doi:10.23689/fidgeo-5805

Sustainable research software for high-quality computational research in the Earth System Sciences: Recommendations for universities, funders and the scientific community in Germany

Petra Döll[1,2], Monika Sester[3], Udo Feuerhake[3], Holger Frahm[3], Bernadette Fritzsch[4], Dominik C. Hezel[1], Boris Kaus[5], Olaf Kolditz[6], Jan Linxweiler[7], Hannes Müller Schmied[1,2], Emmanuel Nyenah[1], Benjamin Risse[8], Ulrich Schielein[1], Tobias Schlauch[9], Thilo Streck[10], Gijs van den Oord[11]

[1] Goethe University Frankfurt (p.doell@em.uni-frankfurt.de)
[2] Senckenberg Leibniz Biodiversity and Climate Research Centre Frankfurt (SBiK-F)
[3] Leibniz University Hannover
[4] Alfred Wegener Institute Bremerhaven
[5] Johannes Gutenberg University Mainz
[6] UFZ Leipzig
[7] Technical University Braunschweig
[8] Westfälische Wilhelms University Münster
[9] DLR Braunschweig
[10] University Hohenheim
[11] Netherlands eScience Center Amsterdam

Abstract. Similar to other research fields, new knowledge in the Earth System Sciences is increasingly produced by computational research. However, the reproducibility of this type of research has been shown to be very limited, and its efficiency and quality need to be improved. Reproducibility requires researchers to publish both their research outcome in the form of a paper and their research workflows, software and data so that other researchers can reproduce the findings without any further support still years later. Efficient and high-quality computational research requires skills beyond programming as well as the capacity for software maintenance. Inspired by a best-practice example from the Netherlands, we provide 15 recommendations for universities, research funders and the scientific community who wish to support the development of sustainable high-quality computational research in Germany. They relate to the training and support of researchers by universities and other research organizations and to research funding. Of particular importance are options for establishing institutional support by research software engineers who are employed in permanent positions, funding of research software as research infrastructure as well as approaches for increasing the scientific merit that is achieved by producing sustainable research software and providing reproducible research output.

1 Introduction

To a large and increasing extent, new knowledge in the Earth System Sciences is obtained through computational research. It can be categorized into 1) the simulation of Earth system processes, 2) the design, processing and analysis of Earth observations (remote sensing and in-situ measurements) as well as of experiments and 3) integrative analyses (Figure 1). The research software[1] that is developed to do computational research in the Earth System Sciences is characterized by the complexity of the underlying models, multifaceted dependencies, the multi-modality of the data and – very often – the size of the data (due to high spatial and temporal resolution and extent), which can impose specific hardware and software requirements. According to Anzt et al. (2021), research software "optimizes existing and enables new research methods, implements and embeds research knowledge, and constitutes an essential research product in itself. Research software must be sustainable in order to understand, (….) reproduce, and build upon existing research or conduct new research effectively." The general principles of FAIR (Findable, Accessible, Interoperable, Reusable) software have been specified for research software as FAIR4RS principles by Barker et al. (2022) and, more detailed, by Chue Hong et al. (2022).

Computational research in the Earth System Sciences

Simulation of Earth system processes by	▪ Earth system models (climate and weather models) and integrated assessment models ▪ sectoral models of, e.g., deep Earth processes, water on the continents, ocean processes, biogeochemical cycles and vegetation
Design, processing and analysis of	▪ Earth observations, e.g., • processing of GRACE satellite signals to derive time series of mass change • geomorphometric analyses of land surface elevations • object identification in satellite images ▪ lab and field observations and experiments, e.g., • luminescence dating • geostatistical analysis
Integrative analysis of	▪ simulation models and Earth observations by, e.g., data assimilation ▪ large databases using statistical analyses or machine learning ("big data" analyses) ▪ stakeholder knowledge by, e.g. multiple-criteria decision analysis or Bayesian networks

Figure 1. Types of computational research in the Earth System Sciences

Like other research, computational research should be reproducible. However, it has been shown that published computational research in various disciplines, including the Earth systems sciences, is very rarely reproducible (e.g., Hutton et al. 2016, Stodden et al. 2018, Stagge et al. 2019). In a workshop organized by members of the German Reproducibility Network in March 2022, "strategies for making reproducible and open science training the norm at research institutions" were identified (Kohrs et al. 2023). While covering the much broader issue of reproducibility mainly from the perspective of

