December 22, 2021

Draft version December 22, 2021
Typeset using LaTeX/show your work! modern style in AASTeX631

The Astropy Project: Sustaining and Growing a Community-oriented Open-source Project and the Latest Major Release (v5.0) of the Core Package

Astropy Collaboration

Submitted to ApJ

ABSTRACT

To be written...

Keywords: Astrophysics - Instrumentation and Methods for Astrophysics — methods: data analysis — methods: miscellaneous

1. INTRODUCTION

The Python programming language is a high-level, interpreted (as opposed to compiled) programming language that has become an industry standard across many computational domains, technological sectors, and fields of research. Despite claims to the contrary (Portegies Zwart 2020), Python enables scalable, time- and energy-efficient code execution (e.g., Augier et al. 2021) with a focus on code readability, ease of use, and interoperability with other languages. Over the last decade, Python has grown enormously in popularity to become a dominant programming language in the astronomical and broader scientific communities. For example, Figure 1 shows the number of yearly full-text mentions of Python as compared to a few other programming languages in refereed articles in the astronomical literature, demonstrating its nearly exponential growth in popularity. The rapid adoption of Python by astronomy researchers, students, observatories, and technical staff combined with an associated increase in awareness and interest about open-source software tools is contributing to

Corresponding author: Astropy Coordination Committee
coordinators@astropy.org

The author list has two parts: the authors that made significant contributions to the writing of the paper in order of contribution, followed by contributors to the Astropy Project in alphabetical order. **The position in the author list does not correspond to contributions to the Astropy Project as a whole.** A more complete list of contributors to the core package can be found in the package repository, and at the Astropy team webpage.

a paradigm shift in the way research is done, data is analyzed, and results are shared in astronomy and beyond.

One of the factors that has led to its rapid ascent in popularity in scientific contexts has been the significant, volunteer-driven effort behind developing community-oriented open-source software tools and fostering communities of users and developers that have grown around these efforts. Today, a broad and feature-diverse "ecosystem" of packages exists in the `Python` scientific computing landscape: Roughly ordered from general-use to domain-specific, this landscape now includes packages that provide core numerical analysis functionality like `numpy` (Harris et al. 2020) and `scipy` (Jones et al. 2001–), visualization frameworks like `matplotlib` (Hunter 2007), machine learning and data analysis packages like `tensorflow` (Abadi et al. 2015), `pymc3` (Salvatier et al. 2016), and `emcee` (Foreman-Mackey et al. 2013), domain-specific libraries like `yt` (Turk et al. 2011), `plasmapy` (Community et al. 2021), `sunpy` (The SunPy Community et al. 2020), `Biopython` (Cock et al. 2009), and `sympy` (?) (to name a few in each category). The `astropy` (Astropy Collaboration et al. 2013, 2018) core package began in this vein, as an effort to consolidate the development of commonly-used functionality needed to perform astronomical research into a community-developed `Python` package.

The `astropy` core package was one of the first large, open-source `Python` packages developed for astronomy and provides, among other things, software functionality for reading and writing astronomy-specific data formats (e.g., FITS), transforming and representing astronomical coordinates, and representing and propagating physical units in code. An early description of the core functionality in `astropy` can be found in the first Astropy paper (Astropy Collaboration et al. 2013) or in detail in the core package documentation.[1] The `astropy` core package is now largely stable in that the software interface does not change without sufficient and significant motivation, and the addition of new features into the core package has slowed significantly as compared to the first years of its development. This is largely driven by the fact that the core package now represents just one piece of the broader astronomy `Python` context, and much new feature development is now happening in more specialized packages that are expanding the capabilities of the Astropy ecosystem by building on top of the foundations laid by the `astropy` core package. Because of this natural expansion, the name Astropy has grown in scope beyond a single `Python` library to become "the Astropy Project."

