Introduction to MADAGASCAR Software Package

Sergey Fomel

Bureau of Economic Geology
The University of Texas at Austin

July 23, 2010

Agenda for Friday, July 23

9:00	– 10:30	Sergey Fomel	Introduction
10:45	– 12:15	Paul Sava	Seismic finite-difference modeling and migration
12:15	– 1:15	Lunch	
1:15	– 2:45	Jim Jennings	Workflows in SCons and automatic testing
3:00	– 4:30	Discussion	
5:30	– 8:00	Dinner and MADAGASCAR 1.0 celebration	

Agenda for Saturday, July 24

9:00 – 9:30	Tariq Alkhalifah	Learning MADAGASCAR
9:00 – 9:30	Jeff Godwin	Programming with MADAGASCAR
10:45 – 11:15	Joe Dellinger	Vplot graphics
11:15 – 11:15	Vladimir Bashkardin	Plotting and HPC
12:15 – 1:15	Lunch	
1:15 – 2:45	Yang Liu	Seismic field data processing
3:00 – 4:30	Discussion	

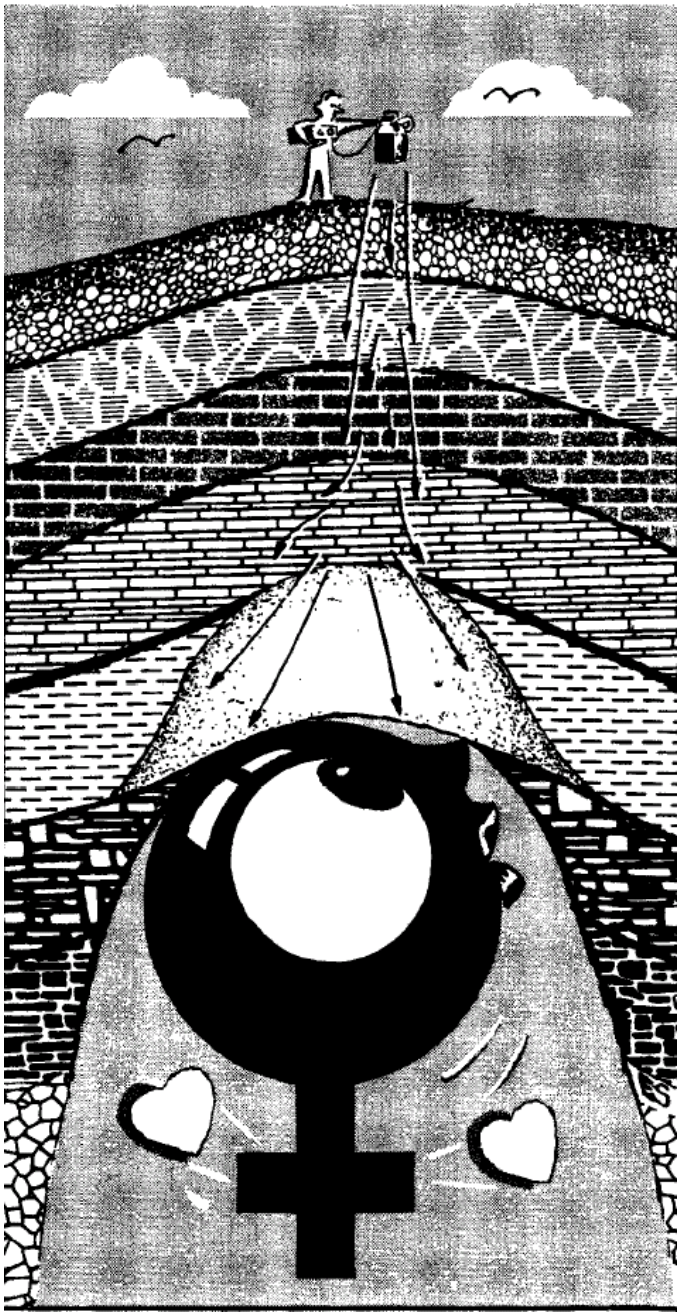
Outline

History of Madagascar

Outline

Outline

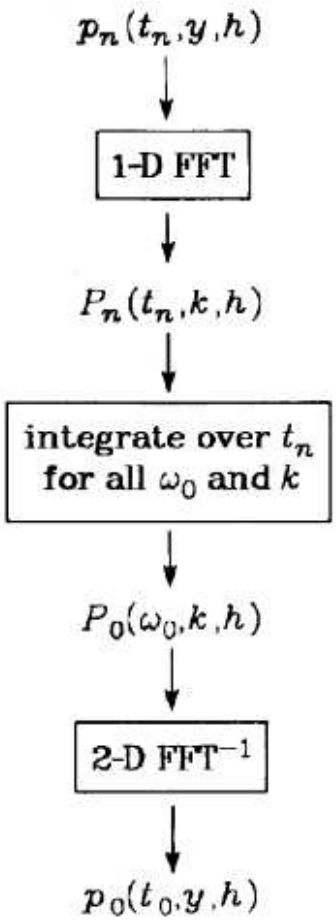
Black Magic in Geophysical Prospecting
L. W. Blau, 1936



Black Magic in Computational Science

