IMPLEMENTING REPRODUCIBLE RESEARCH USING THE MADAGASCAR SOFTWARE PACKAGE

Sergey Fomel
Jackson School of Geosciences
The University of Texas at Austin
Back in 1991

Linux

Python

WWW
Reproducible Research

“It is a big chore for one researcher to reproduce the analysis and computational results of another [...] I discovered that this problem has a simple technological solution: illustrations (figures) in a technical document are made by programs and command scripts that along with required data should be linked to the document itself [...] This is hardly any extra work for the author, but it makes the document much more valuable to readers who possess the document in electronic form because they are able to track down the computations that lead to the illustrations.”

(Claerbout, 1991)
Claerbout’s Principle

“An article about computational science in a scientific publication is not the scholarship itself, it is merely advertising of the scholarship. The actual scholarship is the complete software development environment and the complete set of instructions which generated the figures.”

(Buckheit and Donoho, 1995)
Reproducible Research
Addressing the Need for Data and Code Sharing in Computational Science

By the Yale Law School Roundtable on Data and Code Sharing
In a nutshell, Madagascar...

... has had 12,757 commits made by 97 contributors representing 1,118,533 lines of code
... is mostly written in C with an average number of source code comments
... has a well established, mature codebase maintained by a large development team
... took an estimated 328 years of effort starting with its first commit in May, 2003
Contributors

Basic Earth Imaging

- Imaging in shot-geophone space by Jon F. Claerbout
- Downward continuation by Jon F. Claerbout
- Waves and Fourier sums by Jon F. Claerbout
- Zero-offset migration by Jon F. Claerbout
- Moveout, velocity, and stacking by Jon F. Claerbout
- Waves in strata by Jon F. Claerbout
- Adjoint operators by Jon F. Claerbout
- Field recording geometry by Jon F. Claerbout

Center for Wave Phenomena

- Wide-azimuth angle gathers for wave-equation migration by Paul Sava and Ioan Viad. Geophysics, 76, no. 3, S131-S141, (2011)
Seismic data interpolation using nonlinear shaping regularization

Next: Interpolation using shaping regularization  Up: Theory  Previous: Theory

Review of nonlinear shaping regularization

Supposing \( \mathbf{m} \) is a model vector and \( \mathbf{d} \) is the data after applying a forward operator \( \mathbf{F} \). Nonlinear shaping regularization is used for solving the following equation:

\[
\mathbf{F}[\mathbf{m}] = \mathbf{d},
\]

using an iterative framework:

\[
\mathbf{m}_{n+1} = \mathbf{S}[\mathbf{m}_n + \mathbf{B}[\mathbf{d} - \mathbf{F}[\mathbf{m}_n]]],
\]

where \([ \cdot ]\) means the forward operator \( \mathbf{F} \) is not limited to linear case. \( \mathbf{S} \) is the shaping operator which shapes the model to an admissible model iteratively and \( \mathbf{B} \) is the backward operator which provides an approximate mapping from data space to model space (Fomel, 2008). Specially, when \( \mathbf{B} \) is taken as the adjoint of the \( \mathbf{F} \) (in the linear case) or the adjoint of the Frechet derivative of \( \mathbf{F} \) (in the nonlinear case), and take \( \mathbf{S} \) as an identity operator, iteration 2 becomes a famous Landweber iteration (Landweber, 1951). Iteration 2 can get converged if the spectral radius of the operator on the right hand side is less than one (Collatz, 1966).
Figure 5. Field data demonstration for seismic interpolation using shaping regularization. (a) Original synthetic data. (b) Irregularly sampled section by randomly removing 30% traces. (c) Reconstructed section using shaping regularization after 20 iterations. (d) Reconstructed section using faster shaping regularization with 10 iterations.
Grey('sean-mask','color=j')
Grey('sean-mask2','color=j')

# plotting convergence diagram (dashed -> pocs, solid -> pocs)
Flow('SNR1','snrs1','cat axis=1 ${SOURCES[1:%d]} len(snrs1))
Flow('SNR2','snrs2','cat axis=1 ${SOURCES[1:%d]} len(snrs2))

Flow('SNRs-sean','SNR1 SNR2','cat axis=2 ${SOURCES[1]})
Graph('SNRs-sean','labell="Iteration no. #" label2=SNR unit2=dB dash=0,1')

Flow('SNR2-sean','SNR2','cp')
Graph('SNR2-sean','labell="Iteration no. #" symbol="*" symbolsz=10 label2=SNR unit2=dB dash=1')
End()
Figure 5: Field data demonstration for seismic interpolation. (a) Original synthetic data. (b) Irregularly sampled data after down-sampling 30% traces. (c) Reconstructed section using shaping regularization after 25 iterations. (d) Reconstructed section using faster shaping regularization with 10 iterations.

```
sean/ seann, seann-zero, seann-recon-o, seann-recon
```
Research Pyramid

- Implement
- Apply & Test
- Publish
Madagascar Research Pyramid

- 1,600 Programs
- 8,000 Figures
- 900 Workflows
- 200 Papers
Lessons

• Reproducibility in itself is not the goal.

Lessons

• Reproducibility in itself is not the goal.
• The main beneficiary is the author.

Lessons

• Reproducibility in itself is not the goal.
• The main beneficiary is the author.
• Each computation is a test.

Lessons

• Reproducibility in itself is not the goal.
• The main beneficiary is the author.
• Each computation is a test.
• Reproducibility requires maintenance, maintenance requires an open community.

Obstacles to Adoption

• Resistance to culture change
• Copyright issues
  • Software licenses
  • Publishers
• Publication tools
  • LaTeX2HTML
  • PDF attachments
  • Jupyter notebooks
Conclusions

- Reproducible research: the discipline of attaching software and data to publications
- It works!
  - Continuous maintenance
  - Open community
- Need for a culture change
- Need for publication tools

http://ahay.org