MADAGASCAR

BUREAU OF ECONOMIC GEOLOGY

IMPLEMENTING REPRODUCIBLE RESEARCH USING THE MADAGASCAR SOFTWARE PACKAGE

Sergey Fomel
Jackson School of Geosciences
The University of Texas at Austin

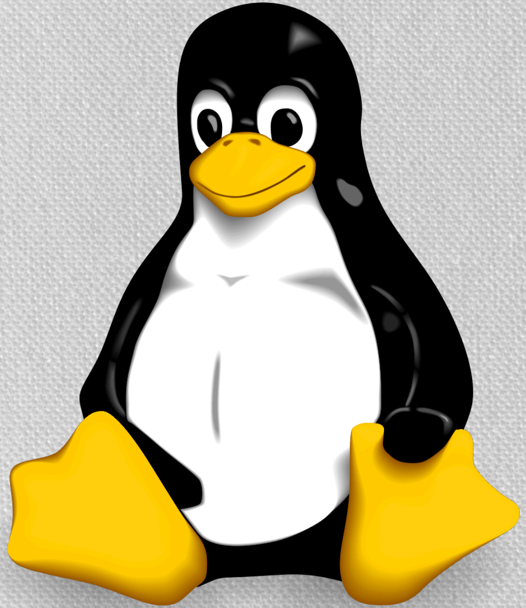
AGU FALL MEETING

San Francisco | 14 – 18 December 2015

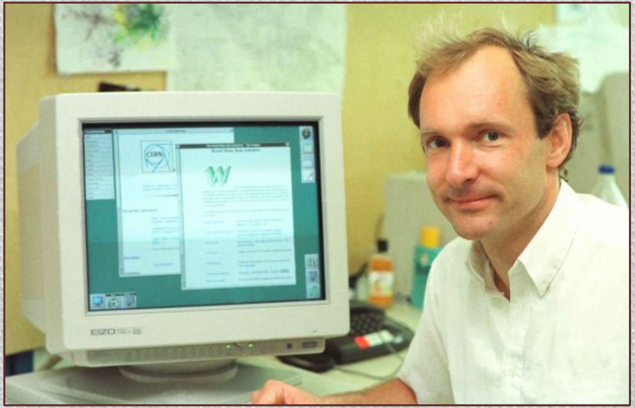
```
[root@sdawes ~]# iostat
Linux 4.1.15-1.1381_SKYN12nnmp (sdawes.sky.net)
T-800 ~]# status     %0.39    %0.89
project:  %user   %nice   %sys %iowait  %idle
           0.04    0.00    0.33   0.39   99.23

Device:      tps   Blk_read/s   Blk_wrtn/s   Blk_read   Blk
sda         7.55       220.83       125.74   1357670665 7730
sdb         0.26         1.72         2.42   10603032   1486009
```

```
0 Dec 19 16:13 execdomains
0 Dec 19 16:13 fb
0 Dec 19 16:13 filesystems
0 Dec 19 16:13 fs
0 Dec 19 16:13 ide
0 Dec 19 16:13 interrupts
0 Dec 19 16:13 iomem
0 Dec 19 16:13 ioports
0 Dec 19 16:13 irq
```

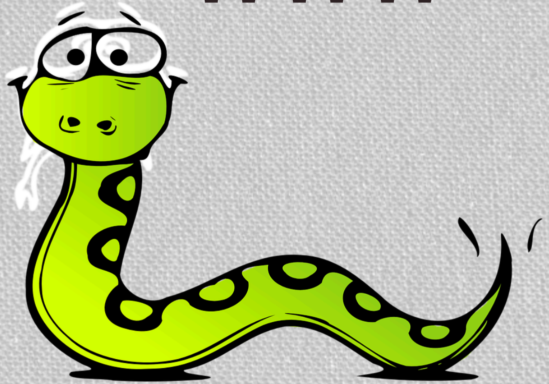
Back in 1991



Linux



WWW



Python

Reproducible Research

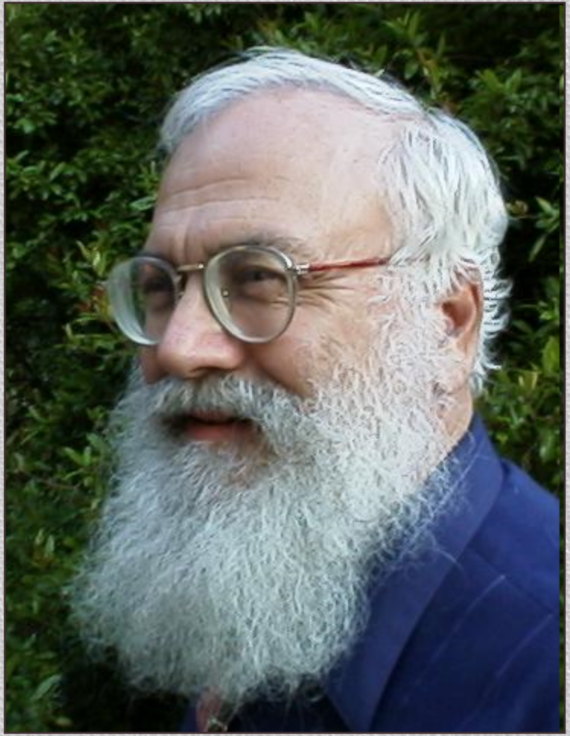
"It is a big chore for one researcher to reproduce the analysis and computational results of another […] I discovered that this problem has a simple technological solution: illustrations (figures) in a technical document are made by **programs and command scripts that along with required data should 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)



Computing in SCIENCE & ENGINEERING
p. 67 Interactive Holography | p. 75 Software Development with pMatlab
REPRODUCIBLE RESEARCH
IEEE
computer society
Computing in SCIENCE & ENGINEERING
p. 49 Testing Scientific Programs | p. 98 Maximized Dataflow Engines
Reproducible Research
IEEE
AIP
computer society
NEWS

