

# Installation Guide for

---



**L\_SU,  
a graphical user interface  
for Seismic Unix (John Stockwell)**

---

**Louisiana State University  
Dept. Geology and Geophysics  
Baton Rouge  
Juan M. Lorenzo  
([gllore@lsu.edu](mailto:gllore@lsu.edu))**

- **Version: 0.4.5 September, 2020**

1.1	Tested operating systems.....	1-4
1.2	Definition of system variables in your computer work environment (all OS's) .....	1-4
1.2.1	Linux .....	1-5
1.2.2	Cygwin/Microsoft Windows .....	1-7
1.3	Notes on installing Seismic Unix.....	1-8
1.3.1	Linux .....	<b>Error! Bookmark not defined.</b>
1.3.2	Cygwin/Microsoft Windows .....	2-16
1.4	Software required by Seismic Unix and L_SU.....	1-8
1.4.1	CentOS .....	1-8
1.4.2	Ubuntu .....	1-9
1.4.3	Debian .....	1-9
1.4.4	Cygwin/Microsoft Windows .....	1-9
1.4.4.1	Libraries.....	1-9
1.4.4.2	General packages .....	1-10
1.4.4.2.1	To aid in compilation of imported codes.....	1-10
1.4.4.2.2	For running a windowed environment .....	1-10
1.4.4.2.3	For general text editing.....	1-10
1.4.4.2.4	Net utilities.....	1-10
1.4.4.2.5	For viewing postscript files .....	1-10
1.5	Perl modules required by L_SU .....	1-11
1.5.1	Linux .....	1-11
1.5.2	Cygwin/Microsoft Windows .....	1-11
1.5.3	Loading cpan .....	1-11
1.5.3.1	CentOS8.....	1-11
1.5.3.2	CentOS7.....	1-11
1.5.3.3	Ubuntu .....	1-11
1.5.3.4	Debian .....	1-11
1.5.3.5	Cygwin/Microsoft Windows.....	1-12
1.5.4	How to load required Perl Modules from the CPAN.....	1-12
1.5.4.1	CentOS 7/8.....	1-13
1.5.4.2	Debian .....	1-13
1.5.4.3	Ubuntu .....	1-13
1.5.4.4	Cygwin/Microsoft Windows.....	1-14
1.6	Required Fortran programs in L_SU (all OS's) .....	2-19
1.6.1	PGPLOT .....	<b>Error! Bookmark not defined.</b>

1.6.2	MMODPG.....	<b>Error! Bookmark not defined.</b>
1.7	L_SU and its modules from github (all OS's) .....	2-19
1.7.1	Linux .....	2-19
1.7.2	Cygwin/Microsoft Windows .....	2-19
1.7.3	Fortran modules in L_SU .....	2-20
1.7.3.1	Linux .....	<b>Error! Bookmark not defined.</b>
1.7.4	Installation of example data sets and flows.....	2-15
1.7.4.1	Linux .....	<b>Error! Bookmark not defined.</b>
1.7.4.2	Cygwin/Microsoft Windows.....	2-18
2.1	Linux.....	2-15
2.2	Cygwin/Microsoft Windows .....	2-15
3	SioSEIS .....	3-23
3.1.1.1	Download SIOSEIS .....	<b>Error! Bookmark not defined.</b>
3.1.1.2	Modify system environmental variables.....	3-23

---

# 1 L\_SU Installation

---

## 1.1 Tested operating systems

All the installation steps have been tested on new, blank systems and have worked without any errors and are the recommended simplest paths to installing the software.

Particular users have their own specialized software installations and we would like to hear from you if you have any difficulties with the installation (gllore@lsu.edu Subject: L\_SU)

We have used the following instructions to install L\_SU under several different operating systems -- sometimes with a little apprehension-- but without any ensuing difficulties.

Linux operating system	Version tested
CentOS	7.7.2003. *.2.2004
Debian	10 (buster)
Ubuntu	18.04.3
RedHat	6.9
Cygwin/Microsoft Windows	10 with cygwin-3.1.7

Across all OS's, most installation problems occur when either (1) the environment variables are not properly set and (2) when the necessary Perl CPAN modules do not loaded correctly. The following installation guide may seem extensive because we try to include many of the available operating systems.

