What does scientific reproducibility and productivity really mean? The dangers and difficulties of a blanket open code policy

John T. Emmert
Space Science Division, U.S. Naval Research Laboratory, Washington, DC
202-767-0467
john.emmert@nrl.navy.mil

Contributing Authors:

Name	Position	Affiliation	Stake (all of the authors primarily study Earth's upper atmosphere and ionosphere)
John T. Emmert	Research Physicist	U.S. Naval Research Lab (NRL)	Data and model analyst; producer of empirical models and reduced data sets
Jens Oberheide	Professor	Clemson University	Data and model analyst; producer of empirical models
Douglas P. Drob	Research Physicist	NRL	Data and model analyst; producer of empirical models
McArthur Jones, Jr.	Research Physicist	NRL	Model and data analyst
Fabrizio Sassi	Research Physicist	NRL	Physics-based model developer; data and model analyst
David E. Siskind	Research Physicist	NRL	Data and model analyst
Kate A. Zawdie	Research Physicist	NRL	Physics-based model developer; data and model analyst

The positions, experiences, and viewpoints expressed below, and the endorsements of the individuals listed in the appendix, are those of the authors and supporters as scientists in the space research community, and are not the official positions of their employing institutions.

Summary: A blanket open code requirement (encompassing data analysis programs written to obtain particular scientific results) would be detrimental to both science quality and productivity, without enhancing scientific reproducibility.

Introduction

In this white paper, we primarily address suggested topic 1 in the call for white papers from the "Best Practices for a Future Open Code Policy for NASA Space Science" Committee: What positive and negative impacts would arise for you, your workplace, your NASA-funded research, science in general, education, commerce, society, and so on, if all future NASA-funded science code were required to be open source? We also address topic 8: Over the long run, what would be the impact on the quality and reproducibility of research if NASA required all NASA-funded, peer-reviewed science papers to include an electronic compendium of (or pointers to) the source codes, inputs, and outputs that produced each scientific claim in the paper?

There is movement among some scientific communities and journals toward full traceability and provenance of published scientific results, including the acquisition and processing of data, model codes, data analysis codes, and final quantitative results (see, e.g., the commentary by *Gil et al.* [2016]). The American Geophysical Union (AGU) has a publications policy that "all data necessary to understand, evaluate, replicate, and build upon the reported research must be made available and accessible whenever possible" (https://publications.agu.org/author-resource-center/publication-policies/data-policy/), where "data" includes "new code/computer software used to generate results or analyses reported in the paper". However, the extent of this requirement is still being debated in the AGU community, including in the Space Physics section that publishes a large amount of research in AGU's Journal of Geophysical Research – Space Physics.

Advocates of open code requirements appear to be motivated by two perceived benefits: 1) facilitation of scientific reproducibility (a key tenet of the scientific method), and 2) increased scientific productivity, via reduction of repetition of code development. Based on our experiences as modelers and analysts in the aeronomy community, we assert that there is no actual benefit for reproducibility, and that productivity is enhanced only under narrowly defined conditions. By contrast, we envision numerous scenarios in which open code will hinder true independent reproducibility and degrade productivity. Therefore, we do not support the establishment of a blanket open code policy by NASA.

Reproducibility

The benefit of open source code to scientific reproducibility is illusory. At best, the re-application of source code by another researcher only demonstrates computational reproducibility on a different computer system, not scientific reproducibility. In order to truly confirm a scientific result, the *method* must be applied *independently*. Thus, the relevant responsibility of a scientist is to describe the methodology, including data processing algorithms and model parameterizations, in sufficient detail that another researcher can carry out the same experiment. This distinction between algorithms and the computer codes written to implement them is also important because codes are specific to the computer language they are written in, whereas an algorithm is a more universal description of the procedure. An even stronger type of scientific reproducibility, which also does not benefit from open code, is the confirmation of a result by a different methodology. In fact, as we discuss below, this most stringent type of reproducibility may be hindered by open code.