Reproducible Research in Signal Processing

[What, why, and how]

[Patrick Vandewalle, Jelena Kovačević, and Martin Vetterli]

Reproducible Research

Addressing the Need for Data and Code Sharing in Computational Science

By the Yale Law School Roundtable on Data and Code Sharing

MADAGASCAR

http://ahay.org

From openhub.net

In a nutshell, Madagascar…

… has had 12,757 commits made by 97 contributors representing 1,118,533 lines of code

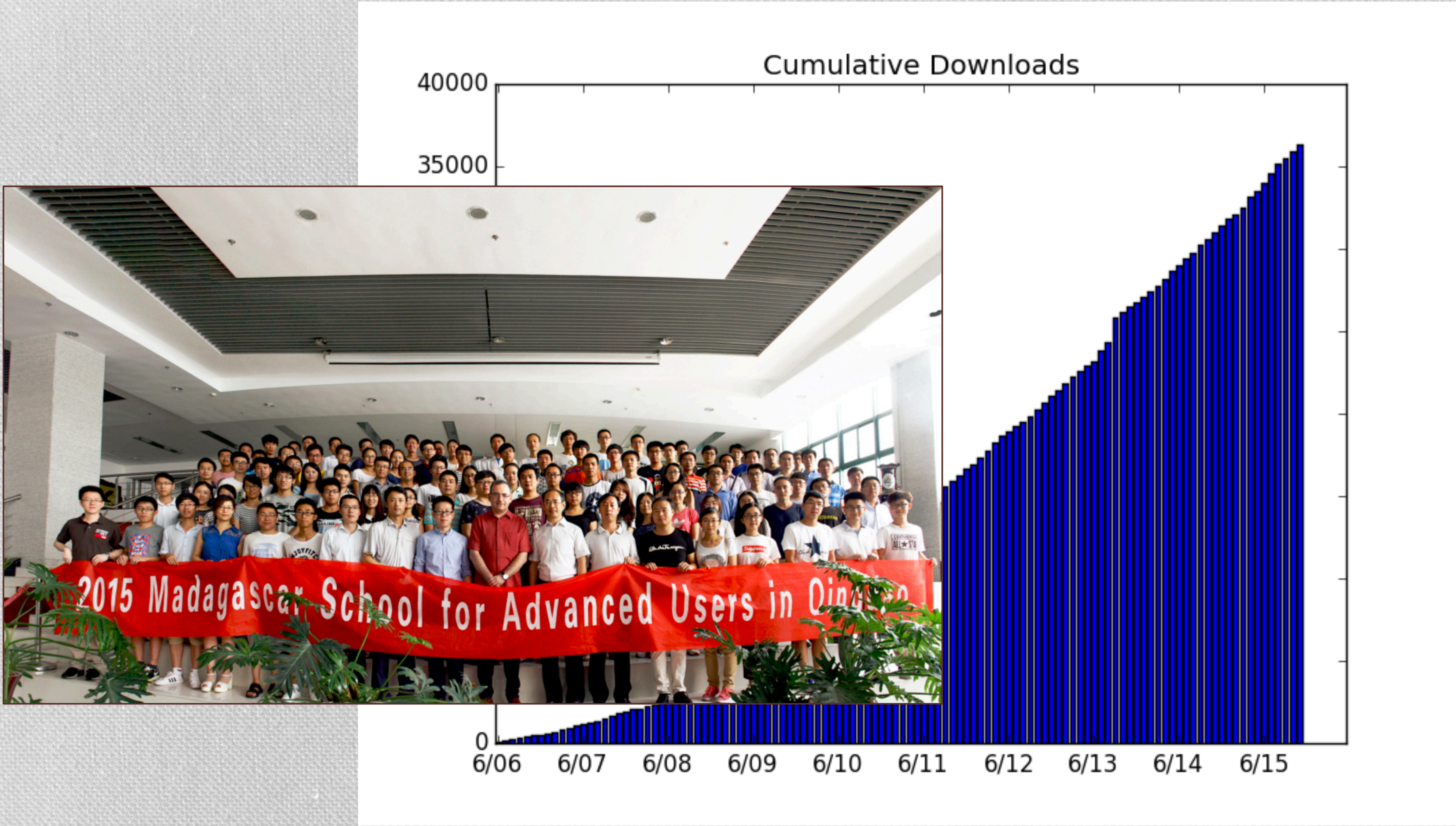
… is mostly written in C with an average number of source code comments

