Reproducibility in Scientific Software

PASC18, Basel, Switzerland

Michael A. Heroux
Senior Scientist, Sandia National Laboratories
Scientist in Residence, St. John’s University, MN
License, citation, and acknowledgments

License and Citation

• This work is licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).


Acknowledgements

• This work was supported by the U.S. Department of Energy Office of Science, Office of Advanced Scientific Computing Research (ASCR), and by the Exascale Computing Project (17-SC-20-SC), a collaborative effort of the U.S. Department of Energy Office of Science and the National Nuclear Security Administration.

• Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy’s National Nuclear Security Administration under contract DE-NA0003525.
Outline

• My Background (for context).
• Increasing focus on reproducibility.
  – Publication requirements.
  – Trustworthiness at Scale.
• Role of
  – Better software practices.
  – Funding Agencies.
  – Community Dynamics.
• Personal Productivity Commitment.
My Software History (Informs My Priorities)

• Cray:
  – Libsci (5 years): Dense, Sparse LA.
  – Eng apps (5 years): FEM, FVM, Solvers.

• Sandia:
  – Trilinos (18 years): Scientific libraries.
  – Mantevo (10 years): Miniapps.
  – HPCG Benchmark (6 years): TOP500 benchmark.
  – IDEAS Productivity (5 years): Productivity & Sustainability.
  – ECP Math Libs (1 year): xSDK (hypre, PETSc, SuperLU, Trilinos, …).
  – ECP SW Director (9 months): Portfolio of 55 projects, full software stack.
Reproducibility Challenge & Terms
Reproducibility

- NY Times highlights “problems”.
- Only one of many cited examples.
- HPC has been spared this “spotlight” (so far).

- Lots of activity:
  - AAAS, ACM initiatives.
  - PPoPP, Supercomputing Conference Series.

Reproducibility Terminology

• **Reviewable Research.** The descriptions of the research methods can be independently assessed and the results judged credible. (This includes both traditional peer review and community review, and does not necessarily imply reproducibility.)

• **Replicable Research.** Tools are made available that would allow one to duplicate the results of the research, for example by running the authors’ code to produce the plots shown in the publication. (Here tools might be limited in scope, e.g., only essential data or executables, and might only be made available to referees or only upon request.)

• **Confirmable Research.** The main conclusions of the research can be attained independently without the use of software provided by the author. (But using the complete description of algorithms and methodology provided in the publication and any supplementary materials.)

• **Auditable Research.** Sufficient records (including data and software) have been archived so that the research can be defended later if necessary or differences between independent confirmations resolved. The archive might be private, as with traditional laboratory notebooks.

• **Open or Reproducible Research.** Auditable research made openly available. This comprised well-documented and fully open code and data that are publicly available that would allow one to (a) fully audit the computational procedure, (b) replicate and also independently reproduce the results of the research, and (c) extend the results or apply the method to new problems.

Several other taxonomies. In particular, Clarbout's.
Increasing Reproducibility Expectations

Trends in Publications
Reproducibility Terminology

- **Reviewable Research.** The descriptions of the research methods can be independently assessed and the results judged credible. (This includes both traditional peer review and community review, and does not necessarily imply reproducibility.)

- **Replicable Research.** Tools are made available that would allow one to duplicate the results of the research, for example by running the authors’ code to produce the plots shown in the publication. (Here tools might be limited in scope, e.g., only essential data or executables, and might only be made available to referees or only upon request.)

- **Confirmable Research.** The main conclusions of the research can be attained independently without the use of software provided by the author. (But using the complete description of algorithms and methodology provided in the publication and any supplementary materials.)

- **Auditable Research.** Sufficient records (including data and software) have been archived so that the research can be defended later if necessary or differences between independent confirmations resolved. The archive might be private, as with traditional laboratory notebooks.

- **Open or Reproducible Research.** Auditable research made openly available. This comprised well-documented and fully open code and data that are publicly available that would allow one to (a) fully audit the computational procedure, (b) replicate and also independently reproduce the results of the research, and (c) extend the results or apply the method to new problems.

---

ACM TOMS Replicated Computational Results (RCR)

• Submission: Optional RCR option.

• Standard reviewer assignment: Nothing changes.

• RCR reviewer assignment:
  – Concurrent with standard reviews.
  – As early as possible in review process.
  – Known to and works with authors during the RCR process.

• RCR process:
  – Multi-faceted approach, Bottom line: Trust the reviewer.