An analogy can be drawn with hardware instruments that make measurements. Science norms require that the builder of an instrument describe (e.g., in a report on the data it produces) the design and construction of the instrument in sufficient detail that another experimentalist could build it. However, the scientist is not required to make available the actual hardware used to implement the design. Similarly, an analyst of data or model output should not be required to make available the source code used to implement his or her analysis algorithm.

Implementation of a blanket open source policy would have dangerous side effects that may actually impede scientific reproducibility. In our experience as data and model analysts, it takes substantial time and effort to decipher and understand analysis codes written by another scientist, even one who works closely with us. Each scientist has a different programming style, and that kind of diversity is a good thing that fosters the long-term robustness of the community's output. Making research-grade analysis codes open source would engender frequent misapplication and misinterpretation of the codes, resulting in false claims of errors or bugs. Even if an actual bug is detected, it may not impact the results but could still be used by the unscrupulous to discredit the results. By diverting time and energy away from the development of truly independent approaches of hypothesis testing in favor of hunting down bugs in other people's programs, open code could well hinder reproducibility.

Another difficulty with a blanket open code policy is that source code is a form of intellectual property, which is retainable under federal law by non-profit and small business grantees. For example, some of us are empirical modelers who make freely available the computer code (properly documented) needed to evaluate a model for user-specified conditions [e.g., *Picone et al.*, 2002; *Drob et al.*, 2015]. However, we view as proprietary the source code used to process the underlying data and estimate the model parameters. In our papers, we describe in reproducible detail the fitting algorithm, but we do not include the fitting source code in our model distribution packages.

In our experience, scientific reproducibility is achieved quite satisfactorily without the need for sharing analysis source code. It often happens that a researcher seeks to independently verify another's result, either out of skepticism or to adopt and/or modify the methodology for another application. When an apparent discrepancy is found, it can usually be cleared up quickly by contacting the authors of the study. In rare cases, inspection of the source code is necessary and sharing of the code may be then be desirable, but it is far more efficient for the developer of the code to search for potential errors.

A more effective way than open code to enhance the integrity of scientific results is to encourage studies that explicitly aim to independently confirm important findings. Some university professors assign their students to reproduce the methodology and results of an earlier study, as a learning exercise that is also a valuable scientific contribution [e.g., *Salawitch et al.*, 2017]. In the space science community, such studies are generally discouraged in peer review of both proposals and papers. Independent confirmation of results is a major (though perhaps unglamorous) scientific pillar that should be supported by the community.

Although we perceive dangers and disadvantages of requiring open code for funded research, that is not to say that preservation of analysis codes is unimportant. While it is unnecessary and generally unrealistic for an outside party to reproduce a researcher's results using the original source code, researchers have the scientific responsibility to maintain an internal archive of code in such a way that they can always reproduce their *own* results. In practice, this means, for example, not altering code (and associated subroutines) used to produce a figure once the figure is published. A natural consequence of this archival requirement is that a researcher's personal library of code quickly becomes vast and complex (even if managed by version control software), which itself poses difficulties for making it open source.

Productivity

The other potential benefit of an open code policy is enhanced scientific productivity of the community as a whole. One way to measure this benefit is the usefulness of codes that have already been made freely available, and one metric of usefulness is the number of citations that the code garners. By this measure, the benefit is well demonstrated for physics-based model codes, including the following upper atmospheric and ionospheric models: TIE-GCM [*Richmond et al.*, 1992, 475 citations as of December 2017], WACCM-X [*Liu et al.*, 2010, 60 citations], and SAMI2 [*Huba et al.*, 2000, 208 citations]. Major empirical models have also proven very useful to the space physics community, including NRLMSISE-00 [*Picone et al.*, 2002, 1191 citations] (the top three cited papers in JGR Space Physics are the MSISE-90, MSIS-86, and NRLMSISE-00 papers), IRI 2000 [*Bilitza et al.*, 2001, 864 citations], and IGRF [*Finlay et al.*, 2010, 546 citations].

