

Dissemination and Management of Computational Science Software

Matthew Knepley^{1,2}

¹Computation Institute
University of Chicago

²Department of Molecular Biology and Physiology
Rush University Medical Center

Sharing Data and Code in Computational Science
New Haven, CT November 21, 2009



What are the barriers
to reproducible computations
with large scientific codes?

Transparency is more than Open source

- Installation
 - Dependencies
- Analysis of output
 - Often partially proprietary
- Understanding the algorithm
 - Knuth
 - PETSc

Transparency is more than Open source

- Installation
 - Dependencies
- Analysis of output
 - Often partially proprietary
- Understanding the algorithm
 - Knuth
 - PETSc

Transparency is more than Open source

- Installation
 - Dependencies
- Analysis of output
 - Often partially proprietary
- Understanding the algorithm
 - Knuth
 - PETSc

Workability is more than Repeatability

- Alter parameters
- Change model
- Looking for limits of the method

- “Code citation”
- Potentially use version control information
- Like the polymath model
- What about good judgment?

- “Code citation”
- Potentially use version control information
- Like the polymath model
- What about good judgment?

- “Code citation”
- Potentially use version control information
- Like the polymath model
- What about good judgment?

- “Code citation”
- Potentially use version control information
- Like the polymath model
- What about good judgment?

- Good tools
- Installed infrastructure
- Good user support

Outline

1 Tools and Infrastructure

Location and Retrieval

“Where’s the Tarball”

- Version Control
 - Mercurial, Git, Subversion
- Hosting
 - BitBucket, GitHub, Launchpad
- Community involvement
 - arXiv

Configuration and Build

“It won’t run on my iPhone”

- Portability
 - PETSc **BuildSystem**, **autoconf**
- Dependencies
 - Does this work with UnsupportedGradStudentAMG?
- Configurable build
 - **SCons**, **Jam**, **CMake**

Testing

“They are identical in the eyeball norm”

- Unit tests
 - `cppUnit`
- Regression tests
 - `buildbot`
- Benchmarks
 - `Cigma`

Big Picture

- **Usability** is paramount
 - Need community buy-in
 - Need complete workflow
- Leverage **existing systems**
 - Adoption is much easier with the familiar
 - **arXiv**, package managers