Implications of a Future NASA SMD Open-Source Policy

Short Summary: We are software developers and users with decades of experience modeling systems and processes important to NASA's Science Mission Directorate. Open source has the potential to enable great progress in many scientific fields by providing community access to sophisticated physical models. However, the potential for inadvertent harm also is significant. A successful NASA open-source policy must include the establishment of NASA programs that will fund long-term scientific software development and maintenance, and provide for community access to this software. Such programs are essential for NASA to achieve its science goals, because large-scale computational modeling has become a central pillar of NASA science, mission design, and mission planning. We point out that an open-source mandate without the resources to develop and maintain the software for community access would have detrimental impacts on those scientists/developers who create large-scale SMD science codes, through negative effects on their resources and their incentives to innovate.

Authors:

C. Richard DeVore
Research Astrophysicist
Space Weather Laboratory (Code 674), NASA Goddard Space Flight Center
Computational astrophysicist – developer/user of large-scale magnetohydrodynamic models

Spiro K. Antiochos
Chief Scientist
Heliophysics Science Division (Code 670), NASA Goddard Space Flight Center
Theoretical and computational astrophysicist – developer/tester of heliophysics theories

Alex Glocer
Research Astrophysicist
Geospace Laboratory (Code 673), NASA Goddard Space Flight Center
Computational astrophysicist – developer/user of large-scale magnetohydrodynamic models

Judith T. Karpen
Laboratory Chief
Space Weather Laboratory (Code 674), NASA Goddard Space Flight Center
Computational astrophysicist – user of hydrodynamic and magnetohydrodynamic models

James E. Leake
Research Astrophysicist
Solar Physics Laboratory (Code 671), NASA Goddard Space Flight Center
Computational astrophysicist – developer/user of large-scale magnetohydrodynamic models

Peter J. MacNeice
Research Astrophysicist
Space Weather Laboratory (Code 674), NASA Goddard Space Flight Center
Computational astrophysicist – developer/user of large-scale magnetohydrodynamic models

Corresponding Author: C. Richard DeVore | c.richard.devore@nasa.gov | (301)-286-7537

Implications of a Future NASA SMD Open-Source Policy

Introduction. The Science Mission Directorate (SMD) of the National Aeronautics and Space Administration (NASA) is contemplating the establishment of a future open-source policy for scientific software developed under NASA research and development (R&D) programs. This policy could require that the source code – i.e., the high-level computer instruction set – used to generate the software must be made available freely under some open-source license (e.g., https://opensource.org/licenses). Such a policy has laudable good intentions of disseminating the fruits of taxpayer-funded R&D to the science community at large, of enhancing the reliability and reproducibility of the results of NASA-funded research investigations, and of expanding the scope and depth of numerical modeling in the service of the nation's space programs.

Stakeholders. We are software developers and users with decades of experience modeling heliophysical and astrophysical phenomena using highly complex, massively computationally parallel codes. There are several important implications of the proposed policy for NASA's research programs and community of the future. The remainder of this document states and explains our views.

Principles. We support the principles of disseminating products of federally funded research to society, and of enhancing the quality and reproducibility of NASA research findings. The former is only just in a democratic republic, and the latter is at the heart of the science enterprise. As a matter of principle and practice, we share our software with students, postdoctoral associates, and other researchers who work with us over a sustained period of time. At the end of their internships, our collaborators are welcome to take the software with them and continue to use it pursuing new research at their new institutions. We emphasize that we are physical scientists, not computer scientists; hence, in our view numerical models are an investment and a means to an end, not ends in themselves. Facility with such models is a major component of our skill set, on par with facilities with analytical mathematics and physical insight. Honing models that are capable of the complexity and flexibility needed for our research requires long-term investments of time and energy. These experiences color our perspective on the implications of an open-source policy and motivate the recommendation expressed below.

Recommendation. We believe that NASA must make a substantial commitment and provide sustained support to a community of physical scientists, computer scientists, and programmers, whose mission will be to develop, test, maintain, support, and regularly update scientific software in open source for use by all interested researchers. Members of this community must also be encouraged to devote a significant fraction of their time to their own research, enabling them to stay abreast of advances in their fields and to be invested in the outcomes of their principal task – producing valuable and accessible scientific software products for the community. Such an initiative could establish an enduring, thriving environment for the creation and provision of software to the research community, supporting innovative advances in computational space science to benefit the agency and the nation.

