# Reproducible Computational Experiments Using MADAGASCAR Software Package

**Sergey Fomel**

**Bureau of Economic Geology**
**Jackson School of Geosciences**
**The University of Texas at Austin**

**May 15, 2008**

# Outline

The Magic of Science

# Outline

# Outline

# Outline

# Outline

# The Magic of Science

- **Verification and Validation**
- **Reproducibility**
- **Peer Review**

# The Magic of Computer Science

- Google
- 

***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 S. Raymond, TAUP, 2004***

# The Magic of 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 J. LeVeque, ICM, 2006***

# (Hale, 1984)

$p_n(t_n, y, h)$

↓

$\boxed{\text{1-D FFT}}$

↓

$P_n(t_n, k, h)$

↓

$\boxed{\text{integrate over } t_n \text{ for all } \omega_0 \text{ and } k}$

↓

$P_0(\omega_0, k, h)$

↓

$\boxed{\text{2-D FFT}^{-1}}$

↓

$p_0(t_0, y, h)$

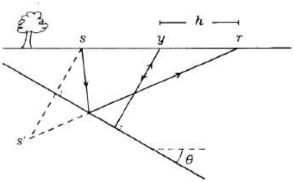



FIG. 1. The seismic experiment, conducted over a simplified subsurface with one dipping reflector. Applying the law of cosines to triangle $s'sr$, one may express the travel time $t$ from source $s$ to receiver $r$ in terms of zero-offset time $t_0$, half-offset $h$, velocity $v$, and dip $\theta$. The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

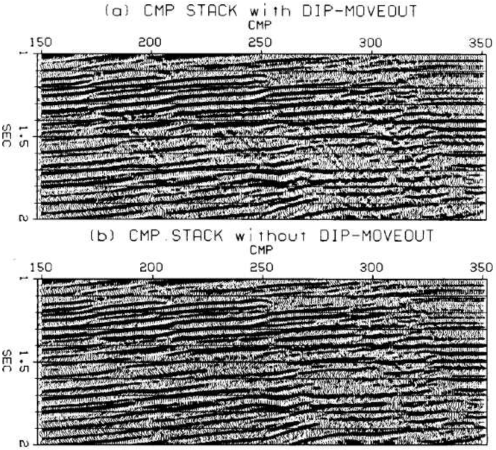
Defining

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

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

# The Magic of Computational Science

***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 B. Buckheit and David L. Donoho, WaveLab, 1995***

# Outline

# Stanford Exploration Project

- **1 proposal, 60 PhDs**
- **Electronic books**
- **(Schwab, Karrenbach, and Claerbout, 2000)**

***The purpose of reproducible research is to facilitate someone going a step further by changing something. The first step that someone will want to make is to be sure that your work is reproducible before they change and improve upon it. Jon F. Claerbout, 1991***

# Follow Up

- **Computational Harmonic Analysis (Donoho et al)**
- **Computational Wave Propagation (LeVeque)**
- **Computational Biostatistics (Gentlemen et al)**
- **Summarized in *CiSE*, Jan/Feb 2009**

# Lessons

- **Reproducibility = Maintenance**
- **Community is important**
- **Tools are important**
  1. **Number crunching**
  2. **Visualization and experiment setup**
  3. **Publications and presentations**
- **Computational experiments are tests**

# MADAGASCAR Package

- **Publicly available since 2006**
- **Version 0.9.6**
- **Open source, open community, open science**
- **Vladimir Bashkardin, Jules Browaeys, Cody Brown, Will Burnett, Maria Cameron, Sergey Fomel, Gilles Hennenfent, Trevor Irons, Jim Jennings, Long Jin, Yang Liu, Doug McCowan, Henryk Modzelewski, Colin Russell, Paul Sava, Jeffrey Shragge, Eduardo Filpo Silva, Ioan Vlad, Jia Yan**
- http://rsf.sourceforge.net

