

Email address

mhaberla@calpoly.edu

Name of Submitter

Matt Haberland

Project Submitting this Proposal

SciPy

Is your project Affiliated or Sponsored?

Affiliated

Proposal Title

SciPy Development Documentation Overhaul

Two Sentence Summary of Proposal

The SciPy library contributes to the work of scientists and engineers worldwide, but for many of its users, learning to contribute back to SciPy can be daunting. I seek to make the SciPy development community more inclusive by improving its development documentation through revision of written instructions and the addition of video tutorials.

Description of Proposal (<750 words, < 4500 chars)

Python package managers make the installation of SciPy for end users as simple as a single terminal command that is common across most platforms. For those who wish to modify SciPy code or contribute to SciPy, however, the process is substantially more challenging: they must use `git` to retrieve the SciPy source code; compile the Fortran, C, and C++ code locally rather than relying on prebuilt binaries; and (usually) install this development version of SciPy in a separate virtual environment. Writing unit tests requires working knowledge of the `pytest` framework, editing and previewing the documentation means writing `reStructuredText` and building it with `Sphinx`, and creating benchmarks takes some understanding of `AirSpeed Velocity`. As SciPy is a collection of numerical routines that support diverse scientific and engineering applications, its would-be developers are not necessarily familiar with these processes already. Some documentation is available from `scipy.org`, but it is spread across half a dozen URLs without conveniently organized links for navigation, and some of the key documentation is written in the context of `NumPy` instead of SciPy. I propose to streamline the process of learning to contribute to SciPy by completing the following tasks.

1) Provide a more convenient means of navigating SciPy development documentation

The SciPy development document “Contributing to SciPy” [1] is a relatively comprehensive starting point for SciPy developers. It contains a narrative describing the process of developing code for SciPy that includes links to more detailed documentation on specific topics. However, the links are not in the order they are needed by a first-time developer, and the links are contained only within the narrative, making

it difficult for new developers to find what they need when they need it. Even as a core developer, I get lost trying to find reference documentation, which is one problem that spurred this proposal. As detailed in GitHub Issue #9951 [2], I will add to [1] a concise list of topics and links ordered according to the needs of a new contributor.

2) Add missing documentation

Docstring fixes would be a gentle way for new contributors to get started - if they only knew how to check what their edits would look like when built by Sphinx (Issue #8966 [3]). AirSpeed Velocity is an immensely valuable tool for tracking the performance of SciPy functionality over time, but documentation of its use with SciPy is hidden in a `readme.rst` file [4]. Information about adding Cython code would be especially useful to scientists and researchers who need to integrate high-performance compiled code with SciPy, but the only source of information is comments in GitHub Issue #8575 [5]. I will add sections about these features to the official SciPy development documentation.

3) Edit existing NumPy tutorial for git to support SciPy developers

The link provided in SciPy's documentation [1] for information about git [6] is written in the context of NumPy; it does not mention SciPy at all. Adapting this information to the context of SciPy is trivial for someone with experience, but the text should be modified explicitly to aid beginners.

4) Create videos to demonstrate important procedures

I have already created videos [7] showing how to set up a SciPy development environment [8, 9] and a typical workflow for fixing a bug and submitting a PR [10]. I aim to make videos about making and previewing changes to documentation with Sphinx, creating and running benchmarks with AirSpeed Velocity, and adding Cython code to SciPy.

References:

[1] "Contributing to SciPy." *SciPy v1.3.0.dev0+4539323 Reference Guide*, <http://scipy.github.io/devdocs/hacking.html>.

[2] @mdhaber. "SciPy Development Documentation #9951." *GitHub*, 16 March 2019, <https://github.com/scipy/scipy/issues/9951>.

[3] @rgommers. "improve docs on building docs #8966." *GitHub*, 23 June 2018, <https://github.com/scipy/scipy/issues/8966>.

[4] "SciPy benchmarks." *scipy/README.rst at master – scipy/scipy*, 17 November 2018, <https://github.com/scipy/scipy/blob/master/benchmarks/README.rst>.