Within the world of science, computation is now rightly seen as a third vertex of a triangle complementing experiment and theory. However, as it is now often practiced, one can make a good case that computing is the last refuge of the scientific scoundrel [...] Where else in science can one get away with publishing observations that are claimed to prove a theory or illustrate the success of a technique without having to give a careful description of the methods used, in sufficient detail that others can attempt to repeat the experiment? Randall LeVeque, ICM, 2006

(Hale, 1984)



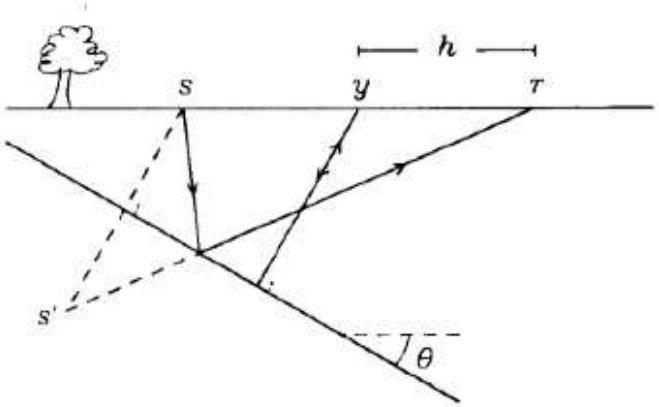


FIG. 1. The seismic experiment, conducted over a simplified subsurface with one dipping reflector. Applying the law of cosines to triangle $s'sr$, one may express the travel time t from source s to receiver r in terms of zero-offset time t_0, half-offset h, velocity v, and dip θ. The result is equation (3) in the text, the

