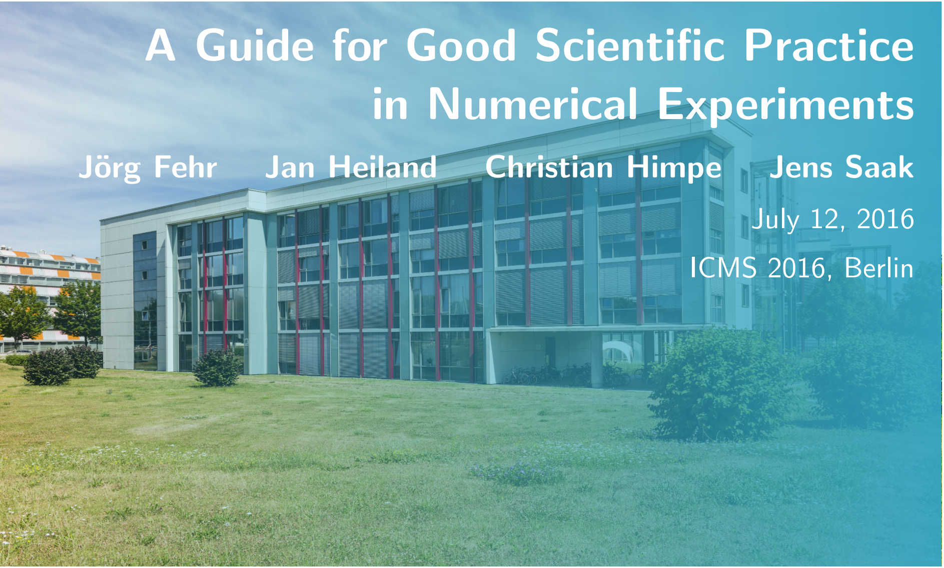
MAX PLANCK INSTITUTE
FOR DYNAMICS OF COMPLEX
TECHNICAL SYSTEMS
MAGDEBURG

CSC
COMPUTATIONAL METHODS IN
SYSTEMS AND CONTROL THEORY



A Guide for Good Scientific Practice in Numerical Experiments

Jörg Fehr Jan Heiland Christian Himpe Jens Saak

July 12, 2016

ICMS 2016, Berlin

Outline

1. Introduction
2. The three $\mathcal{R}$s
3. The Guide

Introduction

Observation (of an inadequacy):

- Numerical experiments gain importance as part of a publication
  - 1971 Nitsche *Über ein Variationsprinzip zur Lösung von Dirichlet-Problemen bei Verwendung von Teilräumen, die keinen Randbedingungen unterworfen sind* — no numerical experiments reported
  - 1986 Saad & Schultz: *GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems* — 2/14 pages are devoted to numerical experiments
  - 2010 Chataranbutat & Sorensen: *Nonlinear Model Reduction via Discrete Empirical Interpolation* — consists to more than 30% of numerical examples or reasonings based on numerical experiments
- still, there are no standards on their presentation, documentation, or publication

Introduction

Observation (of a chance):

- Science builds on previous findings
  - in theory – using e.g. theorems
  - and in practice – using established methods
- Often, the first step of a new scientific endeavor is to reproduce previous results
- Building upon numerical investigations can be incredibly easy
  - software is easy to share and to modify
  - hardware is easily replaceable

Introduction

There are many attempts towards standards for reporting on computer based experiments (CBEx) in view of ensuring reproducibility/reusability

- The *Science Code Manifesto* `sciencecodemanifesto.org`
- The *Recomputation Manifesto* `recomputation.org`
- LeVeque: *Top 10 Reasons to Not Share Your Code (and why you should anyways)*
- Marwick: *Computational Reproducibility in Archaeological Research: Basic Principles and a Case Study of Their Implementation*
- Buckheit & Denoho: *WaveLab and Reproducible Research* (1995)
- Bailey, Borwein, & Stodden: *Facilitating reproducibility in scientific computing: Principles and practice* (2016)

The three $\mathcal{R}$s

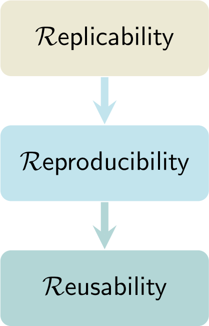
We argue that Good Scientific Practice is all about

- $\mathcal{R}$eplicability,
- $\mathcal{R}$eproducibility, and
- $\mathcal{R}$eusability

and discuss

- what $\mathcal{RRR}$ looks like for CBEx,
- how $\mathcal{RRR}$ forms a hierarchy, and
- how $\mathcal{RRR}$ can be implemented up to a desired or feasible level.

The three $\mathcal{R}$s



From the general concepts of $\mathcal{RRR}$ towards practical guidelines that

- are general enough
  - to be adapted to existing projects or approaches/paradigms/habits in software development
  - to be nonexclusive because of hard requirements like Open Source
- and concrete enough to be substantiated in simple rules that are easy to follow and to implement.

