Comments on a Future Open Code Policy: Potential Problems and Pitfalls

Summary

The open code policy that is being considered by NASA's Science Mission Directorate may have negative impacts on researchers if it is not properly implemented. Several issues with such policy are discussed, including PI time burdens, implementation issues, unauthorized and unacknowledged code use, and commercial use by code developers.

Author

Daniel Weimer, Research Professor
Virginia Tech
dweimer@vt.edu
(757) 325-6908

The open code policy that is being considered by NASA's Science Mission Directorate could have some benefits to the research community. On the other hand, such policy may have substantial negative impacts on researchers if it is not properly implemented. A few potential problems resulting from such requirements are discussed here.

Research Code Development is Unique

One important point to consider is the difference between research grants and contracts. Contracts generally will have specific deliverables that must be provided by the given deadlines. If computer code is one of these deliverables, then it is expected that this code works as expected and the code is well documented. Research grants, on the other hand, tend to list the topics that will be investigated, and in most cases the only expected results are publications. The presentations and publications may be considered as the "deliverable" results of such research grants. The final results of such research are not generally known ahead of time, otherwise the research isn't worth doing.

In order to reach the final results in a research project, a significant number of different computer programs may need to be developed, often by only one or two people. It is a well-known fact that such research code tends to be very poorly documented.
In many cases, several programs are used in sequence, and these programs are often created for one-time use only. If a program doesn't work as intended, then a new

program may be created, orphaning the prior versions. Research code may work for one specific case, but may have bugs that crop up when applied to other cases, as they are often a work in progress. As a result, research programs often have very little documentation, including both embedded comments and separate descriptions.

Research papers often rely heavily on figures, that are often unique for each publication. As a result, many of the research codes that are created may be for the creation of these figures, and sometimes used one time only. These graphics programs often rely on the use of personal utility programs that have been developed over the course of many years, and likewise, these utilities are always being expanded and upgraded, often with little formal documentation.

If it were to be "required that all NASA-funded, peer-reviewed science papers include an electronic compendium of the source codes, inputs, and outputs that produced the results shown" in every paper, then a significant effort would be required to produce the extensive documentation that would be required. This requirement would significantly add to the strain on the investigators' time, and it is already common that research grants do not adequately provide sufficient funding for time and labor.

If an "open code" policy were to be implemented, then a distinction should be made between the different types of code that are expected to be documented. In the case of "models," it might be reasonable to expect their code and libraries to be open source if such models are the specified objective of the research grant, provided that the other issues mentioned later can be adequately addressed (i.e., commercial applications). On the other hand, in the case of programs and graphic routines having only an ancillary role in the research, then the "open code" policy should not be required, or left optional.

While creating a compendium of all files is possible (I've done it on at least one occasion for reproducing specific figures in a paper), it will take a lot of extra time, and in many cases the desired reproducibility is unlikely to be achieved. One reason is that the code may rely on many different input and output files (sometimes numbering in the thousands or using several GB of disk space), that need to be placed in specific directories, and using different syntax on various operating systems. The codes may also require the use of commercial software, such as IDL or MatLab. In the case of C, FORTRAN, and other languages, programs that successfully compile and work on one system may not work on another. Computation times may take many hours, or even days, making reproducibility impracticable in some cases. In essence, researchers may need to spend hours collecting and documenting code that they may not even use again.

Other Issues to Consider

Several questions arise after consideration of an “open code” policy. These are:

1. How thorough does the documentation, if any, need to be?

2. Does the code for all libraries need to be provided, even if used only to draw one graph for a publications?

3. Does the developer need to support the open code, by responding to inquiries and troubleshooting? Is such support expected beyond the end of grant expiration?

4. If such requirements for open code are made retroactive, will investigators need to do this work (document programs used in prior, expired grants) without any new, additional funding? How far back in time will such requirements extend? Will investigators who are now retired be required to comply with retroactive requirements?

5. Where is this additional code and documentation to be stored, particularly in the case of prior grants and publications? Are the publishers expected to be responsible for maintaining the archives, the home institutions, or funding agency?

Use of Code Without Permission or Acknowledgement

In the past there have been experiences with code being used without permission and/or acknowledgement. In the past model code had been provided to the CEDAR data base, and to others upon request. Afterwards, I had seen a publication that obviously had used one of my graphic utility programs, without any acknowledgement. In another case my own model had been used to produce results that were shown in a prominent oral presentation at a conference that I had attended, and passed off as their own commercial product, without any public acknowledgement. As a result of these experiences, on subsequent models I’ve switched to preferring the release of pre-compiled IDL programs, including graphic utilities, without the source code but including instructions. These are distributed upon request by e-mail, rather than publicly available online.

Other people have had similar experiences. A new assistant professor has communicated the experience of releasing open source Matlab GPS data processing codes. This was done with the expectation that when people use it, they will rightly acknowledge the original work, as requested in the source code. This hasn’t happened. She has seen people using snippets of the exact code or figures generated by the code, but no acknowledgment of either. This has happened even with people who had asked for help in understanding the codes!

If all research code were to be made open, then such abuses would likely become more common.

Commercial Applications

While it is rare for space science research to have commercial applications, it is possible in the realm of space weather forecasting. For example, an empirical model of geomagnetic field variations was developed a few years ago under a prior grant from the "National Space Weather Program" at NSF. The software code and required files have been provided to both the NASA CCMC and NOAA SWPC, as was expected. However, since this program may have commercial applications in the electric power industry, this code has not been openly released. Several years later, the Virginia Tech Intellectual Properties office is now getting close to licensing this prediction program to two separate business users. This prospect for commercial use would be less likely to succeed if the code were entirely open, as then it would be possible that people or corporations could profit from reuse without reimbursing the code developers and institutions.

There are other examples of commercial uses in the field of biotechnology and medicine. Research grants in these fields have sometimes led to the formation of very profitable corporations. It would be useful to look into the policies that other agencies, such as the NIH, have in place for research that may have commercial applications.