Open Code Policy for NASA Space Science: A perspective from NASA-supported ocean modeling and ocean data analysis

Sarah Gille[1], Ryan Abernathey[2], Teresa Chereskin[1], Bruce Cornuelle[1], Patrick Heimbach[3], Matthew Mazloff[1] , Dimitris Menemenlis[4], Cesar Rocha[1], Saulo Soares[5], Maike Sonnewald[6], Bia Villas Boas[1], Jinbo Wang[4]

[1]Scripps Institution of Oceanography, University of California San Diego
[2]Lamont-Doherty Earth Observatory, Columbia University
[3]University of Texas at Austin
[4]Jet Propulsion Laboratory
[5]University of Hawaii
[6]Massachusetts Institute of Technology

NASA's science mission directorate has supported a broad range of oceanographic research, including ocean model development as well as development of analysis tools to analyze oceanographic data products (including both satellite observations and model output). Many of these products have been released as open source software, or have been shared via version control hosting services, such as GitHub and BitBucket. The broad consensus of the co-authors of this white paper is that open release of software has significantly sped scientific progress, despite lack of incentive from research institutions and funding agencies.. We note that the topics that we are considering are not subject to ITAR restrictions, so our comments pertain to the value to the community of open code.

Why open source?

While software released within the oceanographic community does not universally meet the high standards expected of "open code", the experiences within our community have highlighted the value of open sharing of source code, which we summarize here:

- *Scientific community support through open source*. By releasing source code, the open source model MITgcm supports the global scientific community, fostering advances throughout the global academic community, including supporting capacity building.
- *Accelerating science through open source*. Sharing source code allows other groups to build off the best ideas from across the scientific community. Two recent examples include the Geophysical Fluid Dynamics Laboratories new ocean model known as MOM6, which uses a new sea ice dynamics code that is based on MITgcm's implementation; and the European NEMO model, which uses sub-ice shelf cavity circulation capabilities and thermodynamic melt rate parameterization, also based on MITgcm code. New users employ available capability in new ways not foreseen by original developers, and ideally improve the code itself.

- *Reproducibility*. The ECCO (version 4) state estimate is the only reanalysis product that can be reproduced by users through provision of source code, required configuration files (compile-time and run-time), and in-out fields. Reproducibility has emerged as a critical issue across all sciences, and increasingly we expect that code reproducibility will become a priority. New users and new applications are invaluable for helping to uncover bugs, code limitations, etc. Ultimately this results in more robust code.
- *Standardization*. Algorithms or software packages that are available open source can define a community standard. For example, the MITgcm has adopted a vertical mixing scheme known as KPP that was originally developed at the National Center for Atmospheric Research and that is now widely used in ocean models. The MITgcm is now considering adopting a new community-developed mixing scheme (CVMix), again benefiting from open release of code. Moving towards interoperability will be helpful, and adopting community wide standards is an important first step.
- *Better code*. An individual investigator's decision to share their code serve as an impetus to develop better tests and more careful documentation, ultimately leading to improved reproducibility and code that is more easily repurposed for other applications.
- *Positive feedback and more citations*. Open source code is a benefit to the entire community, and people who benefit from the code are grateful and often provide positive feedback, as well as finding reason to cite the relevant publications.

Challenges

The oceanographic community has also highlight some challenges to releasing "open code," particularly for smaller projects.

- *LDocumentation.* Releasing code (e.g. for specific analysis activities) is easier if the code developer can explain the code directly to users. For code that will necessarily have a limited user base, resource limitations can make it difficult to justify developing extensive tutorials or documentation without incentive from funding bodies.
- *Maintenance of active code*. While code used for a single analysis can easily be archived for future reference, code that is actively being used by multiple research groups requires systematic maintenance, which has overhead associated with it. In the absence of developer community, a code custodian should be stably funded to screen and quality-control the community's contribution to the open source.

  Perceived lack of structure for acknowledging contributions to open source code. NSF requirements now ask proposal submitters to identify prior products, and this can be used to highlight software or data that have been released with a digital object identifier (doi). However, NASA has not adopted such a strategy, and that can leave code developers uncertain whether their program managers will have a mechanism to ack

- nowledge their contributions for the benefit of the community.
- *Modernising development.* Having a software development infrastructure centered around open source best practices is not yet standard. Creating an open and collaborative development culture, the adoption of good coding practices and contributing code appropriately would naturally follow. This is not possible unless open source code sharing is the norm, and an open source language is used for development.

Examples

For context we first summarize examples of code that has been released in open source format.

Ocean Model. The MITgcm is an ocean general circulation model, the development of which has been supported in part by NASA. It is used for the Estimating the Circulation and Climate of the Ocean (ECCO) and ECCO2 projects, which have also benefited from NASA support. The MITgcm is released as open source code using the Concurrent Versions System (CVS), which is supported by the Free Software Foundation.

Analysis and post-processing codes. In recent years software has been shared via GitHub to facilitate analysis of output from the MITgcm. Here we give two examples. Octopus, developed by Jinbo Wang under NSF funding, is a code to track particle motions in the ocean. GitHub releases by Cesar Rocha provide spectral analysis tools for analyzing output from the MITgcm providing a means for researchers to replicate calculations carried out in published papers.

Collaboration and transparency. Using open source software that is well documented and tractible significantly lowers the bar for interdisciplinary work, and the associated transparency facilitates learning across disciplines and for students. Making shared source code open source has the potential to facilitate such transparency which would greatly benefit the community. Encouraging good coding practices in students through examples such as those listed are key, and further adoption of modern development methods is central to continuing this trend.

Our collective experiences have highlighted the overwhelming benefits of releasing open source code. Releasing useful open source code that is well documented and compliant with best-practices, however, is time consuming. While users of open source code are often immensely grateful for resources that have been shared within the community, code developers can nonetheless be left uncertain whether their efforts will be fully acknowledged. The challenges in supporting the infrastructure to release open source code can ben addressed in part through cultural changes to formally recognize the contributions associated with releasing code, and partly through expanding development tools (such as GitHub, ReadtheDocs, or Zenodo) to help coordinate code release, but ultimately may also require financial investment.