Open Source Tools, Frameworks, and Libraries
Abstracts of Selected Proposals
(NNH20ZDA001N-OSTFL)

Below are the abstracts of proposals selected for funding for the Open Source Tools, Frameworks, and Libraries program. Principal Investigator (PI) name, institution, and proposal title are also included. Sixty-one (61) proposals were received in response to this opportunity. On August 23, 2021, eight proposals were selected for funding.

Philip Brodrick/Jet Propulsion Laboratory
No One Owns the Rainbow: Open Source Imaging Spectroscopy for SBG and Beyond

We will transform the accessibility, sustainability, and utility of the first open source atmospheric correction code for imaging spectroscopy, filling a critical usability gap in preparation for global acquisitions by NASA s Surface Biology and Geology (SBG) decadal designated observable.

In 2021, remote imaging spectroscopy, also known as hyperspectral imagery, is experiencing a renaissance. It is graduating from airborne to orbital platforms through NASA s upcoming Earth Mineral Dust Source Investigation (EMIT) onboard the International Space Station and the SBG investigation anticipating launch in 2027-28. This new global coverage stands to revolutionize our understanding of aquatic and terrestrial ecosystems in a changing climate, with first of kind measurements of properties such as tree canopy chemistry, benthic ecosystem composition, and many more. This capability makes it a critical element of NASA s Decadal Observable portfolio. Yet in stark contrast to multiband sensors such as Landsat and MODIS, the community is not yet ready to make use of SBG data; expertise in imaging spectroscopy is confined to a handful of institutions.

The key barrier to entry is the lack of stable, transparent, and high-quality atmospheric correction. Imaging spectrometers measure the incident illumination at the sensor, so investigators must estimate and remove atmospheric effects before measuring the surface. This requires expertise and access to licensed, proprietary radiative transfer model codes. Atmospheric correction is a complex analysis, and failures of atmospheric correction are the primary performance bottleneck in many surface studies. To realize the full potential of SBG, we must provide this talent pool with good quality data.

Our proposed solution is to realize the first fully open source atmospheric correction code, released in 2018 on github.com as ISOFIT (https://github.com/isofit/isofit). This codebase implements transparent, open source radiative transfer and inversion methods. It facilitates operational estimates of surface reflectance from at-sensor radiance. However, it was developed as a research tool with limited testing, examples, and documentation, making it challenging for non-experts to use. Over the course of a three

year project, we will mature ISOFIT from a JPL research tool to a community-led, self-maintaining project that is accessible for casual users. This will lower the barrier to high quality atmospheric correction, enabling imaging spectrometer data analysis by corporations and university laboratories. We envision a future, enabled by global imaging spectrometers like SBG, in which these revolutionary data are as pervasive and accessible as Landsat is today.

The ISOFIT codebase is the best candidate to achieve this vision. It already has considerable support from multiple NASA projects; it has been adopted by the SBG study team for their initial performance modeling, and will also be used by the EMIT mission and AVIRIS airborne projects, NASA s primary orbital and airborne VSWIR imaging spectrometers, for their reflectance data. These projects will continue to support algorithm development and validation, but there is no provision for improving the user experience or providing the documentation and interface development that is critical for broader public use. Consequently, now is the ideal time in the ISOFIT project to grow beyond the science data system. The core algorithms have undergone significant literature review. Efforts now will ensure project longevity, bringing imaging spectroscopy to a broad audience.

Thomas Caswell/Numfocus, INC.
Revamping Matplotlib for Modern Data Structures

Matplotlib is a comprehensive, community-developed, Python library for creating static, animated, and interactive visualizations, used throughout the SMD community. Matplotlib is an open-source tool and has been developed and maintained for the last 17 years primarily through volunteer effort. Like many other open-source tools, it has become the tool of choice for hundreds of thousands of users, putting significant demands on the volunteer community. In order to support these users, develop a vibrant community of users and developers, and plan for future improvements that can meet modern computing challenges, we are asking for modest support in terms of person hours.

Here we propose full-time support for a software developer and partial support for the Matplotlib Project Lead. This will allow us to take on more complex projects than can be implemented by part-time volunteer effort alone while executing critical day-to-day maintenance tasks required to keep the project healthy. We propose to split the effort on this grant equally between the two primary activities: overhauling our internal data representation with a particular focus on supporting physical units, and general library maintenance and community development.