One issue you may note is that 'end-of-line termination indicators' differ between Linux systems and Windows. These codes are invisible to the user when editing, but cause errors. If you see errors like that contain confusing messages which include "\r" then you have encountered the issue. A solution is to open the text file in an editor. **Notepad++**, or **vim**, are two examples of text editors that can change the codes for you. In Notepad++ there is a menu option to perform the change using a mouse click or two. In **vim** manually set the file format to "unix: with the following command:

```
:set ff=unix
```

## 1.2 Definition of system variables in your computer work environment

In order for Perl to find all the programs that it needs at run time, it will look in pre-defined areas of your hard drive. These pre-defined directories, whether known to the user or not, exist in all or most operating systems. Your system manager usually adds special file locations as needed. Local users can even override the special file locations although that is not a safe practice.

Note for example, that **CWPROOT** is a directory path where the C programs that belong to Seismic Unix are usually installed. In the next Linux example, this path = **“/usr/local/pl/cwp\_su\_all\_44R4”**

If you do not have permission to change your local **“.bashrc”** file then ask your systems manager to make some arrangement that will allow your local **“.bashrc”** file to pointing to a system-wide file that only the administrator control, in which case you can add the following line to your local **“.bashrc”** file:

```
source /PATH/bashrc_system
```

But, you will need to know what ‘PATH’ is and what ‘bashrc\_system’ means. If this sounds confusing, see your administrator or write to me at [gllore@lsu.edu](mailto:gllore@lsu.edu).

### 1.2.1 Linux

For my linux example, I usually place Perl programs under **“/usr/local/pl”**. Then, in order for all the Perl scripts and other programs to run, I have to add several new lines of instructions within my file **“.bashrc”** file, which is located in my home directory:

```
# for L_SU
# for general perl directories
export LOCAL=/usr/local/pl
export PL=$LOCAL/pl

# Also for L_SU
export L_SU=$PL/L_SU
export PERLL5LIB=PERLL5LIB:$L_SU/configs
export PERLL5LIB=$PERLL5LIB:$L_SU/specs
export PERLL5LIB=$PERLL5LIB:$L_SU/sunix

for category in big_streams data datum plot filter header \
    inversion migration model \
    NMO_Vel_Stk par picks shapeNcut shell statsMath transform \
    well
do
    export PERLL5LIB=$PERLL5LIB:$L_SU/configs/$category
    export PERLL5LIB=$PERLL5LIB:$L_SU/specs/$category
done

for category in data datum plot filter header \
    inversion migration model \
    NMO_Vel_Stk par picks shapeNcut shell statsMath transform \
```

```
well
do
  export PERL5LIB=$PERL5LIB:$L_SU/sunix/$category
done

export PERL5LIB=$PERL5LIB:$L_SU/gmt:$L_SU/R:$L_SU/big_streams:$L_SU/messages
export PERL5LIB=$PERL5LIB:$L_SU/misc
export PERL5LIB=$PERL5LIB:$L_SU/reqs:$L_SU/specs:$L_SU/sqlite:$L_SU/streams:$L_SU/geo-
psy:$L_SU/images

# NOTICE: Because there are executable L_SU-related Perl scripts, PATH must be already defined
# within your .bashrc file
# and located somewhere above the current lines, i.e. above the first "for L_SU"
export PATH=$PATH:$L_SU
export PATH=$PATH:$L_SU/big_streams

# for executable L_SU-related compiled and linked programs
export PATH=$PATH:$L_SU/fortran/bin
export PATH=$PATH:$L_SU/c/bin

# for Seismic Unix
export CWPROOT=$LOCAL/cwp_su_all_44R4

# for pgplot
export PGPLOT_DIR=$LOCAL/pgplot
# default display device for pgplot
export PGPLOT_DEV=/XWINDOW
```