[1] Software for generating, processing, analyzing and visualizing research data that was specifically developed in the course of a research process, including source code files, algorithms, scripts, computational workflows and executables that were created during the research process or for a research purpose; software components (e.g., operating systems, libraries, dependencies, packages, scripts, etc.) that are used for research but were not created during or with a clear research intent should be considered software in research and not research software (Barker et al. 2022)

psychologists and biomedical scientists, these strategies are broadly applicable for computational research, too: 1) offering training, 2) adapting research assessment criteria and program requirements and 3) building communities, by specified actors such as instructors and institutional leaders. Computational research was not specifically addressed in this study, the perspective of infrastructure personnel such as software engineers was missing, and the authors expressed the need to explore top-down strategies with administrators in charge of structural changes and funding.

What is needed to safeguard the quality and efficiency of computational research in Earth System Sciences and to make research results that have been generated by research software reproducible? Clearly, not every Earth system researcher must be an expert in developing sustainable software. However, there is a need for a subgroup of researchers, who are able to do so – otherwise, adequate software will not be available for the discipline. Thus, different graduations of software competence are required. It must be rather easy for persons who were not involved in the development of the research software to apply, improve and extend the research software. This requires generating and maintaining research software of high software quality that adheres to the FAIR4RS principles. Characteristics of sustainable high-quality software are a code that is programmed according to standards of modern software design, e.g., readable, modular, maintainable, flexible, portable, reliable, reusable, robust, and tested both at the level of functional units and overall. It has a good test environment and comprehensive software documentation including a user guide. Open-source licensing as well as contributions from the wider research community support the sustainability of research software. However, all of this does not safeguard that software can be run without any further support even after many years, which is necessary for achieving the reproducibility of computational research.

The development of sustainable research software as well as the reproducibility of computational research is hindered by a lack of knowledge, expertise, appreciation and support. Diverse efforts by several actors are required to ensure that computational research in the Earth System Sciences is done in a more sustainable manner. At universities, research software is mostly developed by early career researchers (ECRs) who are domain specialists in their fields and, in their majority, are not trained in software design, are not familiar with the requirements of reproducibility of computational research results and have difficulties finding the time for developing sustainable research software, in particular as they are evaluated based on their scientific innovations. Most likely, their supervisors lack knowledge of sustainable research software, too, and their academic reputation is not yet affected by the sustainability of the research software developed in their groups. Organizing a community-based quality control and software development is costly as it requires personnel for support. An additional difficulty in achieving sustainable research software in a university setting is that new ECRs must continue working with a complex code that many former ECRs have developed, who often cannot be contacted anymore. Legacy codes, i.e. software that has been in development for decades, might need to be re-programmed to become sustainable, which is costly and difficult to achieve without extra funding. Clearly, ECRs as well as their supervisors require more support for developing sustainable research software than is currently available to almost all of them.

In addition to these structural restrictions, high-quality and reproducible computational research is constrained by the type of research funding, which is often dominantly from third-party institutions. Most funding schemes target top-notch developments and provide fixed-time support for individual projects. This implicitly favors quick turnover times of individual projects and does not reflect the need to continue the development of research software beyond the funding period of the respective project. In contrast to publishing a new research software approach in a high-impact journal, providing and maintaining these research tools is usually not considered excellent research in the Earth System

Sciences and beyond. Taken together the structural and organizational conditions hinder the development of sustainable research software at scale.

This paper presents recommendations on how German universities and research funding organizations may support sustainable high-quality computational research in the Earth Systems Sciences and beyond. These recommendations were identified during a DFG round table in June 2023, with contributions by ECRs and senior researchers and instructors from the Earth System Sciences and computer science, research software engineers, chief information officers of two universities and a representative of a funding organization (DFG). In the following, we first present best practice examples that inspired our recommendations.

