



## **Reproducible Paper:**

Writing technical papers using  
Madagascar and L<sup>A</sup>T<sub>E</sub>X

---

**Jiawei Chen\*, Yang Liu**

College of Geo-exploration Sciences and Technology  
Jilin University

chenjw22@mails.jlu.edu.cn, yangliu1979@jlu.edu.cn



**Aug 19, 2025**

# References



## Madagascar L<sup>A</sup>T<sub>E</sub>X and Web tools

Sergey Fomel

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

August 30, 2006

(Fomel, School, 2006)

S. Fomel (UT Austin)

BT<sub>E</sub>X and Web

Vancouver School & Workshop



## Writing a paper using Madagascar

Tariq Alkhalifah

Seismic Analysis Group (SWAG)  
KAUST  
swag.kaust.edu.sa

(Alkhalifah, School, 2011)

Madagascar School for Advanced Users in Qingdao 2015



### Reproducible Paper:

Writing and publishing using L<sup>A</sup>T<sub>E</sub>X and Madagascar

Yang Liu

College of Geo-exploration Sciences and Technology  
Jilin University  
yangliu1979@jlu.edu.cn

Aug 8, 2015

(Liu, School, 2015)

1



Madagascar School  
of Reproducible Computational  
Geophysics Shanghai 2017



同濟大學  
TONGJI UNIVERSITY

## Writing a Reproducible Paper using LaTeX and Madagascar

Jiubing Cheng & Yang Liu



(Cheng and Liu, School, 2017)

# References



首頁 討論

## Main Page

**Madagascar** is an open-source software package for multidimensional data analysis and [reproducible](#) computational experiments. Its mission is to provide

- a convenient and powerful environment
- a convenient technology transfer tool

for researchers working with digital image and data processing "recipes" to be verified, exchanged, and modified by the users.

### Features [編輯]

🕒 Madagascar is a [modern](#) package. It started in 2003 and driven development. The rapid growth of a project of packages such as SEPlib and Seismic Unix. We have borrowed

🕒 Madagascar is a [test-driven](#) package. Test-driven development, the backbone of any real science, to compute working as data filters) and high-level processing flows (describing programming is easily mastered even by beginning students v

🕒 Madagascar is an [open-source](#) package. It is distributed but shared equally among developers. Sharing the responsibility

🕒 Madagascar uses a simple, flexible, and universal data format allows us to share general-purpose data pr

### Latest News [編輯]

**Effects of lateral heterogeneity on time-domain process**  
An old paper is added to the collection of reproducible do  
2024-12-05 20:54:15

**Well log interpolation guided by geologic distance**  
An old paper is added to the collection of reproducible do  
2024-12-05 20:52:35

**Estimation of timeshifts in time-lapse seismic images usi**  
An old paper is added to the collection of reproducible do  
2024-12-05 20:51:13

**Least-squares non-stationary triangle smoothing**  
An old paper is added to the collection of reproducible do  
2024-12-05 20:49:26

**Investigating the possibility of locating microseismic sou**  
An old paper is added to the collection of reproducible do  
2024-12-05 20:45:56

**Using well-seismic mistie to update the velocity model**  
An old paper is added to the collection of reproducible do  
2024-12-05 20:44:26

**Full waveform inversion of passive seismic data for sour**  
An old paper is added to the collection of reproducible do  
2024-12-05 20:42:44

阅读 编辑 查看历史 搜索Madagascar内容

## Conferences

在逆時序順序。

目錄 [跳到]

- 1 Schools, Workshops, and Working Workshops
- 2 Conference presentations
  - 2.1 Online 2021 (SPE)
  - 2.2 Lausanne 2016 (PASC)
  - 2.3 San Francisco 2015 (AGU)
  - 2.4 Stanford 2015 (SIAM Geosciences)
  - 2.5 New York 2013 (NYU Poly)
  - 2.6 Providence 2012 (ICERM)
  - 2.7 Austin 2012 (SciPy)
  - 2.8 Cary 2011 (Interface)
  - 2.9 Long Beach 2011 (SIAM Geosciences)
  - 2.10 Reno 2011 (SIAM CS&E)
  - 2.11 Austin 2010 (SciPy)
  - 2.12 Salt Lake City 2010 (NSF Archive Workshop)
  - 2.13 Düsseldorf 2008 (Berlin 6)
  - 2.14 Austin 2008 (Xes Python Unconference)
  - 2.15 Austin 2008 (Scientific Software Days)
  - 2.16 Rio de Janeiro 2007 (SBGF)
  - 2.17 Vancouver 2007 (AIP)
  - 2.18 Honolulu 2007 (ICASSP)
  - 2.19 Austin 2007 (Scientific Software Day)
  - 2.20 New Orleans 2006 (SEG)
  - 2.21 Vienna 2006 (EAGE)

阅读 编辑 查看历史 搜索Madagascar内容



Schools, Workshops, and Working Workshops [編輯]  
Madagascar Workshop at RAGUIM, Puerto Vallarta 2023  
Madagascar School on Open-Source Geophysics in Hefei 2020  
SEG Working Workshop - Python and Julia for Geophysics, Houston August 8-11, 2018  
SEG Working Workshop in Houston 2017  
Madagascar School of Reproducible Computational Geophysics Shanghai 2017  
Madagascar School at the University of Houston 2017  
SEG Land 3D Seismic Processing Working Workshop Houston 2016  
Madagascar School on Reproducible Computational Geophysics in Zürich 2016  
Workshop - Open-source Software in Applied Geosciences Vienna 2016  
Madagascar School for Advanced Users in Qingdao 2015  
SEG 3D Land Seismic Processing Working Workshop Houston 2015  
School in Harbin 2015  
Second Madagascar Working Workshop Houston 2014  
Workshop at EAGE/St. Petersburg 2014

<https://www.ahay.org>

<https://www.ahay.org/wiki/Conferences>

# Outline

---



- Motivations
- Pre-writing configuration
  - Prerequisites
  - File organization
  - Figure preparation
- SEGTeX
  - Implement steps
  - Rules and tips
- Alternative L<sup>A</sup>T<sub>E</sub>X template
- Simple Example

# Motivations

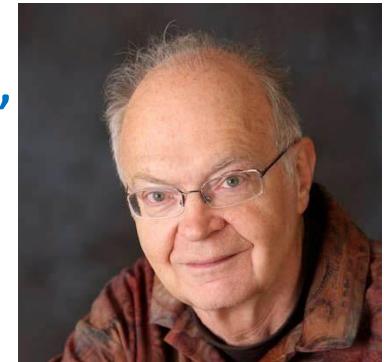


- MS-word--> LaTeX

**"I hope to die before I have to use Microsoft Word"**  
— Donald Ervin Knuth

Reasons:

- 1) Focus on paper structure and content;
- 2) Better mathematic style;
- 3) More efficient (easy comments);
- 4) Better revision for long document;
- 5) Sole high-quality text software in Linux;
- 6) Compiling by command line and supported by Madagascar.



- **"All technical papers should be "reproducible" in the sense that someone of reasonable skill ought to be able to read the paper and then reproduce the results."** — Joe Dellinger
- **Remember, LaTeX is not perfect, either MS-word or LaTeX is just a tool, choose any of them you need.**

# SEGTeX : When LaTeX meets Madagascar



download  
Installation  
GitHub repository  
SEGTeX

Introduction  
Package overview  
Tutorial  
Hands-on tour  
Reproducible documents

User Documentation  
List of programs  
Common programs  
The RSF file format  
Reproducibility with SCons

Developer documentation  
Adding programs  
Contributing programs  
API demo: clipping data  
API demo: explicit finite differences

Community  
User mailing list  
Developer mailing list  
GitHub organization  
SourceForge project  
Conferences  
LinkedIn group  
Development blog  
[www.ahay.org/wiki/Main\\_Page](http://www.ahay.org/wiki/Main_Page)

Page Discussion

Read View source View history

Create account Log in

## SEGTeX

SEGTeX is a LaTeX package for geophysical publications. It consists of

- LaTeX2e class files for [Geophysics](#) papers, SEG expanded abstracts, etc
- BibTeX style files [seg.bst](#)
- BibTeX cumulative bibliography of geophysical publications [SEG.bib](#)
- [Latex2HTML](#) cum



Contents [hid]

- 1 Downloading
- 2 Installation
  - 2.1 Prerequisite
- 3 SEG expanded at
- 4 Documentation
- 5 Tips and tricks
- 6 Communication
- 7 Reproducible pap
- 8 LaTeX references
- 9 News

### Downloadin

Access SEGTeX th

Alternatively, acces

git clone ht

or using Subversio

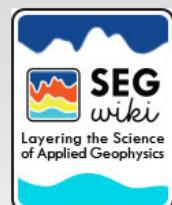
svn co https



Society of Exploration Geophysicists  
The international society of applied geophysics

SEG Membership Events Resources Education Groups & Communities

Print



Layering the Science of Applied Geophysics

Advertise with SEG

Call for papers  
Third International Conference on Geophysical Sciences

15-18 November | Al Ain, UAE



## LaTeX package for paper submission to GEOPHYSICS

A typesetting package is available to help authors prepare papers for GEOPHYSICS. The package consists of a set of macros prepared for LaTeX, a popular document preparation system. The SEG has developed several macros designed specifically for GEOPHYSICS and the SEG Annual Meeting *Expanded Abstracts*.

Authors who use the LaTeX typesetting program to prepare their manuscripts can use the SEG macros (called SEGTeX) to format the text, equations, references, and appendices so that they conform to GEOPHYSICS guidelines for submission. If using BibTeX to create references, authors must run BibTeX before submitting the .tex file and read in or paste the resulting contents of the generated .bbl file within the bibliography section of the .tex file. All LaTeX submissions must include only one .tex file and a PDF of that file. You may send questions concerning LaTeX files to the SEGTeX mailing list.

To submit papers to GEOPHYSICS, follow the procedures described in the SEG [Instructions to Authors](#) (in the January–February issue of GEOPHYSICS and on this Web site at the link above). The manuscript will undergo the standard review process. Once all revisions and changes to the manuscript are made and the manuscript is accepted, LaTeX files will be converted to Microsoft Word documents for production.

<https://library.seg.org/page/gpysa7/ifa/latex-package>

# SEGTeX : When LaTeX meets Madagascar



An example SEG expanded abstract  
Joe Dellinger\*, BP and Sergey Fomel, University of Texas at Austin

**SUMMARY**  
This is an example of using `seg.tex` for writing SEG expanded abstracts.

**INTRODUCTION**  
This is an introduction.  $\text{\LaTeX}$  is a powerful document typesetting system (Lamport, 1994). An excellent reference is (Kopka and Daly, 2001). The new `seg.cls` class complies with the SEG20 standard.

**THEORY**  
This is another section.

**Equations**  
Section headings should be capitalized. Subsection headings should only have the first letter of the first word capitalized.

Here is an example of equations involving vectors and tensors:

$$\mathbf{R} = \begin{pmatrix} R_{XX} & R_{XY} \\ R_{YX} & R_{YY} \end{pmatrix} = \mathbf{P}_{M \times M} \mathbf{D} \mathbf{P}_{M \times M} \quad \mathbf{S}, \quad (1)$$

and

$$R_{jlm}(\omega) = \sum_{k=1}^N P_j^{(m)}(\mathbf{x}_k) D^{(n)}(\omega) P_m^{(n)}(\mathbf{x}_k) \quad (2)$$

Note that the macro for the `Venner` command has been changed to `venner` to have tensors to be held uppercase, in compliance with current SEG submission standards. This is so that documents typed with the `seg.tex` standard will print out according to the new ones: e.g., tensor  $\mathbf{T}$  (note converted to uppercase).

**Figures**

Figure 1 shows what it is about.

**Multiples**

Sometimes it is convenient to put two or more figures from different sections in one array (see Figure 2). Individual plots are Figures 2a and 2b.

The first argument of the `multiple` command specifies the number of plots per row.

**Tables**

The discussion is summarized in Table 1.

Figure 1: This figure is specified in `plot(waveform)(width=6cm)`.

Figure 2: This figure is specified in `[multiple(2)][width=6cm][width=6cm]`.

## Template for SEG abstract

Template for writing SEG abstract.

Sergey Fomel

An example *Geophysics* article,

with a two-line title

(September 18, 2021)

GEO-Example

Running head: *Geophysics* example

## ABSTRACT

This is an example of using `eage.cls` for writing *Geophysics* papers.

## INTRODUCTION

This is an introduction.  $\text{\LaTeX}$  is a powerful document typesetting system (an excellent reference is (Kopka and Daly, 2001). The new `eage.cls` complies with the SEG20 standard).

## THEORY

This is another section.

## Equations

Section headings should be capitalized. Subsection headings should only have the first letter of the first word capitalized.

Here is an example of equations involving vectors and tensors:

$$\mathbf{R} = \begin{pmatrix} R_{XX} & R_{XY} \\ R_{YX} & R_{YY} \end{pmatrix} = \mathbf{P}_{M \times M} \mathbf{D} \mathbf{P}_{M \times M} \quad \mathbf{S}, \quad (1)$$

and

$$R_{jlm}(\omega) = \sum_{k=1}^N P_j^{(m)}(\mathbf{x}_k) D^{(n)}(\omega) P_m^{(n)}(\mathbf{x}_k) \quad (2)$$

Note that the macro for the `Venner` command has been changed to `venner`, in compliance with current SEG submission standards. This is so that documents typed with the `seg.tex` standard will print out according to the new ones: e.g., tensor  $\mathbf{T}$  (note converted to uppercase).

## Figures

Figure 1 shows what it is about.

## Template for GEOPHYSICS

GEOPHYSICS, published by the Society of Geophysicists since 1936, is an archival journal.

Sergey Fomel

## An example EAGE expanded abstract

SEGTeX is a LaTeX package for geophysical publications. This template consists of:

## An example 2-column Geophysics article

This is an example of using `geophysics` for writing two-column Geophysics papers.



# TEX

Example

Geophysics example

Joe Dellinger\* and Sergey Fomel†

**ABSTRACT**  
This is an example of using `geophysics.cls` for writing *Geophysics* reports, with a two-line title

**INTRODUCTION**  
This is an introduction.  $\text{\LaTeX}$  is a powerful document typesetting system (an excellent reference is (Kopka and Daly, 2001). The new `geophysics.cls` class complies with the `seg.tex` standard).

**THEORY**  
This is another section.

**Equations**  
Section headings should be capitalized. Subsection headings should only have the first letter of the first word capitalized.

Here is an example of equations involving vectors and tensors:

$$\mathbf{R} = \begin{pmatrix} R_{XX} & R_{XY} \\ R_{YX} & R_{YY} \end{pmatrix} = \mathbf{P}_{M \times M} \mathbf{D} \mathbf{P}_{M \times M} \quad \mathbf{S}, \quad (1)$$

and

$$R_{jlm}(\omega) = \sum_{k=1}^N P_j^{(m)}(\mathbf{x}_k) D^{(n)}(\omega) P_m^{(n)}(\mathbf{x}_k) \quad (2)$$

Note that the macro for the `Venner` command has been changed to `venner`, in compliance with current SEG submission standards. This is so that documents typed with the `seg.tex` standard will print out according to the new ones: e.g., tensor  $\mathbf{T}$  (note converted to uppercase).

**Figures**

Figure 1: This figure is specified in `plot(waveform)(width=6cm)`.

**Multiples**

Subsections in a subsection will print in an array (see Figures 2a and 2b).

The discussion is common to all the sub-sections, and the number of plots per i

**Tables**

The discussion is common

**ACKNOWLEDGMENT**

I would like to thank from Piero Paoletti for his help with this paper.

**REFERENCES**

See references in the `seg.tex` file.

**FIGURES**

Figure 1 shows what it is about.

**INTRODUCTION**  
This is an introduction.  $\text{\LaTeX}$  is a powerful document typesetting system (Lamport, 1994). An excellent reference is (Kopka and Daly, 2001). The new `geopreport.cls` class complies with the `seg.tex` standard.

**THEORY**  
This is another section.

**Equations**  
Section headings should be capitalized. Subsection headings should only have the first letter of the first word capitalized.

