Additional Topics

SERGEY FOMEL
JULY 11, 2017
Madagascar Source Directory Structure
# Programming Interfaces

![Folder structure](image)

<table>
<thead>
<tr>
<th>Name</th>
<th>Kind</th>
</tr>
</thead>
<tbody>
<tr>
<td>c</td>
<td>Folder</td>
</tr>
<tr>
<td>c++</td>
<td>Folder</td>
</tr>
<tr>
<td>f77</td>
<td>Folder</td>
</tr>
<tr>
<td>f90</td>
<td>Folder</td>
</tr>
<tr>
<td>java</td>
<td>Folder</td>
</tr>
<tr>
<td>julia</td>
<td>Folder</td>
</tr>
<tr>
<td>matlab</td>
<td>Folder</td>
</tr>
<tr>
<td>octave</td>
<td>Folder</td>
</tr>
<tr>
<td>python</td>
<td>Folder</td>
</tr>
<tr>
<td>README.txt</td>
<td>Plain Text</td>
</tr>
</tbody>
</table>
Madagascar
Source
Directory
Structure
Seismic Unix Code in Madagascar
Seismic Unix (SU) is a famous open-source seismic processing package maintained by John Stockwell at the Center for Wave Phenomena, Colorado School of Mines.

SU has been around for 25 years and has attracted many devoted users. If you are one of them, please consider the following:

- Using Seismic Unix is not an excuse for non-reproducible computational experiments. To facilitate reproducibility, you can use Python and SCons with the `rsf.suproj` module supplied by Madagascar. The `book/rsf/su` directory contains many examples of seismic data processing flows using SU and their loose translation to Madagascar analogs. Here is an example `SConstruct` script from
Seismic Unix

You can use SCons with Seismic Unix to do reproducible research
  ◦ examples in \$RSFSRC/book/rsf/su
  ◦ including translations to Madagascar

Convert between RSF and SU formats using sfsuread and sfsuwrite

Madagascar can borrow code from Seismic Unix (not vice versa)
  ◦ examples in \$RSFSRC/su
- Directory Structure
- Seismic Unix
- Tutorials
- Interactivity
- Jupyter Notebooks

MADAGASCAR
Geophysical tutorials

The geophysical tutorial series debuted in the February 2014 issue of the The Leading Edge. The tutorials appear every other month to serve as a "brief exploration of a geophysical topic." According to Matt Hall, the author of the first column, the tutorials "use only open data sets that anyone can download." [1]

List of Tutorials in The Leading Edge  [ edit | edit source ]

- Seismic rock physics by Alessandro Amato del Monte - June 2017
  http://library.seg.org/doi/pdfplus/10.1190/tle36060523.1
- Getting started with controlled-source electromagnetic 1D modeling by Dieter Werthmüller - April 2017
- Step-by-step NMO correction by Leonardo Uieda - February 2017
- Linear inversion by Matt Hall - December 2016
You can find Madagascar version of selected TLE tutorials in $RSFSRC/book/rsf/tutorials

Tutorial on NMO correction

May 8, 2017

The example in rsf/tutorials/nmo reproduces the tutorial from Leonardo Uieda on NMO correction. The tutorial was published in the February 2017 issue of The Leading Edge.
Is there a Graphical User Interface to Madagascar?

July 19, 2010  FAQ

A hardcore Madagascar user does not need anything more than a friendly editor (to edit SConstruct files) and the good old command line (to run scons commands). However, sometimes it is necessary to provide simplified GUIs (Graphical User Interfaces) for inexperienced users. Creating GUIs in Python is quite simple. An example is provided in rsf/rsf/gui. In this example, we obtain a compressed approximation of a piecewise-regular signal with by a wavelet transform. The figure using default parameters is shown below:

![Signal](image)
There are many different libraries for GUI (graphical user interfaces), many of them with Python bindings: PyGTK, PyQt, PySide, etc. Tkinter is one of the oldest Python GUI libraries and is considered to be the standard one. Another popular choice is wxPython, a Python interface for wxWidgets C++ library.
Graphical User Interface

Easy to create using Python GUI packages
- PyGTK, PyQt, Tkinter, Traits, wxPython, etc.

Complete interface (tkMadagascar): sfgui

Can interface with SCons
- $RSFSRC/rsf/rsf/gui
- $RSFSRC/rsf/tutorials/parameters

Can interface with Vplot graphics
- sfzoom
- sfipick
Scientific Python and Jupyter Notebooks

https://scipy2017.scipy.org/
The Jupyter Notebook

The Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations and explanatory text. Uses include: data cleaning and transformation, numerical simulation, statistical modeling, machine learning and much more.
Example Notebook: NMO

To relate measured and true velocities, consider reflection ray tracing in a layered velocity medium. Inside one layer with depth $\Delta z$ and velocity $v$, the time increment is

$$\Delta t = 2 \frac{\Delta z}{v \cos \theta},$$

where $\theta$ is the ray angle.
Jupyter Notebooks

Web browser interface

Easy to integrate text, equations, plots, etc.

Simple to interface with Madagascar
  ◦ %%file filename.scons

Easy to integrate Vplot graphics
  ◦ from m8r import view
  ◦ view(‘result’)

Example of literate programming
Literate Programming

Donald E. Knuth

Emphatic declarations:

* examples: array [sort of small... large], beauty, cool

True confessions:

for random (human) do write (nearly),
while programming := are do
begin incr (position); incr (bag); incr (probability);
incr (unpredictability); incr (quality); incr (salary);
end (happily ever after)

This code is used in theory and practice.
Literate Programming

“The basic idea of literate programming is to take a fundamentally different starting point for the presentation of programs to human readers, without any direct effect on the program as seen by the computer. Rather than to present the program in the form in which it will be compiled (or executed), and to intercalate comments to help humans understand what is going on (and which the compiler will kindly ignore), the presentation focuses on explaining to humans the design and construction of the program, while pieces of actual program code are inserted to make the description precise and to tell the computer what it should do.” (van Leeuwen, 1990)
- Directory Structure
- Seismic Unix
- Tutorials
- Interactivity
- Jupyter Notebooks