Best Practices for Reproducibility

(and Model Reduction)

Christian Himpe (christian.himpe@uni-muenster.de)

WWU Münster
Institute for Computational and Applied Mathematics

MPI Magdeburg
22.04.15

Modus Operandi

Please (dis)agree with me slide by slide!

Utopia

Code should be ...

- Open (Cardware?)
- Documented (Generated XML documentation?)
- Readable (no CRTP?)
- Simple (less than 50 LoC per function?)

What does this mean?

Prior Work

- Science Code Manifesto

  (http://sciencecodemanifesto.org)
- The Recomputation Manifesto

  (http://arxiv.org/pdf/1304.3674v1.pdf)
- Software Evaluation Guide

  (http://www.software.ac.uk/software-evaluation-guide)
- ISO 25010

  (http://ryreitsma.blogspot.de/2011/07/software-has-new-quality-model-iso.html)

Code Availability Section[1]

"**Code Availability**
The source code of the implementations used to compute the presented results can be obtained from `doi:XXXXXXX/XXXXXXXX`"

- Code is treated as part of publication
- Code is expected by default to be available
- Suggests to reflect on importance of code
- Enables replicability

[1] `http://www.nature.com/sdata/for-authors/editorial-and-publishing-policies#code-avail`

Re*ility (for Scientific Computing)

$$\text{Replicability} \rightarrow \text{Reproducibility}^{2} \rightarrow \text{Reusability}$$

[2] Replicability is not Reproducibility: Nor is it Good Science

Open Code

Minimal (Replicability):

- Stable Location (Github, Zenodo, FigShare, RunMyCode, Supplemental Material)
- Licensing (Open Source, Permissive)
- Dependencies (Runtimes, Toolboxes, Libs, Propriety)
- Basic Info (Build, Run, Plot)

Optimal (Reusability):

- Version Control (Git, SVN, ...)
- Core / Experiment Separation (Adaptability)
- Tests (Sanity, Unit, Toy)
- Verification & Validation (Analytical, Numerical, Statistical, Benchmarks)

Software Specifications

Implementation:

- Language (Standard, Version)
- Parameters & Arguments (Config Files, Command Line)
- Benchmarks (Oberwolfach, NICONET, MORwiki, Format)

Environment:

- Compiler (Version, Options)
- Runtime (Version, Options)
- Libs (Version, Flavour, All? BLAS, LAPACK, libc? Dependencies?)
- OS (Version, Flavour)

Accessibility:

- Open Stack (Full / Alternative)
- Virtualization (Containers ie Docker)
- HPC Considerations (Topology, Scaling, Minimal Working Example)

Hardware Specifications

Minimal:

- # FPUs
- FPU SIMD Width
- FPU Clock
- RAM Amount
- RAM Rate, Channels
- RAM Clock

Encoding:

FFFF FF FF FFFF F F FF

Detailed:

- FPU Type (x86, ARM, PowerPC, GPU, Phi)
- # Sockets
- # Cores
- # Threads
- FPU SIMD Width
- FPU Clock[3]
- Caches (L1, L2, L3)
- Instructions Extensions
- RAM Amount
- RAM Rate
- RAM Channels
- RAM Clock
- DD Space
- DD Type
- DD Approx Speed
- NW Speed
- ...

[3] No Turbo Boost, Turbo Core, Dynamic Overclocking

Default Information

- README (Manpage Sections)

  http://stackoverflow.com/a/2304870

- LICENSE
- AUTHORS
- CODE meta data (ini, xml, json)

  http://www.arfon.org/json-ld-for-software-discovery-reuse-and-credit

- CITATION

  http://blog.rtwilson.com/encouraging-citation-of-software-introducing-citation-files/

- RUNME script / executable
- File Header Information

  https://www.gnu.org/software/octave/doc/interpreter/Function-Headers.html

Exchangable Backends

Is a certain BLAS backend calculating correctly / efficiently?
Compare! FlexiBLAS (http://www.mpi-magdeburg.mpg.de/projects/flexiblas)

Is my / a solver calculating correctly?
Compare! Open Interfaces (http://openinterfaces.org)

Is my / a model reduction technique working?
Compare! Benchmarks (http://modelreduction.org)

On Model Reduction

Timing:

- Offline Time / Online Time
- Wall Time / CPU Time

Quality:

- State-Space Norms (L_1, L_2, L_∞)
- Frequency-Space Norms (H_2, H_∞)
- RMS, Scaled H_∞, Special Norms[4]

Properties:

- Stability / Passivity
- Structure Preserving
- ...

[4] Baur/Beattie/Benner/Gugericin'11 "Interpolatory Projection Methods for Parametrized Model Reduction"

More MOR

- How to specify benchmarks? (Matrix Market, Nonlinear, ...)
- How to benchmark? (Test-System)
- What about input / control? (Impulse Response, Chirp, ...)
- Enforced solver for snapshot-based methods?
- Best error / reduced order in preset time / power?
- Gamification by ranking / scoring?

References

- Top 10 Reasons to Not Share Your Code (and why you should anyway)
- 12 Ways to Fool the Masses: Fast Forward to 2011
- Computational science: ...Error
- The Case for Open Computer Programs
- Publish Your Code: It Is Good Enough
- Open code for open science?
- Code Share
- Ctrl alt share
- Best Practices for Scientific Computing
- Ten Simple Rules for the Open Development of Scientific Software
- Reproducible Results: Challenges for Computational Science and the Scientific Method
- Setting the Default to Reproducible
- Reproducibility in Computational Science: Why, What, and How?
- Quo Vadis, Scientific Software?
- Publication and Citation of Scientific Software with Persistent Identifiers
- Best Practices for Computational Science: Software Infrastructure and Environments for Reproducible and Extensible Research
- The Real Software Crisis: Repeatability as a Core Value
- Should code be Released?
- Repeatability, Reproducibility and Rigor in Systems Research
- The Reality of Reproducibility in Computational Science

tl;dl

TODO IMHO:

- (Mandatory) Code Availability Section
- Automated (Standardized) System Info Retrieval
- Software / Reproducibility Questionaire
- Model Reduction Best Practices Guide

Thanks!