Here is an example of equations involving vectors and tensors:

$$\mathbf{R} = \begin{pmatrix} R_{XX} & R_{XY} \\ R_{YX} & R_{YY} \end{pmatrix} = \mathbf{P}_{M \times M} \mathbf{D} \mathbf{P}_{M \times M} \quad \mathbf{S}, \quad (1)$$

and

$$R_{jlm}(\omega) = \sum_{k=1}^N P_j^{(m)}(\mathbf{x}_k) D^{(n)}(\omega) P_m^{(n)}(\mathbf{x}_k) \quad (2)$$

Note that the macro for the `Venner` command has been changed to `venner` to have tensors to be held uppercase, in compliance with current SEG submission standards. This is so that documents typed with the `seg.tex` standard will print out according to the new ones: e.g., tensor  $\mathbf{T}$  (note converted to uppercase).

**Figures**

Figure 1: This figure is specified in `plot(waveform)(width=6cm)`.

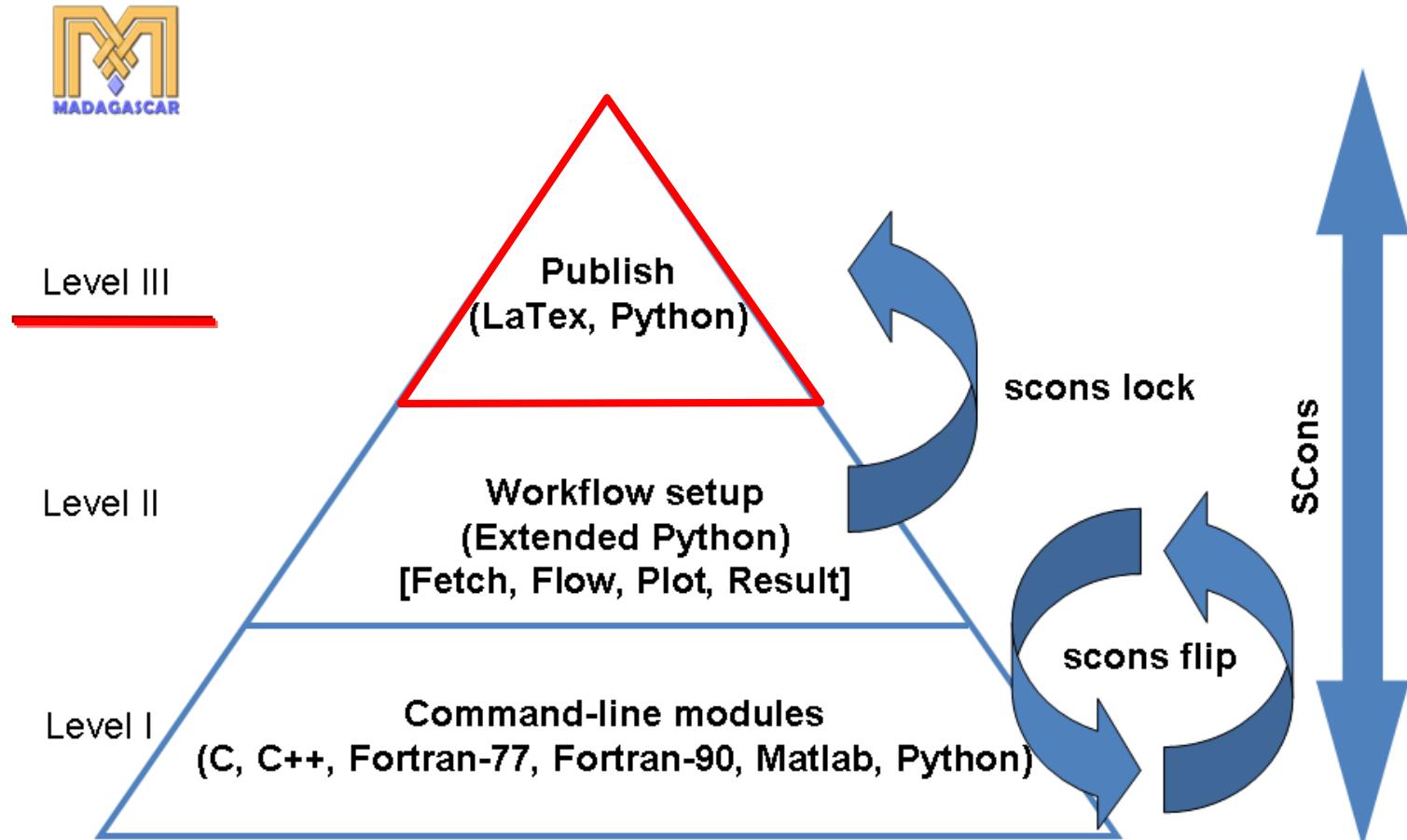
**INTRODUCTION**  
This is an example of using `geopreport.cls` for writing Geophysics reports.

## An example Geophysics report

This is an example of using `geopreport.cls` for writing Geophysics reports.

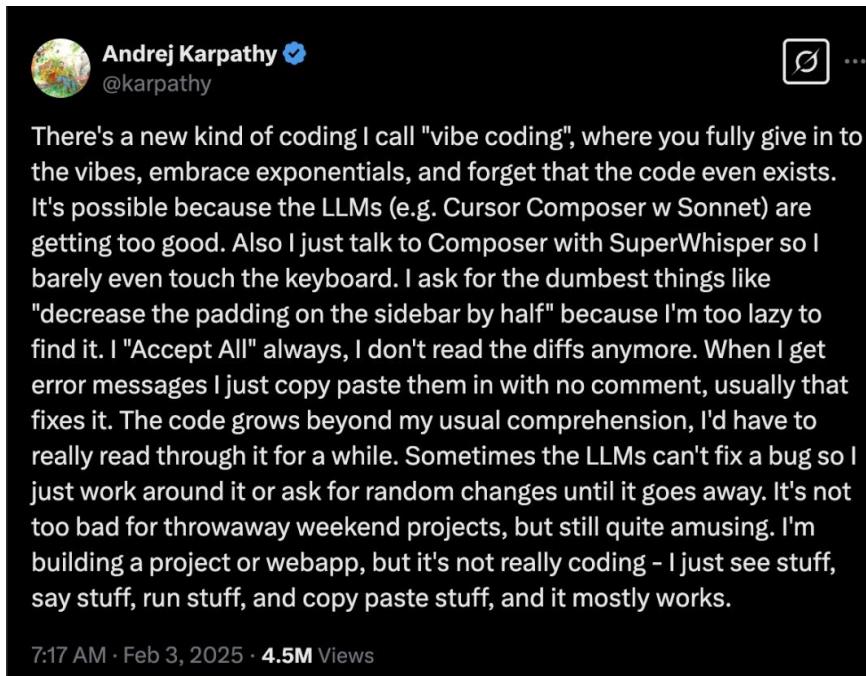
SEGTeX

# Madagascar software architecture



Fomel, S, Sava, P, Vlad, I, Liu, Y, and Bashkardin, V 2013. Madagascar: open-source software project for multidimensional data analysis and reproducible computational experiments. *Journal of Open Research Software* 1(1): e8, DOI: <http://dx.doi.org/10.5334/jors.ag>

# Vibe/LLM Coding makes it easier than ever



A screenshot of a social media post by Andrej Karpathy (@karpathy). The post features a profile picture of Andrej Karpathy, his name, handle, and a small icon with three dots. The main text discusses a new kind of coding called "vibe coding" where one fully gives in to the vibes, embraces exponentials, and forgets about the code's existence. It mentions how LLMs like Cursor Composer and SuperWhisper have become very good, and how the author interacts with them by asking for dumb things like decreasing padding or accepting all changes. The code grows beyond comprehension, requiring multiple reads to understand. The post ends with the author building projects or webapps by seeing stuff, running it, and copying/pasting, which mostly works. The timestamp at the bottom is 7:17 AM · Feb 3, 2025 · 4.5M Views.



- Fully or largely give in to the vibes.
- Embrace exponentials.
- Forget that the code even exists.

# Outline

---



- Motivations
- Pre-writing configuration
  - Prerequisites
  - File organization
  - Figure preparation
- SEGTeX
  - Implement steps
  - Rules and tips
- Alternative L<sup>A</sup>T<sub>E</sub>X template
- Simple Example

# Prerequisites



## 1) Download and install Madagascar

**MADAGASCAR**

下载  
Installation  
GitHub repository  
SEGTeX

Introduction  
Package overview  
Tutorial  
Hands-on tour  
Reproducible documents  
Hall of Fame

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

Developer documentation  
Adding programs  
Contributing programs  
API docstring clipping data  
API docstring explicit finite differences

Community  
Conferences  
User mailing list  
Developer mailing list  
GitHub organization  
LinkedIn group  
Development blog  
Twitter  
Slack

工具  
输入界面  
相关更改  
特殊界面  
打印输出  
固定链接  
页面信息

页面 讨论

安装

Madagascar has been installed and is periodically tested on different platforms

- Linux distributions (Fedora, RedHat, Ubuntu, etc.)
- MacOS X
- Windows under the [Cygwin](#) environment.

目录 [隐藏]

- Precompiled binary packages
- Installation from source
  - Platform-specific instructions
  - Prerequisites
  - General instructions
- Uninstall
- Troubleshooting
  - Alternative build/install procedure
  - Other issues
- Testing and quick start

Precompiled binary packages [编辑]

See the Madagascar download page for the availability of binary packages.

Installation from source [编辑]

Platform-specific instructions [编辑]

Specific instructions are provided for Windows users.

Please consult the Advanced installation guide for other platform-specific instructions.

Prerequisites [编辑]

- C compiler. Any ANSI-compliant compiler such as [GCC](#) should work. GCC usually comes pre-installed on Linux machines.
- Python interpreter. [Python](#) is an interpretable programming language. It is used in Madagascar installation scripts and project management scripts. Python comes pre-installed on some platforms. Madagascar supports Python 3.

Each API option has its own prerequisites.

Prerequisites for optional add-on functionality are platform-specific.

Building under cygwin on Windows requires several preparatory steps.

General instructions [编辑]

- Configuration: Open a terminal window, change to the top Madagascar source directory, and run

```
./configure --prefix=/directory/where/you/want/madagascarinstalled
```

You can examine the config.py file that this command generates. Additional options are available. You can obtain a full list of customizable variables by running `scos -h`. For example, to install Fortran-90 API bindings in addition to the basic package, run

```
./configure API=F90 --prefix=/directory/where/you/want/madagascarinstalled
```



<https://ahay.org/wiki/Installation>

## 2) Download and install LaTeX Package (TeXLive)

### a. Manual download:

<https://tug.org/texlive/>

### b. Package manager (Ubuntu 22.04)

`sudo apt update`

`sudo apt install texlive-full`

install texlive is OK, if you know  
exactly which packages you'll need

Package	Installed Size	Main Dependencies
texlive	~350 Mb	texlive-latex-base, texlive-latex-recommended, texlive-fonts-recommended
texlive-full (recommended)	~5.3 Gb	Entire TeXLive distribution

# Prerequisites



## 3) Download and configure SEGTeX

(1) Download from: <http://www.ahay.org/wiki/SEGTEx>

SEGTEx is a LaTeX package for geophysical publications. It consists of

- LaTeXe class files for Geophysics papers, SEG expanded abstracts, etc
- BibTeX style files `seg.bst`
- BibTeX cumulative bibliography of geophysical publications `SEG.bib`
- `latex2html` customizations

**目录 [隐藏]**

- 1 Downloading
- 2 Installation
  - 2.1 Prerequisites
- 3 Download and Install SEGTEx on Mac
- 4 SEG expanded abstracts
- 5 Documentation
- 6 Tips and tricks
- 7 Communication
- 8 Reproducible papers
- 9 LaTeX references
- 10 News

**Downloading [编辑]**

Access SEGTEx through its SourceForge project by downloading the latest stable release.  
Alternatively, access the current working version by using Git and running

```
git clone https://github.com/SEGTeX/texmf
```

You can also browse the GitHub repository.

**Installation [编辑]**

If your LaTeX installation is missing/incomplete, try installing TeX Live first.  
To install, put the contents of the `texmf` folder where LaTeX can find it. Most systems recognize `$HOME/texmf` as one of the default places. On MacOS X, it can be `$HOME/Library/texmf`. You may need to run `texhash` to tell LaTeX about the new files. For more help on `texmf`, see "Private" installations of files.



SOCIETY OF EXPLORATION  
GEOPHYSICISTS  
Society of Exploration Geophysicists

(2) Add environment variable `TEXMFHOME`:

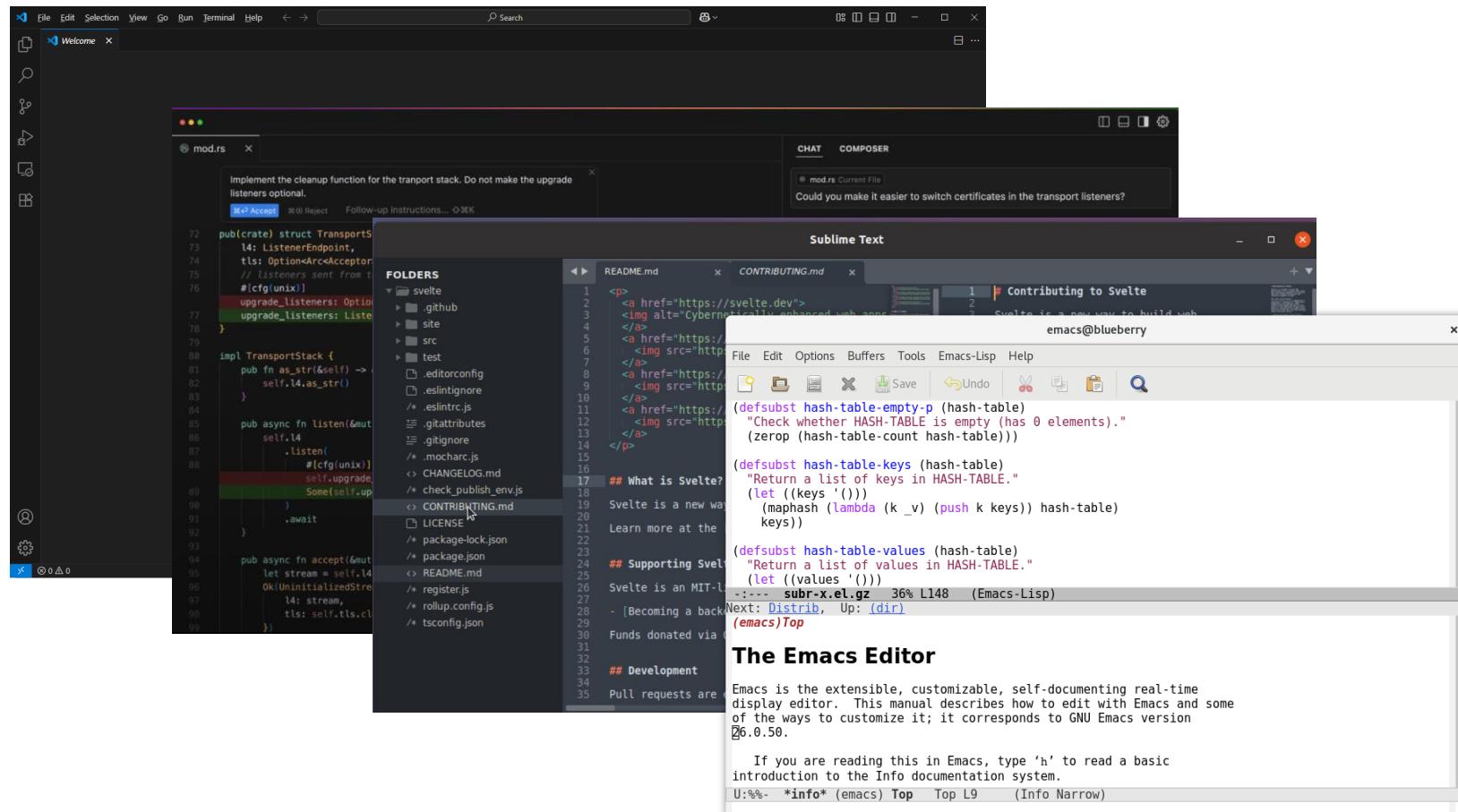
```
echo "export TEXMFHOME=$HOME/texmf" >> ~/.bashrc
```

# Prerequisites



## 4) Choose your favorite text editor

VSCode, Cursor, Sublime Text, Emacs, Gedit, Vim, ...