Implementing this recommendation would greatly strengthen NASA's computational space science capabilities and pave the way to major future advances in fundamental understanding of the Earth, Sun, solar system, and universe. In turn, this understanding would guide the

development of new NASA space missions and the scientific return realized by both ongoing and future missions. Historically, NASA has avoided significant investments in supporting software development specifically. In heliophysics, programs in High-Performance Computing and Communications for Earth and Space Sciences [HPCC-ESS], Applied Information Systems Research [AISR], and Living With a Star Tools and Methods [LWS T&M] – all now defunct or dormant – and LWS Strategic Capability [LWS SC] are the few exceptions. Consequently, much of the currently employed scientific software has been developed under support from agencies other than NASA (the National Science Foundation and the Departments of Energy and Defense, in particular). In many such cases, the focus has not been on creating software suitable for use by non-expert members of the broader community; rather, it has been on addressing critical science objectives of the developers and their close collaborators. This narrowly-focused approach is far from optimal as a model for an effective open-source policy at NASA. The recommendation that we have outlined, in contrast, could provide the foundation required for such a policy to be effective and to realize the potential of open source to advance space science.

The adoption of an open-source policy by NASA would have several consequences for the space science community. Positive impacts on the enterprise include effects on (1) reliability and reproducibility and (2) community-wide collaboration.

Impact 1: Reliability and Reproducibility. The mandated provision of source code to the NASA science community could positively affect the reliability and reproducibility of the results of NASA-funded research. Any published result of a numerical simulation whose validity was questioned could be tested by an independent party, in principle, reducing the likelihood of fraud by investigators by increasing the risk of discovery. It is unclear, however, whether scientific fraud requiring such deterrence has been perpetrated by members of the NASA community. Therefore, researchers are unlikely to place a high priority on double-checking published results, even if enabled by an open-source mandate. On the other hand, performing comparative studies of the outputs of different computational models using nominally identical inputs could have real value, including better understanding of the robustness and sensitivity to the underlying numerical algorithms of the results obtained. Such findings would advance space science and computational science simultaneously.

Impact 2: Community-wide Collaboration. Increasingly, large software projects are developed by wide collaborations across institutions, frequently including students, postdoctoral associates, and junior scientists. Examples of such projects in heliophysics include SolarSoft and SunPy. An open-source policy would provide additional incentives for community-wide collaborations by securing the benefits of future software use, without restriction, to all contributing developers. Such a collaboration also could foster new, joint science investigations among its participants, and enhance the scientific impact of the software development effort, to the benefit of the entire NASA community.

We also believe that if a comprehensive open-source policy were adopted by NASA without associated institutional and financial support for the scientist/developer community, then the impacts on computational space science would be significantly negative. Below, we discuss our concerns about such effects on the (3) resources and (4) incentives available to software developers. These negative impacts would fall most heavily upon the tiny community of

scientist/developers who create large-scale computational models requiring long-term investments of time and energy.

Impact 3: Resources. The mandated provision of source code to the NASA science community is not a zero-cost undertaking. In order to be widely useful, software must be adequately documented, maintained, supported, and updated, all of which consume additional significant resources. Imposing requirements for open-source licensing and release of software would lead to additional resource expenditures on the part of developers, clearly impairing their scientific productivity. The effects of open-source policies would be greater on individuals or small groups of investigators than on large groups, and on developers of extensive, flexible software packages than on those of limited-scope, single-purpose applications. Absent changes in NASA's philosophy and resource allocation toward software development, an open-source mandate by SMD could discourage scientific software innovation in the NASA research community by increasing the cost of that activity to developers.

Impact 4: Incentives. The mandated provision of source code to the NASA science community could negatively affect the motivation of prospective developers to produce new and innovative scientific software. Most such developers are physical scientists, not computer scientists or programmers, who invest their time and energy developing software to advance their research capabilities and their scientific output. Imposing a condition that this investment must be shared across the entire community recalibrates the cost/benefit calculation that the scientist/developer makes about embarking on a software project. This is especially true of long-term efforts to develop large-scale simulation codes, and especially true for young investigators who may be the most innovative thinkers and risk-takers. A possible strategy to mitigate such effects is to allow for a period of exclusive use by the developer before public release of the software, much as is done currently for data derived from some NASA space missions. An additional concern with open source is the misuse of complex software by non-expert users, in which inappropriate inputs produce unexpected, and apparently incorrect, outputs. This could lead to the publication of incorrect findings and consume the developer's time in attempts to correct the record. Therefore, we reiterate that an open-source mandate by SMD could discourage scientific software innovation in the NASA community by decreasing the value of that activity to developers, in the absence of changes of philosophy and resource allocation toward software development.