### 1.2.2 Cygwin/Microsoft Windows

For the Windows example, I usually place Perl programs under “**/usr/local/pl**”. Then, in order for all the Perl scripts and other programs to run, I have to add several new lines of instructions within my file “**.bashrc**” file, which is located in my home directory:

```
# for L_SU
# for general perl directories

# cygwin directory
cygwin=/cygdrive/c/cygwin64

export LOCAL=$cygwin/usr/local
export PL=$LOCAL/pl

# Also for L_SU
export L_SU=$PL/L_SU
export PERL5LIB=PERL5LIB:$L_SU/configs
export PERL5LIB=$PERL5LIB:$L_SU/specs
export PERL5LIB=$PERL5LIB:$L_SU/sunix

for category in big_streams data datum plot filter header \
    inversion migration model \
    NMO_Vel_Stk par picks shapeNcut shell statsMath transform \
    well
do
    export PERL5LIB=$PERL5LIB:$L_SU/configs/$category
    export PERL5LIB=$PERL5LIB:$L_SU/specs/$category
done

for category in data datum plot filter header \
    inversion migration model \
    NMO_Vel_Stk par picks shapeNcut shell statsMath transform \
    well
do
    export PERL5LIB=$PERL5LIB:$L_SU/sunix/$category
done

export PERL5LIB=$PERL5LIB:$L_SU/gmt:$L_SU/R:$L_SU/big_streams:$L_SU/messages
export PERL5LIB=$PERL5LIB:$L_SU/misc
```

```
export PERL5LIB=$PERL5LIB:$L_SU/reqs:$L_SU/specs:$L_SU/sqlite:$L_SU/streams:$L_SU/geo-  
psy:$L_SU/images  
  
# NOTICE: Because there are executable L_SU-related Perl scripts, PATH must be already defined  
# within your .bashrc file  
# and located somewhere above the current lines, i.e. above the first "for L_SU"  
export PATH=$PATH:$L_SU  
export PATH=$PATH:$L_SU/big_streams  
  
# for executable L_SU-related compiled and linked programs  
export PATH=$PATH:$L_SU/fortran/bin  
export PATH=$PATH:$L_SU/c/bin  
  
# for Seismic Unix  
export CWPROOT=$LOCAL/cwp_su_all_44R4  
  
# for pgplot  
export PGPLOT_DIR=$LOCAL/pgplot  
  
# default display device for pgplot  
export PGPLOT_DEV=/XWINDOW  
  
# for pgplot binary  
export PATH=$PATH:$LOCAL/pgplot  
  
##### FOR SHARED LIBRARIES ##  
# with pgplot libraries  
export LD_LIBRARY_PATH=/usr/lib64  
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$PGPLOT_DIR
```

## 1.3 System software required by Seismic Unix and L\_SU

In order to install **Seismic Unix** properly you will need to make sure the following libraries and standalone programs are installed.

All OS's will need:

- **evince**

### 1.3.1 CentOS

#### 1.3.1.1 CentOS7x

If you are working under CentOS7x, and you need help with library requirements,



please e-mail [glore@lsu.edu](mailto:glore@lsu.edu) for help.

Install the following programs and libraries:

```
% sudo yum install evince
```

#### 1.3.1.1 CentOS8x

```
% sudo dnf install evince
% sudo dnf install libX11-devel
% sudo dnf install libpng-devel
% sudo dnf install libgfortran
```

### 1.3.2 Ubuntu

Some users experience problems when installing Tk modules. Often this occurs because of missing libraries. For example, some missing libraries such as the following can be installed manually with the following command:

```
% sudo apt-get install dpkg-dev
% sudo apt-get install libx11-dev libfreetype6-dev libxft-dev
% sudo apt-get install aptitude libpng-dev libz-dev libjpeg-dev
% sudo apt-get install evince
```

### 1.3.3 Debian

If you are working under **Debian**, pre-install the following libraries:

```
% sudo apt-get install dpkg-dev
% sudo apt-get install libx11-dev
% sudo apt-get install libpng-dev libjpeg-dev
% sudo apt-get install evince
```

### 1.3.4 Cygwin/Microsoft Windows

We have successfully tried **cygwin**, a linux environment emulator for Windows. When we install **cygwin** we take a minimalist approach by, installing only the smallest system, and adding essential elements later, manually.

#### 1.3.4.1 Libraries

Use **setup-x86\_64** to install libraries:

- Perl-Log-Log4Perl
- libnetpbm-devel
- libnetpbm10
- libgcc1
- libgd-devel

- *libtirpc-devel*
- *libQtOpenGL4*
- *libQtOpenGL4-devel*
- *libXaw-devel*
- *libXaw7*
- *libX11-devel*
- *libcrypt-devel*
- *libglut*
- *libglut-devel*
- *libXi6 and libXi-devel*
- *libGL1*
- *libGL-devel*
- *libGLU-devel*
- *libGLU1*
- *libXm-devel*

#### 1.3.4.2 General packages

Use **setup-x86\_64** to install the following:

##### 1.3.4.2.1 To aid in compilation of imported codes

- *make*
- *gcc-core*
- *gcc-g++*
- *gcc-fortran*
- *rpcbind*
- *curl*

##### 1.3.4.2.2 For running a windowed environment

- *xinit*
- *xorg-server*
- *gnome-flashback*
- *xlaunch*

##### 1.3.4.2.3 For general text editing

- *vim*

##### 1.3.4.2.4 Net utilities

- *ncftp*

##### 1.3.4.2.5 For viewing postscript files

- *evince* (*may have problems—do check*)

## 1.4 Perl modules required by L\_SU

Before any further installation steps you first must have installed the Perl language on your linux box. Most linux-type systems come automatically with the Perl language. You can check to see if you have Perl installed and its version at the same time, by entering the following command:

```
% perl -v
```

### 1.4.1 Linux

If you install Perl as a regular user the process will create “**perl5**”, as a sub-directory in your home directory. You will have to accept these modifications. You do not need to do anything. But, as a result, you will find several lines of code installed also automatically inside your local “**.bashrc**” file.

I recommend that all software be installed with superuser permissions (**sudo**).

### 1.4.2 Cygwin/Microsoft Windows

Use **setup-x86\_64** to install **perl** using administrator privileges.

### 1.4.3 Loading cpan

For all types of operating systems, we recommend that the easiest way to install public Perl modules is to use **cpan**. First of all, **cpan**, a utility that is written in Perl. You will have to have root privileges, at least temporarily when you install **cpan** and the following modules. Later you will be able to use them in L\_SU as a regular user. (**cpan** can be downloaded from <https://www.cpan.org/>)

#### 1.4.3.1 CentOS8

If you are working under **CentOS8x**, install **cpan** as follows:

```
% sudo dnf install perl-CPAN
```

#### 1.4.3.2 CentOS7

If you are working under CentOS7x, install **cpan** as follows:

```
% sudo yum install cpan
```

#### 1.4.3.3 Ubuntu

If you are working under **Ubuntu**, install **cpan** as follows:

```
% sudo apt-get upgrade
```

```
% sudo cpan -v
```

(Hint: choose to configure **cpan** automatically)

#### 1.4.3.4 Debian

If you are working under **Debian**, it is easier to install **cpan** as follows:

```
% sudo cpan -v
```

To help during **cpan** installations also install Log::Log4perl:

```
% sudo cpan Log::Log4perl
```

#### 1.4.3.5 Cygwin/Microsoft Windows

If you are using **cygwin**, then **cpan** comes already installed.

However, when you first use **cpan** answer:

“yes” to automatic configuration,

and “local::lib” as your preferred installation privileges. Both these are the default answers.

### 1.4.4 How to load required Perl modules from the CPAN

For the general case of any operating system, you will use **cpan** to install some or all of the following required Perl modules:

- ***MIME::Base64***
- ***Shell***
- ***Tk***
- ***Tk::JFileDialog***
- ***Clone***
- ***Tk::Pod***
- ***Moose***

The following packages below have been tested under the following operating systems

Perl module name in cpan format	Ubuntu (18.x)	Debian 9.9	Versions tested under CentOS7x	cygwin
MIME::Base64	3.15			3.15
Perl	5.26.1	5.24.1	5.16.3	5.26.3
Shell	0.73.1	0.73	0.73	0.73
Tk::JFileDialog	2.20?	1.62	1.62	2.20
Tk or PerlTk	804.034	804.033	804.034	804.034
Tk::Pod	5.41	5.41	5.41	5.41
Moose	2.18	2.187	2.2010	2.2012
Clone	0.41	0.38	0.39	0.43

For each of the above packages use the following commands to install each of them.

**1.4.4.1 CentOS 7/8**

```
% sudo cpan MIME::Base64
% sudo cpan Shell
% sudo cpan Tk
% sudo cpan Tk::JFileDialog
% sudo cpan Clone
% sudo cpan Tk::Pod
% sudo cpan Moose
% sudo cpan PDL::IO:FlexRaw
```

**1.4.4.2 Debian**

```
% sudo cpan Tk (e.g., 804.034)
% sudo cpan MIME::Base64 (e.g., 3.15)
% sudo cpan Config::Simple (e.g., 4.58 installed)
% sudo cpan Shell (e.g., 0.73 installed)
% sudo cpan Clone (e.g., 0.43 installed)
% sudo cpan Tk::JFileDialog (e.g., 2.20 installed)
% sudo cpan Tk::Pod (.9943 installed)
% sudo cpan Moose (installed 2.2012)
```