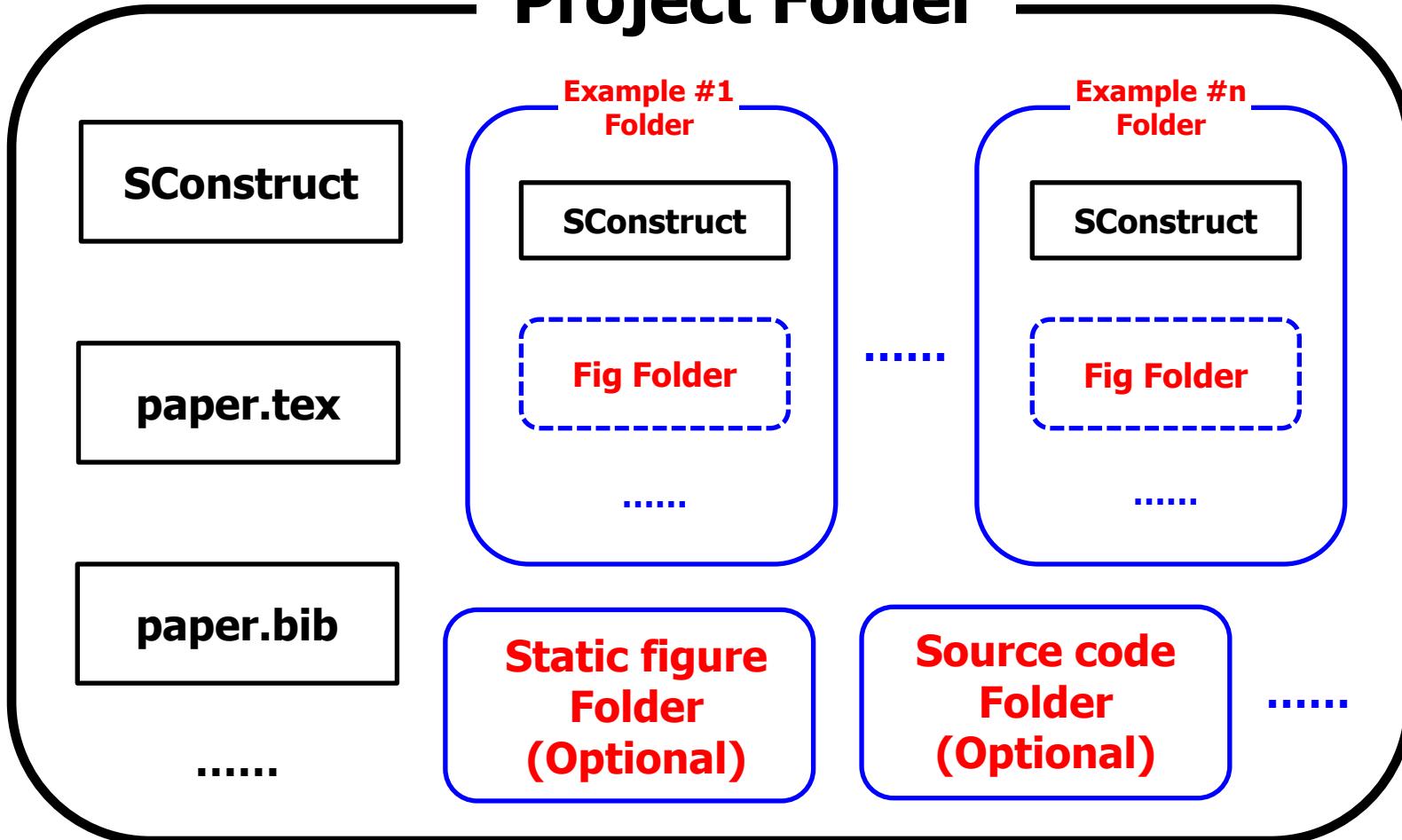
# Outline

---



- Motivations
- Pre-writing configuration
  - Prerequisites
  - File organization
  - Figure preparation
- SEGTeX
  - Implement steps
  - Rules and tips
- Alternative L<sup>A</sup>T<sub>E</sub>X template
- Simple Example

## Project Folder

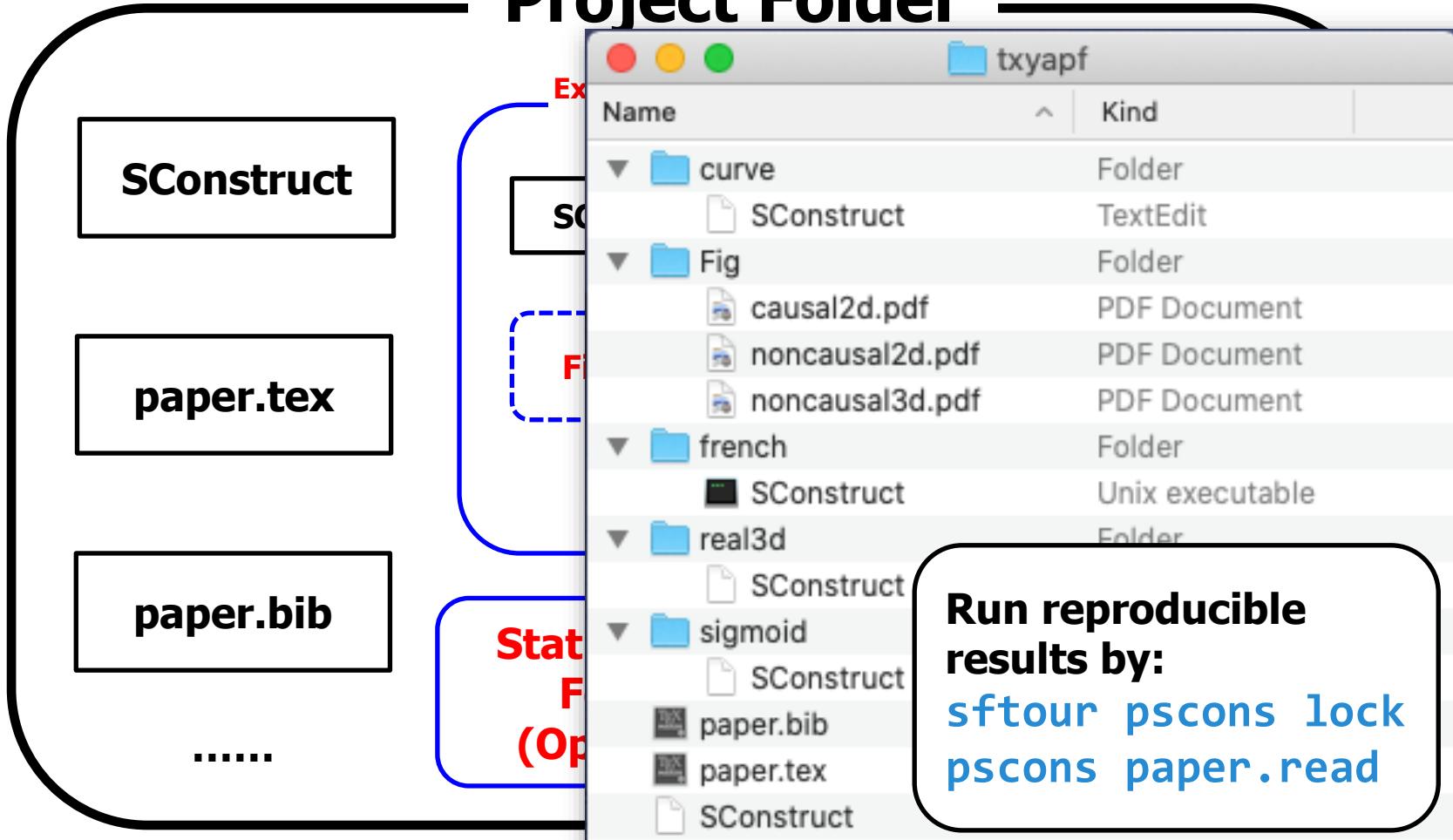


**Look inside example: \$RSFSRC/book/jlu/txsopf**

# File organization



## Project Folder



Look inside example: **\$RSFSRC/book/jlu/txsopf**

# Outline

---



- Motivations
- Pre-writing configuration
  - Prerequisites
  - File organization
  - Figure preparation
- SEGTeX
  - Implement steps
  - Rules and tips
- Alternative L<sup>A</sup>T<sub>E</sub>X template
- Simple Example

# Figure preparation



## a. Static figures:

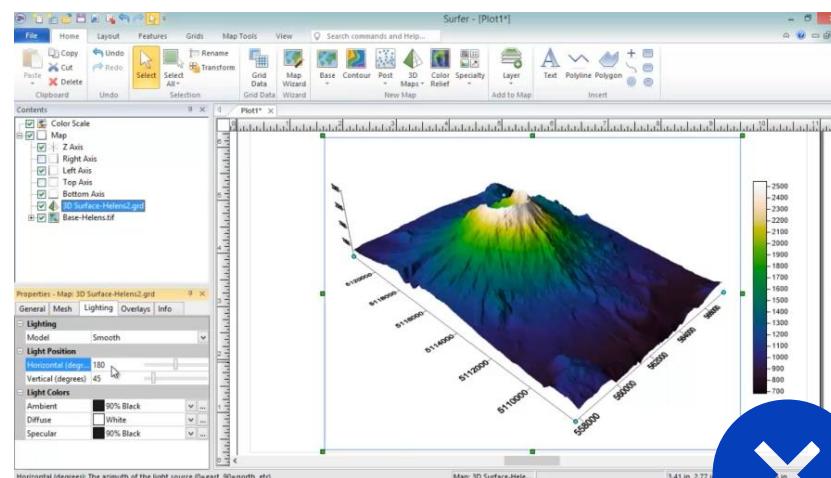
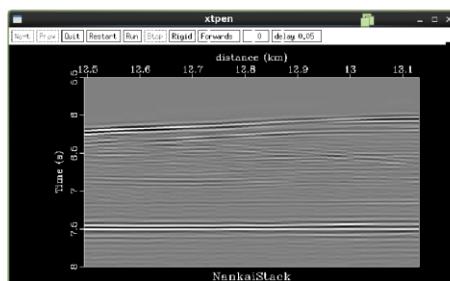
**latex-friendly format: pdf, eps, ...**

## b. Reproducible results:

**Use code/script to generate figures: Madagascar, Python,**

**Matlab, GNU Octave, GMT, ...**

```
sfsuread <Nstack.su suxdr=y \
| sput d2=0.0125 unit2=km label2=distance title=NankaiStack \
| sfwindow min1=5.5 max1=8 f2=1000 n2=51 \
| sfbandpass flo=5 fhi=20 >stk.rsf
<stk.rsf sfgray | sfpn
```



# Figure preparation



## Plotting with Madagascar

- Plotting tools: rsf → vpl

- sfgraph
  - sfgraph3
  - sfcontour
  - sfcontour3
  - swiggle
  - sfgrey
  - sfgrey3
  - sfgrey4

- Pens: vpl → jpg/png/tif/gif...

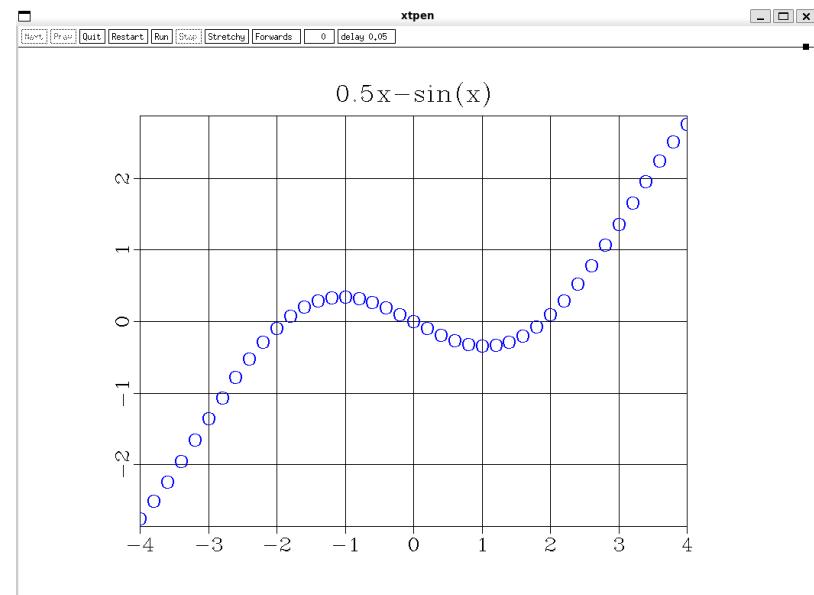
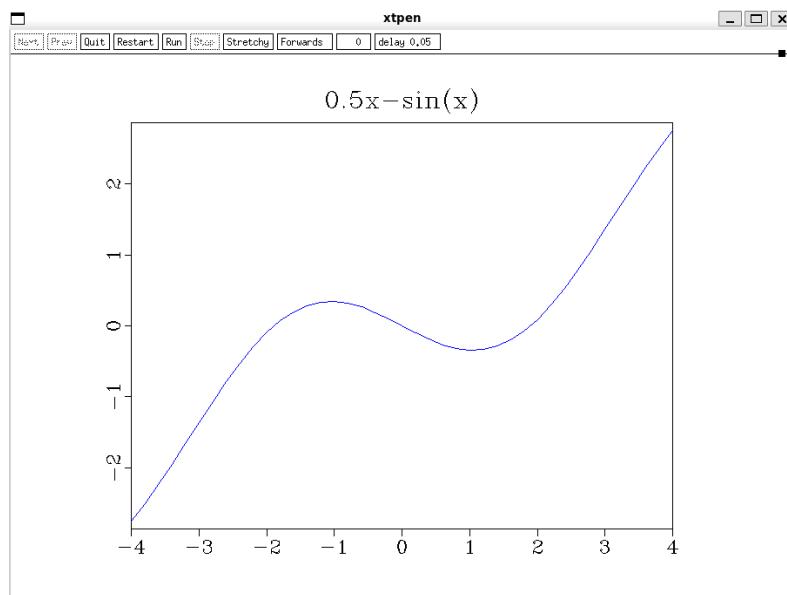
- vppen
  - pspen
  - xtpen
  - .....

