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] , 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 models for the purpose of synthesizing and interpreting its diverse observational data streams and tools to analyze satellite observations, model output, and model-data syntheses. Many of these products have been released as open source software, or have been shared publicly 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 oceanographic topics that we study are not subject to export control 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 public sharing of source code, which we summarize here:

- *Scientific community support through open source*. Releasing source code, for example the open source MIT general circulation model (MITgcm), supports the global scientific community, fostering advances and capacity building throughout the global academic community.
- *Accelerating science through open source*. Sharing source code allows other groups to build off the best ideas from across the scientific community. Recent examples include the Geophysical Fluid Dynamics Laboratory's (GFDL) new ocean model, MOM6, which uses a new sea ice dynamics code that is based on MITgcm's C-grid implementation of an Elastic-Viscous-Plastic (EVP) rheology (Losch et al. 2010; A. Adcroft, pers. comm. 2017); and the European NEMO model, which uses sub-ice shelf cavity circulation capabilities and thermodynamic melt rate

parameterization, also based on MITgcm code (Losch 2008; Mathiot et al. 2017). Meanwhile, the MITgcm has recently adopted a biogeochemical package that was originally developed for GFDL's MOM4 (Verdy and Mazloff, 2017). New users employ available capabilities in new ways not foreseen by original developers, and ideally improve the code itself. Similarly, just as sharing model routines advances modeling efforts, sharing analysis code can reduce duplication of effort and enable researchers to build directly on others' work, speeding the progress of new discoveries.

- *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 (Forget et al. 2015). Reproducibility has emerged as a critical issue across the 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 employs the KPP vertical mixing scheme, which was originally developed at the National Center for Atmospheric Research and is now widely used in ocean models. MITgcm developers are considering implementing a new community-developed mixing scheme (CVMix), again making use of open-source code (http://cvmix.github.io). 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 serves 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.
- *Collaborations and transparency*. Using open source software that is well documented, with clear traceability, facilitates interdisciplinary work. This supports learning by students and across disciplines, and it also allows non-specialists to delve into the research process. Making source code public enhances transparency in science, benefits the community, and provides information that can increase public trust.
- *Positive feedback and more citations*. Scientists who benefit from open source code are grateful and often provide positive feedback. Building these relationships nurtures a collaborative environment, leading to better exposure and potentially even publication citations.

Challenges

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

- *Lack of structure for acknowledging contributions to open source code*. NSF requires principal investigators to identify products from prior research, and this can be used to highlight software or data that have been released with a digital object identifier (doi). NASA, however, has not adopted such a strategy, and that can leave code developers uncertain whether their program managers will have a mechanism to acknowledge their contributions for the benefit of the community.
- *Documentation.* 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.
- *Support for maintaining 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 a developer community, a code custodian should be stably funded to screen and quality-control the community's contributions to the open source.
- *Modernizing 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 summarize several 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 a central component in 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. Because of the open source approach, it has gained worldwide recognition with a wide-spread user base. Acting as a pool of knowledge not only for the specific model but for numerical modeling in general, the MITgcm repository clearly demonstrates the benefit of open source.

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 three examples. (1) Octopus, developed by Jinbo Wang under NSF funding, is a code to track particle motions in the ocean. The package has been used by graduate students at Scripps Institution of Oceanography as a learning tool to explore

Lagrangian methods and used as a research tool in several studies (e.g. Tamsitt et al, 2017). (2) 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. The open-source Github code has standardized the methodology used by Rocha et al (2016) and energized discussions in the community (3) The community-developed python packages xmitgcm and xgcm are increasingly being adopted for MITgcm post-processing and analysis (Abernathey et al., 2017).

Discussion and recommendations

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 be left uncertain whether their efforts will be fully acknowledged or benefit their career advancement. The challenges in supporting the infrastructure to release open source code can be 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.

A direct way to contribute towards these cultural changes is to raise awareness about open source software among NASA-funded researchers. For example, when applicable applications for NASA grants could provide an opportunity to comment on "benefits to the open source community" or to provide a "Software Management Plan" (similar to the now standard “Data Management Plan”), and these points should be taken into account by review panels. In this context, new community-driven initiatives for open-source scientific software such as the Pangeo Project may be valuable partners for NASA in establishing best practices around open source software.

Expanding the formal expectations for open code will require educating the next generation of scientists, as well as training established scientists in modern software development tools. Training can perhaps build off of existing lessons developed by organizations such as Software Carpentry, though field-specific training is also likely to be important.

Finally, we note that several NASA projects and scientists already rely on open-source code such as the scientific python packages Numpy, Scipy and Matplotlib. NASA should consider supporting the maintenance and development of this valuable software via dedicated developer time and contributions to the non-profit NumFocus Foundation.

References

Abernathey, R., A. Cimatoribus, L. Brannigan, and G. Sérazin, 2017. xgcm/xmitgcm: v0.2.1 (Version v0.2.1). Zenodo. , May 31, doi:10.5281/zenodo.801476

Forget, G., J.-M. Campin, P. Heimbach, C. N. Hill, R. M. Ponte, and C. Wunsch, 2015. ECCO version 4: an integrated framework for non-linear inverse modeling and global ocean state estimation. *Geoscientific Model Development*, **8**(10), 3071–3104, doi:10.5194/gmd-8-3071-2015.

Losch, M., 2008. Modeling ice shelf cavities in a z-coordinate ocean general circulation model. *Journal of Geophysical Research*, **113**(C8), doi:10.1029/2007JC004368.

Losch, M., D. Menemenlis, J-M. Campin, P. Heimbach, and C. Hill, 2010. On the formulation of sea-ice models. Part 1: Effects of different solver implementations and parameterizations. *Ocean Modelling*, **33**(1-2), 129–144, doi:10.1016/j.ocemod.2009.12.008.

Mathiot, P., A. Jenkins, C. Harris, and G. Madec, 2017. Explicit representation and parametrised impacts of under ice shelf seas in the z* coordinate ocean model NEMO 3.6, *Geosci. Model Dev.*, **10**, 2849-2874, doi:10.5194/gmd-10-2849-2017.

Rocha, C., T. K. Chereskin, S. T. Gille, and D. Menemenlis, 2016. Mesoscale to submesoscale wavenumber spectra in Drake Passage, *J. Phys. Oceanogr.*, **46**, 601-620.

Tamsitt, V., H. Drake, A. K. Morrison, L. D. Talley, C. O. Dufour, A. R. Gray, S. M Griffies, M. R. Mazloff, J. L. Sarmiento, J. Wang, and W. Weijer, 2017. Spiraling up: pathways of global deep waters to the surface of the Southern Ocean. *Nature Communications*, **8,** doi:10.1038/s41467-017-00197-0.

Verdy, A, and M. R. Mazloff, 2017. A data assimilating model for estimating Southern Ocean biogeochemistry. *Journal of Geophysical Research*. **122**, doi:10.1002/2016JC012650