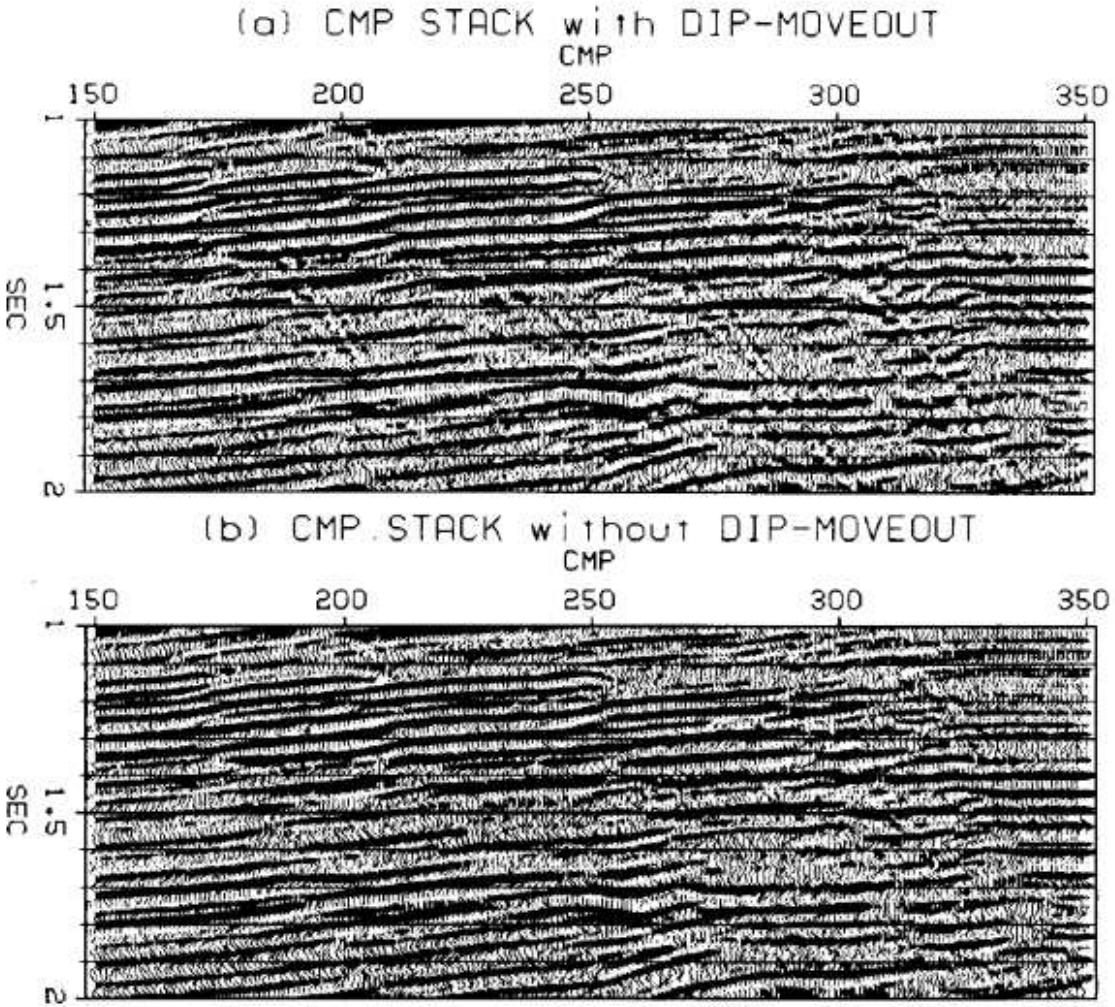
Defining

$$A \equiv \frac{dt_n}{dt_0} = \frac{t_0}{t_n} = \left[1 + \left(\frac{\Delta t_0}{\Delta y}\right)^2 \frac{h^2}{t_n^2}\right]^{1/2},$$

and using equation (10) to replace $p_0(\sqrt{t_n^2 + (\Delta t_0/\Delta y)^2 h^2}, y, h) = p_n(t_n, y, h)$, the Fourier transform becomes

$$P_0(\omega_0, k, h) = \int dt_n\, A^{-1}\, e^{i\omega_0 t_n A} \int dy\, e^{-iky} p_n(t_n, y, h). \qquad (12a)$$

(Hale, 1984)


(a) CMP STACK with DIP-MOVEOUT
CMP
150
200
250
300
350
1
1.5
2
SEC
(b) CMP STACK without DIP-MOVEOUT
CMP
150
200
250
300
350
1
1.5
2
SEC

What is Science?

What is Science?

Science *is the systematic enterprise of gathering knowledge about the universe and organizing and condensing that knowledge into testable laws and theories. The success and credibility of science are anchored in the willingness of scientists to* ***independent testing and replication*** *by other scientists. This requires the* ***complete and open exchange of data, procedures and materials****.*

American Physical Society, What is Science?

From Science to Open-Source Software

Abandoning the habit of secrecy ***in favor of process transparency and peer review was the crucial step by which alchemy became chemistry. In the same way, it is beginning to appear that open-source development may signal the long-awaited maturation of software development as a discipline.***

Eric Raymond, TAUP, 2004

What is Reproducible Research?

- Attaching software code and data to publications
- Communicating computational results to a skeptic

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. Jon Buckheit and David Donoho, WaveLab

Reproducible Research Discussions

- http://www.reproducibleresearch.net

ICASSP 2007

Berlin-6 2008

CiSE 2009

- Donoho et al.
- LeVeque
- Ping & Eckel
- Stodden

IEEE Signal Processing Magazine 2009

- Vandewalle et al.