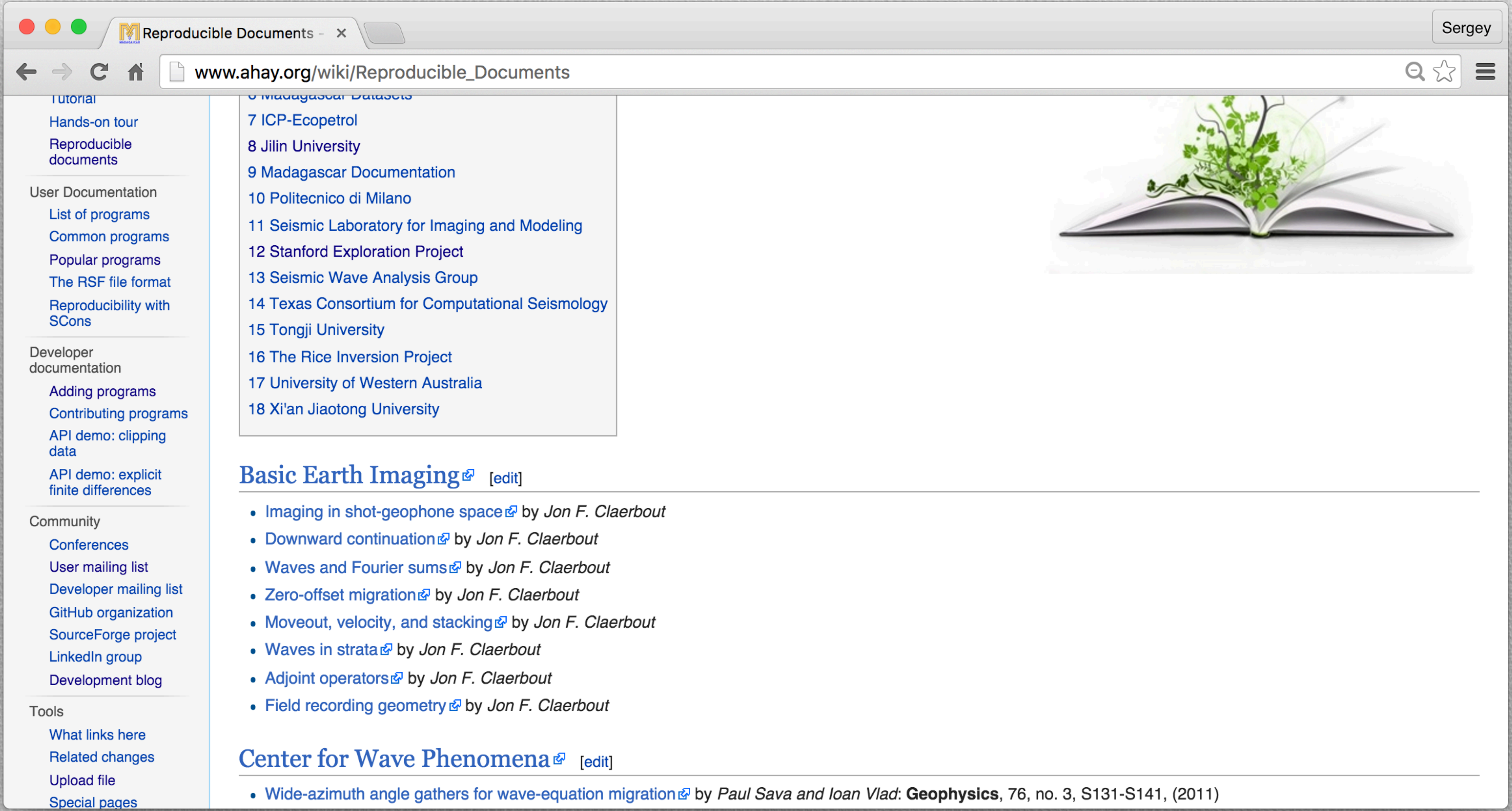
… **has a well established, mature codebase maintained by a large development team**

… took an estimated 328 years of effort starting with its first commit in May, 2003

Contributors

Salah Al-Hadab, Tariq Alkhalifah, Vladimir Bashkardin, Filippo Broggini, Jules Browaeys, Cody Brown, William Burnett, Yihua Cai, Maria Cameron, Lorenzo Casasanta, Yangkang Chen, Zhonghuan Chen, Jiubing Cheng, Luke Decker, Joseph Dellinger, Sjoerd de Ridder, Esteban Diaz, Yuting Duan, Gang Fang, Mehdi Far, Sergey Fomel, Jeff Godwin, Gilles Hennenfent, Jie Hou, Jingwei Hu, Yin Huang, Trevor Irons, Detchai Ittharat, Jim Jennings, Jun Ji, Long Jin, Parvaneh Karimi, Roman Kazinnik, Alexander Klokov, Siwei Li, Guochang Liu, Yang Liu, Yujin Liu, Xuxin Ma, Douglas McCowan, Henryk Modzelewski, Jorge Monsegny, Francesco Perrone, Jack Poulson, Kelly Regimbal, James Rickett, Daniel Rocha, Sean Ross-Ross, Colin Russell, Christos Saragiotis, Paul Sava, Karl Schleicher, Reza Shahidi, Jeffrey Shragge, Eduardo Filpo Silva, Xiaolei Song, Yanadet Sripanich, Junzhe Sun, Ioan Vlad, Hui Wang, Robin Weiss, Zedong Wu, Zhiguang Xue, Jia Yan, Pengliang Yang, Lexing Ying, Zhendong Zhang, Hejun Zhu


Cumulative Downloads
40000
35000
0
6/06
6/07
6/08
6/09
6/10
6/11
6/12
6/13
6/14
6/15
2015 Madagascar School for Advanced Users in Qingdao



- Tutorial
- Hands-on tour
- Reproducible documents

User Documentation

- List of programs
- Common programs
- Popular programs
- The RSF file format
- Reproducibility with SCons

Developer documentation

- Adding programs
- Contributing programs
- API demo: clipping data
- API demo: explicit finite differences

Community

- Conferences
- User mailing list
- Developer mailing list
- GitHub organization
- SourceForge project
- LinkedIn group
- Development blog