# MADAGASCAR Tools

1. **Number crunching**
   - **Main programs (C, Fortran, C++, etc)**
   - **500 modules**
2. **Visualization and experiment setup**
   - **Data processing flows (Python/SCons)**
   - **600 scripts, 1,700 tests**
3. **Publications and presentations**
   - **Books and papers (LaTeX and Python/SCons)**
   - **100 papers**

# Outline

SourceForge.net Repository – [rsf] View of /trunk/filt/proc/Mcanny.c

http://rsf.svn.sourceforge.net/viewvc/rsf/trunk/filt/proc/Mcanny.c?view=markup

SOURCEFORGE.NET®

# View of /trunk/filt/proc/Mcanny.c

Parent Directory | Revision Log

Revision **2362** - (**download**) (**annotate**)
*Thu Nov 9 01:41:19 2006 UTC* (18 months ago) by *sfomel*
File size: 4228 byte(s)

update

```
/* Canny-like edge detector. */
/*
  Copyright (C) 2004 University of Texas at Austin

  This program is free software; you can redistribute it and/or modify
  it under the terms of the GNU General Public License as published by
  the Free Software Foundation; either version 2 of the License, or
  (at your option) any later version.

  This program is distributed in the hope that it will be useful,
  but WITHOUT ANY WARRANTY; without even the implied warranty of
  MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
  GNU General Public License for more details.

  You should have received a copy of the GNU General Public License
  along with this program; if not, write to the Free Software
  Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA  02111-1307  USA
*/
#include <rsf.h>

int main(int argc, char* argv[])
{
    int n1, n2, n3, i1, i2, i3, j1, j2, k1, k2, nedge, nold, nmin, nmax, n12;
    int **edge;
    float **pp, **ww, **w1, **w2, g1, g2, w, min, max;
    sf_file in, out;

    sf_init(argc,argv);

    in = sf_input("in");
    out = sf_output("out");

    if (!sf_histint(in,"n1",&n1)) sf_error("No n1= in input");
    if (!sf_histint(in,"n2",&n2)) sf_error("No n2= in input");
    n3 = sf_leftsize(in,2);

    if (!sf_getfloat("min",&min)) min=5.0;
    /* minimum threshold */
```

Done

SConstruct

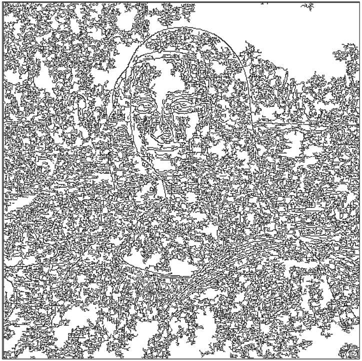
```
# Download data
Fetch('mona.img','imgs')

# Convert to standard format
Flow('mona','mona.img',
     '''
     echo n1=512 n2=513 in=$SOURCE data_format=native_uchar |
     dd type=float
     ''',stdin=0)

Result('mona',
       '''
       grey transp=n allpos=y title="Mona Lisa"
       color=b screenratio=1 wantaxis=n
       ''')

Result('edge','mona',
       '''
       canny | dd type=float |
       grey allpos=y title="Edge Detector" transp=n screenratio=1 wantaxis=n
       ''')
```

```
1:--  SConstruct      3% (21,0)     (Python)
Find file: ~/RSF/book/rsf/rsf/
```

xterm

```
wireless-128-62-143-29:~/RSF/book/rsf/rsf/mona sfomel$ scons view
scons: Reading SConscript files ...
scons: done reading SConscript files.
scons: Building targets ...
retrieve(["mona.img"], [])
echo n1=512 n2=513 in=mona.img data_format=native_uchar | /Users/sfomel/RSFROOT/
bin/sfdd type=float > mona.rsf
< mona.rsf /Users/sfomel/RSFROOT/bin/sfgrey transp=n allpos=y title="Mona Lisa"
color=b screenratio=1 wantaxis=n > Fig/mona.vpl
/Users/sfomel/RSFROOT/bin/xtpen Fig/mona.vpl
< mona.rsf /Users/sfomel/RSFROOT/bin/sfcanny | /Users/sfomel/RSFROOT/bin/sfdd ty
pe=float | /Users/sfomel/RSFROOT/bin/sfgrey allpos=y title="Edge Detector" trans
p=n screenratio=1 wantaxis=n > Fig/edge.vpl
/Users/sfomel/RSFROOT/bin/xtpen Fig/edge.vpl
```