Yale Roundtable 2009

NSF Archive Workshop 2010

Jon Claerbout's Story

1987 Sunview experience
- Interactive programs are slavery

1992 LaTeX + `cake`
- Building books by a single command

1990s Ph.D. students
- `cake` to `make`, CD-Rom to WWW

2001 Reproducible research paper in *CiSE*
- **The principal beneficiary is the author**

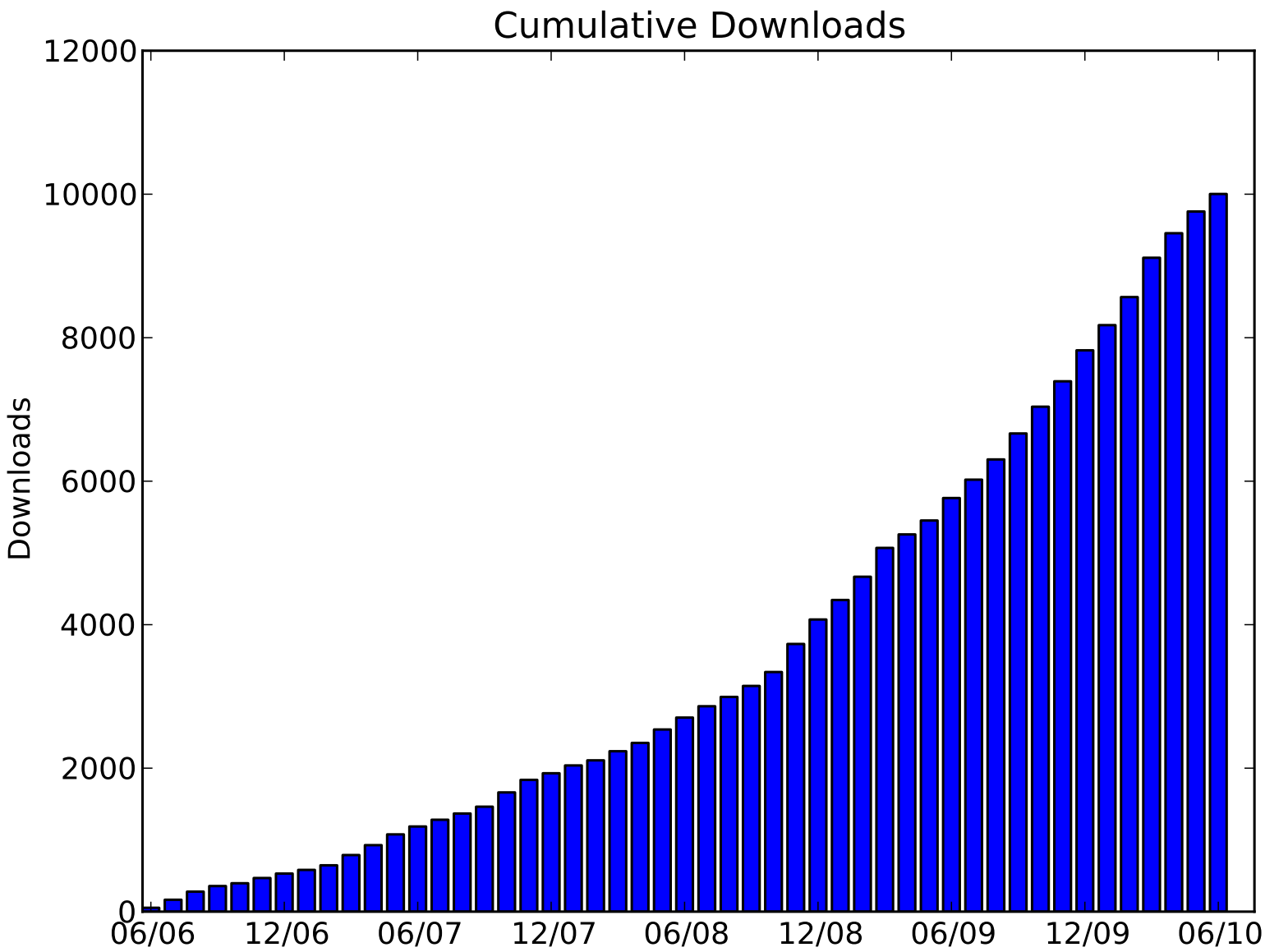
Reproducibility Laws

- The principal beneficiary is the author
- Software code requires continuous maintenance
- Maintenance requires an open community