To install evince, a viewer for postscript files, although it may already be present:

```
% sudo apt-get install evince
```

**1.4.4.3 Ubuntu**

To help during **cpan** installations:

```
% sudo cpan Log::Log4perl
```

Then, continue to install the following:

```
% sudo cpan Tk (e.g., V804.034 installed)
% sudo cpan MIME::Base64 (e.g., V3.15 installed)
% sudo cpan Config::Simple (e.g., V4.58 installed )
% sudo cpan Shell (e.g., V0.73 installed)
% sudo cpan Clone (e.g., V0.43 installed)
% sudo cpan Tk::JFileDialog (e.g., V2.20 installed)
% sudo cpan Tk::Pod (e.g., V2.9943 installed)
% sudo cpan Moose (e.g., V2.2012 installed)
```

To install **evince**, which is a viewer for postscript files and often already present in your system:

```
% sudo apt-get install evince
```

To help when building Perl modules, install the following:

```
% sudo cpan Module::Build          (e.g., V0.4229 installed)
% sudo cpan TAP::Harness           (e.g., V3.42 installed)
```

**cpan** will find dependencies for the above packages and install them as well, so you may see a lot of additional packages installed during the process.

#### 1.4.4.4 Cygwin/Microsoft Windows

First, load as many of the needed Perl modules in the simplest fashion. Use **setup-x86\_64** to install the needed libraries:

##### **perl-Clone**

For the following two make sure you have an XLaunch running and an X-window in the background, or the tests cannot be run

##### **perl-Tk**

**perl-Tk\*** (\* is a wildcard for everything that starts with perl-Tk, including perl-Tk-Pod, e.g., and equivalent to Tk::JFileDialog )

Then, use **cpan** from the command line in the **cygwin** window and install the following modules in the order shown. I am recommending that you **force (-fi)** the installation despite the failure of a few tests:

```
% cpan Shell
% cpan YAML
% cpan -fi Module::Build
% cpan -fi Module::Runtime
% cpan -fi Dist::CheckConflict
% cpan -fi Moose
% cpan -fi PDL
```

Moose and PDL are bigger-than-normal packages that can take tens of minutes (or less) to install, depending on the power of your PC.

---

## 2 -Seismic Unix installation from github

---

At present, we recommend that you download Seismic Unix and install the program as per the git hub site set up by John Stockwell at: "<https://github.com/JohnWStockwellJr/SeisUnix>"

If you are familiar with the program **git** (must be installed on your OS), the following is an example of my installation procedure, contained within a shell-script file. This file can be called what you please, e.g., "**clone\_SU.sh**"

There are small differences between the script for linux and for Windows, that depend only on file locations.

Seismic Unix is available at: <https://github.com/JohnWStockwellJr/SeisUnix> and can be installed anywhere on your machine, as long as you prescribe their location to the operating system (See 1.2).

In order to download these files from the github site you can run the following shell script. You must have administrator privileges. The following example script will place the Seismic Unix package within the “/usr/local/ directory” that has the path: “/usr/local/cwp\_su\_all\_44R4”. The script below is saved as a file with any name you choose, e.g., “SU\_clone.sh”. You can create this file inside your home directory.

After files have been cloned you will have to read through the installation instructions and prepare Seismic Unix to compile, link to libraries and install on your local machine. Instructions are found in the directory: “~cwp\_su\_all\_44R4/src”

## 2.1 Linux

### 2.1.1.1 Clone of Seismic Unix

Run the following instructions individually and in sequence from the command line, or place the following instructions into a script, e.g., “clone\_SU.sh” as follows:

```
#!/bin/bash
# my name is clone_SU.sh

# give a name to directory
installation_directory_for_SU= /usr/local/cwp_su_all_44R4

# create installation directory
mkdir $installation_directory_for_SU

# change into the installation directory
cd $installation_directory_for_SU

# clone the directory from the remote site on to your computer
git clone https://github.com/gllore/L_SU.git

# move files from one directory into the current directory
mv SeisUnix/* ./
mv SeisUnix/. * ./

# git status
```

```
git status
```

Execute the script contained in “**clone\_SU.sh**”, while using administrative privileges as **sudo**, e.g.:

```
% sudo sh clone.sh
```

#### 2.1.1.1 Preparation of Makefile.config

Please follow the instructions in all the “**README**” files but first make the following changes to the

“/usr/local/cwp\_su\_all44\_R4/src/**Makefile.config**”:

##### 2.1.1.1.1 CentOS 8

```
IX11 = /usr/include/X11
LX11 = /usr/lib
IMOTIF = /usr/include/Xm
LMOTIF = /usr/lib
```

#### 2.1.1.1 Compilation and Installation

Stay in the current directory:

“/usr/local/cwp\_su\_all44\_R4/src” and run the following commands one at a time:

```
sudo make install
sudo make xtinstall
```

#### 2.1.1.2 Installation of Demos

Once the installation of project: **L\_SU**, is complete on your system, you can copy some of the accompanying demonstration data sets to your home directory, where, for example, “/home/user” is the complete path to the location of a user’s (“**user**”) home directory. Put these instructions in another convenient shell script called, e.g., “**copy\_demos.sh**”

```
#!/bin/bash
# my name is copy_demos.sh

# give a name to directory
installation_directory_for_L_SU=/usr/local/pl

cp -R $installation_directory_for_L_SU/L_SU/Servilleta_demos /home/user/
cp -R $installation_directory_for_L_SU/L_SU/LSBB /home/user/
```



The L\_SU tutorial manual makes use of these two demonstration projects. “**Servilleta\_demos**” contains files from the 2018 IRIS internship orientation program and LSBB contains files from Pau University in France, courtesy of Dominique Rousset and Guy Sénéchal, both extensive contributors to the improvement of Seismic Unix.

## 2.2 Cygwin/Microsoft Windows

### 2.2.1 Clone of Seismic Unix

Run the following instructions individually and in sequence from the command line, or place the following instructions into a script, e.g., “**clone\_SU.sh**” as follows:

```
#!/bin/bash

# my name is clone_SU.sh
# cygwyn directory
cygwyn=/cygdrive/c/cygwin64

# give a name to directory
installation_directory_for_SU=$cygwyn/usr/local/cwp_su_all_44R4

# create installation directory
mkdir $installation_directory_for_SU

# move into the installation directory
cd $installation_directory_for_SU

# change into the installation directory
cd $installation_directory_for_SU

# clone
git clone https://github.com/JohnWStockwellJr/SeisUnix

# move files from one directory into the current directory
mv SeisUnix/* ./
mv SeisUnix/. * ./

git status
```

In order to execute the script contained in “**clone\_SU.sh**”:

```
% sh clone.sh
```

### 2.2.2 Preparation of Makefile.config

Please follow the instructions in all the “README” files but first make the following changes to the

“/usr/local/cwp\_su\_all44\_R4/src/Makefile.config”:

```
IX11 = /usr/include/X11
LX11 = /usr/lib
IMOTIF = /usr/include/Xm
LMOTIF = /usr/lib
```

### 2.2.3 Compilation and installation

Stay in the current directory:

“/usr/local/cwp\_su\_all44\_R4/src” and run the following commands one at a time:

```
make install
make xtinstall
```

## 2.3 Installation of example data sets and flows

### 2.3.1.1 Cygwin/Microsoft Windows

Once the installation of project: **L\_SU**, is complete on your system, you can copy some of the accompanying demonstration data sets to your home directory, where, for example, “/cygdrive/c/cygwin64/home/user” is the complete path to the location of a user’s (“user”) home directory

```
#!/bin/bash
# my name is copy_demos.sh

# cygwin directory
cygwin=/cygdrive/c/cygwin64

# L_SU directory
installation_directory_for_L_SU=$cygwin/usr/local/pl

# path to user's directory
user_path=$cygwin/home/user

cp -R $installation_directory_for_L_SU/L_SU/Servilleta_demos $user_path
cp -R $installation_directory_for_L_SU/L_SU/LSBB $user_path
```

Because L\_SU is visual in nature, you will need to have **X** running in the background. Use all the defaults and start by selecting from the Windows Start Menu: **Cygwin-X->XLaunch** to establish your preferences and run **X**.

## 2.4 Clone of L\_SU from github and Installation

Most of the L\_SU scripts, are written in Perl, available at: [www.github.com/gllore](http://www.github.com/gllore) and can be installed anywhere on your machine, as long as you prescribe their location to the operating system (See 1.3.4).