The three $\mathcal{R}$s

Definition

The attribute **Replicability** describes the ability to repeat a CBEx and to come to the same (in a numerical sense) results. Sometimes the equivalent term **Repeatability** is used for this experimental property.

- $\mathcal{R}$eplicability is a basic requirement of reliable software as well as of its result as it shows a certain robustness of the procedure against
  - statistical influences
  - and bias of the observer.
- Also, only replicable CBEx can serve as a benchmark to which new methods can be compared, cf. [Vitek & Kalibera '11].

$\mathcal{R}$eproducibility

Definition

Reproducibility of a CBEx means that it can be repeated by a different researcher in a different computer environment.

This is an adaption of the general concept of $\mathcal{R}$eproducibility that

- is key in science that relies on experiments,
- is a subject in the theory of science, and
- which absence in a significant fraction of publications in many research areas has shaped the term *reproducibility crisis* in recent years[1]; cf. also [Collberg, Proebsting, & Warren '04] on reproducibility in computer science.

[1] newyorker.com/tech/elements/the-crisis-in-social-psychology-that-isnt

$\mathcal{R}$eusability

Definition

In the sphere of CBEx, **Reusability** refers to the possibility to reuse the software or parts thereof for different purposes, in different environments, and by researchers other than the original authors.

- In particular, $\mathcal{R}$eusability enables the utilization of the test setup or parts of it for other experiments or related applications.
- Although theoretically, any bit of a software can be reused for different purposes, here, $\mathcal{R}$eusability applies only for reproducible parts.

The Guide

- **Replicability**

  Required: Basic Documentation
  Recommended: Automation & Testing

- **Reproducibility**

  Required: Extensive Documentation
  Recommended: Availability

- **Reusability**

  Required: Accessibility
  Recommended: Modularity, Software Management & Licensing

Further Reading

Cornell University Library

and The Alliance of Science Organisations in G

arXiv.org > cs > arXiv:1607.01191

Search or Article-id

Computer Science > Mathematical Software

Best Practices for Replicability, Reproducibility and Reusability of Computer-Based Experiments Exemplified by Model Reduction Software

Jörg Fehr, Jan Heiland, Christian Himpe, Jens Saak

(Submitted on 5 Jul 2016)

Over the recent years the importance of numerical experiments has gradually been more recognized. Nonetheless, sufficient documentation of how computational results have been obtained is often not available. Especially in the scientific computing and applied mathematics domain this is crucial, since numerical experiments are usually employed to verify the proposed hypothesis in a publication. This work aims to propose standards and best practices for the setup and publication of numerical experiments. Naturally, this amounts to a guideline for development, maintenance, and publication of numerical research

Arxiv preprint 1607.01191 featuring

- a list of related literature, websites, and blog posts,
- a general and CBEx specific discussion on $\mathcal{RRR}$,
- a blueprint for a *basic* and *extensive* documentation,
- link to an example repository.

References I

D. Bailey, J. Borwein, and V. Stodden.
Facilitating reproducibility in scientific computing: Principles and practice.
In H. Atmanspacher and S. Maasen, editors, *Reproducibility: Principles, Problems, Practices, and Prospects*, pages 205–232. Wiley, July 2016.
available from www.davidhbailey.com/dhbpapers/reprod.pdf.

J. B. Buckheit and D. L. Donoho.
WaveLab and Reproducible Research.
In A. Antoniadis and G. Oppenheim, editors, *Wavelets and Statistics*, volume 103 of *Lecture Notes in Statist.*, pages 55–81. Springer, New York, 1995.

References II

C. Collberg, T. Proebsting, and A. M. Warren.
Repeatability and Benefaction in Computer Systems Research.
Technical report, University of Arizona, 2014.

J. Fehr, J. Heiland, C. Himpe, and J. Saak.
Best Practices for Replicability, Reproducibility and Reusability of Computer-Based Experiments Exemplified by Model Reduction Software.
ArXiv e-prints, July 2016.
available from `arxiv.org/abs/1607.01191`.

R. J. LeVeque.
Top Ten Reasons To Not Share Your Code (and why you should anyway).
SIAM News, April 2013.

References III

B. Marwick.
Computational Reproducibility in Archaeological Research: Basic Principles and a Case Study of Their Implementation.
Journal of Archaeological Method and Theory, pages 1–27, 2016.

J. Vitek and T. Kalibera.
Repeatability, reproducibility, and rigor in systems research.
In *Proceedings of the 9th ACM international conference on Embedded software*, pages 33–38, 2011.

Thank you!

Thank you for your attention!

I am always open for discussion

heiland@mpi-magdeburg.mpg.de
www.janheiland.de
github.com/highlando