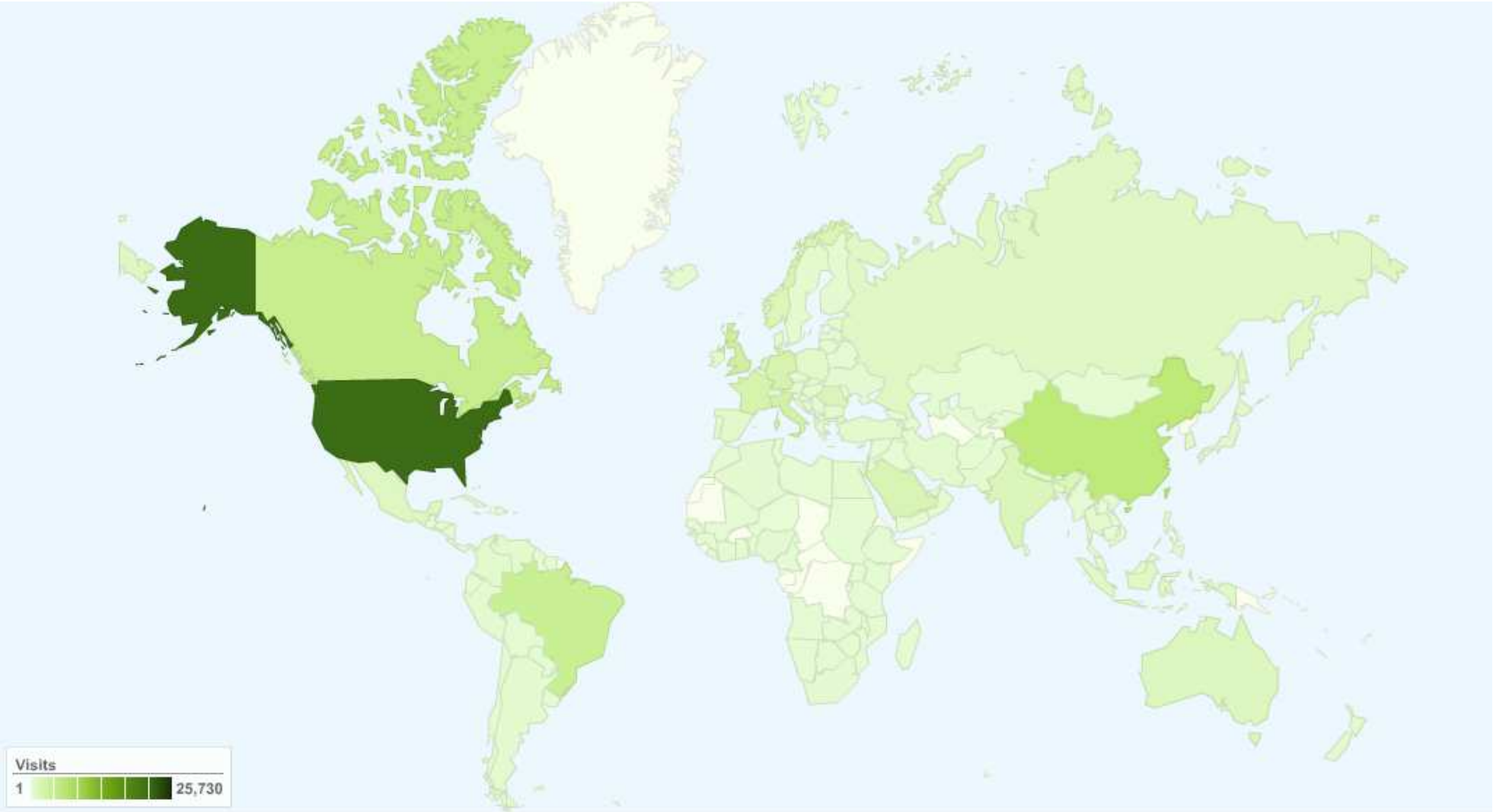
http://reproducibility.org/
http://ahay.org/
http://m8r.info/

Basic Facts about MADAGASCAR

- Publicly available since June 12, 2006
- GPL licensed
- 1.0 version released on July 22, 2010
- 25+ developers
- 250,000+ lines of code
- 10,000+ downloads from SourceForge
- http://www.ahay.org/wiki/Reproducible_Documents