• Publication:
  – Replicated Computational Results Designation.
  – The RCR referee acknowledged.
  – Review report appears with published manuscript.
RCR Process: Two Basic Approaches

1. Independent replication (3 options):
   A. Transfer of, or pointer to, author’s software.
   B. Guest account, access to author’s software.
   C. Observation of authors replicating results.

Or (Now used in SC18 Conference)

2. Review of computational results artifacts:
   – Results may be from an unavailable system.
   – Leadership class computing system.
   – In this situation:
     • Careful documentation of the process.
     • Software should have its own substantial V&V process.

TOMS:
• First RCR paper in TOMS issue 41:3
  – Editorial introduction.
  – van Zee & van de Geijn, BLIS paper.
  – Referee report.
• Second: TOMS 42:1
  – Hogg & Scott.
• Third: TOMS 42:4.
• More in the meantime.

TOMACS
• Similar.
Big Picture of ACM RCR

• Improve science.
  – Quality of prose: Good.
  – Quality of data: Poor.

• So bad now:
  – Trust comes from seeing a “cloud” of similar papers with similar results.
  – Which could still be wrong (built on a common bad piece).
  – Replicability: First step toward improvement.

• Engage a “dark portion” of the R&D community.
  – Reviewers not among typical reviewer pool.
  – Practitioners, users. Expert at use of Math SW.
  – Community overlaps with RSEs.

Thank you for taking the time to consider our paper for your journal.

XXX has agreed to undergo the RCR process should the paper proceed far enough in the review process to qualify. To make this easier we have preserved the exact copy of the code used for the results (including additional code for generating detailed statistics that is not in the library version of the code).
SC18 Reproducibility Initiative

• Two appendices:
  – Artifact description (AD).
    • Blue print for setting up your computational experiment.
    • Makes it easier to rerun computations in future.
    • AD appendix will be mandatory for SC19 paper submissions.
  – Artifact Evaluation (AE).
    • Targets "boutique" environments.
    • Improves trustworthiness when re-running hard, impossible.

• Details:
  – https://collegeville.github.io/sc-reproducibility/
Reproducibility and Supercomputing

Scenario:
You compute a “hero” calculation using 5M core-hours on Mira and submit your results for publication. During the review process, a referee questions the validity of your results. What options are feasible:

- The reviewer re-runs your code on a laptop or cluster.
- The reviewer re-runs your code on Mira.
- You re-run your code on Mira.
- Your results are rejected.
- Your results are accepted, but with risk.

One Approach to improve trustworthiness: Meta-computations.
Meta-computations to Improve Trustworthiness

Focus of Artifact Evaluation (AE) Appendix for Supercomputing Conference

• Synthetic operators with known:
  – Spectrum (Huge diagonals).
  – Rank (by constructions).

• Invariant subspaces:
  – Example: Positional/rotational invariance (structures).

• Conservation principles:
  – Example: Flux through a finite volume.

• General:
  – Pre-conditions, post-conditions, invariants.

Can you think of something for your problems?
Coming to Your World Now (or Soon): Reproducibility Requirements

• These conferences expect artifact evaluation appendices (most optionally):
  – CGO, PPoPP, PACT, RTSS and SC.
  – http://fursin.net/reproducibility.html

• ACM Replicated Computational Results (RCR).
  – ACM TOMS, TOMACS.
  – http://toms.acm.org/replicated-computational-results.cfm

• ACM Badging.
  – https://www.acm.org/publications/policies/artifact-review-badging

How can you prepare?
Common statement: “I would love to do a better software job, but I need to:
• Get this paper submitted.
• Complete this project task.
• Do something my employer and community values more.

Goal: Change incentives to include value of better software, improve reproducibility.
Productivity and Sustainability

What do we mean?
Objectives

• Productivity – Output per unit input.
• Sustainability – The future cost of maintaining usability.

• Productivity improvements realized sooner.
• Sustainability later.
Tradeoffs: Better, faster, cheaper

• “Better, faster, cheaper: Pick two of the three.”

– Scenario: (Today)
  You are behind in developing a sophisticated new model in your software that you want to use for results in an upcoming paper.

– Which of these could be reasonable choices?
  • Develop a simpler model for the paper.
  • Set other work aside and spend more time on development.
  • Ask for an extension on the paper deadline.
  • Develop sophisticated model, but don’t test its correctness.
  • Develop sophisticated model, but don’t document it or check it in.
Impact of Improved Developer Productivity


Scenario: (6 months later)

After investing in developer productivity improvements, you are on time in developing a sophisticated new model in your software that you want to use for results in an upcoming paper.

