SpiceyPy, A Complete Pythonic Interface For Spice.

A. M. Annex[1], [1]Department of Earth and Planetary Sciences, Johns Hopkins University, Baltimore, MD 21218 (annex@jhu.edu).

Introduction: The SPICE library provided by the Navigation and Ancillary Information Facility (NAIF) is a powerful tool used by scientists and engineers for determining geometric conditions of planets and spacecraft [1]. The NAIF provides SPICE interfaces for interpreted languages such as MATLAB and IDL. However, a Python interface to SPICE was not available until the development of the open source package SpiceyPy with the release of version 1.0.0 in March 2016. SpiceyPy is open sourced under the MIT software license and is hosted on GitHub*[2]. This abstract serves as a status update on the project, utilization in the field, and future development plans.

Background: SpiceyPy uses the ctypes foreign function interface (FFI) which is part of the Python standard library to interface directly with a compiled shared library of CSPICE. Interface functions are written in pure Python code for each CSPICE function enabling API refinement and convenience functionality. This allowed a more "pythonic" interface to CSPICE to be crafted, which is a desirable trait for python packages [3]. Functions are all extensively tested with coverage reports collected from weekly test builds using the Travis Continuous Integration (CI) service. Currently, the code coverage from tests is reported to be 99.795%† of the full codebase. SpiceyPy is available through the Python Package Index (PyPI) and the community supported Conda-forge for Anaconda Python users. SpiceyPy can be installed on all 64bit systems on major operating systems by running "pip install spiceypy" or "conda install spiceypy --channel conda-forge".

Development History: Development on SpiceyPy began in early 2014 due to the desire of the author to have cross-platform access to SPICE. After two years of development, SpiceyPy version 1.0.0 was released in March of 2016. Version 2.0.0 was released soon after the release of SPICE N66 and included many of the new CSPICE functions and DSK capabilities of that release in June 2017. The rest of the interfaces to the new N66 SPICE functions were added later that year in November 2017 with the release of version 2.1.0. Afterward, development on SpiceyPy primarily focused on resolving installation issues, bug fixes, documentation improvements, completion of the API coverage, and adopting better community standards such as a code of conduct culminating with the release of version 2.2.0 in February 2019 [2]. Releases after 2.0.0 have received many substantial community contributions, demonstrating increased community envolvement in the project.

API details: SpiceyPy provides a nearly complete interface to CSPICE N66; all but eight functions out of about 585 have interfaces, and all of these functions have unit tests. This is a substantive increase in test coverage and function coverage since the release of SpiceyPy 2.0.0 that is in large thanks to community contributions. Functions for Event Kernels (EK) and geometry finding routines that utilize C function callbacks are now all supported in SpiceyPy. The remaining unwrapped functions are either experimental DSK related functions (that are considered deprecated as of the N66 release) or are not particularly useful to a Python user (such as "maxd_c").

Usage by the Community: Gauging the direct usage and impact of SpiceyPy is difficult given the inconsistent usage of DOIs for SpiceyPy and possible exclusion from references. However, there are many proxies for popularity and usage that can be used. One of these metrics is the number of forks and stars the GitHub repository has, which forms a rough gauge of popularity among those with GitHub accounts [4], see figure 1. Since 2018, SpiceyPy has also been cited in 11 documents that include technical reports, theses, peer-reviewed papers, and conference abstracts. These documents originated from a variety of engineering and scientific fields and were identified from a Google Scholar query‡. SpiceyPy is also increasingly utilized in other libraries as a dependency, including Heliopy and spiops and some projects by the USGS astrogeology organization.

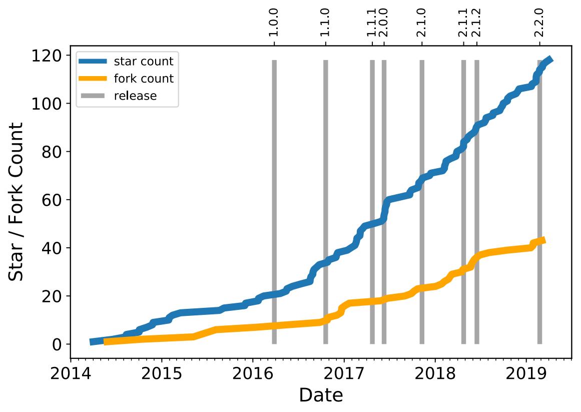

Figure 1 GitHub cumulative star and fork counts over time, release dates plotted as vertical gray lines. Data from GitHub.