Matplotlib is based on visualizing data stored in Numpy arrays, which is a simple yet flexible data model that has been the underpinning of data visualization for the last few decades. However, modern data structures are much richer; they can carry additional meta-data or describe massive data sets, allowing piece-wise operations on data sets too

large to load into memory. We propose to refactor Matplotlib's internal data representation to exploit these developments.

An immediate motivation for this refactor is to improve Matplotlib's handling of data with physical units attached; currently, physical units are largely but inconsistently supported within the library. This refactor will allow us to consolidate unit handling code and standardize its behavior across the library. The importance of handling physical units in computation is well-recognized in the Scientific Python Ecosystem; core libraries including NumPy (NEP40-43), pandas, and xarray are working to add support for physical units to their foundational data structures. The proposed work will ensure that Matplotlib can take advantage of such efforts, for seamless handling of data with units from computation to visualization. Additionally, this refactor will lay the groundwork for future enhancements such as smart resampling, native consumption of structured data, out-of-memory data, seamless updating, and alternative data sources such as analytic functions or database queries.

To sustain Matplotlib and continue to support the hundreds of thousands of users and hundreds of downstream packages will require continued maintenance of the library and its community. This includes tasks like promptly fixing critical bugs and regressions as they are reported, producing regular releases, answering user questions, facilitating discussions, community management, and general project management. Dedicated full-time developers help ensure that crucial tasks happen in a consistent and timely manner, and provide the roadmaps that will steer the future of the library

Matplotlib is used throughout the SMD community, including flagship missions like the Hubble Space Telescope and the James Webb Space Telescope. Domain-specific libraries, such as AstroPy (astrophysics), sunpy (heliophysics), and Cartopy (earth science), extend Matplotlib to provide specialized plotting. Libraries like Matplotlib were identified for funding in a National Academies of Science report. Given the breadth of Matplotlib usage across the SMD divisions, small investments in Matplotlib will have returns across the entire SMD portfolio.

Scott Henderson/University Of Washington, Seattle
Enhancing analysis of NASA data with the open-source Python XArray Library

The accelerating data deluge from modern sensors presents an unprecedented challenge for all four NASA Science Mission Directorate (SMD) science divisions. SMD collectively stores over 100 Petabytes (PB) of data and estimates generating an additional 100 PB per year within the next five years. Consequently, science today requires software that enables expressive and easily-parallelized workflows on gigabyte to petabyte sized datasets. Xarray is an actively developed open source library that reduces the barriers to analyzing NASA datasets at scale, leading to greater scientific return and faster discoveries.

Xarray s impact on the scientific community is significant, growing steadily since the library s public release in 2014. Xarray provides scientists with a powerful interface for parallelized computation with multi-dimensional raster datasets (e.g. image stacks), which are prevalent today across all scientific domains. The scalability that Xarray unlocks plays an essential role in NASA s transition to hosting large data archives on public cloud computing infrastructure. High-impact scientific studies relying on Xarray s capabilities span many topics including ocean dynamics, glacier dynamics, atmospheric and climatological change, and satellite-derived snow properties. We aim to significantly expand the use of Xarray for research using NASA data and sustain development of the library through specific maintenance and outreach activities:

* Science users of Xarray integrate multiple Python libraries to achieve results, requiring integration and benchmarking tests to ensure error-free, stable, and performant code. We will greatly expand an existing limited benchmarking test suite that focuses on performance of Xarray for common operations on NASA data.

* Domain-specific extensions of Xarray are necessary to meet the needs of unique NASA remote sensing data (imagery swaths, laser altimeters, etc.). We will decouple and expand upon geoscience-specific functionality such as Coordinate Reference System (CRS) management, reprojection, and clipping in the RioXarray extension library.

* There is a critical need for documentation of complete workflows that illustrate both core functionality and domain-specific applications of Xarray. We will create novel, interactive documentation for scientific workflows that focus on scalable distributed computing for domain-specific analysis tasks with NASA data.

* The paradigm shift towards cloud-computing and community-developed software is a major change for scientists accustomed to operating on small local files with custom code. We will address this socio-technical challenge by hosting public "Xarray for NASA data" monthly virtual office hours along with a regular online Xarray tutorial series.