Invest in developer tools, processes, practices.
Impact of Improved Software Sustainability


Scenario: (3 years later)

After investing in **software sustainability improvements**, you are on time in developing **several** sophisticated new models in your software that you want to use for results in upcoming papers.

Invest in testing, documentation, integration for long-term software usability.
Which of These Enhance Reproducibility?

- Code written by first-year, untrained grad student.
- Tuning for high performance.
- Dynamic parallelism of modern processors.
- Better software testing.
- Source code and versioning management.
- Investing in developer productivity.
- Investing in software sustainability.
Common statement: “I would love to do a better software job, but I need to:
• Get this paper submitted.
• Complete this project task.
• Do something my employer and community values more.

Goal: Change incentives to include value of better software, improve reproducibility.
Software Quality Requirements

Getting Funding Agencies Involved
DOE SW Productivity and Sustainability Plan (SW PSP).

• Key Entities:
  – DOE Biological and Environmental Research (BER).
  – DOE Advanced Scientific Computing Research (ASCR)
  – IDEAS Project

• Milestone:
  – SW Productivity and Sustainability Plan.
DOE BER SW PSP Requirements

• Describe overall SW development process.
  – Software lifecycle, testing, documentation and training.

• Development tools and processes:
  – source management, issue tracking, regression testing, SW distribution.

• Training and transition:
  – New and departing team members.

• Continuous process improvement:
  – Getting better at productivity and sustainability.

Importance: Could be used to prioritize otherwise-similar proposals.
Promoting and Rewarding Improved Practices
Better Scientific Software (BSSW)

Scientific software has emerged as an essential discipline in its own right. Because computational models, computer architectures, and scientific software projects have become extremely complex, the Computational Science & Engineering (CSE) community now has a unique opportunity—and an implicit mandate—to address pressing challenges in scientific software productivity, quality, and sustainability.

We want and need contributions from the community … Join us!

Collaborative content development on general topics related to developer productivity and software sustainability for CSE.
BSSw Fellowship 2019

Start Fall 2018 we are looking for applications from people with the following characteristics:

• Passionate about scientific software.
• Interested in contributing powerful ideas, tools, methodologies, and more that improve the quality of scientific software.
• Able to use the fellowship to broadly benefit the scientific software community.
• Willing to participate as an alum in subsequent years to guide selection of future fellows and promote better scientific software in their community.

Experience:
• Describe your work relevant to scientific software (1000 - 1500 characters).
• Describe your background and experience relevant to being a BSSw Fellow (1000 - 1500 characters).

Proposed work and impact:
• What would you do as a BSSw Fellow? (1000 - 1500 characters).
• What impact do you foresee from your efforts? (1000 - 1500 characters).

Look for announcement for 2019 class in Fall 2018
Personal Expectations

Calling out the best in team members
A Few Concrete Recommendations

Show me the person making the most commits on an undisciplined software project and I will show you the person who is injecting the most technical debt.

• GitHub stats: Easy to find who made the most commits.
  – Some people: Pride in their high ranking.

• Instead, be the person who ranks high in these ways:
  – Writes up requirements, analysis and design, even if simple.
  – Writes good GitHub issues, tracks their progress to completion.
  – Comments on, tests and accepts pull requests.
  – Provide good wiki, gh-pages content, responses to user issues.
(Personal) Productivity++ Initiative
Ask: Is My Work _______ ?

https://github.com/trilinos/Trilinos/wiki/Productivity---Initiative
Summary

• Reproducibility demands are here and growing.
  – Conferences first, journals slower.

• HPC software is particularly challenging:
  – Hardware variation.
  – Code optimization.
  – Dynamic parallelism.

• Better software practices:
  – Improve chances for reproducibility.
  – Lower its cost.

• Funding agencies, community dynamics can help.
  – Can drive productivity & sustainability requirements.
  – Will further enable reproducibility efforts.

• Each of us can do our part personally, make a commitment.
Other resources


*Enhancing Reproducibility for Computational Methods.* Victoria Stodden, Marcia McNutt, David H. Bailey, Ewa Deelman, Yolanda Gil, Brooks Hanson, Michael A. Heroux, John P.A. Ioannidis, Michela Taufer *Science* (09 Dec 2016), pp. 1240-1241