The Astropy Project is a community effort that represents the union of the `astropy` core package, the ecosystem of astronomy-specific software tools that are interoperable with `astropy` (Astropy Affiliated Packages), *and* the community of users, developers, and maintainers that participate in Astropy efforts. However, there is no institution responsible for managing the Astropy Project, for funding or maintain-

[1] https://docs.astropy.org/

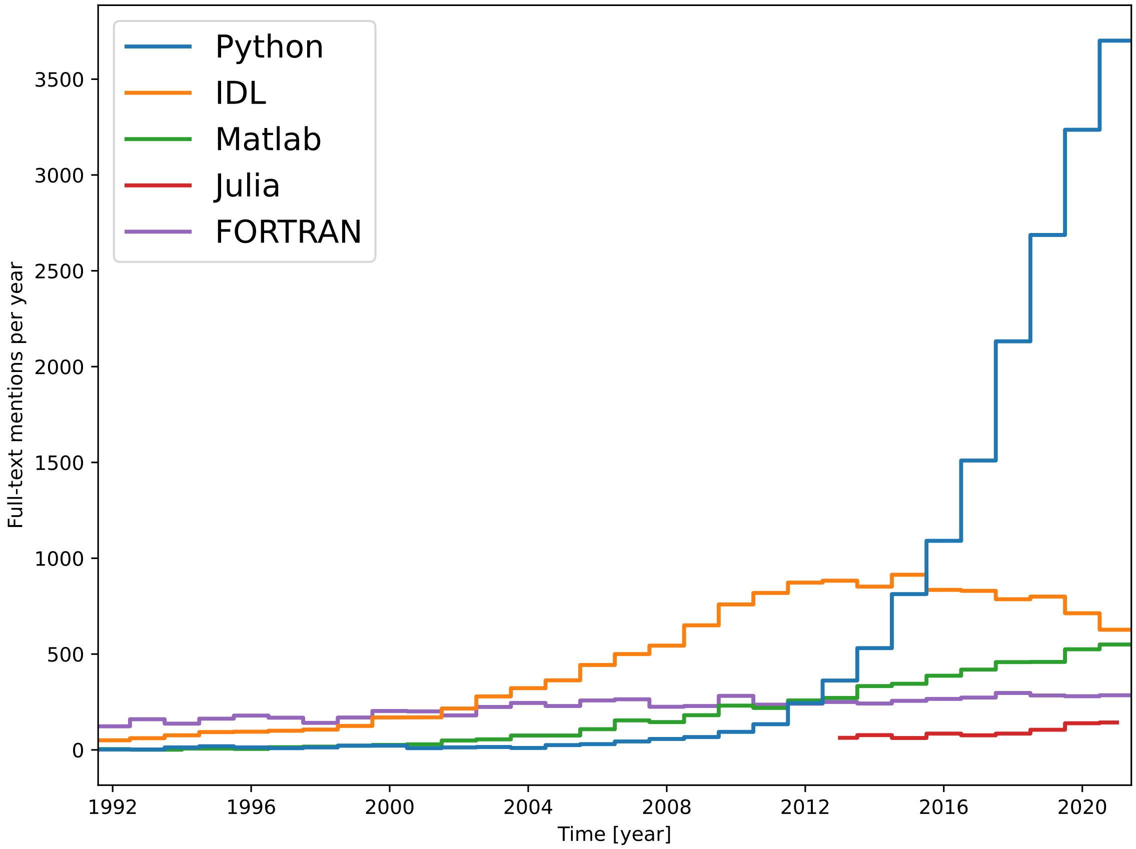

Figure 1. Yearly full-text mentions of programming languages (indicated in the figure legend) in refereed publications in the astronomical literature database in the Astrophysics Data System (ADS; ?). `Python` has rapidly become the dominant programming language mentioned in refereed articles over the last 10 years.

ing its development, or sustaining it into the future: The Project is maintained and coordinated largely by volunteers. While new Astropy-affiliated packages are being developed that expand upon the core functionality in the `astropy` package, representing a natural expansion of the Astropy Project ecosystem, the needs of and challenges faced by the Project are evolving. In particular, the transition from focusing our energy on development and maintenance of a single core package, to instead sustaining the core package and fostering the development of the community and its expansion has been a key issue faced by the Astropy Project in the last several years.

In this Article, we briefly describe recent key updates in the `astropy` core package since the last Astropy paper ("Paper II"; Astropy Collaboration et al. 2018), major updates in the governance, contributor base, and funding of the Project, and discuss some of the future plans and challenges faced by the Astropy Project.