Find doc also from sfstdplot:  
<http://ahay.org/RSF/sfstdplot.html>

```
sfmath n1=41 o1=-4 d1=.2 output=".5*x1-sin(x1)" > y.rsf
```

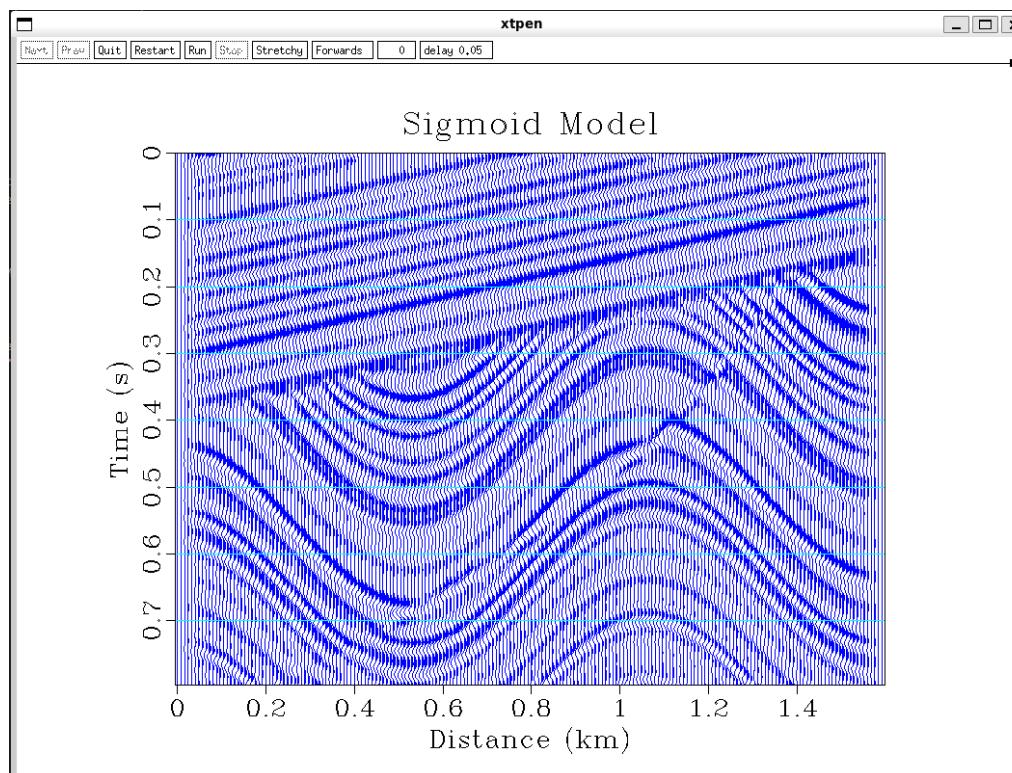
```
< y.rsf sfgraph title="0.5x-sin(x)" min1=-4 max1=4 | sfpn
```

```
< y.rsf sfgraph title="0.5x-sin(x)" symbol=o symbolsz=12 grid=y min1=-4 max1=4 | sfpn
```

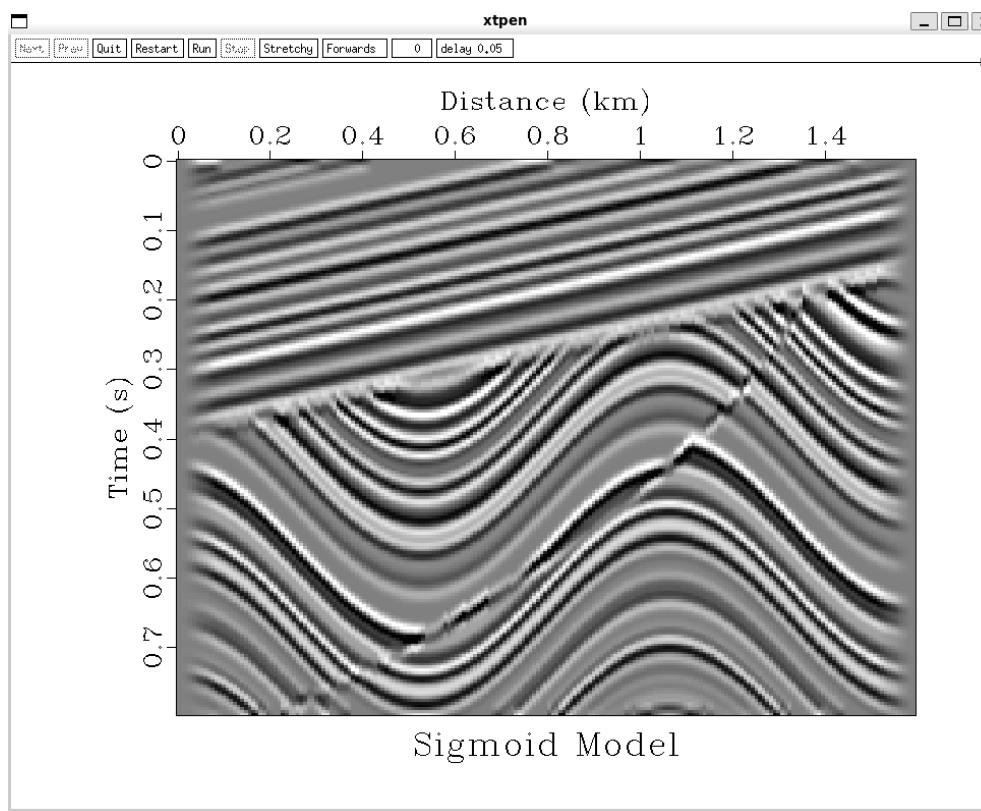


**export SFPENOPTS='bgcolor=w'**

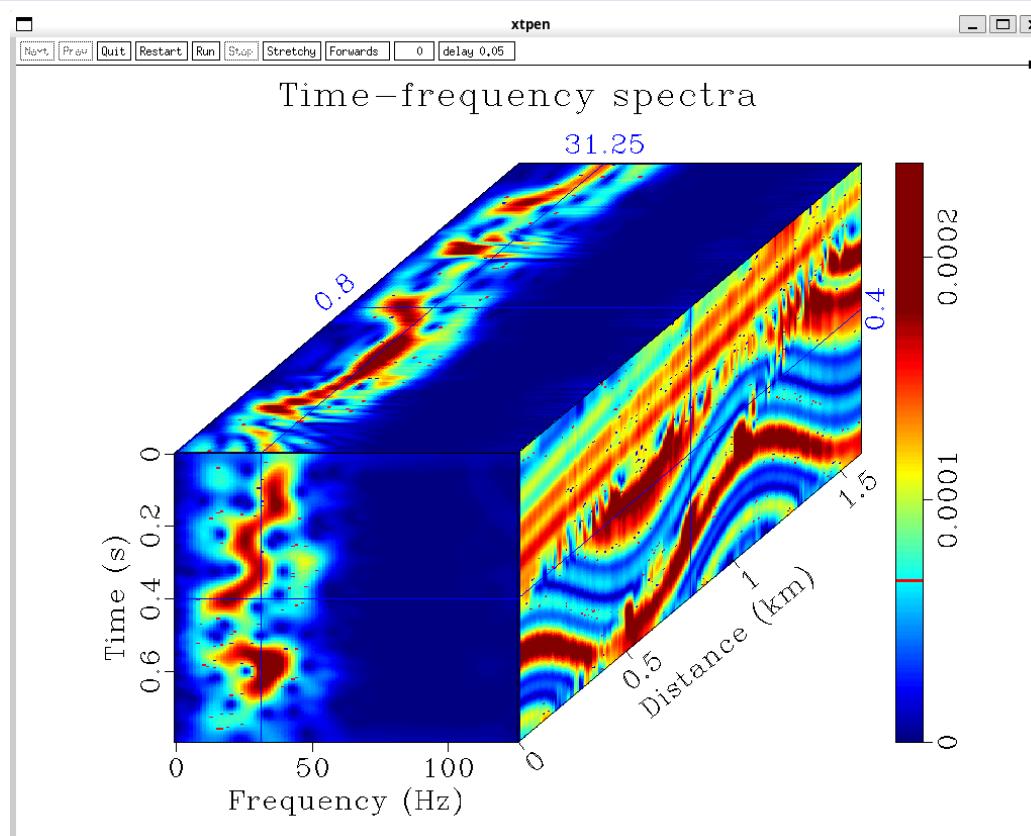
```
sfsigmoid n1=200 n2=200 d1=.004 d2=.008 \
| sfsmooth rect1=3 diff1=1 | sfsmooth rect1=3 \
| sfput label2=Distance > model.rsf
< model.rsf sfwiggle yreverse=y transp=y poly=y title="Sigmoid Model" | sfpn
```



```
sfsigmoid n1=200 n2=200 d1=.004 d2=.008 \
| sfsmooth rect1=3 diff1=1 | sfsmooth rect1=3 \
| sfput label2=Distance > model.rsf
< model.rsf sfgrey title="Sigmoid Model" | sfpen
```



```
< model.rsf sflfft niter=10 rect=5 | sfcabs > lfft.rsf
< lfft.rsf sfbyte bar=bar.rsf allpos=y clip=2e-4 \
| sfgrey3 title="Time-frequency spectra" bar=bar.rsf scalebar=y \
color=j frame1=100 frame2=25 frame3=100 flat=n | sfpen
```



# sfdoc stdplot



```
NAME
    sfdstdplot
DESCRIPTION
    Setting up frames for a generic plot.
SYNOPSIS
    sfstdplot backcol= fillcol= dash= plotfat= plotcol= xreverse=xreverse1 yreverse=yreverse1 pad=pad1 scalebar=n barmove=n tickscale=0.5
    tickscale1=tickscale tickscale2=tickscale tickscale3=tickscale tickscale4=tickscale min1=umin1 min2=umin2 max1=umax1 max2=umax2 font=-1 screen
    ratio=VP_SCREEN_RATIO screenht=VP_STANDARD_HEIGHT screenwd=screenht / screenratio crowd=0.75 xinch= crowd1=crowd yinch= crowd2=crowd xll= xur=
    yll= yur= barwidth=0.36 axiscol=VP_WHITE framelabelcol=VP_YELLOW cubelinecol=framelabelcol labelsz=8. larnersz=0.0f labelrot=n grid1=transp?
    false: grid grid2=transp? grid: false gridcol=grid? VP_RED: framecol gridfat=1 griddash=0.0f titlesz=10. barlabelsz= framelabel1=(bool) (NULL
    != label1) framelabel2=(bool) (NULL != label2) framelabel3=(bool) (NULL != label3) axisfat=0 axiscol=7 labelfat=0 labelsz=8. wantaxis= screen
    ratio=VP_SCREEN_RATIO screenht=VP_STANDARD_HEIGHT screenwd=screenht / screenratio crowd=0.75 xinch= crowd1=crowd yinch= crowd2=crowd xll= xur=
    yll= yur= transp=transpi xreverse=xreverse1 yreverse=yreverse1 labelrot=n min1= min2= max1= max2= titlefat=0 titlesz=10. wantaxis= wantaxis
    sl= wantaxis2= wantaxis3= labelfat= label1= label1= unit1= unit3= label2= unit2= nbartic= dbarnum= obarnum= whereartics= nltic= dlnum= olnum=
    n2tic= d2num= o2num= n3tic= d3num= o3num= n4tic= d4num= o4num= wheretics= grid1= glnum0= glnum= grid2= g2num0= g2num= title= barlabelfat= bar
    label= barunit= bartype= wherexlabel= whereylabel= formatbar= format2= format1= format3= wheretitle= wherebarlabel=
PARAMETERS
    int    axiscol=7
    int    axisfat=0
    floats backcol=          [3]
    string barlabel=          ( barlabel bar label )(bar label)
    int    barlabelfat=        bar label fatness
    float  barlabelsz=         bar label font size
    bool   barmove=n [y/n] adjust scalebar position, if bartype=h
    string bartype=            [v,h] vertical or horizontal bar (default is v)
    string barunit=            ( barunit bar unit )(bar unit)
    float  barwidth=0.36      scale bar size
    float  crowd=0.75
    float  crowd1=crowd
    float  crowd2=crowd
    int    cubelinecol=framelabelcol      cube lines color
    float  d1num=    axis1 tic increment
    float  d2num=    axis2 tic increment
    float  d3num=    axis3 tic increment
    float  d4num=    axis4 tic increment
    floats dash=      line dash type
        0 continuos (default)
        1 fine dash
        2 fine dot
        3 dash
        4 large dash
        5 dot dash
        6 large dash small dash
        7 double dot
        8 double dash
        9 loose dash The part after the decimal point determines the pattern repetition interval [n2]
    string dbarnum=           scalebar tic increment
    floats fillcol=            [3]
    int    font=-1             font to use in text
    string format1=            tick mark format
    string format2=            tickmark format ()
```

line 1 (press h for help or q to quit)

# Tips for M8R plotting



## 1) sfdoc stdplot

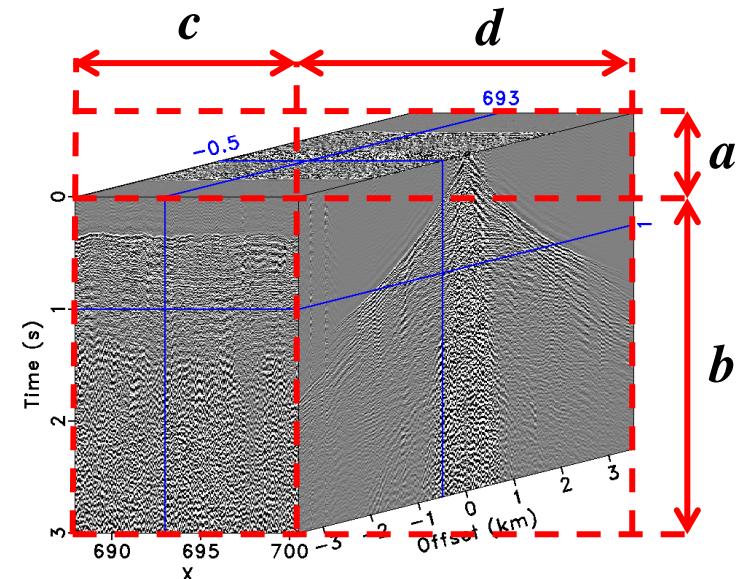
## 2) Aspect ratio settings for sfgrey3

$$\text{point1} = \frac{b}{a+b} = 0.8$$

$$\text{point2} = \frac{c}{c+d} = 0.4$$

## 3) Element color

Lines, labels, frames, grids...

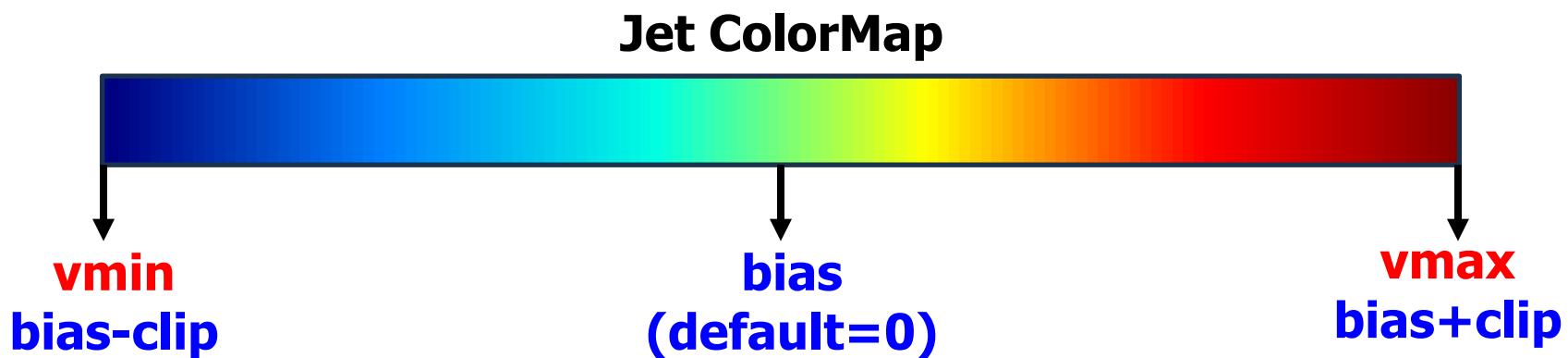


Index	0	1	2	3	4	5	6	7
bgcolor=w	white	yellow	cyan	green	magenta	red	blue	black
bgcolor=b	black	blue	red	magenta	green	cyan	yellow	white

# Tips for M8R plotting



## 4) Clip/bias (m8r), vmin/vmax (Python) and caxis/CLim (Matlab)



**allpos=y:** ignore **bias=**, **vmin=0**, **vmax=clip**

**Matlab:** `caxis([vmin, vmax]);`  
**or:** `ax=gca; ax.CLim = [vmin, vmax];`

# Tips for M8R plotting



## 5) Figure combinations

```
Plot('subfig1', 'model', 'grey')
```

```
Plot('subfig2', 'model ', 'wiggle transp=y yreverse=y poly=y')
```

```
Result('combined','subfig1 subfig2','SideBySideAniso')
```

### Combinations

Movie

Overlay

SideBySideAniso

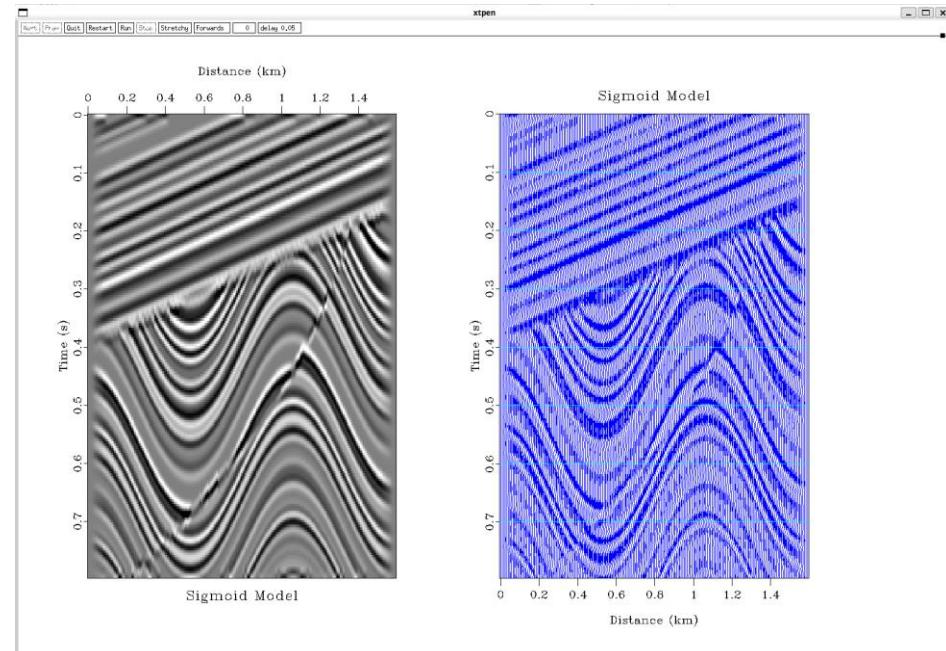
OverUnderAniso

SideBySidelso

OverUnderlso

TwoRows

TwoColumns



# Plotting with Python: sfmatplotlib



## Plotting with matplotlib

September 12, 2019    Programs    1 comment



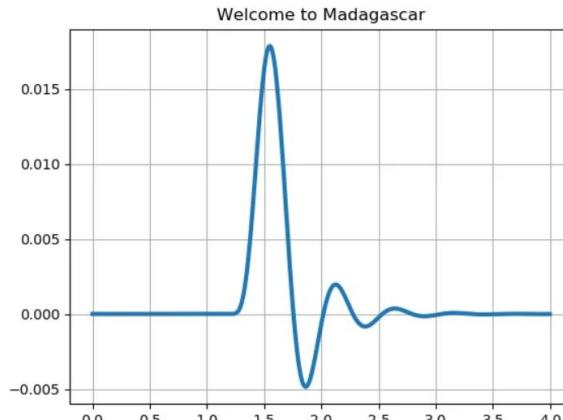
Matplotlib is a popular Python library for 2D plotting, which emulates (and exceeds) the plotting functionality of MATLAB.