Tools

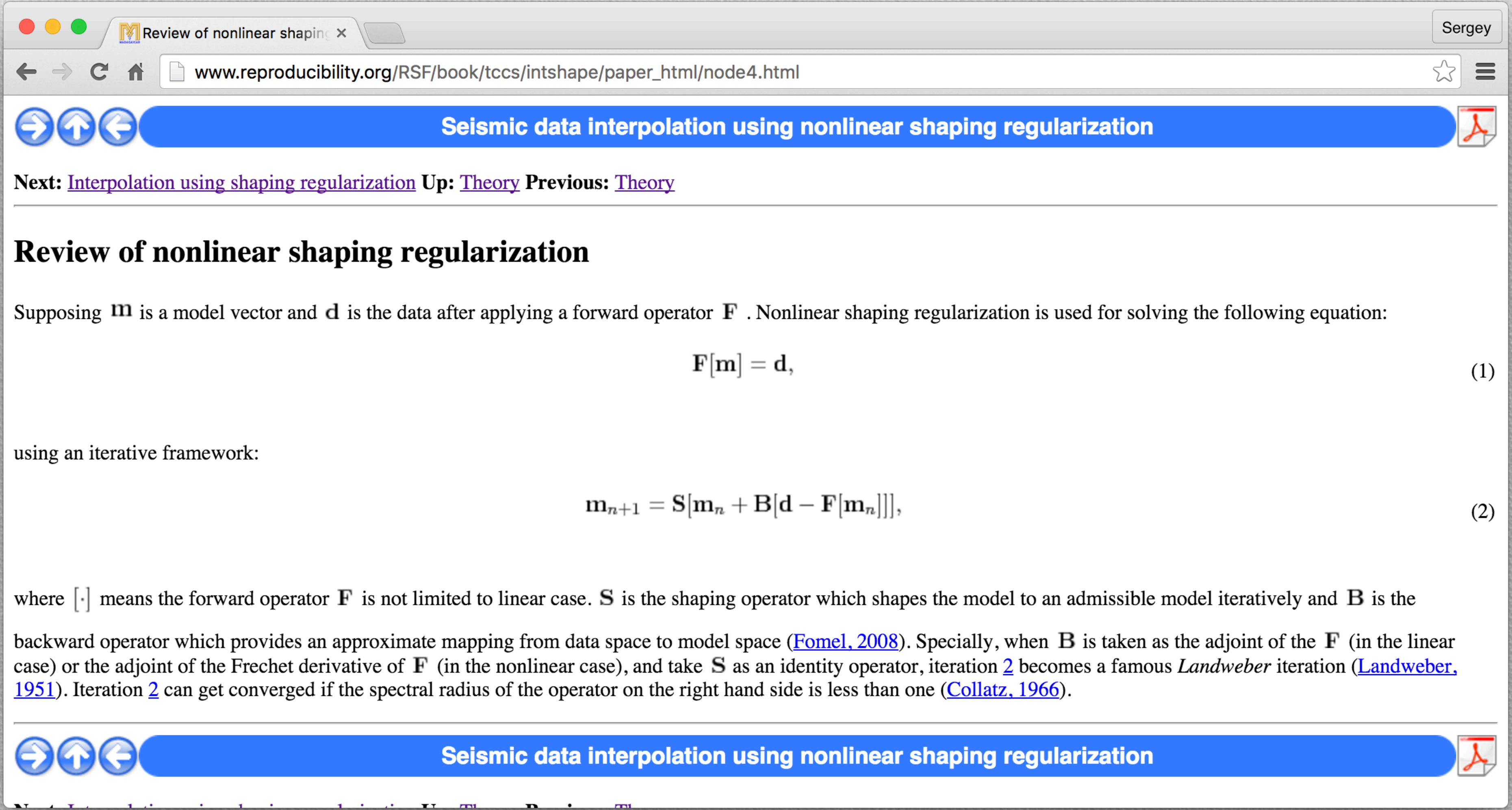
- What links here
- Related changes
- Upload file
- Special pages

Basic Earth Imaging [edit]

- 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 [edit]

- Wide-azimuth angle gathers for wave-equation migration by *Paul Sava and Ioan Vlad*: **Geophysics**, 76, no. 3, S131-S141, (2011)


Review of nonlinear shaping ×
Sergey
www.reproducibility.org/RSF/book/tccs/intshape/paper_html/node4.html

Review of nonlinear shaping regularization

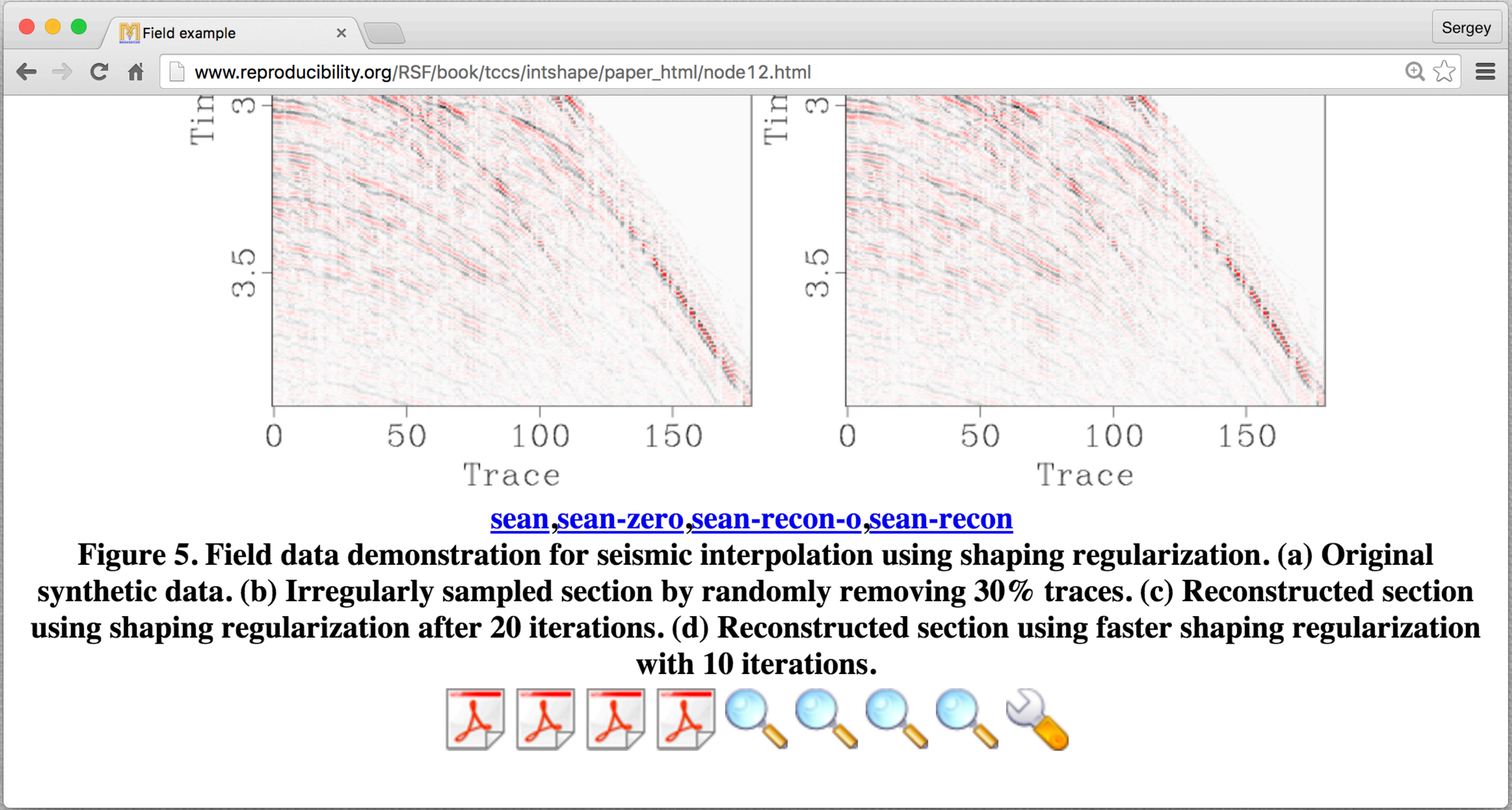
Supposing $\mathbf{m}$ is a model vector and $\mathbf{d}$ is the data after applying a forward operator $\mathbf{F}$. Nonlinear shaping regularization is used for solving the following equation:

$$\mathbf{F}[\mathbf{m}] = \mathbf{d}, \tag{1}$$

using an iterative framework:

$$\mathbf{m}_{n+1} = \mathbf{S}[\mathbf{m}_n + \mathbf{B}[\mathbf{d} - \mathbf{F}[\mathbf{m}_n]]], \tag{2}$$

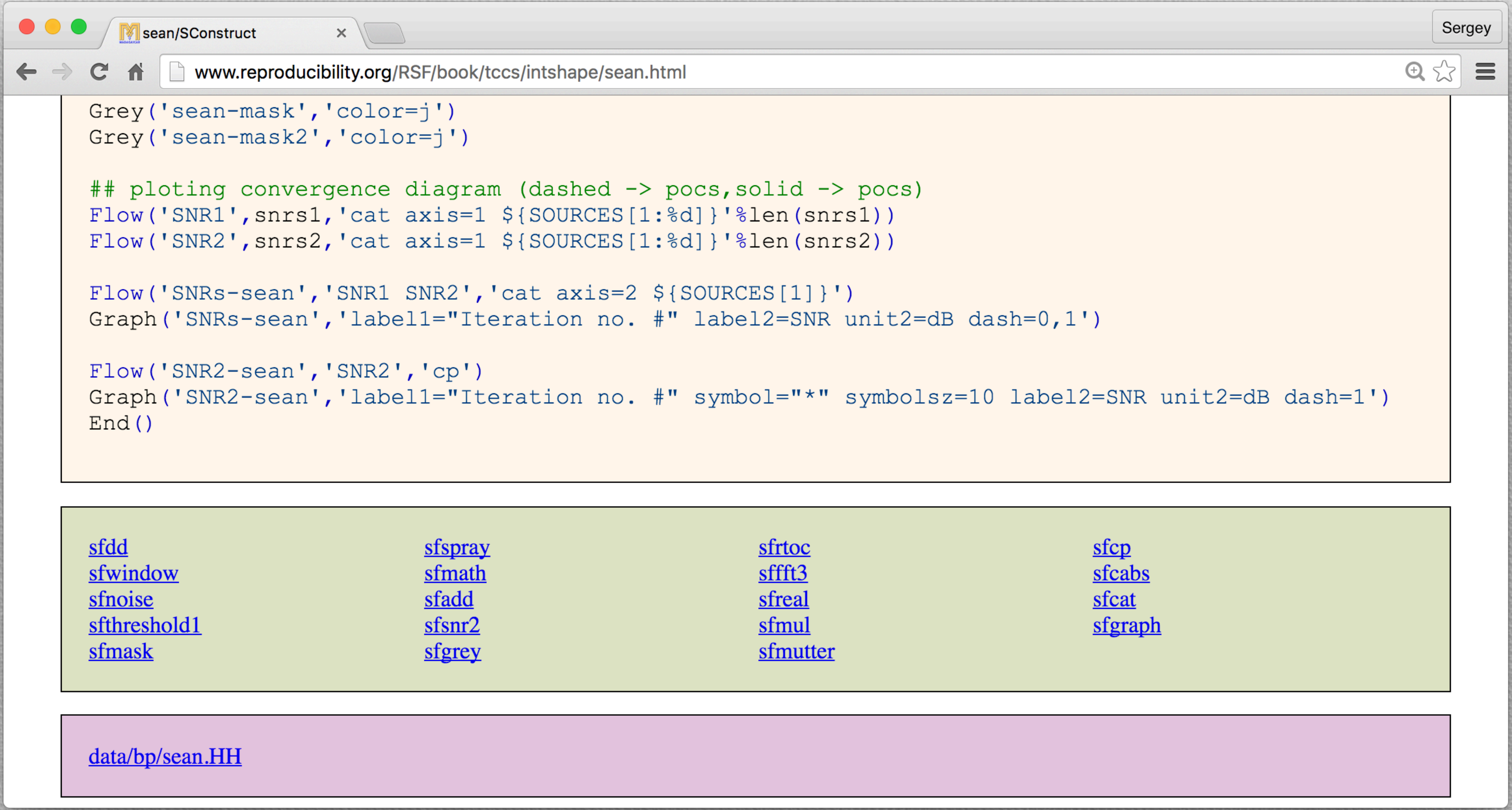
where $[\cdot]$ means the forward operator $\mathbf{F}$ is not limited to linear case. $\mathbf{S}$ is the shaping operator which shapes the model to an admissible model iteratively and $\mathbf{B}$ is the backward operator which provides an approximate mapping from data space to model space (Fomel, 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).