Mona Lisa



Edge Detector

SConstruct

```python
# Edge preserving smoothing

rect=80 # smoothing radius

Flow('mona2','mona','impl2 rect1=%d rect2=%d tau=1' % (rect,rect))

Result('mona2',
       '''
       grey transp=n allpos=y title="Smoothed"
       color=b screenratio=1 wantaxis=n
       ''')

Result('edge2','mona2',
       '''
       canny | dd type=float |
       grey allpos=y title="Edge Detector" transp=n screenratio=1 wantaxis=n
       ''')

End()
```

1:-- SConstruct Bot (28,0) (Python)

xterm

```
wireless-128-62-143-29:~/RSF/book/rsf/rsf/mona sfomel$ scons
scons: Reading SConscript files ...
scons: done reading SConscript files.
scons: Building targets ...
scons: `Fig/mona.vpl' is up to date.
scons: `Fig/edge.vpl' is up to date.
< mona.rsf /Users/sfomel/RSFROOT/bin/sfimpl2 rect1=80 rect2=80 tau=1 > mona2.rsf
sfimpl2: build/filt/lib/impl2.c: nstep=533 tau=(1.00047,1.00047) nclip=131327
< mona2.rsf /Users/sfomel/RSFROOT/bin/sfgrey transp=n allpos=y title="Smoothed"
color=b screenratio=1 wantaxis=n > Fig/mona2.vpl
< mona2.rsf /Users/sfomel/RSFROOT/bin/sfcanny | /Users/sfomel/RSFROOT/bin/sfdd t
ype=float | /Users/sfomel/RSFROOT/bin/sfgrey allpos=y title="Edge Detector" tran
sp=n screenratio=1 wantaxis=n > Fig/edge2.vpl
scons: done building targets.
wireless-128-62-143-29:~/RSF/book/rsf/rsf/mona sfomel$ 
```

Smoothed

Edge Detector

SConstruct<3>

```
\begin{frame}
  \MadLogo
  \begin{center}
    \includegraphics[width=0.45\textwidth]{mona/Fig/mona2}
    \hfill
    \includegraphics[width=0.45\textwidth]{mona/Fig/canny2}
  \end{center}
\end{frame}

1:--  ssd2.tex      88% (259,0)    (PDFLaTeX/F Ref BCite)
Paper('ssd2',
      lclass='beamer',
      use='helvet,hyperref,listings',
      include=r'''
               \mode<presentation>{\usetheme{Szeged}}
               \usepackage[absolute,overlay]{textpos}
               \setlength{\TPHorizModule}{1mm}
               \setlength{\TPVertModule}{1mm}
               \newcommand{\MadLogo}{%
               \begin{textblock}{30}(115,-25)
               \includegraphics[width=30\TPHorizModule]{Fig/Madagascar}
-:--  SConstruct<3>   81% (82,34)   SVN:3226   (Python)
Find file: ~/RSF/book/rsf/rsf/
```

# MADAGASCAR Tools

1. **Number crunching**
   - **Main programs (C, Fortran, C++, etc)**
   - **500 modules**
2. **Visualization and experiment setup**
   - **Data processing flows (Python/SCons)**
   - **600 scripts, 1,700 tests**
3. **Publications and presentations**
   - **Books and papers (LaTeX and Python/SCons)**
   - **100 papers**

# Outline

# Conclusions

- **Reproducible computational experiments**
- **Reproducibility = maintenance**
- **MADAGASCAR software package**

# Help Needed

- **Automatic testing**
- **Efficient parallelization**
- **Visualization standards**
- **Graphical User Interface**
- **...**

http://rsf.sourceforge.net