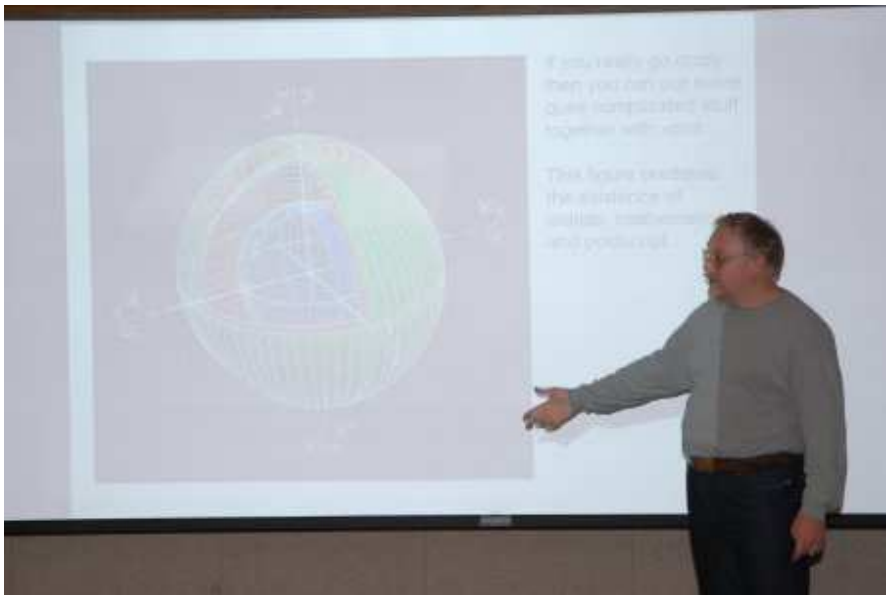

Cumulative Downloads
Downloads
12000
10000
8000
6000
4000
2000
0
06/06
12/06
06/07
12/07
06/08
12/08
06/09
12/09
06/10

Access Geography



School and Workshop: Vancouver 2006

School: Austin 2007



Developer Workshop: Golden 2008

School and Workshop: Delft 2009

- More than 50 people
- 30 organizations (20 universities and 10 companies)
- 15 countries

Outline

Obtaining MADAGASCAR

- **Download stable release from SourceForge**
- **Use `svn` (Subversion) to download unstable (current development) release**
  - `svn checkout`
  - `svn update`
  - `svn commit` **(developers)**

Installing MADAGASCAR from Source

- `./configure --prefix=/directory/name`
- `make`
- `make install`

Full details at http://www.ahay.org/wiki/Installation
http://www.ahay.org/wiki/Advanced_Installation

One Week Technology Transfer

Monday:	**Get an idea**
Tuesday:	**Implement it**
Wednesday:	**Test it**
Thursday:	**Communicate it**
Friday:	**Apply it in practice**

MADAGASCAR Components

Tuesday: Implement it

- Main programs (C, C++, Fortran, Python, etc)
- 750 modules

Wednesday: Test it

- Data processing flows (Python/SCons)
- 350 scripts $\longrightarrow$ 3,200 figures

Thursday: Communicate it

- Books and papers (LaTeX/SCons)
- 100 papers

MADAGASCAR Objectives

- To make computational research efficient
- To make it easy to share computational results
- To promote an open community