www.reproducibility.org/RSF/book/tccs/intshape/paper_html/node12.html

sean,sean-zero,sean-recon-o,sean-recon

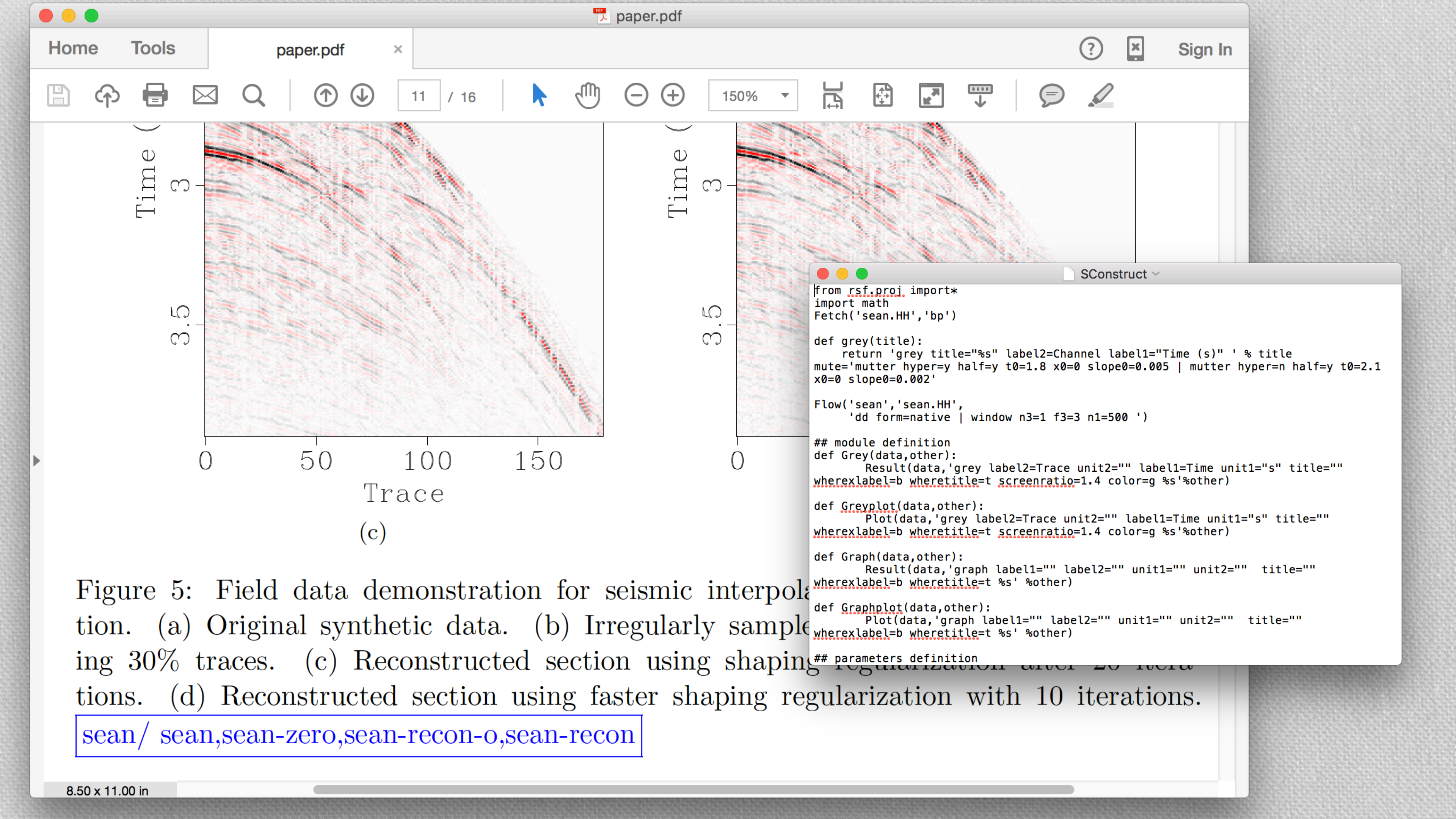
Figure 5. Field data demonstration for seismic interpolation using shaping regularization. (a) Original synthetic data. (b) Irregularly sampled section by randomly removing 30% traces. (c) Reconstructed section using shaping regularization after 20 iterations. (d) Reconstructed section using faster shaping regularization with 10 iterations.



```
Grey('sean-mask','color=j')
Grey('sean-mask2','color=j')

## ploting 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','label1="Iteration no. #" label2=SNR unit2=dB dash=0,1')

Flow('SNR2-sean','SNR2','cp')
Graph('SNR2-sean','label1="Iteration no. #" symbol="*" symbolsz=10 label2=SNR unit2=dB dash=1')
End()
```

sfdd
sfwindow
sfnoise
sfthreshold1
sfmask
sfspray
sfmath
sfadd
sfsnr2
sfgrey
sfrtoc
sffft3
sfreal
sfmul
sfmutter
sfcp
sfcabs
sfcat
sfgraph