2 Best practice examples for supporting sustainable high-quality computational research

The Netherlands is at the forefront of institutional support for research software (Box 1). In 2012, the Netherlands eScience Center was established as an independent foundation; to keep the Netherlands at the cutting edge of research, it builds and applies software to enhance the use of computing and digital technologies in academic research, by empowering researchers through innovative software (www.esciencecenter.nl). The well-staffed center is mainly funded by the Dutch national research funder NWO. Similarly, many Dutch universities employ groups of research software engineers (RSEs) to support computational research. The term RSE refers to a professional who combines expertise in software development and methodology with deep knowledge of one or more research fields. RSEs focus on developing and/or maintaining research software, to answer research questions within their research disciplines; at universities, they can be either part of the academic or the research support staff (Netherlands eScience Center, 2023).

Box 1: Institutional support for sustainable high-quality research software in the Netherlands

The Netherlands eScience Center is a Dutch national research organization dedicated to applying research software to answer research questions in any scientific domain. Its approximately 75 RSEs work with universities and research organizations in the Netherlands and Europe by collaboratively designing sustainable software and by building digital skills and expertise. The RSEs are mostly domain scientists with extensive previous experience in software development. A main approach is to release calls for proposals for research requiring software skills that can be provided by its RSEs; the proposals are then reviewed externally.

The RSEs of the eScience Center write high-quality software as well as software tests and documentation while also teaching the involved domain scientists by example (e.g. via code review). They also write papers and acquire external research funding, e.g., from the EU. In addition, the eScience Center organizes workshops for researchers, makes all of its software and training materials openly available online and provides a fellowship program for people who wish to improve the use of research software within their organization or discipline. It also maintains a research software directory (https://research-software-directory.org).

In addition to the Netherlands eScience Center, many Dutch Universities employ a group of RSEs to support their students and researchers in the development of sustainable research software, in addition to data management. One example is Utrecht University (https://www.uu.nl/en/research/research-data-management), where about ten RSEs work together with students to professors at Utrecht University to develop innovative research software and help build digital skills in academia, by providing free consultancy and free short-term support including own coding, and by participating in the writing of research proposals (https://utrechtuniversity.github.io/research-engineering/).

Different from the Netherlands eScience Center, the UK Software Sustainability Institute does not employ a large number of RSEs, and it is not a national center but is based at the universities of Edinburgh, Manchester, Southampton and Oxford (https://software.ac.uk). It focuses on advancing the role of research software and on promoting communities of practice through its fellowship program but has also participated in joint research projects with domain researchers. In addition, the institute coordinates the UK activities of The Carpentries (carpentries.org) regarding workshops on sustainable research software.

In Germany, the Scientific Software Center of Heidelberg University currently employs a team of five RSEs (www.ssc.uni-heidelberg.de), who offer similar consulting and coding services as Utrecht University. For research groups under the Helmholtz umbrella, HIFIS provides training and consulting services for software development as well as a research software directory (https://www.hifis.net/services/software-overview.html). High-performance computing in Earth system modeling is supported by Joint Lab ExaESM (https://www.exaesm.de/structure/) as well as by natESM (www.nat-esm.de). The Society of Research Software (de-RSE.org), a German association of RSEs, promotes research software and its role in science and scientific careers, e.g. by writing position papers such as Anzt et al. (2021).

3 Recommendations

The 15 recommendations are structured along the three groups that should implement them, universities, research funders and the scientific community. They are to support students and (early career) researchers in learning about, developing and maintaining sustainable high-quality research software and in doing reproducible computational research.

3.1 Universities together with German research funders should urgently set up structures for the employment of groups of RSEs with permanent positions

A tiered structure where 1) individual universities (or often preferably small groups of universities) and 2) national center(s) employ groups of RSEs with permanent positions, like in the Netherlands, might be a good option. Local centers can provide more direct consultancy and technological support, while at national centers, disciplinary groups of RSEs related to specific research domains, e.g., the Earth system sciences, can create a domain-specific computational research (support) hub. A national research software center could be organized concerning research domains similar to the German NFDI (Nationale Forschungsdateninfrastuktur, www.nfdi.de), where NFDI4Earth supports research data management in the Earth system sciences.

High-quality RSEs can only be attracted by permanent positions. To be attractive and effective, it is necessary to employ, at any location, not just one or two but a group of RSEs with different research domains. With such an organizational structure, RSEs will be enticed by interesting topics and innovative technologies. In addition, institutionalizing the profession of RSEs will prevent the loss of talent of university domain researchers that have become RSEs during their Ph.D. or postdoc phase.