2. MAJOR UPDATES TO THE ASTROPY CORE PACKAGE

2.1. *New Long-term Support (LTS) Version: v5.0*

2.2. *Highlighted Feature Development*

- Support for representing and transforming velocity data in coordinates, epoch propagation (v3.0)
- Improved support for astronomical time series: TimeSeries object (v3.2), Box Least Squares periodogram (v3.1)
- Overall improved support of Quantity throughout numpy (v4.0) and scipy
- Native support for Time, Quantity, and SkyCoord objects in Astropy tables
- TODO: something about WCS, SpectralCoord?

3. MAJOR UPDATES IN THE ASTROPY PROJECT

3.1. *Project governance*

New CoCo and election overview.

3.2. *Contributor base*

Overview and statistics of contributors. Changes since v2.0.

3.3. *Funding*

New funding sources.

4. SUPPORTING THE ECOSYSTEM OF ASTRONOMICAL PYTHON SOFTWARE

4.1. *Community-oriented infrastructure*

Infrastructure packages that exist to support and help others maintain their package infrastructure.

4.2. *Affiliated packages*

New packages and major updates!

4.3. *Connections with observatories*

JWST (motivated by excitement around the launch?)
Gemini

5. FUTURE PLANS FOR THE ASTROPY PROJECT

5.1. *Roadmap*

6. LEARN ASTROPY

Current status and scope
Vision for the future (review what we said in v2.0 paper)
User forums and engagement with user base

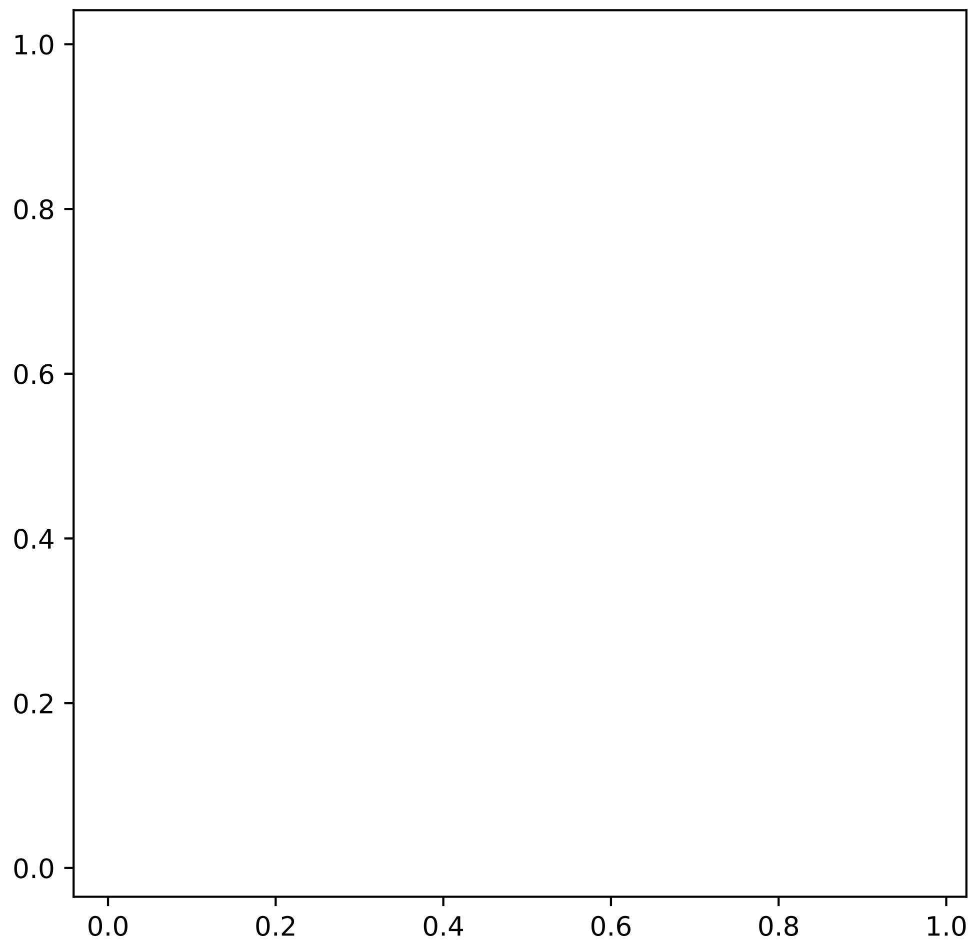

Figure 2. Placeholder figure!