data/bp/sean.HH

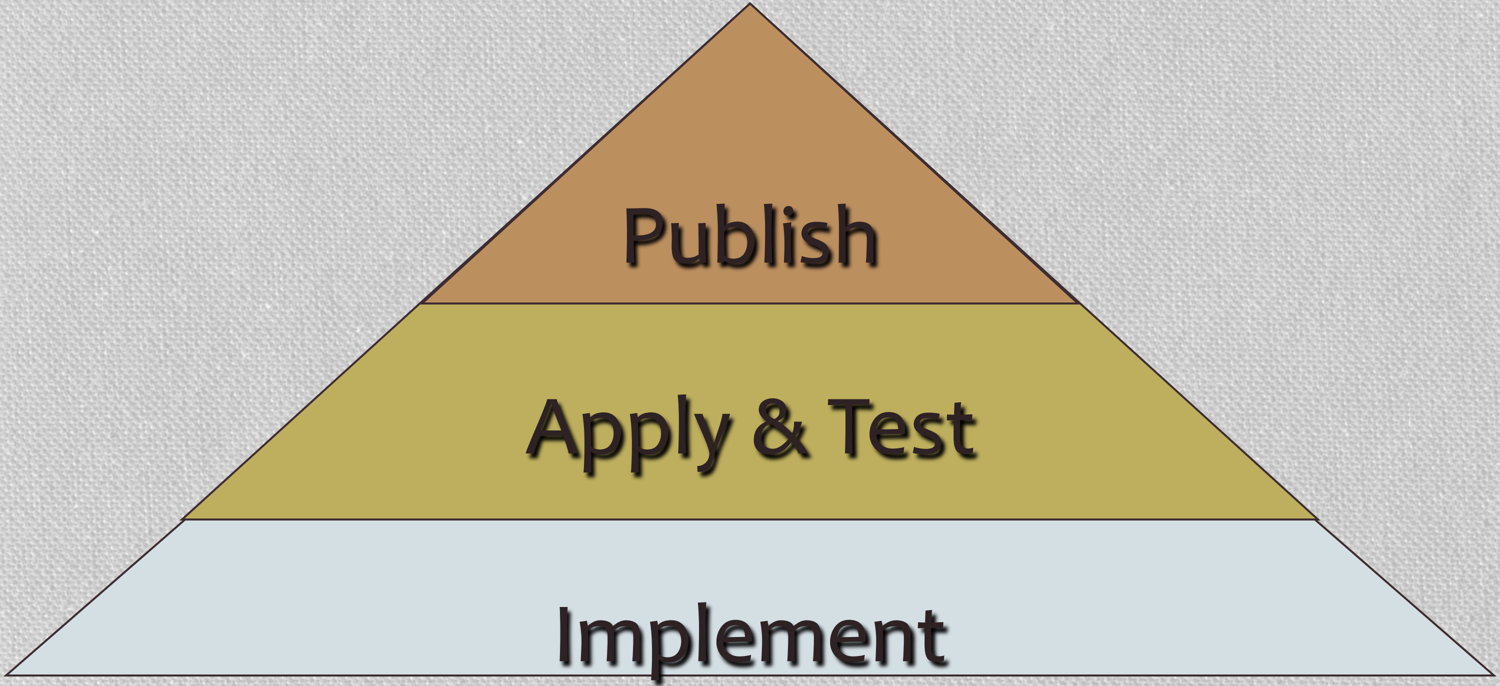


(c)

Figure 5: Field data demonstration for seismic interpola
tion. (a) Original synthetic data. (b) Irregularly sample
ing 30% traces. (c) Reconstructed section using shaping regularization after 20 itera
tions. (d) Reconstructed section using faster shaping regularization with 10 iterations.
sean/ sean,sean-zero,sean-recon-o,sean-recon

```
from rsf.proj import*
import math
Fetch('sean.HH','bp')

def grey(title):
    return 'grey title="%s" label2=Channel label1="Time (s)" ' % title
mute='mutter hyper=y half=y t0=1.8 x0=0 slope0=0.005 | mutter hyper=n half=y t0=2.1
x0=0 slope0=0.002'

Flow('sean','sean.HH',
     'dd form=native | window n3=1 f3=3 n1=500 ')

## module definition
def Grey(data,other):
       Result(data,'grey label2=Trace unit2="" label1=Time unit1="s" title=""
wherexlabel=b wheretitle=t screenratio=1.4 color=g %s'%other)

def Greyplot(data,other):
       Plot(data,'grey label2=Trace unit2="" label1=Time unit1="s" title=""
wherexlabel=b wheretitle=t screenratio=1.4 color=g %s'%other)

def Graph(data,other):
       Result(data,'graph label1="" label2="" unit1="" unit2=""  title=""
wherexlabel=b wheretitle=t %s' %other)

def Graphplot(data,other):
       Plot(data,'graph label1="" label2="" unit1="" unit2=""  title=""
wherexlabel=b wheretitle=t %s' %other)

## parameters definition
```