Tasks and services of research software centers should include teaching/training, consultancy and software development (compare Box 1). The establishment of fellowships for researchers interested in promoting sustainable software development was shown to be beneficial for creating communities for sustainable software development (Sufi and Jay 2018).

Funding of permanent RSEs at a national center could be achieved, like in the case of the Netherlands eScience Center, by central funding from the German research funders DFG and BMBF. A process for creating and maintaining permanent positions for RSE pools at universities could be supported by 1) funders providing seed money for university research software centers and 2) including, in research proposals, a module in which funds for the contribution by the RSEs are requested. Research funders could establish calls for research software centers (similar to BMBF calls for data centers) for which universities or consortia of universities could apply with an implementation plan and obtain seed money for their centers.

When focusing on supporting computational research in the Earth System Sciences, there is an alternative to multi-disciplinary research software support. One of the large non-university research organizations strongly involved in the Earth System Sciences (such as Helmholtz, which already provides software services to its researchers, see Section 2) could provide a platform and support for computational research in the Earth System Sciences that is also open to university researchers. Appropriate funding for such a community service is required.

3.2 Recommendations for German research funders

3.2.1 Research funders should survey needs regarding the sustainability of research software among universities and other research organizations

This would be an informative basis for evaluating the options of research funders for supporting the sustainability of research software.

3.2.2 Research funders, in particular DFG and BMBF, should establish explicit "sustainable research software" modules for research proposals

Explicit "sustainable research software" modules in proposals would have the following benefits:

- The high visibility of such modules makes applicants aware of the possibility of financing RSEs
- Such a proposal component makes requested shares for scientific innovation and sustainable research software, e.g., for university RSE pool, transparent
- It increases awareness and acceptance of the high value of software development in the scientific community, including the reviewers and bodies deciding on funding (e.g., DFG-Fachkollegien)

However, it may be difficult to plan the work of RSE pools without sufficient project-independent base funding; next to project-related work, RSEs need time to keep up with computer science innovations.

3.2.3 Research funders should require software management plans as part of research proposals

Proposals for funding computational research should be required to include a software management plan. Before this requirement starts, sufficient guidance needs to be given on the generation and format of such a plan, comparable to the guidance of the Netherlands eScience Center (Martinez-Ortiz et al. 2022). To keep software available over a longer period of time requires additional funding.

3.2.4 Research funders should regularly release specific calls for supporting the sustainability of research software

In 2016, 2019 and 2022, DFG released calls for proposals that aimed at increasing the sustainability of research software. While 28% of the 68 proposals in 2019 were funded, only 20% of the funding requested by ca. 132 proposals in 2022 could be funded due to limited funds. The percentage of proposals from the Earth System Sciences increased from 9% in 2019 to 14% in 2022. An increased need and willingness of researchers to enable excellent research by high-quality research software are clearly recognizable, underlining the need for regular well-endowed calls for sustainable research software, not only by DFG.

3.2.5 Research funders should provide follow-up funding for finalizing sustainable research software of high interest to the user community

Follow-up projects or project extensions of, e.g. one year, in which research software that has resulted in innovative scientific outcomes, is further developed into a fully sustainable software, should be financed if the potential user community shows interest. As an example, financing could be done with the DFG-instrument “transfer project”.

3.2.6 Research funders should contribute to the “Amsterdam Declaration on Funding Research Software Sustainability” and then implement its recommendations

The “Amsterdam Declaration on Funding Research Software Sustainability” (Barker et al. 2023) is being developed based on a workshop with representatives of national research funders from many countries, organized in November 2022 by the Netherlands eScience Center and the Research Software Alliance (ReSA). “The aim of the declaration is to raise awareness of the role of funding practice in the sustainability of research software, and to improve that practice“ (Barker et al. 2023). We commend that DFG has become, in July 2023, a member of ReSA and participates in the “Funders Forum” to harmonize how funders deal with research software issues[2]. It is ReSA’s vision that research software (and its developers) is appropriately acknowledged as an important contribution to research. DFG should continue its participation in ReSA and the Amsterdam declaration, and BMBF and other funders should get involved in the finalization of the declaration, sign it and implement its recommendations. However, no recommendation regarding permanent positions for RSEs is currently included.