In order to download these files from the github site you can run the following shell script. You must have administrator privileges. The following example script will place the L\_SU package within the “pl directory” that has the path: “/usr/local/pl”. The script below is saved as a file with any name you choose, e.g., “**clone.sh**”. You can create this file inside your home directory. Then to activate the instructions on the command line enter the following:

```
% bash clone.sh
```

### 2.4.1 Linux

The contents of the file “clone.sh” are as follows:

```
#!/bin/bash
# my name is clone.sh

# give a name to directory
installation_directory_for_L_SU= /usr/local/pl/

# create installation directory
mkdir $installation_directory_for_L_SU

# change into the installation directory
cd $installation_directory_for_L_SU

# clone the directory from the remote site on to your computer
git clone https://github.com/gllore/L_SU.git

# git status
git status
```

### 2.4.2 Cygwin/Microsoft Windows

The contents of the file “clone.sh” are as follows:

```
#!/bin/bash
# my name is clone.sh

# give a name to directory
installation_directory_for_L_SU=/cygdrive/c/cygwin64/usr/local/pl

# create installation directory
mkdir $installation_directory_for_L_SU

# change into the installation directory
cd $installation_directory_for_L_SU

# clone the directory from the remote site to your computer
git clone https://github.com/gllore/L_SU.git

# git status
git status
```

In order to run L\_SU you will need an Xserver. Do so by:

- Under the menu select: selecting Cygwin-X ->XLaunch or ->XWin Server
- export DISPLAY=:0.0

### 2.4.3 Fortran modules in L\_SU

There are Tools in L\_SU that communicate with programs written in Fortran (e.g., immodpg). These L\_SU modules (e.g., for raytracing reflections and refracted arrivals) are written in Fortran and some as well, use fortran plotting libraries for interactive forward modeling.

You will first need to install the graphic libraries (PGPLOT)

#### 2.4.3.1 pgplot

The principal plotting library is “**pgplot**” which has to be downloaded and compiled separately.

Extensive instructions on the use and installation of pgplot are available from <https://www.astro.caltech.edu/~tjp/pgplot/>.

In brief:

- Download the library, as follows, from your home directory to your home directory:

```
% ncftpget ftp://astro.caltech.edu/pub/pgplot/pgplot5.2.tar.gz
```

- Decompress the downloaded file: pgplot5.2.tar.gz and reconstitute the directories and files it contains within a general directory: “**/usr/local/src/pgplot**”

```
% mkdir /usr/local/src
```

```
% tar -xvzf pgplot5.2.tar.gz
% mv pgplot /usr/local/src/
```

- Prepare and compile code (A good explanation is available at [https://www.gnu.org/software/gnuastro/manual/html\\_node/PGPLOT.html#FOOT220](https://www.gnu.org/software/gnuastro/manual/html_node/PGPLOT.html#FOOT220) )

```
% cd /usr/local/src/pgplot
$ gedit drivers.list
```

Remove the “!” for the following lines, save and close the file:

```
XWDRIV 1 /XWINDOW
XWDRIV 2 /XSERVE
```

Specify the type of fortran compiler to use by opening the following file:  
/usr/local/src/pgplot/sys\_linux/g77\_gcc.conf file:

```
% gedit /usr/local/src/pgplot/sys_linux/g77_gcc.conf
```

Change the following lines from:

```
FCOMPL="g77"
```

to

```
FCOMPL="gfortran",
```

and from :

```
XINCL="-I/usr/X11R6/include"
```

to

```
XINCL="-I/usr/include/X11"
```

and from:

```
LIBS="-L/usr/X11R6/lib -lX11"
```

to:

```
LIBS="-L/usr/lib -lX11"
```

and save it. This is a very important step during the compilation of the code if you are in GNU/Linux.

Create a folder within “/usr/local” and copy the modified “**drivers.list**” to the new folder.

```
% mkdir /usr/local/pgplot
```

(Remember that your unpacked address is: **/usr/local/src/pgplot**)

```
% cd /usr/local/pgplot
% cp /usr/local/src/pgplot/drivers.list ./
```

Type the following command:

```
% /usr/local/src/pgplot/makemake /usr/local/src/pgplot linux g77_gcc  
to make the “Makefile”, but make sure you are still in the directory (“/usr/local/pgplot”).
```

If all goes well, you should see: “**Determining object file dependencies**”.  
Finish up by running the following these three commands in order:

```
% make libpgplot.a  
% make grfont.dat  
% make pgxwin_server  
% make prog  
% make clean
```

Modify your .bashrc file to include necessary variable definitions (Section 1.2.1 for Linux and 1.2.2 for Windows)