Such models and tools enhance scientific productivity because they allow researchers to address their own lines of inquiry without having to develop the necessary sophisticated physical models or to conduct the extensive data reductions encapsulated in empirical models. But in order to be useful, the codes must be specifically written for widespread use, tested for robustness, and documented. This “publishing” of software requires substantial effort by and funding of the individuals and groups that develop such codes, and very few institutions have the resources needed to publish, maintain, and support open-source physics-based modeling code. A desirable balance should therefore be found between usefulness and publishing costs (which reduce the scientific productivity). Model diversity is another important consideration, because the reliance on a small number of models can lead to erroneous results and reduce long-term productivity; the Climate Model Intercomparison Project (CMIP) [*Meehl et al.*, 2000] illustrates the benefits of diversity for scientific productivity.

While the free availability of sophisticated physics-based and empirical model codes enhances productivity, at some point (on a scale of diminishing sophistication and applicability), the cost of making the code useable by the general community outweighs the productivity gains. We assert that most data and model output analysis codes (i.e., codes tailored to process data to obtain a specific scientific result) fall into this category. From personal experience, we estimate that the time it takes to publish analysis code (i.e., make it usable by others) is 1 to 4 times the initial effort of developing the code for internal or personal use. Since analysis is the backbone of scientific results, a blanket open code requirement would thus have a severe negative impact on scientific productivity.

Publishing all analysis codes could also degrade science quality. It enables the re-use of others’ analysis codes without understanding and thereby fosters sloppy results. In contrast, writing one’s own analysis code is an invaluable (and undervalued) way of learning methodological techniques and algorithms. The benefits of a creative diversity of analysis approaches and interpretations would be greatly diminished if the community were to use the same code over and over. Additionally, we expect that free availability of analysis code would lead to a more sequestered scientific community, rather than one that is rich with collaboration. Each researcher and institution cultivates their own sets of expertise on a range of topics, which is embodied in their analysis codes. Open code would remove a key incentive for collaboration, and would thus degrade the effective employment of the community’s collective skill set.

Furthermore, a blanket publishing requirement would inhibit scientific creativity and thus impact science quality. As data and model analysts, we often find that standard statistical and processing techniques and codes are inadequate for a particular science question. When we get ideas for new approaches to probe data or model output, we prefer to code them up as quickly as possible to test their suitability and to make new discoveries. But a publishing requirement would make us less likely to act on new ideas (especially risky ones), knowing that we either have to make the code publishable up front (delaying the discovery) or later on (delaying future discoveries).

The option of making analysis code available as-is (i.e., without documentation) is also not likely to enhance scientific productivity. First, as previously noted, it takes substantial effort to read and understand another person's "raw" code. For the same reason, scientists typically do not publish either their initial back-of-the envelope calculations or their notebook entries; rather, they communicate via polished papers. In most cases, it is probably quicker for researchers to write their own analysis code, especially since they likely need to tailor it for their own purposes. Second, whereas the published physics-based and empirical model codes listed above are all written in Fortran (the lingua franca of scientific programming), analysis codes are typically written in IDL, Matlab, or Python (which have sophisticated graphics capabilities), and most researchers use only one or two of those languages. Thus, a given set of analysis code is directly useable only by a portion of the community. Effectively, disseminating raw analysis code pushes the responsibility for translation and publication from the producer to the user. Furthermore, a blanket open code requirement would flood the community with a confusing sea of bespoke codes, thereby possibly reducing both scientific productivity and quality.

Conclusions and Recommendations

In conclusion, based on our experience as data and model analysts, and as empirical and physics-based model developers, we assert that a blanket open code requirement would be detrimental to both science quality and productivity, without enhancing scientific reproducibility. However, there is a benefit to productivity from the free availability of sophisticated physics-based and empirical model codes, provided such codes are robust, well documented, maintained, and supported (all of which requires substantial funding).

Finally, to address topic 4 ("What lessons can be drawn from your experience with open data policies that might help inform future open source policies?"), we have found that the requirement in Heliophysics programs to use only publicly available data, and to primarily use only NASA-produced data, in funded research has restricted the types of scientific projects we can pursue, and has therefore inhibited our scientific output. There are many examples in space science of "one person's noise is another's signal", and it is a mistake to not consider such data in scientific inquiry. If a corresponding requirement were imposed on model codes (i.e., funded research must only use openly available code), scientific productivity would be further diminished.