*: https:github.com/AndrewAnnex/SpiceyPy
†: https://coveralls.io/builds/22493730
‡: https://scholar.google.com/scholar?as_vis=0&q=spiceypy&hl=en&as_sdt=1,21

Download counts do not detail how SpiceyPy is used but can be considered a low precision gauge of interest. For PyPI, download counts are queryable through a publically available Google BigQuery™ table "the-psf:pypi.downloads" [5]. A query for PyPI downloads that used the pip command was run against this table limited between 01/01/2017 and 04/04/2019 which processed 852.2 GB of data. The results are reported in table 1 and aggregate the download counts grouped by the version of SpiceyPy and the major revision number of Python to distinguish Python 2 users from those using Python 3. Additionally, Conda-forge downloads were reported in table 1, although this distribution of SpiceyPy has only been available since December of 2017. Downloads of version numbers older than 2.0.0 were excluded from the table.

Version	2.0.0	2.1.0	2.1.1	2.1.2	2.2.0
Python 3.X downloads	275	1363	647	2800	500
Legacy Python (2.X) downloads	453	230	65	714	108
Conda-forge downloads	1211	1659	NA	1944	NA

Table 1: Counts of SpiceyPy downloads for versions 2.0.0 and up from pip grouped by version and Python version (3.X vs. 2.X) between 01/01/2017 and 04/04/2019. See figure 1 for the dates of specific releases. "NA"s indicate missing versions from Conda Forge, counts which began approximately in December 2017.

By comparing these numbers to similar download counts reported in a prior abstract, download counts jumped substantially [6]. This jump may indicate both increased interest and utilization of SpiceyPy, although this is an imperfect measure of both properties.

Future Directions: Development of SpiceyPy will for the foreseeable future focus on usability improvements, documentation improvements, and bug fixes because all of the important CSPICE API functions are available in SpiceyPy. Future function additions are primarily limited to any new functions added in future unreleased versions of SPICE, such as N67. Previously discussed future projects such as CFFI [7] based reimplementation have been prototyped, but the need has not been demonstrated yet. New Python 3 features such as type annotations [8] should be investigated for enhanced type safety in SpiceyPy. However, this could break the legacy Python (2.x) compatibility. Legacy Python, despite the upcoming deprecation in 2020, is still in use by a nonnegligible percentage of SpiceyPy users, see table 1. Deprecation of legacy python support in SpiceyPy would result in some code maintenance burden reductions and documentation improvements. Further education of the SpiceyPy user base on the need to upgrade to modern python versions (preferably 3.7.X or newer) is needed to help make these necessary changes possible.

Another development that will likely happen within the next year is the migration of SpiceyPy to an independent GitHub organization apart from the authors GitHub profile. This development is the result of conversations with interested parties at various organizations that partially desire more redundancy in the continued maintenance and support of SpiceyPy into the future. The author will in all likelihood remain involved in future development on SpiceyPy, but it is in the interest of the project to spread the development burden broadly. This transition will require that a suitable GitHub organization is identified. This organization must have adopted a good governance model to ensure credit is correctly attributed and that the primary author remains involved.

Conclusions: SpiceyPy is now a feature complete open source wrapper for SPICE implemented by the community. SpiceyPy is being used in an increasing number of open source projects as well as peer-reviewed research. SpiceyPy remains supported by the community and will be updated as necessary upon future NAIF releases of SPICE. The API for SpiceyPy has reached a high level of stability and maturity, enabling researchers and engineers alike to utilize the full extent of the CSPICE API within Python.

Acknowledgements: The author would like to thank all the users and contributors to SpiceyPy for their continued comments, suggestions, bug reports, and anecdotes. These have all helped to drive the project forward and make it a joy. To the users, please feel free at any time to write in to say how SpiceyPy is aiding their work.

References: [1] C. Acton, N. Bachman, B. Semenov, E. Wright; *A look toward the future in the handling of space science mission geometry*. P&SS. (2017) DOI: 10.1016/j.pss.2017.02.013. [2] A. Annex, *et al. SpiceyPy.* (2019) DOI: 10.5281/zenodo.593914. [3] C. Alexandru, J. Merchante, S. Panichella, S. Proksch, H. Gall, and G Robles. *On the usage of pythonic idioms.* (2018) DOI: 10.1145/3276954.3276960 [4] T. Jason, L. Dabbish, and J. Herbsleb. *Influence of social and technical factors for evaluating contribution in GitHub.* ACM. (2014) DOI: 10.1145/2568225.2568315 [5] D. Stufft, *Publicly Queryable Statistics.* (2016) mail.python.org/pipermail/distutils-sig/2016-May/028986.html. [6] A. Annex. *SpiceyPy, a Python Wrapper for SPICE.* 3rd Planetary Data Workshop. (2017) 2017LPICo1986.7081A. [7] R. Armin, *et al. CFFI Documentation* cffi.readthedocs.io. [8] G. Rossum, J. Lehtosalo, and Ł. Langa. *PEP 484 Type Hints.* (2014) www.python.org/dev/peps/pep-0484/.