3.2.7 Research funders, in particular DFG, should make it possible for major research software to be financed like a research infrastructure

Major research software should be financed like a research infrastructure (“Großgeräteantrag”, with combined funding that the federal state and DFG, covering the total cost of ownership (including all money for maintenance for the first 5-7 years, with staff costs). The scientific community would have to contribute to the criteria for “major research software”. While the funding rate for major infrastructure is very high, staff is currently not financed with this instrument, which would be indispensable in the case of major research software. For finding RSEs, permanent positions at research software centers would be required.

3.3 Recommendations for German universities

3.3.1 Universities should define standards for sustainable research software and reproducibility of computational research and provide support to researchers to implement these standards

These standards should be developed jointly by the German universities and should be consistent with and included in their open science policy (as open as possible but as closed as necessary). A framework for collaborating with private industry to enable the sustainability of research software (and open science) should be designed. This setting of standards needs to be combined with ample support for researchers, which includes infrastructure for software development and software and data storage as well as guidance, e.g., on software publication and licensing.

[2] https://www.kooperation-international.de/aktuelles/nachrichten/detail/info/staerkung-der-internationalen-forschungssoftwarelandschaft-dfg-ist-neues-mitglied-der-research-software-alliance-resa

3.3.2 Universities should make training on sustainable research software and reproducible computational research accessible to all its members.

Students and researchers can certainly learn from the literature how to best produce sustainable research software and how to best do reproducible computational research. Wilson et al. (2017) presented a set of good computing practices that every researcher can adopt, regardless of their current level of computational skill, while Wilson et al. (2014) describe practices for more advanced research software developers. However, for most researchers, it will be very difficult to follow the very advanced skills required to understand the "ten simple rules for writing Dockerfiles for reproducible data science" compiled by Nüst et al. (2020). Therefore, some training beyond programming courses should be made accessible to all members of the university involved in computational research.

While such training may be best organized by a university research software center (comp. 3.3.1) or by computer science departments, it can otherwise be provided to ECRs by collaborating with external institutions, e.g., at the level of federal states (e.g. HeFDI - Hessian Research Data Infrastructures) or by international initiatives like The Carpentries. The Carpentries latter do not provide instructors but train potential instructors and provide course material (https://carpentries.org/instructors/). In particular, courses for university teaching staff are necessary so that they can teach the subject to Bachelor's and Master's students.

3.3.3 Universities should include digital literacy including programming and sustainable research software development in their Earth System Sciences curricula

Already at the BSc level, all students should achieve some degree of digital literacy, e.g., understand the relevance of software-generated results and obtain programming experience. Universities should offer courses that cover aspects of sustainable research software development and reproducible research for Bachelor's and Master's students as part of disciplinary curricula. The scientific community should derive recommendations for learning outcomes and curricula.

For particularly interested students, universities should offer a micro-degree/certificate in sustainable research software development and reproducibility of computational research or, more broadly, in, e.g., sustainable data science, for which students take additional courses with about 10-25 credit points.

3.3.4 The requirements for a cumulative PhD dissertation should be modified by replacing one of three scientific papers with the production of a peer-reviewed sustainable and high-quality software that enables reproducible computational research

Such a modification would strongly enhance the development of sustainable research software and foster the reproducibility of computational research, as it would allow PhD students to spend more time on the software and its publication. Since 2022, DFG has accepted published research software as a "normal" publication. Software development should be viewed in the same way as developing a new lab method that is described in a paper and then counted towards the PhD.

3.4 Recommendations for the scientific community

Some of the above recommendations for universities and research funders require input from the disciplinary scientific communities in particular regarding the following.

3.4.1 The scientific community should support the definition of criteria for "major research software" to enable major research software can be financed like a research infrastructure

(see Section 3.2.7)

3.4.2 The scientific community should define which digital skills students and ECRs in their fields should acquire

Each field of science needs to define a coherent set of digital skills that are required in their field to execute research. This includes skills in data management, software development (architecture, coding, agile methods) and IT security. The disciplinary scientific communities should support universities in adapting the curricula such that that students can acquire the necessary digital skills (see Sections 3.3.2 and 3.3.3).