At the [Madagascar Working Workshop](#) in 2018, a couple of new scripts were added to simplify plotting with Matplotlib in Madagascar.

- `sfmatplotlib` can be used to plot RSF files. You can use it as follows:

```
</>
sfmatplotlib <matplotlib function> <plot options> [format=eps] < inp.rsf [ > out.eps]
```

Without the standard output, the figure is displayed on the screen.



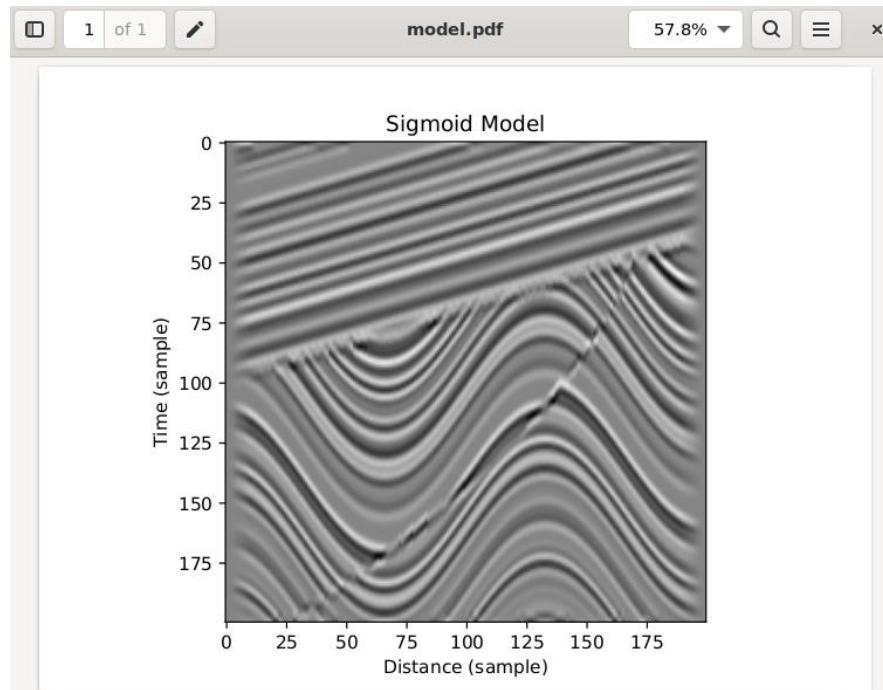
The figure above was generated with

```
</>
sfspike n1=1000 k1=300 | sfbandpass fhi=2 phase=y| sfmatplotlib plot title="Welcome to Maagascar" grid=y linewidth=3 format=png > figure.png
```

# Plotting with Python: sfmatplotlib



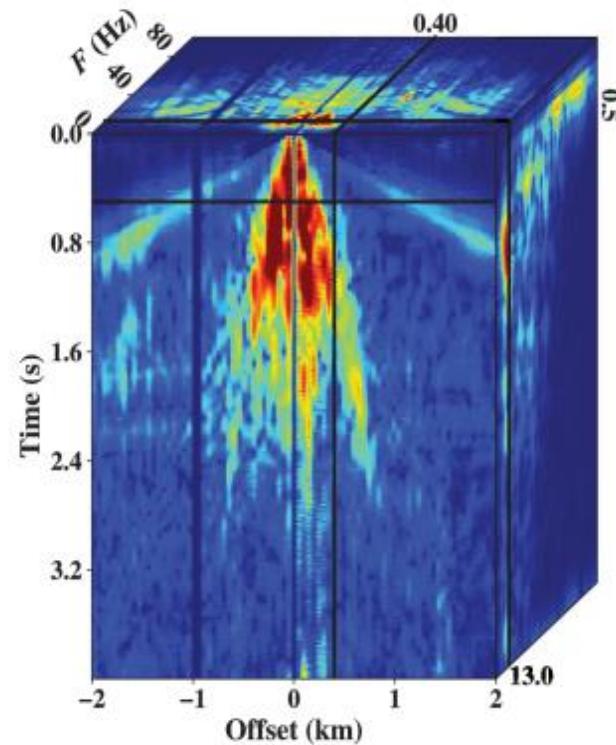
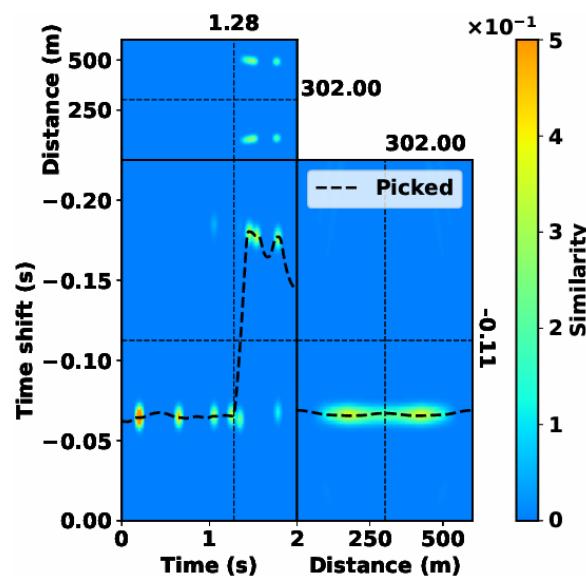
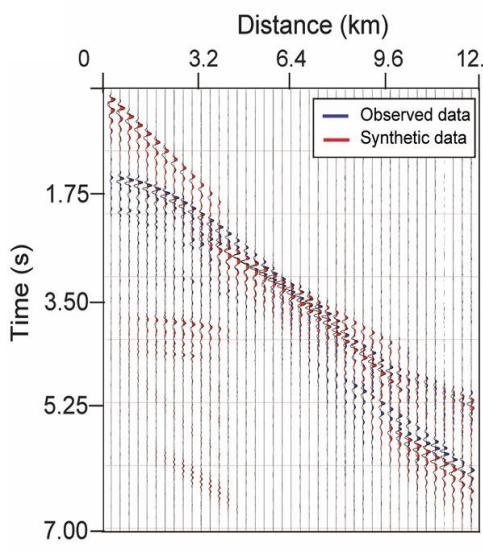
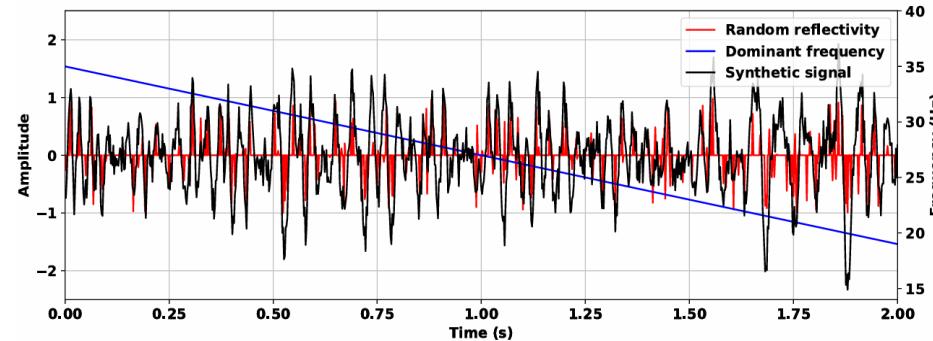
```
sfsigmoid n1=200 n2=200 d1=.004 d2=.008 \
| sfsmooth rect1=3 diff1=1 | sfsmooth rect1=3 \
| sfput label2=Distance > model.rsf
< model.rsf sftransp | sfmatplotlib imshow cmap=gray title="Sigmoid Model" xlabel="Distance
(sample)" ylabel="Time (sample)" format=pdf > model.pdf
```



# Plotting with Python: write your own script



## Gallery



## How to implement

**1) Choose a command-line style.**

**2) Write your script.**

- **Read your data.**
- **Plot and visualize.**
- **Save your figure.**



## 1) Choose a command-line style:

### a. No arguments and no stdin/stdout

```
python your_script.py
```

### b. No arguments but stdin/stdout

```
< data.rsf python your_script.py > figure.pdf
```

### c. No stdin/stdout but arguments

```
python your_script.py input=data.rsf \
    output=figure.pdf
```

### d. Both arguments and stdin/stdout

```
< data.rsf python your_script.py arg1=val1 \
    arg2=val2 ... > figure.pdf
```

## 1) Choose a command-line style:

### a. No arguments and no stdin/stdout

```
Result("figure", "data", "python your_script.py", stdin=0, stdout=-1, suffix=".pdf")
```

### b. No arguments but stdin/stdout

```
Result("figure", "data", "python your_script.py", suffix=".pdf")
```

### c. No stdin/stdout but arguments

```
Result("figure", "data", """  
python your_script.py input=${SOURCES[0]} output=${TARGETS[0]}  
""", stdin=0, stdout=-1, suffix=".pdf")
```

### d. Both arguments and stdin/stdout

```
Result("figure", "data", """  
python your_script.py arg1=val1 arg2=val2 ...  
""", suffix=".pdf")
```

## 2) Write your script:

- ✓ Use **m8r** to read rsf data.
- ✓ Use **numpy** to deal with data volume and calculation.
- ✓ Use **matplotlib** to visualize your results.
- ✓ (Optionally) use AI to assist your coding.



```
from rsf.proj import *

# Sigmoid model
Flow('model',None,
  ''
    sigmoid n1=200 n2=200 d1=.004 d2=.008 |
    smooth rect1=3 diff1=1 | smooth rect1=3 |
    put label2=Distance title='''Sigmoid Model'''
  '')
Result('figure', 'model', 'python script.py', suffix='.pdf')
End()
```

# Plotting with Python: simple example

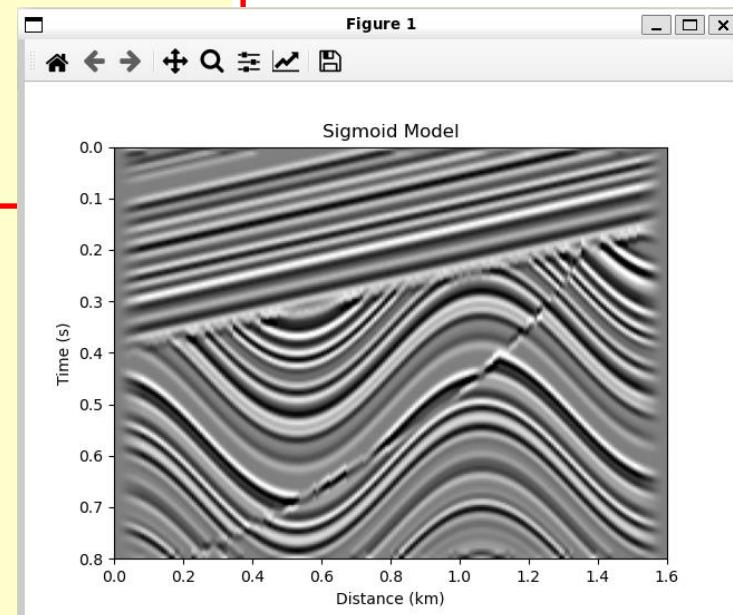


```
import m8r, numpy, sys
import matplotlib.pyplot as plt

if __name__ == "__main__":
    # check stdin
    if sys.stdin.isatty():
        print("Usage: python greystest.py < data.rsf [> output.pdf]",
file=sys.stderr)
        sys.exit(1)

    # read data from stdin      Use m8r.Input() to read rsf data and header.
    inp = m8r.Input()
    # read data and axes
    n1, n2 = inp.int('n1'), inp.int('n2')
    d1, d2 = inp.float('d1'), inp.float('d2')
    o1, o2 = inp.float('o1'), inp.float('o2')
    l1, l2 = inp.string('label1'), inp.string('label2')
    u1, u2 = inp.string('unit1'), inp.string('unit2')
    title = inp.string('title')
    data = inp.read(datatype=numpy.float32).T
    # calculate clip
    clip = numpy.percentile(numpy.abs(data), 99)
    # display data
    plt.imshow(data, aspect='auto', cmap='gray',
               vmin=-clip, vmax=clip,
               extent=(o2, o2 + d2 * n2, o1 + d1 * n1, o1))
    plt.ylabel(f"{l1} ({u1})")
    plt.xlabel(f"{l2} ({u2})")
    if title: plt.title(title)
    # check stdout
    if sys.stdout.isatty(): plt.show()
    else: plt.savefig(sys.stdout.buffer, format='pdf')
    sys.exit(0)
```

Always float32 for rsf float data.