Ultimately the success of this project will be measured by increasing adoption of Xarray among scientists using NASA data for research. This is a transitional moment in which the scientific community can either be stymied by data management or empowered with enhanced open-source software tools such as Xarray. Our tasks will enable the SMD community to fully unlock Xarray s potential for efficiently exploring petabyte-scale NASA data, accelerating scientific discoveries in this age of cloud-computing and big data.

Steven Hughes/NASA Goddard Space Flight Center
Improved Sustainment, Alignment, and Impact of Open Source Flight Dynamics Tools for SMD Science Objectives

We propose to improve the utility and sustainability of several existing and active high-value, open-source tools and libraries that have made significant impacts to the SMD science community. These tools -- the General Mission Analysis Tool (GMAT), the Collocation Stand Alone Library and Toolkit (CSALT), the Evolutionary Mission Trajectory Generator (EMTG), the Orbit Determination Toolbox (ODTBX), and Kamodo, a Space Weather Modelling tool which is part of the Community Coordinated Modeling Center (CCMC) -- are software packages developed and maintained at Goddard Space Flight Center (GSFC) that have and continue to make vital contributions to SMD via support of numerous SMD projects. Examples include OSIRIS-REx, LRO, Fermi, Lucy, TESS, ACE, Wind, SDO, SOHO, and LISA. CSALT and EMTG are trajectory optimization systems used to design and fly SMD missions. GMAT is used in both trajectory optimization and orbit determination roles and interfaces with CSALT. ODTBX is an orbit determination library used for navigation analysis for future SMD missions. Kamado is a library for accessing space weather models maintained by the Community Coordinate Modelling Center (CCMC)

To ensure this effort results in both improved sustainability and utility, we will improve both developer and user documentation and refactor components of GMAT, CSALT, and EMTG to eliminate duplication and promote standardization and sharing of models and algorithms between these tools. To dramatically enhance utility, we will focus refactorization and improvements on challenging new use cases to ensure new capability, including small body swarm optimization and rapid, onboard mission re-planning for opportunistic science. To improve utility and synergy between engineering tools used to support SMD and science models we will also integrating with navigation software with CCMC space weather models to improve modeling and prediction of orbits during periods of high solar activity.

This effort is directly relevant to SMD's science strategic plan elements 1.1, 1.3, 2.2, 3.1, and 4.1. Additionally, the tools in this effort directly address "TIER I: Unique Capabilities Essential to the Science Mission Directorate (SMD) identified in SMD's "GSFC Long-term Capability/Capacity Strategy". The software in scope in this effort is open source and is routinely used to support SMD proposals (including decadal studies) and projects across the mission lifecycle and across divisions. Benefits of this effort include reduced cost and risk due to refactored, unified components and algorithms for ease of maintenance, re-use, and sharing, as well as improved documentation. Additional benefits include increased capability and impact by focusing refactoring and extensions on challenging use cases and improved synergy by interfacing engineering and science tools that support SMD.

Dharhas Pothina/Quansight, LLC
Reinforcing the Foundations of Scientific Python

Large parts of the scientific community use and rely on the scientific Python ecosystem. Array and dataframe data structures, provided by NumPy and Pandas, and general scientific and machine learning algorithms, provided by SciPy and Scikit-learn, lie at the heart of that ecosystem. We will address key challenges in the maintenance of these four projects, as well as implement technical improvements with a focus on computational performance of and tighter integration between them.

Maintaining continuous integration (CI) and extending it to, e.g. new Python versions and more hardware platforms, is one of the most time-consuming activities for each project. We intend to have an infrastructure engineer work on this, sharing expertise between projects. NumPy, Pandas, SciPy and Scikit-learn all make regular releases. Providing high-quality packages for Linux, Windows and macOS - and more recently also ARM64 and PowerPC - is critical for end users. Automating this process and integration testing between the different projects will save maintainers a lot of time, prevent regressions, and allow more frequent releases to be made easily.

The volume of both open issues and proposed contributions for each project is large - between 1300 (SciPy) and 3400 (Pandas) issues, and between 180 (Pandas) and 740 (Scikit-learn) pull requests. The bulk of that work is volunteer effort. We will perform targeted issue triaging and code review to be able to prioritize the most important bug fixes and contributions, and get them integrated into the projects.