3.4.3 The scientific community should acknowledge the value of open and sustainable software and reproducible research

The community should also acknowledge the contribution of RSEs to their scientific accomplishments by offering co-authorship of publications or engaging in collaborative efforts to publish their open software.

4 Conclusions

The importance of *research data management* for the quality and sustainability of research is now widely acknowledged — in the future, it must also ensured that research relying on computational methods is of high quality, sustainable and reproducible, and that a sound *research software management* is established. This means changes at different levels:

- Funding agencies must understand and support software (development) as an indispensable tool for high-quality and reproducible research because particularly in the Earth System Sciences, research increasingly builds on computational methods.
- Universities need to better support their researchers and students such that they can produce sustainable high-quality research software, including software-related infrastructure and guidance, teaching and assistance by RSEs.
- The scientific communities must deliberate which digital skills are essential in their fields for both students and ECRs.

Acknowledgments

The authors thank the Deutsche Forschungsgemeinschaft (DFG) for funding a meeting on 22/23 June 2023 in Hannover, Germany, (Rundgespräch, grant DO 737/22-1), during which the ideas for this paper were developed.

References

Anzt, H., et al. (2021): An environment for sustainable research software in Germany and beyond: current state, open challenges, and call for action. [version 2; peer review: 2 approved]. F1000Res2021, 9, 295. https://doi.org/10.12688/f1000research.23224.2

Barker, M., Chue Hong, N.P., van Eijnatten, J., Katz, D.S. (2023): Amsterdam Declaration on Funding Research Software Sustainability (0.3). Zenodo. https://doi.org/10.5281/zenodo.7740084

Barker, M. et al. (2022): Introducing the FAIR Principles for research software, Scientific Data, 9, 622. https://doi.org/10.1038/s41597-022-01710-x

Chue Hong, N.P., et al. (2022). FAIR Principles for Research Software (FAIR4RS Principles) (1.0). https://doi.org/10.15497/RDA00068

Hutton, C., et al. (2016): Most computational hydrology is not reproducible, so is it really science? Water Resour. Res., 52, 7548–55. https://doi.org/10.1002/2016WR019285

Kohrs, F.E., et al. (2023): Eleven strategies for making reproducible research and open science raining the norm at research institutions. OSF Preprints. May 28. https://doi.org/doi:10.31219/osf.io/kcvra.

Martinez-Ortiz, C., Martinez Lavanchy, P., Sesink, L., Olivier, B.G., Meakin, J., de Jong, M., Cruz, M. (2022): Practical guide to Software Management Plans (1.0). Zenodo. https://doi.org/10.5281/zenodo.7248877

Netherlands eScience Center (2023): What is a Research Software Engineer? A definition by the Netherlands eScience Center. Zenodo. https://doi.org/10.5281/zenodo.7994286

Nüst, D., et al. (2020): Ten simple rules for writing Dockerfiles for reproducible data science. PLoS Comput Biol 16(11): e1008316. https://doi.org/10.1371/journal.pcbi.1008316

Stagge, J.H., et al. (2019): Assessing data availability and research reproducibility in hydrology and water resources, Scientific Data, 6:190030. https://doi.org/10.1038/sdata.2019.30

Stodden, V., Zeiler, J., Ma, Z. (2018): An empirical analysis of journal policy effectiveness for computational reproducibility. PNAS, 115, 2584-2589. https://doi.org/10.1073/pnas.1708290115

Sufi, S., Jay, C. (2018): Raising the status of software in research: A survey-based evaluation of the Software Sustainability Institute Fellowship Programme [version 1; peer review: 3 approved with reservations]. F1000Research 2018, 7:1599. https://doi.org/10.12688/f1000research.16231.1

Wilson, G., Bryan, J., Cranston, K., Kitzes, J., Nederbragt, L., Teal, T.K. (2017): Good enough practices in scientific computing. PLoS Comput Biol 13(6): e1005510. https://doi.org/10.1371/journal.pcbi.1005510

Wilson, G., et al. (2014): Best Practices for Scientific Computing. PLoS Biol 12(1): e1001745. https://doi.org/10.1371/journal.pbio.1001745