Should NASA decide to establish an open code policy, our recommendations are:

1. Require open source codes for models (physical and empirical) only when the development of such models is new or previously undocumented and is the primary objective or a major task of the funded research.
2. Require open source codes for data processing techniques only when the development of such techniques is the primary objective of the funded research.
3. For science papers resulting from funded research, strongly encourage inclusion, citation, and/or complete description of all inputs, algorithms, and output values that produced each claim, but do not require inclusion of or pointers to source codes.

4. Funded researchers should be strongly reminded to follow scientific norms by maintaining an internal archive of source code, so that their results can be reproduced by them in the event that a discrepancy is discovered.
5. Do not place restrictions on the type of modeling and analysis codes that can be used in funded research proposals.
6. Provide extra funding (above the standard grant amount) for the task of documenting the code to be shared as part of the funded research.
7. Support existing model enhancement and documentation activities with a dedicated program (like the Heliophysics Data Environment Enhancements program).
8. Support studies that explicitly aim to independently reproduce important scientific results, or that aim to resolve discrepancies among existing results.

References

Bilitza, D. (2001), International Reference Ionosphere 2000, *Radio Sci., 36*, 261–275.

Drob, D. P., et al. (2015), An update to the Horizontal Wind Model (HWM): The quiet time thermosphere, *Earth and Space Science*, *2*, doi:10.1002/2014EA000089.

Finlay, C.C., S. Maus, C.D. Beggan, et al. (2010), International Geomagnetic Reference Field: The eleventh generation, *Geophys. J. Int., 183*, 1216–1230.

Gil, Y., et al. (2016), Toward the Geoscience Paper of the Future: Best practices for documenting and sharing research from data to software to provenance, *Earth and Space Science, 3*, 388–415, doi:10.1002/2015EA000136.

Huba, J. D., G. Joyce, and J. A. Fedder (2000), Sami2 is Another Model of the Ionosphere (SAMI2): A new low-latitude ionosphere model, *J. Geophys. Res.*, *105*, 23,035–23,054.

Liu, H.-L., et al. (2010), Thermosphere extension of the Whole Atmosphere Community Climate Model, *J. Geophys Res., 115*, A12302, doi:10.1029/2010JA015586.

Meehl, G. A., G. J. Boer, C. Covey, M. Latif, and R. J. Stouffer (2000), The Coupled Model Intercomparison Project (CMIP), *Bull. Amer. Meteor. Soc., 81*, 313-318.

Picone, J. M., A. E. Hedin, D. P. Drob, and A. C. Aikin (2002), NRLMSISE-00 empirical model of the atmosphere: Statistical comparisons and scientific issues, *J. Geophys Res., 107*, doi:10.1029/2002JA009430.

Richmond, A. D., E. C. Ridley, and R. G. Roble (1992), A thermosphere/ionosphere general circulation model with coupled electrodynamics, *Geophys. Res. Lett., 19*, 601–604.

Salawitch, R. J., T. P. Canty, A. P. Hope, W. R. Tribett, and B. F. Bennett (2017), *Paris Climate Agreement: Beacon of Hope*, Springer, doi:10.1007/978-3-319-46939-3.

Appendix: Supporters

Name	Position	Affiliation	Stake
1. Hanli Liu	Sr. Scientist	NCAR	Model developer & analyst
2. Sarah McDonald	Res. Physicist	NRL	Model developer / data, model analyst
3. Ludger Scherliess	Professor	Utah St. Univ.	Physics-based, empirical model developer
4. Alan Segerman	Aerospace Eng.	NRL	Research astrodynamicist / data analyst
5. Stanley Solomon	Sr. Scientist	NCAR	Model developer & analyst
6. Jeffrey Thayer	Professor	U. Colorado	Data & model analyst
7. Harry Warren	Astrophysicist	NRL	Data analyst / model developer
8. Daniel Weimer	Res. Professor	Virginia Tech	Data analyst / model developer