[5] @sschnug. "Documentation: how to contribute mixed-language code (build; deployment)?" *GitHub*, 18 March 2018, <https://github.com/scipy/scipy/issues/8575>.

[6] "Working with NumPy source code." *Numpy v1.16 Manual*, 31 January 2019, <https://docs.scipy.org/doc/numpy/dev/gitwash/>.

[7] @mdhaber. "SciPy Dev Install Videos." *GitHub*, 1 February 2018, <https://github.com/scipy/scipy/issues/8353>.

[8] Haberland, Matt. "Anaconda SciPy Dev: Part I (macOS)." *YouTube*, 4 February 2018, <https://www.youtube.com/watch?v=1rPOSNdOULI>.

[9] Haberland, Matt. "Anaconda SciPy Dev: Part II (macOS)." *YouTube*, 6 February 2018, <https://www.youtube.com/watch?v=Faz29u5xIZc>.

[10] Haberland, Matt. "SciPy Development Workflow." *YouTube*, 3 April 2018, <https://www.youtube.com/watch?v=HgU01gJbzMY>.

Benefit to Project/Community (<400 words, < 2500 chars)

By improving the documentation about modifying and contributing to SciPy, this work will make SciPy development work more inclusive. Specifically, making the development documentation accessible to researchers and engineers with diverse backgrounds will enable them to inspect and modify the code to meet their needs. This will broaden the base of potential SciPy code contributors and reviewers, helping SciPy to better meet the numerical computing needs of the broader scientific and research community.

Amount Requested

\$4,274

Brief Budget Justification - How will the money be spent?

\$2,866 (\$2,605 salary; \$261 fringe benefits/payroll taxes) will compensate the proposer for 39 hours of overload/summer work: 2 hours for task 1, 18 hours for task 2 (6 hours per topic), 1 hour for task 3, and 18 hours for task 4 (6 hours per video).

\$220 will cover 10 hours of a graduate student's time (\$200 wages, \$20 fringe benefits/employer payroll taxes). A graduate student who is already familiar with numerical computing in Python will use the revised documentation to learn to make their first contribution to SciPy. Afterwards, they will submit feedback on the development documentation, which will be used to make final improvements.

The \$1,188 is for Cal Poly recovery of indirect costs.

The salary and wage rates are based on the California Polytechnic State University (CPSU) and Cal Poly Corporation (CPC), jointly Cal Poly, established salary and wage rates paid during the 2018-2019 Fiscal year (July 1 – June 30). Faculty duties at CPSU consist of a full fifteen units each of three Academic quarters per nine month Academic year. The salary and wage rates for faculty and non-student staff includes a projected 4.5% salary increase per year. The rates shown are for budgetary purposes; the rates in effect at the time the work is performed will be charged to the project.

Benefits for CPSU Faculty summer and overload work include FICA, SUI and Workers Compensation and are calculated at the DHHS pooled rate of 10%. CPC graduate student fringe benefits include SUI and Worker's Compensation which would result in the pooled rate of 4.5%. CPC graduate students convert to intermittent employees if the graduate student is not fully enrolled when the work is performed, resulting in the addition of FICA to fringe benefits and the current intermittent fringe benefit rate of 10%. Cal Poly elects to budget graduate student fringe benefits at the 10% intermittent rate, assuming that the graduate students will not be fully enrolled. It is not feasible to assess enrollment status at the time of proposal submission.

Cal Poly's federally negotiated indirect rate is 38.5% of modified total direct costs, effective July 1, 2018. Modified total direct costs exclude equipment, capital expenditures, charges for patient care, tuition remission, rental costs of off-site facilities, scholarships, and fellowships, participant support costs, and the portion of each subaward in excess of \$25,000.

Timeline of Deliverables

4/19/2019 – Acceptance notification

Late April – Development documentation navigation improvement, git documentation edits

Early May – Missing development documentation drafts

Late May – Development video drafts

Early June – Graduate student learns to make SciPy contribution using revised documentation, graduate student and community feedback

Late June – Final revisions implemented, changes merged

Project Team

Matt Haberland

I agree to submit a grant report-back if my proposal is selected for funding.

I agree.