Computational speed improvements, lower memory usage and the ability to parallelize algorithms are beneficial to most use cases, and were at or near the top of desired enhancements in two recent end user surveys. Therefore performance will be a main theme of the technical work we propose. Our key objectives are:
Better cross-project integration, in particular via data-type compatibility (e.g. reimplement Pandas dtypes on top of NumPy's new dtype extension mechanism) and zero-copy protocols (e.g., enable scikit-learn to consume Pandas dataframes in more situations).

Adopt and implement a uniform API per project for parallel execution. This will build on the `n_jobs` and `workers` patterns in Scikit-learn and SciPy respectively.

Extend the use of accelerator technologies. Pandas recently started using Numba as an optional dependency which we will extend to a larger part of the API; all projects have scope for significant performance improvements via heavier use of Cython.

More significant performance gains, and the ability to scale up to larger data sets, will be unlocked by enabling the use of GPU and distributed arrays in SciPy and Scikit-learn. Initial experimental work has proven that this is feasible, via use of NumPy array protocols plus CuPy for GPU support and Dask for distributed support. We will solve the

remaining technical hurdles to ensure that users are able to smoothly use CuPy and Dask in at least one complete SciPy and one Scikit-learn submodule.

The proposed work will increase both the health and the capabilities of these four projects, which is not only beneficial to the tens of millions of users they have, but also to the wider Python ecosystem - including space science specific libraries like AstroPy and SunPy - and the NASA missions that rely on scientific computing with Python.

Eleonora Presani/Cornell University
Enhancing arXiv interoperability: steps towards interdisciplinary research

arXiv is the largest e-print server, with about 1.7 million articles available and 175,000 new submissions every year, with a steady growth of almost 20% YoY. arXiv serves a global and diverse community, with authors and readers distributed worldwide. While arXiv started in physics, and rapidly expanded to astrophysics, mathematics, statistics and computer science, it is now growing in disciplines outside the original scope of the repository: the addition of quantitative biology, quantitative finance, economics and engineering are examples. This wealth of academic literature is a valuable asset for third parties who can use it to improve discoverability in different disciplines, use Machine Learning tools to extract useful information and to provide aggregated data. Examples of this interoperability are the arXiv dataset made available in kaggle.com (https://www.kaggle.com/Cornell-University/arxiv); double linking of articles and underlying software with Papers With Code, and enhanced search and discovery capabilities with NASA's Astrophysical Data System (NASA-ADS). arXiv provides critical infrastructure services to NASA-ADS.

arXiv seeks to enhance and sustain the newly launched framework arXiv Labs , that allows third parties to develop software that is either based on arXiv data or enhances arXiv functionality. This would be done in coordination with a re-platforming effort to bring arXiv codebase to a more modern architecture.
This would require:
* Implementation of the new arXiv architecture based on micro services that allow for an enhanced interoperability.
* Ensure all new code is Open Source and available on GitHub (all code of the new architecture is available at: https://github.com/arXiv and currently represents about 40% of the total codebase).
* Development of an enriched and machine readable arXiv record, including relational data model to link to entities such as authors and institutions, as well as additional media such as data, software, and notebooks.
* Collaborate with organizations such as ORCID, INSPIRE-HEP and NASA-ADS to share and improve author and record metadata, with the purpose of improving author discovery and attribution.
* Consolidate relationships with existing partners while extending new relationships, especially with international and diverse stakeholders.

* Develop compatibility with additional submission formats to cater needs of new communities such as planetary sciences, meteorology, oceanography, ecology, and additional fields of engineering.
* Develop necessary APIs for sharing data with third parties.
* Improve accessibility and mobile experience of the site, including support for HTML5 and PDF/A.

All of this effort would consolidate arXiv as the reference platform for sharing preprint research, while expanding the scope and the technological services it provides.