# Plotting with Python: simple example



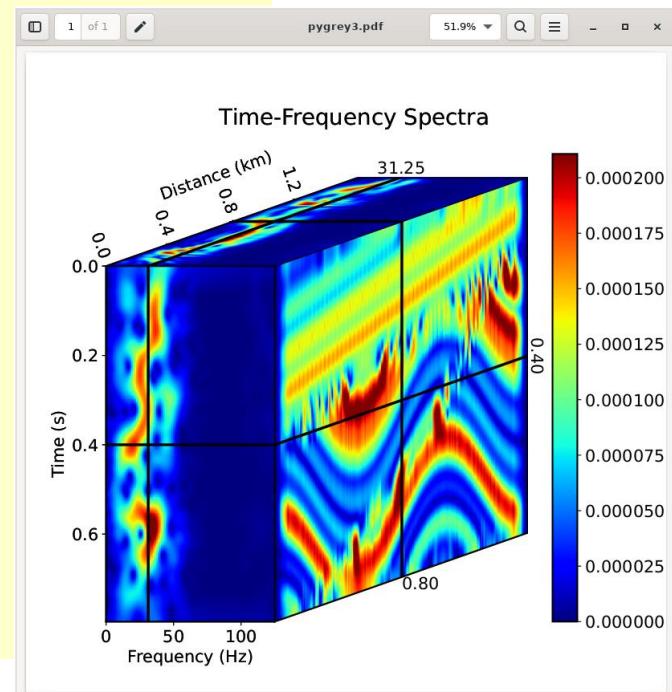
```
....  
# read header from stdin  
inp = m8r.Input()  
n1, n2, n3 = inp.int('n1'), inp.int('n2'), inp.int('n3')  
d1, d2, d3 = inp.float('d1'), inp.float('d2'), inp.float('d3')  
o1, o2, o3 = inp.float('o1'), inp.float('o2'), inp.float('o3')  
l1, l2, l3 = inp.string('label1'), inp.string('label2'), inp.string('label3')  
u1, u2, u3 = inp.string('unit1'), inp.string('unit2'), inp.string('unit3')  
title = inp.string('title')
```

Use `m8r.Input()` to read parameters

```
# Read parameters from command line  
pars = m8r.Par()  
frame1 = pars.int('frame1', 0)  
frame2 = pars.int('frame2', 0)  
frame3 = pars.int('frame3', 0)  
point1 = pars.float('point1', 0.8)  
point2 = pars.float('point2', 0.4)  
clip = pars.float('clip', None)  
pclip = pars.float('pclip', 99.)  
isflat = pars.bool('flat', True)  
allpos = pars.bool('allpos', False)  
colorbar = pars.bool('scalebar', False)  
color = pars.string('color', 'gray')  
title = pars.string('title', title)
```

`pars.par_type(par_key, default)`

```
....  
69 Result('pygrey3',[ 'ltft', grey3script],  
70     ''  
71     | ${SOURCES[1]} frame1=100 frame2=25 frame3=100  
72     | point1=0.8 point2=0.4 clip=2e-4 flat=n scalebar=y  
73     | title="Time-Frequency Spectra" allpos=y color=jet  
74     | ''', suffix='.pdf')
```



**For example, let's ask the Copilot to generate a wiggle plot:**

*Here is a python script used to generate a grey-scale image. Please use the framework and make it generate a wiggle plot instead: import m8r..... (dump the previous code here)*

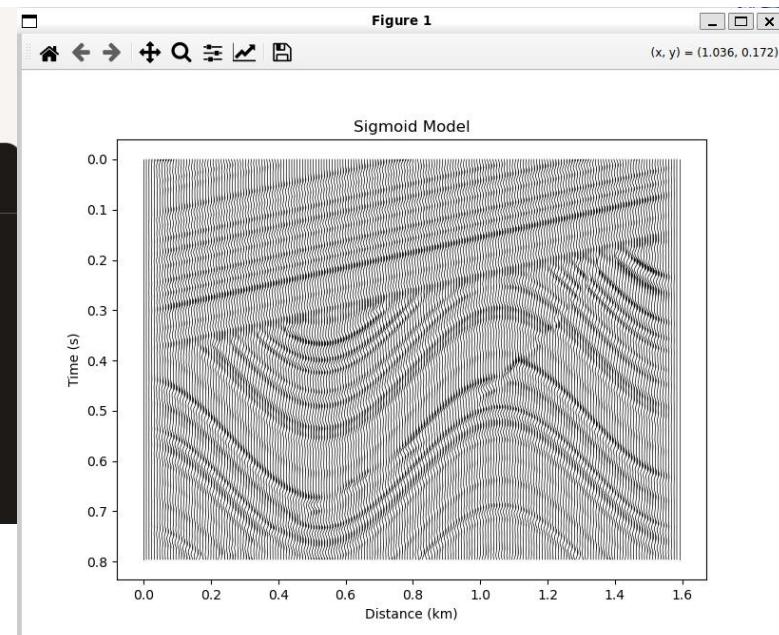
**And the Copilot replies:**

Here's a drop-in replacement of your script that reads the same RSF data but renders a classic seismic-style wiggle plot instead of a grey-scale image. You can adjust the `scale_factor` or `fill_positive` flags to taste.

```
Python
```

```
import m8r, numpy, sys
import matplotlib.pyplot as plt

def wiggle_plot(data, time, o2, d2, scale_factor=1.0,
                color='black', linewidth=0.5, fill_positive=True):
    """
    Draws a wiggle plot for 2D array `data` of shape (n_samples, n_traces).
    time : 1D array of length n_samples
    o2, d2 : origin and sampling interval on the horizontal axis
    scale_factor : multiplies amplitudes before plotting
    """
    ....
```



## Avoid directly asking AI like this:

*Please write a python script that read rsf data from stdin and then plot its grey-scale image. Save the image as a pdf to stdout.*

## And the Copilot replies:

Here's a self-contained Python script that reads an RSF volume from stdin, plots it in greyscale with Matplotlib, and writes the PDF image to stdout:

```
Python □ 复制  
  
#!/usr/bin/env python3  
import sys  
import numpy as np  
import matplotlib.pyplot as plt  
from rsf.api import rsf_input  
  
def main():  
    # Read RSF from stdin  
    a = rsf_input()          # by default reads from sys.stdin  
    n1 = a.getint('n1')      # number of samples in fast axis  
    n2 = a.getint('n2', 1)    # number of samples in slow axis (default 1)  
  
    # Load the full volume into a 2D numpy array  
    data = a.read()          # shape: (n2, n1) or (n1,) if 1D  
  
    # If 1D, force into a 2D for imshow  
    if data.ndim == 1:  
        data = data[np.newaxis, :]
```

**When there isn't precise context, AI often won't ask you for additional information and instead tends to outright fabricate things that don't exist.**

.....

# Plot with Matlab

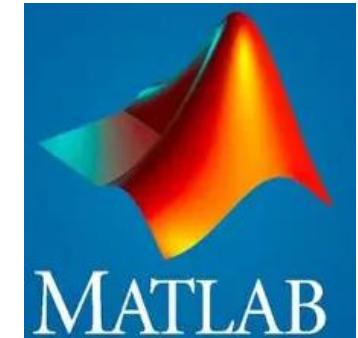


## a. Use Madagascar Matlab api to read rsf data.

- **Require proper installation.**

Check by `ls $RSFROOT/lib/*.mexa64`

- **GNU Octave unsupported.**
- **Matlab version > R2019a.**



```
from rsf.proj import *

# Sigmoid model
Flow('model',None,
    ...
    sigmoid n1=200 n2=200 d1=.004 d2=.008 |
    smooth rect1=3 diff1=1 | smooth rect1=3 |
    put label2=Distance title='''Sigmoid Model'''
    ''))
    
Result('matgrey','model',
    ...
    matlab -batch "matgrey('${SOURCES[0]}', '${TARGETS[0]}', 'Sigmoid Model')"
    '', suffix='.pdf', stdin=0, stdout=-1)

End()
```

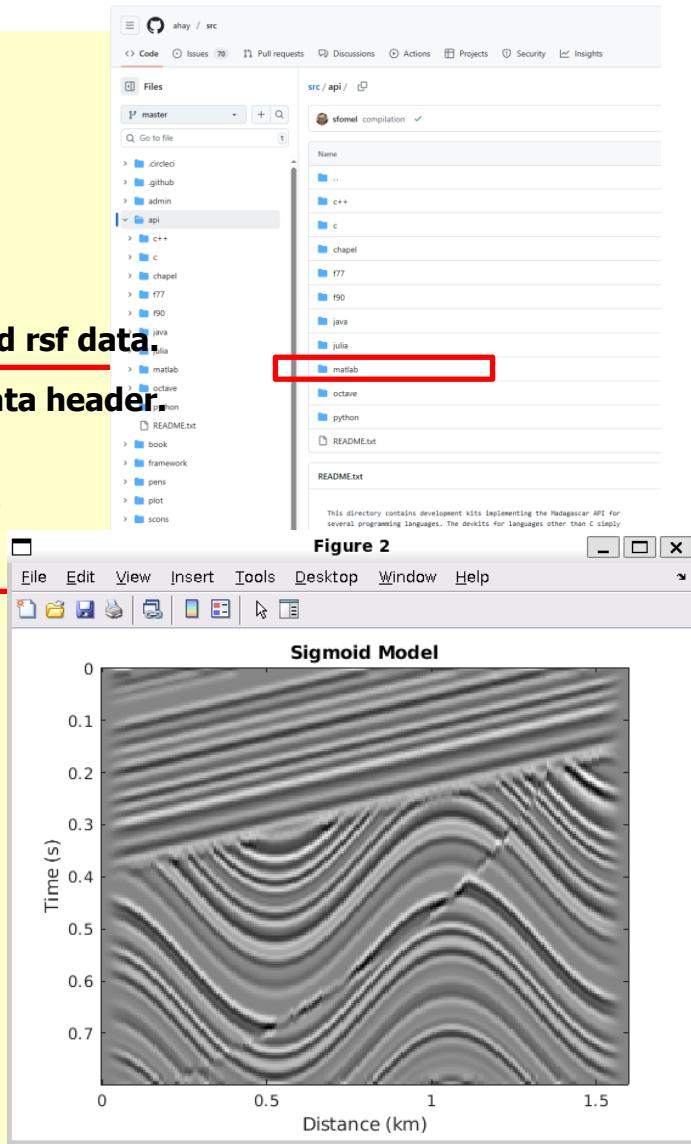
# Plotting with Matlab: simple example



```
function matgrey(rsf_file, outfig, figtitle)
    % Require 2 or 3 input arguments
    narginchk(2,3);
    if isempty(outfig)
        error('The output filename "outfig" cannot be empty.');
    end
    if nargin < 3
        figtitle = '';
    end
    %--- 1. Read RSF header and data
    % Add the RSF Library to MATLAB path
    addpath(fullfile(getenv('RSFROOT'), 'lib'));
    [shape, delta, origin, label, unit] = rsf_read_header(rsf_file);
    data = zeros(shape);
    rsf_read(data, rsf_file);
    time_axis = origin(1) + (0:shape(1)-1) * delta(1);
    spatial_axis = origin(2) + (0:shape(2)-1) * delta(2);
    %--- 2. Display the data in a grayscale figure
    hFig = figure('Color', 'w');
    imagesc(spatial_axis, time_axis, data);
    colormap gray; axis tight;
    xlabel(sprintf('%s (%s)', label{2}, unit{2}));
    ylabel(sprintf('%s (%s)', label{1}, unit{1}));
    if ~isempty(figtitle)
        title(figtitle, 'Interpreter', 'none');
    end
    %--- 3. Save the figure as a PDF
    exportgraphics(hFig, outfig, 'ContentType', 'vector');
    fprintf('Saved figure to %s\n', outfig);
end
```

Use rsf\_read to read rsf data.

Use rsf\_read\_header to read data header.



## b. Read rsf data manually.

- Write your own I/O module.
- Minimum requirements.



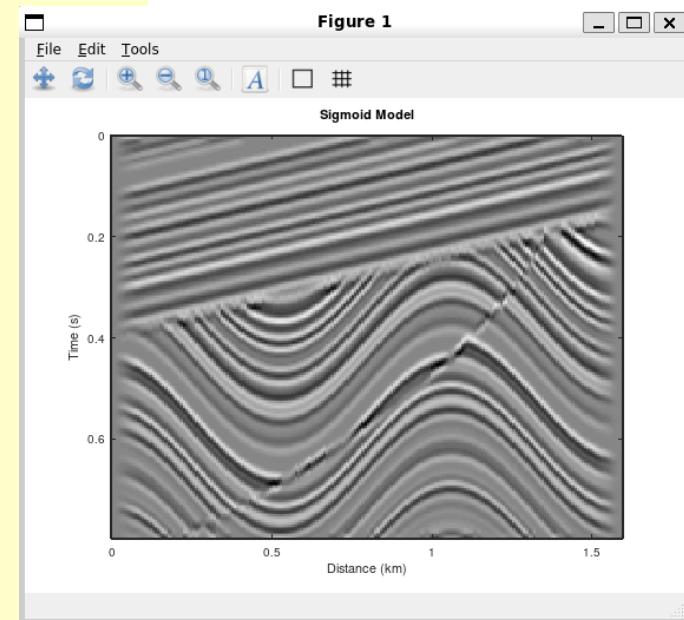
```
from rsf.proj import *
# Sigmoid model
Flow('model',None,
...
    sigmoid n1=200 n2=200 d1=.004 d2=.008 |
    smooth rect1=3 diff1=1 | smooth rect1=3 |
    put label2=Distance title='''Sigmoid Model'''
    ''')
# Matlab version
Result('matgrey','model',
...
    matlab -batch "matgrey('${SOURCES[0]}', '${TARGETS[0]}', 'Sigmoid Model')"
    '', suffix='.pdf', stdin=0, stdout=-1)
# Octave version
Result('octgrey','model',
...
    octave --eval "matgrey('${SOURCES[0]}', '${TARGETS[0]}', 'Sigmoid Model'); exit;"
    '', suffix='.pdf', stdin=0, stdout=-1)
End()
```

# Plotting with Matlab/Octave: simple example



```
function matgrey(rsf_file, outfig, figtitle)
    narginchk(2,3); % Require 2 or 3 input arguments
    if isempty(outfig)
        error('The output filename "outfig" cannot be empty.');
    end
    if nargin < 3
        figtitle = '';
    end
    %--- 1. Read RSF header and data  Use read_rsf to read data and header.
    [data, header] = read_rsf(rsf_file);

    taxis = header.o1 + (0:header.n1-1) * header.d1;
    xaxis = header.o2 + (0:header.n2-1) * header.d2;
    %--- 2. Display the data in a grayscale figure
    hFig = figure('Color', 'w');
    imagesc(xaxis, taxis, data);
    colormap gray; axis tight;
    xlabel(sprintf('%s (%s)', header.label2, header.unit2));
    ylabel(sprintf('%s (%s)', header.label1, header.unit1));
    if ~isempty(figtitle)
        title(figtitle, 'Interpreter', 'none');
    end
    %--- 3. Save the figure as a PDF
    if exist('OCTAVE_VERSION','builtin') ~= 0
        print(hFig, outfig, '-dpdf', '-painters');
    else
        exportgraphics(hFig, outfig, 'ContentType', 'vector');
    end
    fprintf('Saved figure to %s\n', outfig);
end
```



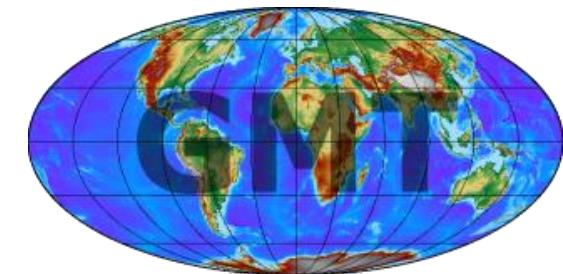
# Plot with GMT: GMT example 02



```
from rsf.proj import *

Result('gmt', 'gmt.sh', 'bash ${SOURCES[0]}',
suffix='.pdf', stdin=0, stdout=-1)

End()
```

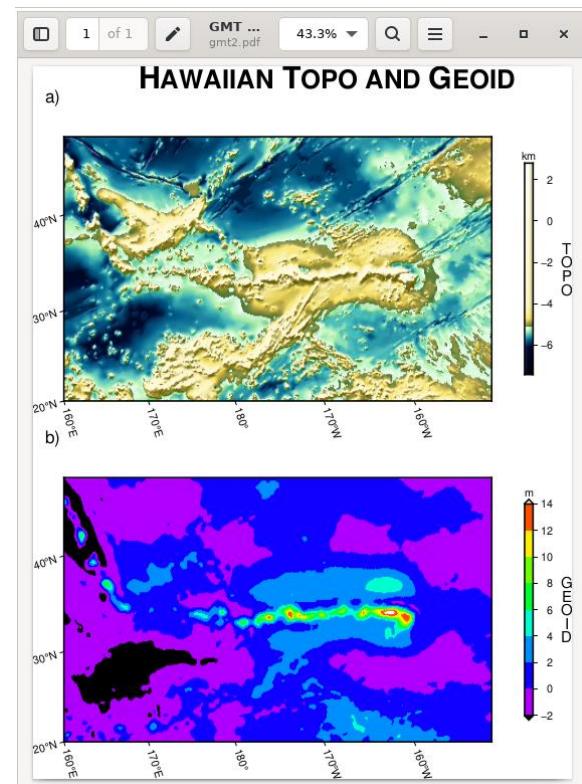


## (2) Image presentations

As our second example we will demonstrate how to make color images from gridded data sets (again, we will defer the actual making of grid files to later examples). We have prepared two 2-D grid files of bathymetry and Geosat geoid heights from global grids and will put the two images on the same page. The region of interest is the Hawaiian Islands, and due to the oblique trend of the island chain we prefer to rotate our geographical data sets using an oblique Mercator projection defined by the hotspot pole at (68°W, 69°N). We choose the point (190, 25.5) to be the center of our projection (e.g., the local origin), and we want to image a rectangular region defined by the longitudes and latitudes of the lower left and upper right corner of region. In our case we choose (160, 20) and (220, 30) as the corners. We twice use `grdimage` to make the illustration:

```
#!/usr/bin/env bash
# GMT EXAMPLE 02
#
# Purpose: Make two color images based gridded data
# GMT modules: set, grd2cpt, grdimage, makecpt, colorbar, subplot
#
gmt begin 02
gmt set GMTOBJOUT separate
gmt subplot begin 2x1 -A+JTL -Fsi6c/9c -M0 -R160/20/220/30+r -J0c190/25.5/292/69/16c -B10 -T"HI@#hawaiian@# T@#opo and @#G@#eoid@#"
gmt subplot set 0,0 -Ce3c
gmt grd2cpt @HI_topo_02.nc -Crelief -Z
gmt grdimage @HI_topo_02.nc -I+a@0
gmt colorbar -D3RM+01c/0+mc -I0.3 -Bx2+lTOPO -By+1km

gmt subplot set 1,0 -Ce3c
gmt makecpt -Crainbow -T-2/14/2
gmt grdimage @HI_geoid_02.nc
gmt colorbar -D3RM+01c/0+e+mc -Bx2+lGEOID -By+1m
gmt subplot end
gmt end show
```



