SpiceyPy: a Pythonic Wrapper for the SPICE Toolkit

Andrew M. Annex¹, Ben Pearson², Benoît Seignovert³, Brian T. Carcich⁴, Helge Eichhorn⁵, Jesse A. Mapel⁶, Johan L. Freiherr von Forstner⁷, Jonathan McAuliffe⁸, Jorge Diaz del Rio⁹, Kristin L. Berry⁶, K.-Michael Aye¹⁰, Marcel Stefko¹¹, Miguel de Val-Borro¹², Shankar Kulumani¹³, and Shin-ya Murakami¹⁴

1 Johns Hopkins University 2 None 3 Jet Propulsion Laboratory, California Institute of Technology 4 Latchmoor Services, LLC 5 Planetary Transportation Systems GmbH 6 USGS Astrogeology Science Center 7 Institute of Experimental and Applied Physics, University of Kiel 8 DLR Gesellschaft für Raumfahrtanwendungen (GfR) mbH 9 ODC Space 10 Laboratory for Atmospheric and Space Physics, University of Colorado 11 ETH Zurich 12 Planetary Science Institute 13 Collins Aerospace 14 GFD Dennou Club

Statement of Need

Operating in space necessitates quantifying the positions, velocities, geometries, and other properties of spacecraft and planetary bodies through time. Scientists and engineers working with robotic planetary spacecraft missions use the Spacecraft, Planet, Instrument, Camera-matrix, Events (SPICE) Toolkit (Acton, Bachman, Semenov, & Wright, 2018) to help plan observations and to quantify the positions of planetary bodies and spacecraft through time. SPICE is developed at the Jet Propulsion Laboratory by NASA’s Navigation and Ancillary Information Facility (NAIF). Scientists also use SPICE to analyze data returned by these missions and to plan hypothetical orbital trajectories for future missions (Acton et al., 2018). For example, SPICE can calculate future occultations of planets relative to a camera on a rover or spacecraft. The NAIF provides SPICE in Fortran 77, C, and they also provide Matlab and IDL wrappers; however, as of 2014, they did not offer a Python interface. The growth of Python and movement away from proprietary interpreted languages (Burrell et al., 2018) motivated the development of SpiceyPy so that planetary scientists and engineers can use SPICE within Python.

Summary

SpiceyPy is an open-source, MIT licensed Python package that provides a pythonic interface to nearly all of the C SPICE toolkit N66. SpiceyPy was developed in Python using the ctypes module of the CPython standard library to wrap the underlying C SPICE shared library. Developing SpiceyPy in Python enabled the SpiceyPy API to expose simplified and more pythonic interactions with the underlying C API for SPICE. SpiceyPy relies on the NumPy library for numeric arrays and tight integration with the SciPy stack.

SpiceyPy is extensively tested using a combination of unit and integration tests, which run using continuous integration services. The tests also serve as code examples translated from the NAIF documentation. Continuous deployment updates documentation and deploys artifacts of releases to PyPI and the conda-forge. Every SPICE function wrapper in SpiceyPy contains docstrings that provide short descriptions of the function duplicated from the SPICE documentation. Docstrings in SpiceyPy also contain links to the corresponding CSPICE documentation page hosted by the NAIF to provide additional details regarding the function.
SpiceyPy enables scientists to utilize the full functionality of SPICE within Python and the ecosystem of visualization and scientific packages available. SpiceyPy has been utilized in peer-reviewed research (Attree et al., 2019; Behar, Nilsson, Alho, Goetz, & Tsurutani, 2017; Behar et al., 2016; Porter et al., 2018; Zangari, Finley, Alan Stern, & Tapley, 2018), masters and doctoral theses (Albin, 2019; Hackett, 2019), spacecraft mission operations, as a dependency in other Python libraries (Stansby, Rai, Broll, Shaw, & others, 2019), and for a variety of other projects (Costa & Grass, 2018; Wilson, 2017; Wilson & Xiong, 2016).

Acknowledgements

The authors would like to acknowledge members of the NAIF (Charles Acton, Ed Wright, Boris Semenov, Nat Bachman) for continued support for SpiceyPy and for providing to users a SpiceyPy translation of their excellent “Hands-on” lessons. The first author also thanks all of the contributors and users of SpiceyPy; they motivate further improvements to the project. Co-authors other than the first author are ordered solely alphabetically by their first name.

References