Albert Shih/NASA Goddard Space Flight Center
Strengthening the Foundations of the SunPy Ecosystem

The SunPy project facilitates and promotes the use and development of community-led, free, and open-source data analysis software for solar physics based on the scientific Python environment (SunPy Community et al. 2020). To date, the SunPy project has largely relied on unpaid, volunteer efforts from early-career scientists, especially for developing SunPy s core library (released under the BSD-2-clause license). Even so, a wide variety of science projects already depend on the SunPy ecosystem for data pipelines or analysis workflows. To continue the successful expansion of this ecosystem as a broad community resource, we propose three targeted efforts to strengthen its foundations, ensuring robust support for the analysis of complex data from next-generation missions and simulations:

* Improve the technical infrastructure for maintaining the core library and for nurturing its ecosystem of affiliated packages
* Augment science-enabling functionality in key areas relevant for newly available solar-physics data sets
* Provide training and outreach to the solar-physics community on how to most effectively use and contribute to SunPy

Improving the technical infrastructure:
SunPy has a rich and growing ecosystem of affiliated packages that provide interoperable instrument-specific and science-task-specific capabilities that are enabled by and supplement the core library. We will improve the long-term sustainability of the SunPy ecosystem by improving the development infrastructure (e.g., task automation) and associated documentation.

Augmenting science-enabling functionality:
We will augment the tools currently available in SunPy s ecosystem to meet new challenges and evolving methodologies in analyzing data from the Heliophysics System Observatory. In particular, we will improve the frameworks for efficiently working with large data sets, such as those produced by the Solar Dynamics Observatory (SDO) and the Daniel K. Inouye Solar Telescope (DKIST), and we will improve support for working

with the multi-point, multi-instrument observations being pioneered by Parker Solar Probe (PSP) and Solar Orbiter in combination with Earth-based observatories (both in space and on the ground).

Providing training and outreach:
In order to ensure a broad and growing user base and developer community for SunPy, we will produce and deliver a variety of resources to train users in effectively using SunPy and the fuller Python ecosystem in their heliophysics workflows. This will include online tutorials, focused workshops, and materials targeted to imparting and instilling efficient programming methodologies to graduate students and other early-career researchers. We will direct specific outreach efforts to instrument teams and package developers in order to encourage the ongoing incorporation of new functionality into the SunPy ecosystem (e.g., as an affiliated package).

Sustaining/improving the health of the SunPy ecosystem supports SMD science goals and is directly aligned with recommendations in the latest guiding documents for NASA (e.g., the Midterm Assessment for the most recent heliophysics Decadal Survey).

Erik Tollerud/Numfocus, INC.
Sustaining the Astropy Project

The Python language has seen increasingly wide use in astronomy, including astrophysics, cosmology, and planetary, solar, and space sciences. This has both led to and been influenced by the Astropy Project. Astropy was founded in 2011 with two major objectives: to develop a single core Python package to provide the foundational tools for astronomy research; and to build an ecosystem of usable, interoperable, and collaborative astronomy Python packages. With these goals, it has developed into a true community effort, with participants from a wide range of backgrounds, that is not the product of one particular institution, group, or mission. It has over 20,000 dependent repositories on GitHub alone; at least 5,000 peer-reviewed publications have used it; and it is critical for a range of NASA missions including Chandra, Hubble, and JWST. However, supporting such a wide user base through community effort alone presents a number of sustainability challenges. In particular, most development is done by a few people, infrastructure requires ongoing maintenance, and the number of active core developers is decreasing.

This proposal aims to address these concerns and improve the robustness of the Astropy Project s infrastructure by funding a set of tasks identified by the Project s contributors as particularly difficult to find volunteer effort to pursue. This includes a mix of Project-wide infrastructure improvements, targeted work on specific areas of the code that are in need of support, enhancements to the Astropy education materials to encourage community engagement and with the hope of fostering more contributions, and better support for the affiliated packages, independent Python packages that collaborate with Astropy and are peer-reviewed for coding, documentation and testing standards. We

propose to continue using our established funding model of primarily supporting contributors as part-time contractors, an approach that has been shown to successfully fund work by both existing and new contributors. In short, this proposal aims to provide support for the Astropy ecosystem and ensure that it is healthy and able to continue fulfilling its vision to provide a core package and an associated collaborative ecosystem. While the Project did not require dedicated funding to earn its role in the community, it does now need funding to continue to provide the production-level code and educational materials that the community has come to not just expect from us, but to rely on us to provide.