<https://docs.generic-mapping-tools.org/6.6/>

# Plot with GMT: GMT example 13



```
from rsf.proj import *

Result('gmt', 'gmt.sh', 'bash ${SOURCES[0]}',
suffix='.pdf', stdin=0, stdout=-1)

End()
```

[Home](#) / Illustration Gallery / (13) Plotting of vector fields

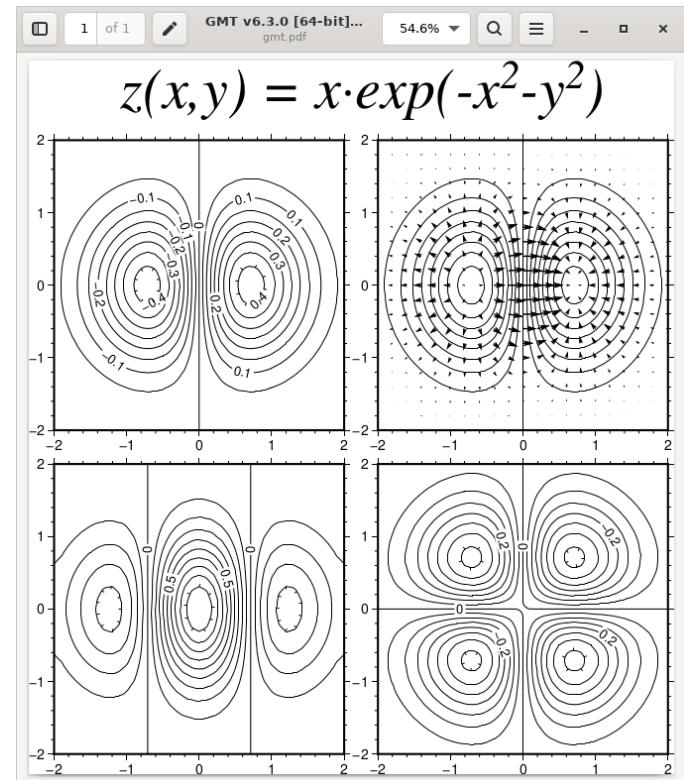
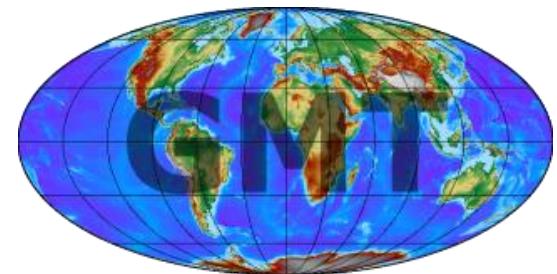
[Edit on GitHub](#)

## (13) Plotting of vector fields

In many areas, such as fluid dynamics and elasticity, it is desirable to plot vector fields of various kinds. GMT provides a way to illustrate 2-component vector fields using the `grdvector` utility. The two components of the field (Cartesian or polar components) are stored in separate grid files. In this example we use `grdmath` to generate a surface  $z(x, y) = x \cdot \exp(-x^2 - y^2)$  and to calculate  $\nabla z$  by returning the  $x$ - and  $y$ -derivatives separately. We superpose the gradient vector field and the surface  $z$  and also plot the components of the gradient in separate windows. A `text` call to place a header finishes the plot

```
#!/usr/bin/env bash
# GMT EXAMPLE 13
#
# Purpose: Illustrate vectors and contouring
# GMT modules: set, grdmath, grdcontour, grdvector, subplot
# Unix progs: rm
#
gmt begin example
gmt set FONT_HEADING 40p,Times-Italic
gmt grdmath -R-2/2/-2/2 -I0.1 X Y R2 NEG EXP X MULT = z.nc
gmt grdmath z.nc DDX = dzdx.nc
gmt grdmath z.nc DDY = dzdy.nc
gmt subplot begin 2x2 -M0.1c -FF15c -BW5ne -T"z(x,y) = x*exp(-x@+2@+-y@+2@+)"
gmt grdcontour z.nc -C0.05 -A0.1 -Gd5c -S4 -T+d8p/2p -c0,0
gmt grdcontour z.nc -C0.05 -Gd5c -S4 -c0,1
gmt grdvector dzdx.nc dzdy.nc -I0.2 -Q0.25e+e+0.25i/0+h0.5 -Gblack -W1p -S2c
gmt grdcontour dzdx.nc -C0.10 -A0.5 -Gd5c -S4 -T+d8p/2p -c1,0
gmt grdcontour dzdy.nc -C0.05 -A0.2 -Gd5c -S4 -T+d8p/2p -c1,1
gmt subplot end
gmt end small
rm -f z.nc dzdx.nc dzdy.nc
```

<https://docs.generic-mapping-tools.org/6.6/>



# Outline

---



- Motivations
- Pre-writing configuration
  - Prerequisites
  - File organization
  - Figure preparation
- SEGTeX
  - Implement steps
  - Rules and tips
- Alternative L<sup>A</sup>T<sub>E</sub>X template
- Simple Example

# Implement steps



**1) Ideas**



**2) Implement ideas in Madagascar and lock Figures  
(scons fig.lock)**

**3) Write a paper text by following SEGTeX rules**

**4) Insert Figures into paper by following SEGTeX rules**

# Outline

---

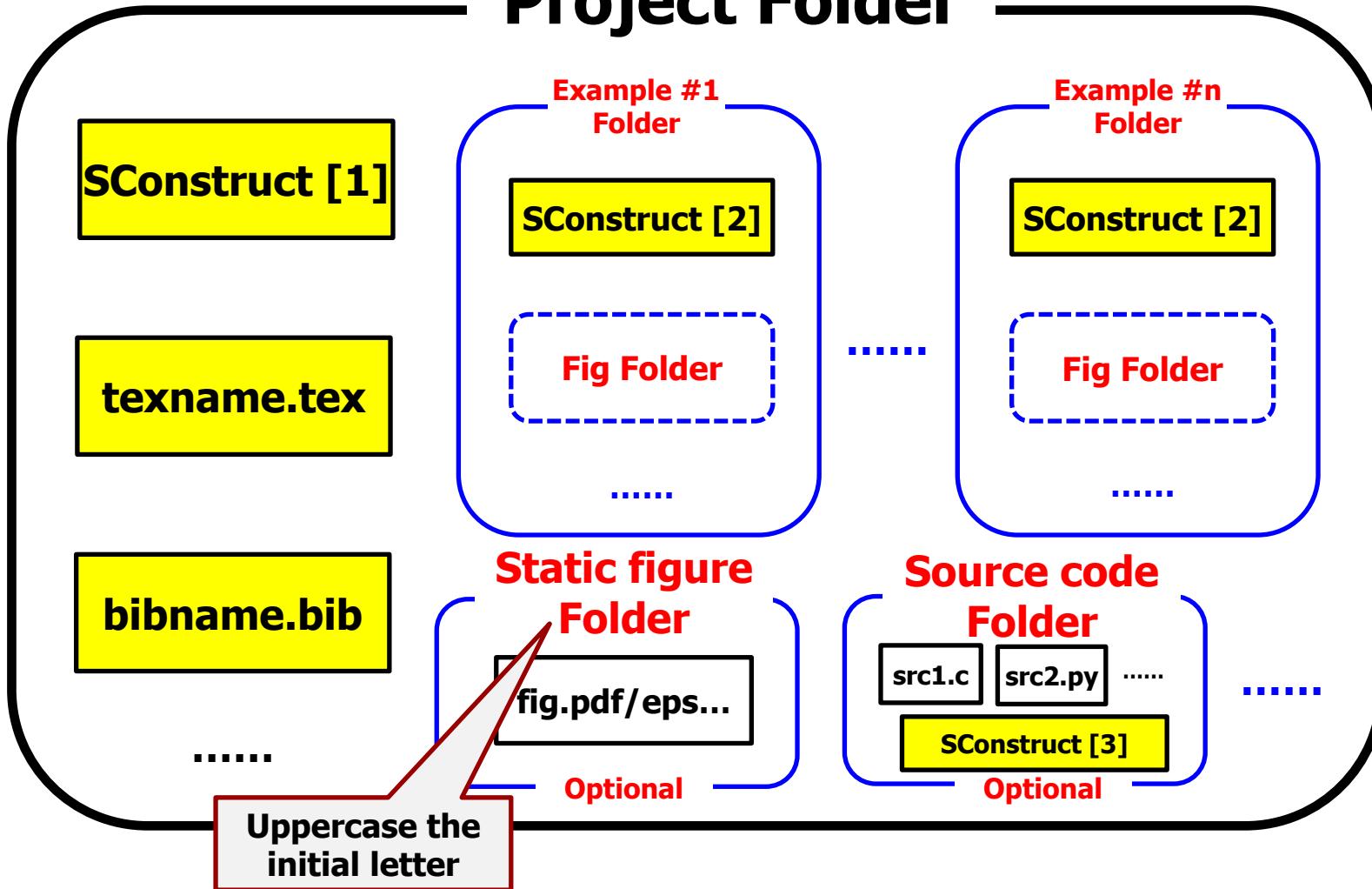


- Motivations
- Pre-writing configuration
  - Prerequisites
  - File organization
  - Figure preparation
- SEGTeX
  - Implement steps
  - Rules and tips
- Alternative L<sup>A</sup>T<sub>E</sub>X template
- Simple Example

# Rules for project structure



## Project Folder



# Rules for the tex file (structure)



**Similar to common L<sup>A</sup>T<sub>E</sub>X format, but with a few differences**

## L<sup>A</sup>T<sub>E</sub>X tex file

### Preamble

```
\documentclass[options]{class}
```

```
\usepackage[options]{package}
```

### Custom Definitions \newcommand,...

```
\begin{document}  
Document Body  
Bibliography  
\end{document}
```

## Madagascar tex file

Document Body  
Bibliography

# Rules for SConstruct [1]



- Always starts with: `from rsf.tex import *`
- Configure your paper by:

```
Paper( 'pdfname' , 'texname' ,  
       lclass='geophysics' ,  
       options='manuscript,revised'  
       use='''amsmath empheq graphicx  
             float threeparttable multirow  
             algorithm [caption=false]subfig  
             '''.split()  
       include='')
```

**texname.tex**

**Document class**

**options**

**Used packages  
\usepackage**

**Custom definitions**

- Ends with `End()`
- Specially, if paper name is “paper.tex”, you can directly use `End()` similar to `Paper()`.

`\usepackage[caption=false]{subfig}`

## 1) SEP report:

**no need lclass and options**

## 2) Geophysics:

```
lclass='geophysics', options='manuscript'
```

## 3) Geophysical Prospecting:

```
lclass='geophysics', options='manuscript,a4paper'
```

## 4) SEG abstract

```
lclass='segabs'
```

## 5) EAGE abstract

```
lclass='name'
```

**Need edit name.cls**

**See template "texmf/tex/latex/cwp/adam2009.cls"**

**Revised mode: options='manuscript,revised'**

# Rules for SConstruct [2]



➤ Always starts with: `from rsf.proj import *`, and ends with `End()`.

➤ Use `Fetch()` to fetch open datasets in Madagascar server:

```
Fetch('Txx.HH', 'septour')
```

➤ Use `Flow()` to handle processing flow:

```
Flow('windowed', 'Txx.HH',
      ...
      window n2=10 min1=0.4 max1=0.8
      ''')
```

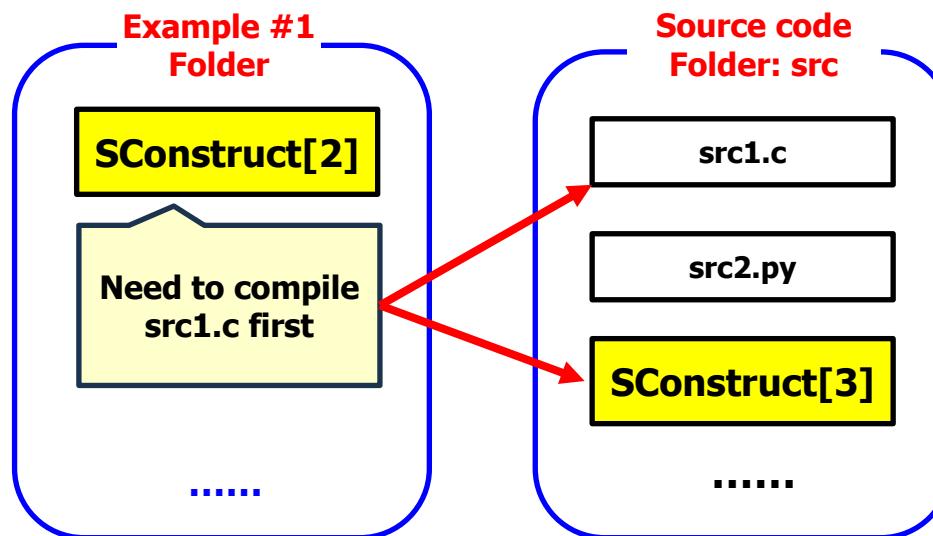
➤ Use `Result()` or `Plot()` to plot results:

```
Result('wiggle', 'windowed',
       ...
       transp=y poly=y yreverse=y pclip=100 nc=100 allpos=n
       '')
```

# Rules for SConstruct [2]



- [Optional] Use SConscript() for multi-layer dependency configuration.



```
SConscript('..../src/SConstruct')
```

# Example of SConstruct [2]



```
from rsf.proj import *
Fetch('Txx.HH','septour')
Result('wiggle0','Txx.HH','wiggle')
Flow('windowed','Txx.HH','window n2=10 min1=0.4 max1=0.8')
plotpar = '''
    transp=y poly=y yreverse=y pclip=100 nc=100 allpos=n'''
for plot in ('wiggle','contour','grey'):
    Result(plot,'windowed',plot + plotpar)
End()
```

Python tricks  
work here.

# Rules for SConstruct [3]



- Always starts with: `from rsf.proj import *`, and ends with `End()`.
- [Optional] Use `Program()` to compile source codes (C/C++):

```
Program(target='Mprogram.exe',  
        source=['Mprogram.c', 'other.c'])
```

- [Optional] Scripts written in interpreted languages (Python, GMT, Shell, ...) do not require compilation. However, we can use a shebang (`#!`) to simplify their commands. Put `#! /usr/bin/env python` at the very top of a python script file:

`python script.py`      ➔      `./script.exe`

```
Flow('script.exe', 'script.py', '''  
    echo "#! /usr/bin/env python" > ${TARGETS[0]} &&  
    cat ${SOURCES[0]} >> ${TARGETS[0]} &&  
    chmod +x ${TARGETS[0]}  
    ', rsfflow=0, stdin=0, stdout=-1)
```

# Example of Geophysics paper



## 1) Title page and abstract

```
\title{paper title}

\address{\footnotemark[1] address1 \footnotemark[2]
address2 \footnotemark[3] address3}

\author{Name1\footnotemark[1], Name2\footnotemark[2], and
Name3\footnotemark[3]}

\footer{GEO-2020-XXXX}

\lefthead{Surname1 et al.}
\righthead{short title}

\maketitle

\begin{abstract}
Write paper abstract here ...
\end{abstract}
```

# Example of Geophysics paper



## 2) Sections, citations and equations

`\section{section name, e.g., Introduction}` (Refname1, year)

The paper starts from old idea `\cite[]{}{refname1}.`

`\cite{refname2}` also improve it. ...

