Comments on Sharing and Archiving of Software:

We refer to the proposed NASA plan regarding software access - Part C, p. 17 in
https://www.nasa.gov/sites/default/files/atoms/files/nasa_ocs_public_access_plan_may_2023.pdf

The motivation is reasonable, i.e., increase access to NASA-supported research outcomes, particularly to enable reproducibility of results, but the proposed measures are not well-reasoned. In our opinion, the established scientific process - including peer review and inherent competition in scientific endeavors - is the mechanism to achieve this (and there is a long evidentiary history to support this). However, there is admittedly widespread public perception that most scientific results are not reproducible. Unfortunately, there is some truth in this perception. The problem appears to stem largely from the social sciences (but also from the medical and biological sciences), and there are many underlying causes (such as poor statistical methodologies and the hype-tendencies of social media). However, that there is actually such a problem with NASA-supported research and that the proposed remedy (at least the part about sharing software) will actually be a meaningful remedy is entirely unclear. The document does not give any persuasive reasoning to make this case; it seems to be wishful thinking or just “let's do something and hope it works”.

We are thus concerned about NASA's planned requirements for sharing, documenting, and supporting software. We already are seeing anecdotal evidence of troubling developments in NASA-supported science related to software practices.

1. Sharing of Software Can Tend to Discourage Innovation:
Many researchers use established software packages to save time, and this is appropriate when done with knowledge of the limitations of the tools. However, the sharing requirements encouraged by NASA will tend to spread this practice to avoid having to document new code. It is unknown what innovations might not occur if those rare truly original thinkers might take that path instead.

2. Sharing of Software Can Corrupt Scientific Investigations:
Instead of developing new code specific to novel science investigations, researchers take the easy path of writing minimal wrappers to established software tools. This practice, without mastery of the underlying characteristics and limitations of those tools, can introduce biases, flaws, and even gross errors in new investigations.

3. Sharing of Software Inhibits Competition:
The skills and accomplishments of a research group are their stock in trade in competition for research grant awards. If their software can be adopted freely by their competitors, then the distinction between research proposals is blurred. The NASA panelists on the webcasts did not seem to appreciate the intensity of competition for grant funds, and instead described a bewildering campaign of social re-engineering toward an ideal of communal collaboration. The unintended but predictable consequence of such a policy is elimination of competition and a lowering of the rate of scientific progress.

4. Sharing of Software Can Pre-empt Fundamental Education:

Students are increasingly using large, established simulation and modeling software tools without learning the fundamentals underlying the subject. Furthermore, the rare, truly innovative students may be discouraged from developing new methods by the pressure of time and the tendency of the community to accept more readily results derived with known tools. This also will have the predictable consequence of lowering the rate of scientific progress.

5. Sharing of Software "as-is" for Peer Scrutiny is Appropriate:
The scientific process requires demonstration of the validity of claimed findings, and the currently established peer-review process for small-to-modest-scale projects accommodates such rigor, including for projects that develop and use in-house software tools. (An example of this process working well is found in the refutation by Mutel & Fix (JGR 109, 1171 (2003), doi:10.1029/2002JA009391) of the claim for a huge population of small comets near Earth that turned out to be due to a bug in a Fortran program.) So making one's source code available as-is to scrutiny by other researchers is logical and appropriate. However, see #6 below.

6. Sharing of Software for general users is Burdensome to Individual Investigator Research:
To have to fully document all software and support it for others to use beyond its original purpose would be a disproportionately high burden on individual-investigator research projects. Code written for one purpose with implicit assumptions about boundary conditions and parametric cases will require considerable work to become generalized. That required level of technical support is not commonly available to individual researchers with small grants. The NASA webinar panelists were not convincing in the promise that additional funds will be made available for such support because they cannot predict or control future allocations of funds. Moreover, requiring individual investigators to support software for others would take away from their research time. A reasonable approach may be to adopt a software documentation, sharing and support requirement for projects that are focused on the development of software tools and for "large projects", e.g., those of funding exceeding $10M per year or exceeding $100M in total. Such projects are likely to have sufficient personnel to comply with such software policy requirements.

7. Sharing of Software Will Benefit Preferentially Large Projects:
Taking advantage of new, large, sophisticated shared packages and tools may require human resources beyond the scale of small-group projects and individual investigators. Source code may be in an unfamiliar language, compilation may depend on unfamiliar resources, and installation may require system upgrades. Many R&A grants are too small to accommodate such efforts.

8. Sharing of Software Will Not Be Counted Equally with Peer Reviewed Publications:
The NASA webinar panelists were not convincing that academic committees for hiring, promotion, and tenure will comply with directives to count service to the scientific community (such as software documentation and support) equally with traditional metrics. That scale of cultural engineering goes far beyond NASA's charter.