

Installation Guide for

SeismicUnixGui: a graphical user interface for Seismic Unix*

**Juan M. Lorenzo
Nathan W. Benton**

Version: 0.87.5 SeismicUnixGui, June, 2025.

The latest installation guide is downloaded by the following linux instruction:

```
wget https://github.com/gllore/AppSeismicUnixGui/tree/V0.87.5/lib/App/SeismicUnixGui/doc/  
SeismicUnixGuiInstallationGuide0.87.5.pdf
```

*Stockwell, J. W. (1999). The CWP/SU: Seismic Unix package, Computers & Geosciences, 25, 415-419.

Table of Contents

1	Installation Background and Prerequisites	3
1.1	Docker installation for Microsoft Windows (*Version 11)	3
1.1.1	Regular use of SeismicUnixGui once the installations are complete	6
1.2	Docker installation for Mac OS (Intel/AMD chipsets).....	7
1.3	Tested operating systems	7
1.4	Definition of system variables in your computer work environment	8
1.5	Software required by Seismic Unix and SeismicUnixGui	10
1.5.1	Seismic Unix	10
1.5.2	SeismicUnixGui.....	10
2	Automated installation of SeismicUnixGui by cpan	10
2.1.1	Linux.....	11
2.1.2	Loading cpan	11
2.1.3	How to load required Perl modules from the CPAN.....	12
3	Installation of SeismicUnixGui from github	13
4	Fortran modules in SeismicUnixGui.....	14
4.1	pgplot (*Ubuntu)	14
4.2	mmodpg (*Ubuntu)	16
4.3	SioSEIS	17
4.3.1	Linux.....	17
5	C-based modules in SeismicUnixGui.....	18
5.1	Synseis.....	18
6	Seismic Unix installation	18
6.1	Direct download of Seismic Unix from wiki.seismic-unix.org (ubuntu 20.x).....	18
6.1.1	Preparation of Makefile.config.....	19
1.2	Download of Seismic Unix from github.com (ubuntu 20.x)	20
1.2.1	Preparation of Makefile.config.....	21

1 Installation Background and Prerequisites

1.1 Docker installation for Microsoft Windows (*Version 11)

We have currently introduced an automated installation procedure using Docker “containerization” software. The current Docker image of SeismicUnixGui (SUG) also includes pre-compiled, third-party software, including *Seismic Unix* (version 26.1.4), *mmodpg*, *pgplot*, and *SioSEIS*. We have tested this installation with the Windows Subsystem for Linux (WSL-2) backend. In the case that the user does not have WSL-2 (or WSL-1) on their machine and/or is not familiar with it, this document will also provide a quick, simple way to set up WSL-2.

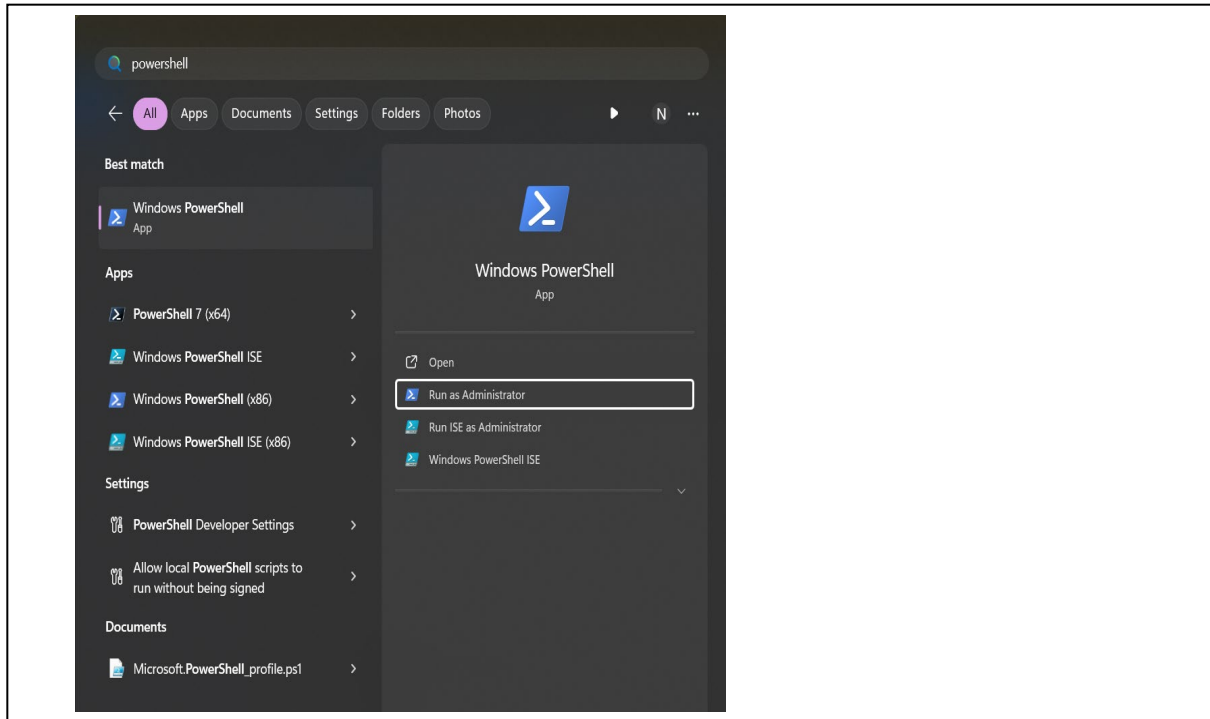
The complete set of following instructions (Steps 1 through 9) will remove all prior Docker installations on your machine. If Docker is already present and working on your machine and you also have the latest version of WSL running, and you do not want to reinstall them both, execute only the instruction in Steps 10 and beyond. All scripts and corresponding steps should be executed using the “Windows PowerShell” application with Administrator privileges.