Refname2 (year)

`\section{section name, e.g., Theory}`

`\subsection{subsection name}`

The theory is shown as follows ...

`\begin{equation}` Write equation label here

`\label{eq: eq1}`

$$\mathbf{C} = \frac{\mathbf{A}}{b},$$

`\end{equation}`

Refer to the equation by its label

Refer to equation~`\ref{eq: eq1}.`

# Example of Geophysics paper



## 3) Figures and their references

```
\section{section name, e.g., Synthetic examples}
```

The paper has several synthetic tests.

```
\inputdir{example#1 folder name}
```

Declare figure folder first

```
\multiplot[2]{fig1,fig2}{width=0.5\textwidth}{figure1  
name (a) and figure2 name (b).}
```

One can also refer to different figures, e.g.,

```
Figure~\ref{fig:fig1,fig2}a ...
```

Figure 1a

```
\section{section name, e.g., Field data tests}
```

The method is also used to deal with field data  
(Figure~\ref{fig:fig3}).

```
\inputdir{example#2 folder name}
```

```
\plot{fig3}{width=0.75\textwidth}{figure3 name.}
```

# Example of Geophysics paper



## 3) Ending and bibliography

\section{Conclusion}

Write conclusion here...

\section{Acknowledgments}

Should thank anyone for useful help...

\appendix

\section{Appendix: name}

Write the content of appendix here...

\bibliographystyle{seg}

\bibliography{bibname}

- **Revised mode:** `options='manuscript, revised'`
- **Use `\new{New contents}` and `\old{Old contents}` to indicate “revised contents” and “~~replaced contents~~” corresponding to SConstruct[1] “options='revised' ”**

## ✓ Revised mode on

You can use `\new` and `\old` commands to mark the changes in the revised version.

For example, the revised text is *revised text* and the replaced text is ~~replaced text~~.

## ✗ Revised mode off

You can use `\new` and `\old` commands to mark the changes in the revised version.

For example, the revised text is revised text and the replaced text is .

# Rules for revision



- Equations, citations, etc. don't work inside `\new` and `\old`.  
However, you can enclose citations in `\mbox` and equations in  
`\parbox` or `minipage` environment:

```
\old{wrong citation \cite{wrong}}
\new{revised citation \cite{revised}}
```



```
\old{wrong citation \mbox{\cite{wrong}}}
\new{revised citation \mbox{\cite{revised}}}
```



# Rules for revision



➤ Equations, citations, etc. don't work inside `\new` and `\old`.

However, you can enclose citations in `\mbox` and equations in `\parbox` or `minipage` environment:

```
\old{wrong equation  
\begin{equation}  
2*2 = 5  
\end{equation}}
```



```
\new{revised equation }\\  
\begin{equation}  
2*2 = 4  
\end{equation}
```



```
\old{wrong equation }\\  
\begin{minipage}{\textwidth}  
\begin{equation}  
2*2 = 5  
\end{equation}\end{minipage}
```

```
\new{revised equation }\\  
\parbox{\textwidth}{  
\begin{equation}  
2*2 = 4  
\end{equation}}
```



# Example of bibliography file



```
@Article{Abma05,
  author  = {R[] Abma and N[] Kabir},
  title   = {Comparisons of interpolation methods},
  journal = {The Leading Edge},
  year    = 2005,
  volume  = 24,
  pages   = {984-989}
}

@Article{Naghizadeh09,
  author  = {M[ostafa] Naghizadeh and M[auricio] D[] Sacchi},
  title   = {f-x adaptive seismic-trace interpolation},
  journal = {Geophysics},
  year    = 2009,
  volume  = 74,
  pages   = {V9-V16}
}
```

# Example of bibliography file



```
@book{Claerbout92,
    author      = {J[on] F[] Claerbout},
    publisher   = {Blackwell Scientific Publications},
    title       = {{Earth Soundings Analysis: Processing Versus
Inversion}},
    year        = {1992}
}

@inproceedings{Curry04,
    author      = {W[illiam] Curry},
    title       = {Interpolation with multi-shifted-scale
prediction-error filters},
    booktitle   = {74th Annual International Meeting},
    year        = {2004},
    publisher   = {SEG, Expanded Abstracts},
    pages       = {2005-2008}
}
```

**Check “texmf/bibtex/bib/seg/SEG.bib” before you start to type the references.**

# Download citations



It is also very convenient to download bibtex citations, for example:

Google 学术搜索

Madagascar: open-source software project for multidimensional data analysis

文章

时间不限  
2025以来  
2024以来  
2021以来  
自定义范围...

按相关性排序  
按日期排序

不限语言  
中文网页  
简体中文网页

类型不限  
评论性文章

包括专利  
✓ 包含引用

1

引用

GB/T 7714 Fomel S, Sava P, Vlad I, et al. Madagascar: Open-source Software project for multidimensional data analysis and reproducible computational experiments[J]. Journal of Open Research Software, 2013, 1(1): e8-e8.

MLA Fomel, Sergey, et al. "Madagascar: Open-source software project for multidimensional data analysis and reproducible computational experiments." *Journal of Open Research Software*, 1(1), e8-e8.

APA Fomel, S., Sava, P., Vlad, I., Liu, Y., & Bashkardan, V. (2013). Madagascar: Open-source software project for multidimensional data analysis and reproducible computational experiments. *Journal of Open Research Software*, 1(1), e8-e8.

BibTeX EndNote RefMan RefWorks

2

SEG LIBRARY

GEOPHYSICS

PREVIOUS NEXT

No Access | GEOPHYSICS | Volume 84, Issue 3

A large-scale framework for symbolic implementations of seismic inversion algorithms in Julia

Check for updates

Authors: Philipp A. Witte, Matthias Louboutin, Navjot Kukreja, Fabio Luporini, Michael Lange, Gerard J. Cormier, and Felix J. Herrmann

https://doi.org/10.1190/geo2018-0174.1

Sections Full Text PDF/ePub Tools Share ADD TO FAVORITES DOWNLOAD CITATIONS

ABSTRACT

Download Citation

If you have the appropriate software installed, you can download the citation data to the citation manager of your choice in RIS format by simply clicking on download. RIS format can be imported by all major citation management software, including EndNote, ProCite, RefWorks and Reference Manager.

Format

RIS (ProCite, Reference Manager)  
EndNote  
BibTeX  
Medline  
RefWorks

Direct Import

Tips for downloading citations

DOWNLOAD PUBLICATION CITATION DATA

Download publication citation data for:

1  
2  
3  
4

# Tips for Madagascar plot modules



## 1) Label parameter:

- <http://ahay.org/blog/2007/10/27/how-do-i-change-fonts-in-vplot-graphics/>
- <http://ahay.org/blog/2008/03/26/a-journal-requires-tick-labels-on-my-plots-to-be-oriented-vertically-and-aligned-on-the-left-how-do-i-achieve-that/>
- <http://ahay.org/blog/2008/10/15/a-journal-requires-a-particular-font-e-g-arial-or-helvetica-in-figure-labels-how-do-i-achieve-that/>

## 2) Figure color:

- <http://ahay.org/blog/2005/03/28/color-schemes/>
- <http://ahay.org/blog/2013/03/19/color-plettes/>
- <http://ahay.org/blog/2013/08/23/cube-helix-color-palette/>
- <http://ahay.org/blog/2014/05/15/light-bartlein-color-palette/>
- <http://ahay.org/blog/2014/10/18/tutorial-on-colormaps/>
- <http://ahay.org/blog/2015/07/12/more-colormaps/>

# Tips for Madagascar plot modules



## 3) Control figure color and resolution (SConstruct [1]):

```
Paper('name',lclass='geophysics', options='manuscript',
      use='listings',hires='fig1 fig2 fig3',color='fig4 fig5 fig6')
```

## 4) Insert only grey figures in LaTeX (SConstruct [1]):

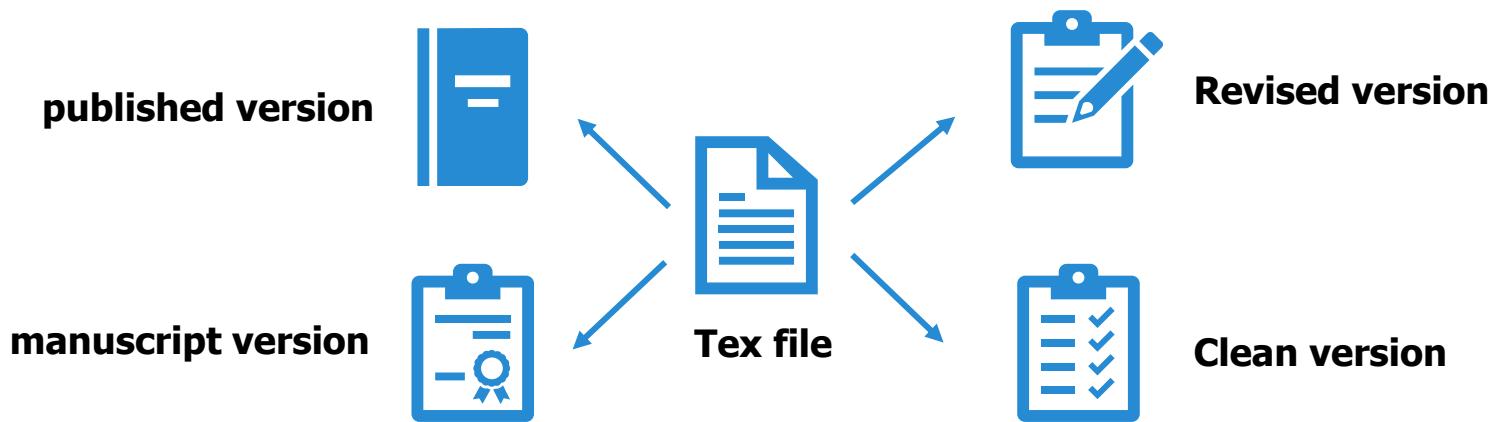
```
from rsf.tex import *
import os
os.environ['PSTEXPENOPTS'] = 'color=n'
```

```
End(use='amsmath,hyperref',options='manuscript')
```

# Tips for SCons scripts



- You can write multiple `Paper()` with different options in `SConstruct[1]` to build different papers.



- Use `pscons` or `scons -j #` for parallel building.

# Tips for SCons scripts



- Most parameters in the SCons file are specified as **strings**.

Therefore, Python's **string manipulation capabilities** are **extremely useful** when writing SCons scripts.

```
plotpar = ''  
        transp=y poly=y yreverse=y pclip=100 nc=100 allpos=n'''  
for plot in ('wiggle','contour','grey'):  
    Result(plot,'windowed',plot + plotpar)
```

```
f'{plot} {plotpar}'  
'%s %s'%(plot, plotpar)
```

# Outline

---



- Motivations
- Pre-writing configuration
  - Prerequisites
  - File organization
  - Figure preparation
- SEGTeX
  - Implement steps
  - Rules and tips
- Alternative L<sup>A</sup>T<sub>E</sub>X template
- Simple Example

# How to use alternative templates

---



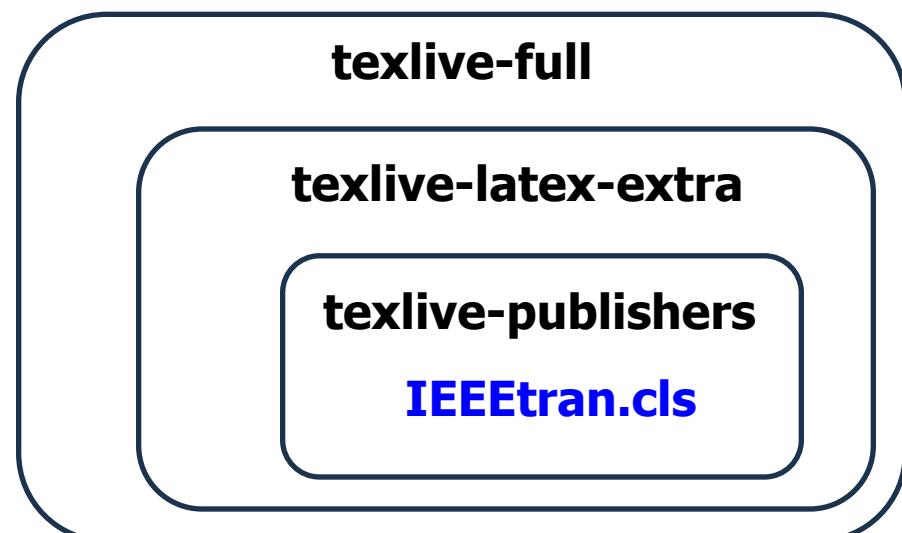
- 1) Download a LaTeX template file and corresponding class file.**
- 2) Open the template file. According to the template .tex file, configure `Paper()` in `SConstruct` [1] following its **rules**.**
- 3) Remove all preamble contents (before `\begin{document}`) including `\begin{document}` and `\end{document}`.**
- 4) `scons paper.read`.**

## 1 Download template file and class file

Asking AI helps

- a. The class file **IEEEtran.cls** is included in TeXLive packages:  
**texlive-publishers**, **texlive-latex-extra** and **texlive-full**.

**sudo apt update**  
**sudo apt install texlive-full**



- b. Or manually download it from <https://template-selector.ieee.org/> and put it under your project directory.

# IEEE Transactions L<sup>A</sup>T<sub>E</sub>X Template



```
Paper('paper','school2025_tgrs',
      lclass='IEEEtran',
      options='journal, twoside',
      use='''amsmath,amsfonts algorithmic array
[caption=false,font=normalsize,labelfont=sf,textfont=sf]subfig
textcomp stfloats url verbatim graphicx balance
'''.split(),
      include=r'''\\hyphenation{op-tical net-works semi-conduc-tor IEEE-Xplore}
\\def\\BibTeX{{\\rm B}\\kern-.05em{\\rm{ }}\\rm i\\kern-.025em b}\\kern-.08em
T\\kern-.1667em}lower.7ex\\hbox{E}\\kern-.125emX}'''')
```

The screenshot shows a LaTeX editor window with the following elements:

- Top Bar:** Shows the file name "New\_IEEEtran.How-to.tex" and a search bar.
- Code Area:** Displays the LaTeX code for the template. A red box highlights the preamble area (lines 1-15), and a red circle with the number "2" points to the word "Configure".
- Output Area:** Shows a PDF preview titled "school2025\_tgrs.pdf" with the text "JOURNAL OF L<sup>A</sup>T<sub>E</sub>X CLASS FILES, VOL. 18, NO. 9, SEPTEMBER 2020".
- Bottom Area:** Contains sections like "How to Use the IEEEtran L<sup>A</sup>T<sub>E</sub>X Templates" and "III. L<sup>A</sup>T<sub>E</sub>X DISTRIBUTIONS: WHERE TO GET THEM".

**Annotations:**

- 2 Configure Paper()**: Points to the title of the document being generated.
- 3 Remove preamble**: Points to the preamble section of the LaTeX code.
- 4 scons paper.read**: Points to the command used to build the document.

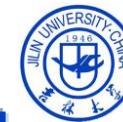
# Outline

---



- Motivations
- Pre-writing configuration
  - Prerequisites
  - File organization
  - Figure preparation
- SEGTeX
  - Implement steps
  - Rules and tips
- Alternative L<sup>A</sup>T<sub>E</sub>X template
- Simple Example

# Simple Example



## Never start from scratch!

### STEP 1:

```
$git clone https://github.com/cplusv2023/school2025.git ~/school2025  
(alternative: https://gitee.com/chenjw2318/school2025.git)
```

### STEP 2:

```
$cd ~/school2025  
$sftour pscons lock
```

### STEP 3:

```
$scons school2025.read
```

### STEP 4:

- ✓ Change examples in SConstruct[2] and LaTeX files
- ✓ Change parameters in SConstruct[1] to get different template
- ✓ Loop over STEP 2-4



Asking AI helps!

# **THANKS FOR ATTENTION AND HAVE A FUN!**

**Stand on each other's shoulders  
(not each other's toes)!**

**Modify/add your own data tests,  
write your own papers ...**