6.1. *Challenges*

Attracting new contributors when the code has become quite complex, Contributor to maintainer mentoring, Long-term / sustained funding for maintaining infrastructure, ...

We would like to thank the members of the community that have contributed to Astropy, that have opened issues and provided feedback, and have supported the project in a number of different ways.

The Astropy community is supported by and makes use of a number of organizations and services outside the traditional academic community. We thank Google for financing and organizing the Google Summer of Code (GSoC) program, that has funded severals students per year to work on Astropy related projects over the summer. These students often turn into long-term contributors. We also thank NumFOCUS and the Python Software Foundation for financial support. Within the academic community, we thank institutions that make it possible that astronomers and other developers on their staff can contribute their time to the development of Astropy projects. We would like acknowledge the support of the Space Telescope Science Institute, HarvardSmithsonian Center for Astrophysics, and the South African Astronomical Observatory.

Furthermore, the Astropy packages would not exist in their current form without a number of web services for code hosting, continuous integration, and documentation; in particular, Astropy heavily relies on GitHub, Travis CI, Appveyor, CircleCI, and Read the Docs.

`astropy` interfaces with the SIMBAD database, operated at CDS, Strasbourg, France. It also makes use of the ERFA library (Tollerud et al. 2017), which in turn derives from the IAU SOFA Collection[2] developed by the International Astronomical Union Standards of Fundamental Astronomy (Hohenkerk 2011).

Software: `astropy` (Astropy Collaboration et al. 2013, 2018), `numpy` (Harris et al. 2020), `scipy` (Jones et al. 2001–), `matplotlib` (Hunter 2007), `Cython` (Behnel et al. 2011).

REFERENCES

Abadi, M., Agarwal, A., Barham, P., et al. 2015, TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. https://www.tensorflow.org/

Astropy Collaboration, Robitaille, T. P., Tollerud, E. J., et al. 2013, A&A, 558, A33, doi: 10.1051/0004-6361/201322068

Astropy Collaboration, Price-Whelan, A. M., Sipőcz, B. M., et al. 2018, AJ, 156, 123, doi: 10.3847/1538-3881/aabc4f

Augier, P., Bolz-Tereick, C. F., Guelton, S., & Mohanan, A. V. 2021, Nature Astronomy, 5, 334, doi: 10.1038/s41550-021-01342-y

Behnel, S., Bradshaw, R., Citro, C., et al. 2011, Computing in Science Engineering, 13, 31 , doi: 10.1109/MCSE.2010.118

Cock, P. J. A., Antao, T., Chang, J. T., et al. 2009, Bioinformatics, 25, 1422, doi: 10.1093/bioinformatics/btp163

Community, P., Everson, E., Staczak, D., et al. 2021, PlasmaPy, 0.6.0, Zenodo, doi: 10.5281/zenodo.4602818

Foreman-Mackey, D., Hogg, D. W., Lang, D., & Goodman, J. 2013, PASP, 125, 306, doi: 10.1086/670067

Harris, C. R., Millman, K. J., van der Walt, S. J., et al. 2020, Nature, 585, 357, doi: 10.1038/s41586-020-2649-2

Hohenkerk, C. 2011, Scholarpedia, 6, 11404, doi: 10.4249/scholarpedia.11404

Hunter, J. D. 2007, Computing In Science & Engineering, 9, 90, doi: 10.1109/MCSE.2007.55

Jones, E., Oliphant, T., Peterson, P., et al. 2001–, SciPy: Open source scientific tools for Python. http://www.scipy.org/

Portegies Zwart, S. 2020, Nature Astronomy, 4, 819, doi: 10.1038/s41550-020-1208-y

Salvatier, J., Wiecki, T. V., & Fonnesbeck, C. 2016, PeerJ Computer Science, 2, e55, doi: 10.7717/peerj-cs.55

The SunPy Community, Barnes, W. T., Bobra, M. G., et al. 2020, The Astrophysical Journal, 890, 68, doi: 10.3847/1538-4357/ab4f7a

Tollerud, E., Pascual, S., Nair, P., et al. 2017, doi: 10.5281/zenodo.1021149

Turk, M. J., Smith, B. D., Oishi, J. S., et al. 2011, ApJS, 192, 9, doi: 10.1088/0067-0049/192/1/9