Research Pyramid

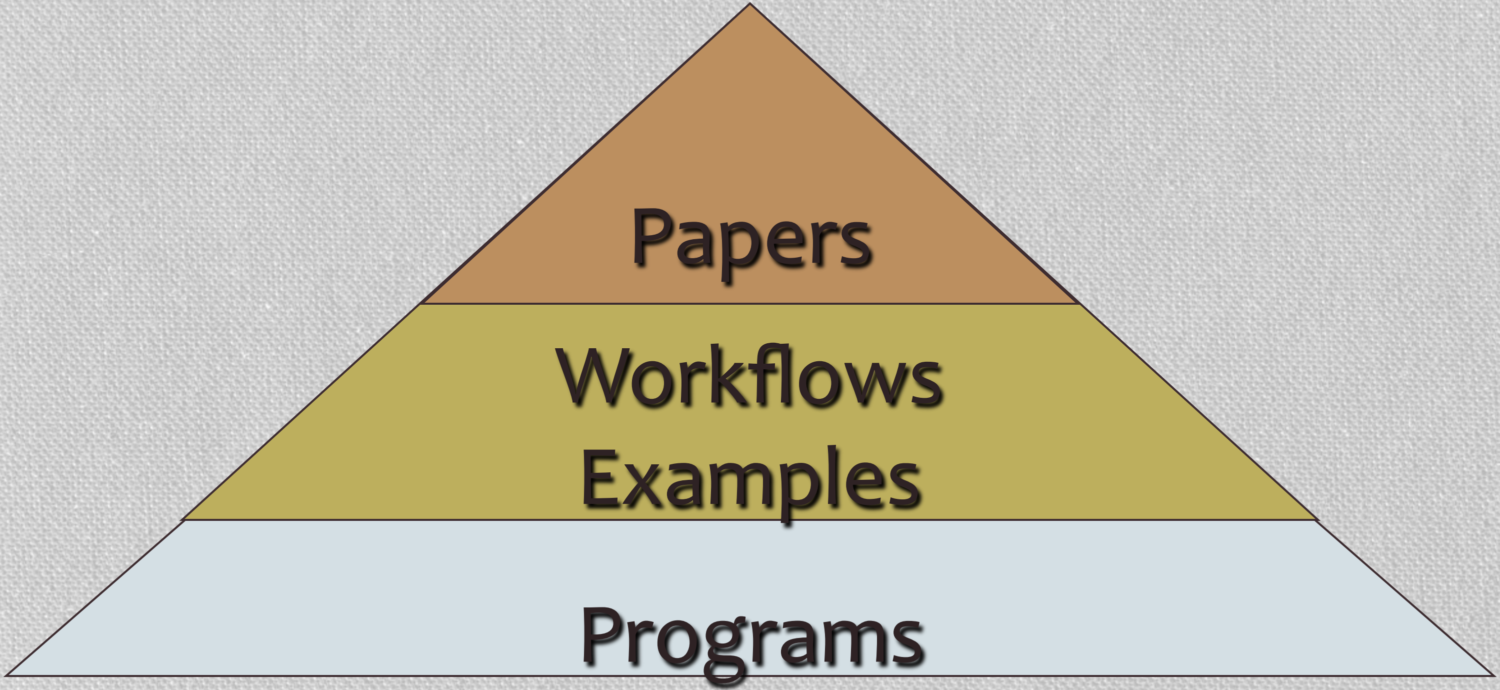


Publish

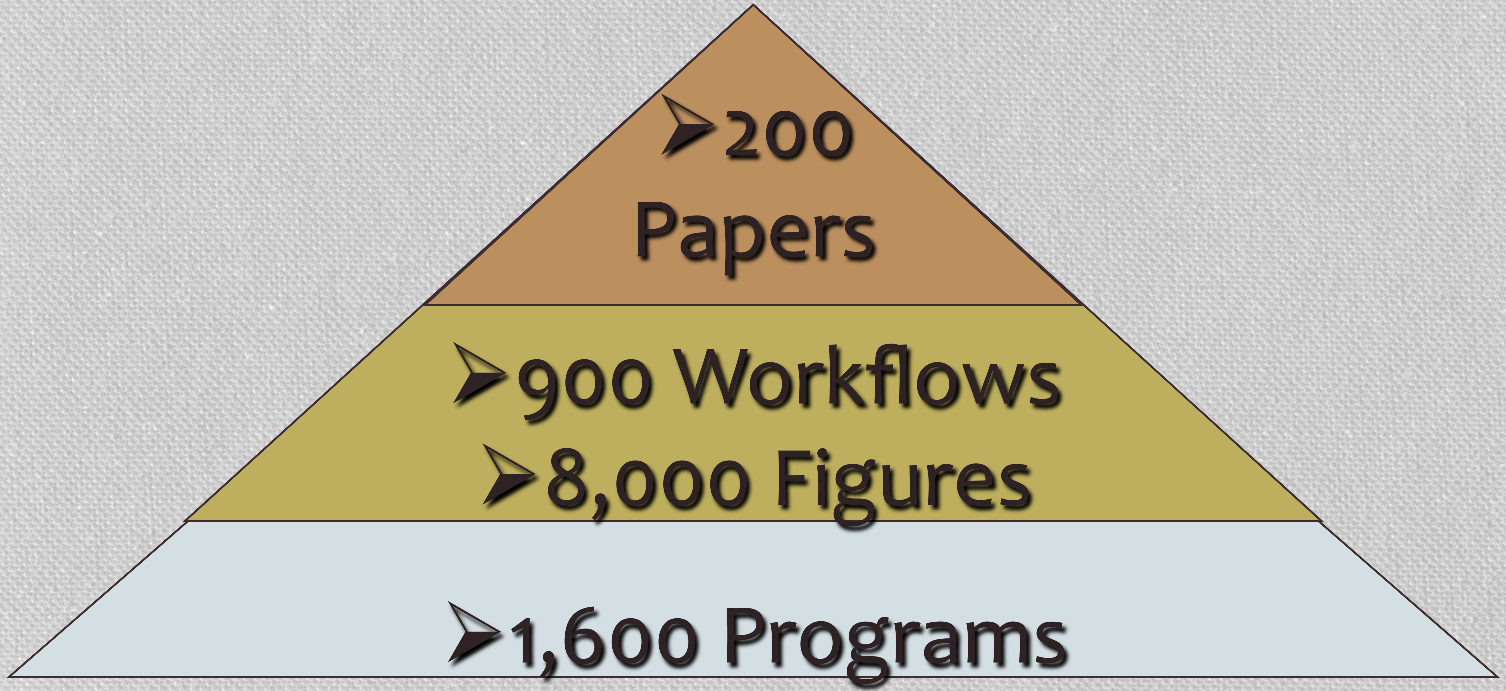
Apply & Test

Implement

Research Pyramid



Madagascar Research Pyramid



➢200 Papers

➢900 Workflows

➢8,000 Figures

➢1,600 Programs

Lessons

- **Reproducibility in itself is not the goal.**

S. Fomel, 2015, Reproducible research as a community effort: Lessons from the Madagascar project: Computing in Science and Engineering, v. 17, 20-26.

Lessons

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

S. Fomel, 2015, Reproducible research as a community effort: Lessons from the Madagascar project: Computing in Science and Engineering, v. 17, 20-26.

Lessons

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

S. Fomel, 2015, Reproducible research as a community effort: Lessons from the Madagascar project: Computing in Science and Engineering, v. 17, 20-26.

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.

S. Fomel, 2015, Reproducible research as a community effort: Lessons from the Madagascar project: Computing in Science and Engineering, v. 17, 20-26.

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

MADAGASCAR

http://ahay.org