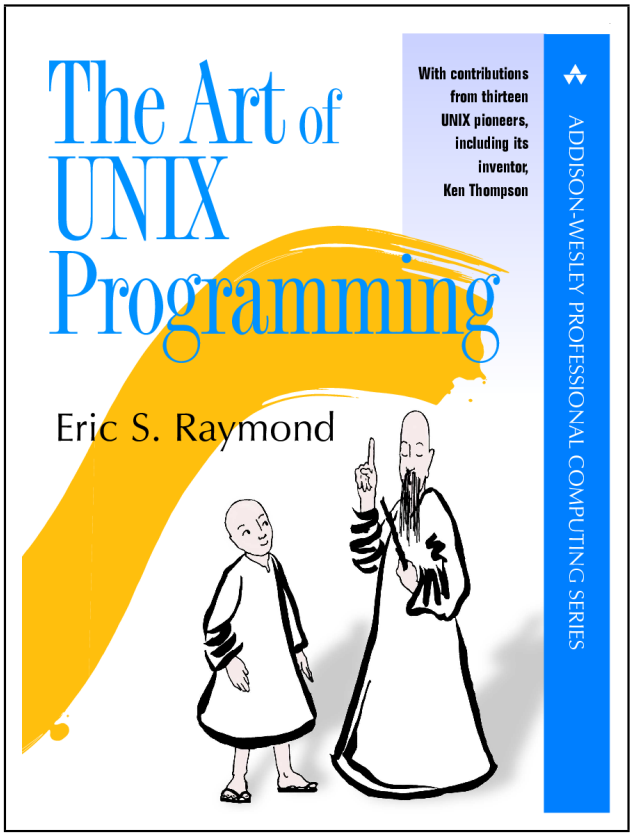
MADAGASCAR Design Principles

- Document computational experiments and use them in the future as regression tests
- Reproducible research
- YAGNI (You Ain't Gonna Need It)
- Encapsulation and modularity

Always implement things when you actually need them, never when you just foresee that you need them. Ron Jeffries, YAGNI

Write programs that do one thing and do it well. Write programs to work together. Write programs to handle text streams, because that is a universal interface. Doug McIlroy, Unix

RSF File Format



- Multidimensional arrays as files
- Simple universal data file format
  - mostly compatible with SEPlib
- Data separated from text headers
- Conceptually N-D hypercubes
- Multiple files for irregular data

If you feel an urge to design a complex binary file format, or a complex binary application protocol, it is generally wise to lie down until the feeling passes.
Eric Raymond, TAUP

MADAGASCAR filter in C

```
#include <rsf.h>

int main(int argc, char* argv[])
{
    int n1, n2, i1, i2;
    float clip, *trace;
    sf_file in, out;

    sf_init(argc,argv);
    in  = sf_input("in");
    out = sf_output("out");

    sf_histint(in,"n1",&n1); /* trace length */
    n2 = sf_leftsize(in,1);  /* number of traces */
    if (!sf_getfloat("clip",&clip)) sf_error("Need clip=");

    trace = sf_floatalloc (n1);
    for (i2=0; i2 < n2; i2++) {
        sf_floatread(trace,n1,in);
        for (i1=0; i1 < n1; i1++) {
            if      (trace[i1] >  clip) trace[i1]= clip;
            else if (trace[i1] < -clip) trace[i1]=-clip;
        }
        sf_floatwrite(trace,n1,out);
    }

    exit(0);
}
```

MADAGASCAR filter in Python

```
#!/usr/bin/env python

import numpy
import m8r

par = m8r.Par()
input  = m8r.Input()
output = m8r.Output()

n1 = input.int("n1") # trace length
n2 = input.size(1)   # number of traces

clip = par.float("clip")

trace = numpy.zeros(n1,'f')
for i2 in xrange(n2): # loop over traces
    input.read(trace)
    trace = numpy.clip(trace,-clip,clip)
    output.write(trace)
```

MADAGASCAR SConstruct script

```
from rsf.proj import Flow

Flow('spike',None,'spike n1=1000 n2=100')
Flow('cliped','spike','clip clip=0.5')
```

```
bash$ scons
scons: Building targets ...
sfspike n1=1000 n2=100 > spike.rsf
< spike.rsf sfclip clip=0.5 > cliped.rsf
scons: Done building targets.
bash$ sed s/0.5/0.25/ < SConstruct > SConstruct2
bash$ mv SConstruct2 SConstruct
bash$ scons
scons: Building targets ...
< spike.rsf sfclip clip=0.25 > cliped.rsf
scons: Done building targets.
```

- http://www.scons.org/

Goal for MADAGASCAR 1.0

- **Automatic Testing**

Goals for MADAGASCAR 2.0

- High-performance computing
- Seismic field data processing examples
- Applications beyond seismic

Conclusions

- Reproducible research
  - Attaching software and data to publications
  - Computational experiments communicated to a skeptic
  - Continuous maintenance requires an open community
- MADAGASCAR Objectives
  - To make computational research efficient
  - To make it easy to share computational results
  - To promote an open community