose, I., Unterborn, C., Dannberg, J., & Gassmoeller,
 197 R. (2023). BurnMan – a python toolkit for planetary geophysics, geochemistry and
 198 thermodynamics. *Journal of Open Source Software*, 8(87), 5389. <https://doi.org/10.21105/joss.05389>
- 199 Neveu, M. (2025). IcyDwarf: A coupled physical-chemical simulator of icy dwarf planets or
 200 moons. In *GitHub repository*. GitHub. <https://github.com/MarcNeveu/IcyDwarf>
- 201 Qin, C., Zhong, S., & Wahr, J. (2014). A perturbation method and its application: Elastic tidal
 202 response of a laterally heterogeneous planet. *Geophysical Journal International*, 199(2),
 203 631–647. <https://doi.org/10.1093/gji/ggu279>
- 204 Renaud, J. P. (2022). CyRK - ODE Integrator Implemented in Cython and Numba. Zenodo.
 205 <https://doi.org/10.5281/zenodo.7093266>
- 206 Renaud, Joe P., & Henning, W. G. (2018). Increased Tidal Dissipation Using Advanced
 207 Rheological Models: Implications for Io and Tidally Active Exoplanets. *The Astrophysical
 208 Journal*, 857(2), 98. <https://doi.org/10.3847/1538-4357/aab784>
- 209 Renaud, Joe P., Henning, W. G., Saxena, P., Neveu, M., Bagheri, A., Mandell, A., & Hurford,
 210 T. (2021). Tidal Dissipation in Dual-body, Highly Eccentric, and Nonsynchronously
 211 Rotating Systems: Applications to Pluto-Charon and the Exoplanet TRAPPIST-1e. 2(1),
 212 213 214

- 215 4. <https://doi.org/10.3847/PSJ/abc0f3>
- 216 Renaud, Joe P., Lopez, E., Brande, J., Cruz-Arce, C. E., Kelahan, C., Susemehl, N., Cristy,
217 D., Hostetter, C., Moore, M. D., Patel, A., & Mandell, A. M. (2022). The Exoplanet
218 Modeling and Analysis Center at NASA Goddard. *Research Notes of the AAS*, 6(9), 185.
219 <https://doi.org/10.3847/2515-5172/ac9060>
- 220 Rollo, C. (2020). Cmcrameri: Python wrapper around fabio crameri's perceptually uniform
221 colormaps. In *GitHub repository*. GitHub. <https://github.com/callumrollo/cmcrameri>
- 222 Rovira-Navarro, M., Matsuyama, I., & Berne, A. (2024). A Spectral Method to Compute
223 the Tides of Laterally Heterogeneous Bodies. *The Planetary Science Journal*, 5(5), 129.
224 <https://doi.org/10.3847/PSJ/ad381f>
- 225 Saikiran Tharimena, D. M., Marshall J. Styczinski. (2024). PyALMA3: A pythonized version
226 of ALMA 3. In *GitHub repository*. GitHub. <https://github.com/drsairikant88/PyALMA3>
- 227 Shida, T. (1912). *On the elasticity of the Earth and the Earth's crust*.
- 228 Takeuchi, H., & Saito, M. (1972). Seismic Surface Waves. In *Methods in Computational
229 Physics: Advances in Research and Applications* (Vol. 11, pp. 217–295). <https://doi.org/10.1016/B978-0-12-460811-5.50010-6>
- 231 Vidal, J., & Cébron, D. (2025). Did lunar tides sustain the early Earth's geodynamo? *arXiv
232 e-Prints*, arXiv:2506.19039. <https://doi.org/10.48550/arXiv.2506.19039>
- 233 Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D.,
234 Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson,
235 J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., ... SciPy
236 1.0 Contributors. (2020). SciPy 1.0: Fundamental Algorithms for Scientific Computing in
237 Python. *Nature Methods*, 17, 261–272. <https://doi.org/10.1038/s41592-019-0686-2>
- 238 Wagner, N. L., & James, P. B. (2025). New Geophysical Constraints for Intrusive Magmatism
239 at Large Martian Volcanoes: Implications for Crustal Thickness and Volatile Outgassing.
240 *Journal of Geophysical Research (Planets)*, 130(8), e2025JE008959. <https://doi.org/10.1029/2025JE008959>