



MAX PLANCK INSTITUTE
FOR DYNAMICS OF COMPLEX
TECHNICAL SYSTEMS
MAGDEBURG



COMPUTATIONAL METHODS IN
SYSTEMS AND CONTROL THEORY

RRR

Replicability, Reproducibility, Reusability

Jörg Fehr, Jan Heiland, Christian Himpe, Jens Saak

2016-11-22

Helmholtz Open Science Workshop
“Access and reuse of scientific software”



1. About Us
2. Replicability, Reproducibility, Reusability
3. Best Practices



Who are we?

- Prof. Dr. Ing. Jörg Fehr (Uni Stuttgart)
- Dr. rer. nat. Jan Heiland (MPI Magdeburg)
- Dipl. Math. Christian Himpe (MPI Magdeburg; formerly: Uni Münster)
- Dr. rer. nat. Jens Saak (MPI Magdeburg)

What we have in common:

- Model Reduction (Applied Mathematics)
- Scientific Computing
- Small community, many methods
- Underwhelmed by published numerical results



Improve Computer-Based Experiments (CBEx):

- Define terminology
- Establish best-practices
- Ensure scientificity
- Discipline-agnostic guidelines
- A practical guide



*Review***Best practices for replicability, reproducibility and reusability of computer-based experiments exemplified by model reduction software****Jörg Fehr¹, Jan Heiland², Christian Himpe^{3,*}, and Jens Saak²**An open-access review article, [short-doi:bsb2](#)

Definition:

“The attribute **Replicability** describes the ability to repeat a CBEx and to come to the same (in a numerical sense) results.”

In practice:

- Requirement: Basic Documentation
- Recommendation: Automation & Testing

Definition:

“**Reproducibility** of a CBE_x means that it can be repeated by a different researcher in a different compute environment.”

In practice:

- Requirement: Extensive Documentation
- Recommendation: Availability

Definition:

“**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 practice:

- Requirement: Accessibility
- Recommendation: Modularity, Software Management & Licensing

Summary:

- Replicability ← This is a sanity check
- Reproducibility ← This makes it science
- Reusability ← This is a competitive advantage

Interrelations:

- Reproducibility requires Replicability
- Reusability requires Reproducibility



Outline:

- (Existence)
- (Function)
- Availability
- Usability
- (Comparability)

- Introduced by NATURE ([doi:10.1038/514536a](https://doi.org/10.1038/514536a), [doi:10.1038/sdata.2015.4](https://doi.org/10.1038/sdata.2015.4)).
- An additional section stating the availability of source code.
- Code should be shared [LeVeque'13]
- Obvious at a glance.
- No excuses.

Code Availability Section

The source code of the implementations used to compute the presented results can be obtained from:

`doi:????????/????????` and is authored by: X Y, A B.

Please contact X Y for licensing information.

- README - Every code should have a README file
 - Title, Version, Release-Date, Summary, Table-of-Contents, ...
- RUNME - Every scientific code should have a RUNME reproducing results
- LICENSE - Licensing contents
- AUTHORS - List of authors and contributors
- CITATION - Citation for the code
- DEPENDENCIES - required hardware & software
- CODE - Code meta-data (see next slide)
- More: CHANGELOG, FAQ, INSTALL, TODO
- Source file headers:
 - Project, Authors, Summary, ...

All in plain text!

Proposed by: [Katz & Smith'15]

We suggest: `.ini` format

Sample keys:

- name
- shortname
- version
- release-date
- id
- id-type
- authors
- orcids
- associated
- topic
- type
- license
- license-type
- repository
- repository-type
- languages
- dependencies
- systems
- website
- keywords

Got suggestions for additional keys?



```
name: Empirical Gramian Framework
shortname: emgr
version: 5.0
release-date: 2016-10-20
id: 10.5281/zenodo.162135
id-type: doi
author: Christian Himpe
orcid: 0000-0003-2194-6754
topic: Science, Mathematics, Model Reduction
type: Toolbox
license: 2-Clause BSD
license-type: open
repository: github.com/gramian/emgr
repository-type: git
language: Matlab
dependencies: Octave >=4.0, Matlab >=2016b
systems: Linux, Windows
website: gramian.de
keywords: Controllability, Observability, Model Reduction,
Reduced Order Modelling, Model Order Reduction
```



Feedback?

- Is there something you fundamentally disagree with?
- What is missing? i.e.: Compute environment specification ...
- Is it practical enough?

Future Activities?

- We need common terminology.
- Actively discourage greenfielding.
- No code (availability section), no acceptance.

<http://himpe.science>