#### 2.4.3.2 mmodpg

For interactive, forward modeling of the arrival times of refraction and reflections we use **mmodpg** (Vera, 1994). Go the directory where you installed L\_SU and inside the directory called “**fortran**” run an installation script called “**run\_me\_only.sh**”. The following examples should work across several operating systems:

You can compile the fortran programs in the fortran directory:

```
% cd /usr/local/pl/L_SU/fortran
```

The instruction for compiling and linking the programs is:

```
% sh run_me_only.sh
```

### 2.4.4 C-based modules in L\_SU

#### 2.4.4.1 Synseis

For simple interactive, normal-incidence synthetic seismogram modeling we use Synseis (under “**Tools**” tab). If you want to be able to use this tool proceed as follows:

- Go to the directory where you installed L\_SU and once inside move into the directory called “c/synseis” run an installation script called “run\_me\_only.sh”. The following examples should work across several operating systems:

You can compile synseis in the c directory:

```
% cd /usr/local/pl/L_SU/c/syneis
```

The instruction for compiling and linking the programs is:

```
% sh run_me_only.sh
```

---

## 3 SioSEIS installation

---

From the SIOSEIS Website: " SIOSEIS is a software package for enhancing and manipulating marine seismic reflection and refraction data, sponsored by the National Science Foundation (NSF) and the Scripps Industrial Associates. The system currently runs on Mac OSX (PowerPC and Intel) and PCs (Linux and CYGWIN) E-mail phenkart@gmail.com for inquires. Open source can be downloaded from "<http://sioseis.ucsd.edu/index.html>"

I recommend you read the documentation at this website for many details on this valuable software.

L\_SU integrates some of the functionality of SIOSEIS in order to convert data written in a SEG2 format into SU formatted data.

You can use your browser to navigate to that website and download the file or you can directly load it into your folder by the following command:

### 3.1 Linux

```
% wget http://sioseis.ucsd.edu/src/sioseis-2016.3.1.tar.bz2
```

After you untar and decompact this software read the README file to learn how to install the programs while using root privileges. Later, when L\_SU looks for Sioseis you should have the path to the binary defined.

In order to decompact use bunzip as follows:

```
% bunzip sioseis-2016.3.1.tar.bz2
```

Then, you can untar the directory

```
% tar -xvf sioseis-2016.3.1.tar
```

After you compile the programs, move the directory and all of its contents to your preferred system location:

```
% mv sioseis-2016.3.1 /usr/local/ sioseis
```

Note that since gfortran10 you must include the following line together with the other FFLAGS in the "makefile".

FFLAGS += w -fallow-argument-mismatch -O2

Use of this additional switch during compilation will prevent errors.

#### 3.1.1 System environmental variables

If you use the common bash shell, the file ".bashrc" should contain the following command when SIOSEIS is installed within the directory "/usr/local/sioseis":

```
export $PATH=$PATH:/usr/local/sioseis
```

Commonly, your ".bashrc" file can contain other general definitions to achieve an identical result, for example:

```
export LOCAL=/usr/local
export SIOSEIS=$LOCAL/sioseis
export $PATH=$PATH:$SIOSEIS
```

## 3.2 Cygwin/Microsoft Windows

```
% wget http://sioseis.ucsd.edu/src/sioseis-2016.3.1.tar.bz2
```

After you untar and decompact this software read the README file to learn how to install the programs while using root privileges. Later, when L\_SU looks for Sioseis you should have the path to the binary defined.

In order to decompact use bunzip as follows:

```
% bunzip sioseis-2016.3.1.tar.bz2
```

Then, you can untar the directory

```
% tar -xvf sioseis-2016.3.1.tar
```

After you compile the programs, move the directory and all of its contents to your preferred system location:

```
% mv sioseis-2016.3.1 /usr/local/ sioseis
```

Note that if you use a gfortran version greater than 9x you will get compilation errors. Try to replace your gfortran compiler with the slightly earlier version.

### 3.2.1 System environmental variables

If you use the common bash shell, the file “**.bashrc**” should contain the following command when **SIOSEIS** is installed within the directory “**/usr/local/sioseis**”:

```
export $PATH=$PATH:/usr/local/sioseis
```

Commonly, your “**.bashrc**” file can contain other general definitions to achieve an identical result, for example:

```
export LOCAL=/usr/local
export SIOSEIS=$LOCAL/sioseis
export $PATH=$PATH:$SIOSEIS
```