To download the necessary scripts used in the following instructions, in your web browser navigate to: **https://github.com/gllore/SUG_WSL_Installation**
Click on each of the six files, one at a time, and download them each to your computer in their “Raw” format.

Completion time (e.g. seconds to 20 minutes) for each step depends on multiple factors such as your networking speed and the hardware you are using. Ideally, the instructions can be combined into a single program, but because your machine will be restarted three times, that option is less feasible.

Step 1:

To start a PowerShell, under the Windows icon, enter “PowerShell” (not **CMD**), select **Administrator** privileges, and then run the application.



Go ahead and copy the scripts that you will need into the local working directory of the PowerShell. Enter the following commands:

Define where you downloaded your files from the github. Note that this path and username, will differ on your computer. Enter the following command:

```
$Path="C:\Users\USERNAME\Downloads\*.ps1"
```

Define your destination for copying scripts. Enter the following command:

```
$Destination="C:\WINDOWS\system32\"
```

Copy all the files. Enter the following command:

```
copy -Path $Path -Destination $Destination -PassThru
```

Make the scripts you just copied executable for the current session only and answer “Y” to the question. Enter the following command:

```
Set-ExecutionPolicy -Scope Process -ExecutionPolicy Bypass
```

Step 2:

Stop and remove any Docker software and meta-data, if they exist. Enter the following command:

```
remove_docker.ps1
```

Step 3:

Check whether WSL (and specifically WSL-2) is present and working in your Windows OS. Enter the following command:

```
check_wsl.ps1
```

Step 4:

Remove and disable all container or virtual machine software from your Windows OS. This step will automatically restart your machine. Enter the following command:

```
disable_wsl_vm.ps1
```

Step 5:

Wait for the machine to restart and log back in. After that, launch the Windows PowerShell application, again with **Administrator** privileges.

Make the scripts you just copied earlier executable for the current session only and answer “Yes” to the question. Enter the following command:

```
Set-ExecutionPolicy -Scope Process -ExecutionPolicy Bypass
```

Step 6:

Enable the Docker container and virtual machine software for user’s Windows OS. This step will automatically restart your machine. Enter the following command:

```
enable_wsl_vm.ps1
```

Step 7:

Wait until your machine restarts and launch the Windows PowerShell application with **Administrator** privileges.

Make the scripts you copied earlier executable for the current session only and answer “Y” to the question. Enter the following command:

```
Set-ExecutionPolicy -Scope Process -ExecutionPolicy Bypass
```

Step 8:

Identical to Step 3, this script checks, installs, and configures the latest version of WSL-2:

```
check_wsl.ps1
```

Step 9:

Download, configure, and run the Docker Desktop application. **After downloading, the Docker application startup sequence will launch automatically**, and when it does, you may skip all the (optional) registration sections. This step takes the longest. Enter the following command:

```
docker_setup.ps1
```

For some users/machines, the Docker Desktop may not automatically start or restart **for later use**, so in case it isn’t running or doesn’t start, start the Docker Desktop manually, using **Administrator privileges**, i.e., search for the application on your machine and click on the icon.

Step 10:

Download and launch the X-server for SeismicUnixGui. If the application does not automatically render a screen, you may need to click/access it via the task bar-- look for a large black "X"). Execute the following command:

```
setup_xserver.ps1
```

Step 11:

Open a PowerShell window with only **regular user privileges**. Download the SeismicUnixGUI (sug) image using the Docker Desktop or with the following command in the PowerShell Window:

```
docker pull nathanbenton/sug:custom
```

Step 12:

Start using the Docker container. The command prompt changes to something like the following: sug_user@nnnnnnnnnnnn:~\$

Execute the following command:

```
docker run -v C:/sug_user:/home/sug_user -it nathanbenton/sug:custom
```

Step 13: Start SeismicUnixGui (SUG) from within the running Docker container via the following command. Once that successfully runs, SeismicUnixGui will render on the X-Server window. Run the following command:

```
SeismicUnixGui
```

1.1.1 Regular use of SeismicUnixGui once the installations are complete

If the installation of the docker image is complete, each time that you want to return to running **SeismicUnixGui**,

- (a) Open a PowerShell window with **regular user privileges**,
- (b) Run the Docker Desktop (manually) in the background.
- (c) Run the Windows X-server application as follows:

```
& 'C:\Program Files\VcXsrv\vcxsrv.exe' :0 -ac -multiwindow -clipboard
```

- (d) Enter the following command:

```
docker run -v C:/sug_user:/home/sug_user -it nathanbenton/sug:custom
```

- (e) Once in your home directory, start the Seismic Unix application as follows:

```
SeismicUnixGui
```

In future, you can download the latest versions of the Docker image (Step 11) with the following instruction- - first make sure that the Docker desktop is running.

```
docker pull nathanbenton/sug:custom
```

1.2 Docker installation for Mac OS (Intel/AMD chipsets)

To evaluate whether your local machine uses an CPU processor manufactured by Advanced Micro Devices (AMD), open a terminal and run the following instruction from the command line:

```
uname -m
```

If this command returns this output: **x86_64**, your platform uses an AMD processor. If the command returns **arm64**, your will not be able to use the available Docker image version for SUG software. We are currently working to build a Docker image for the ARM processor.

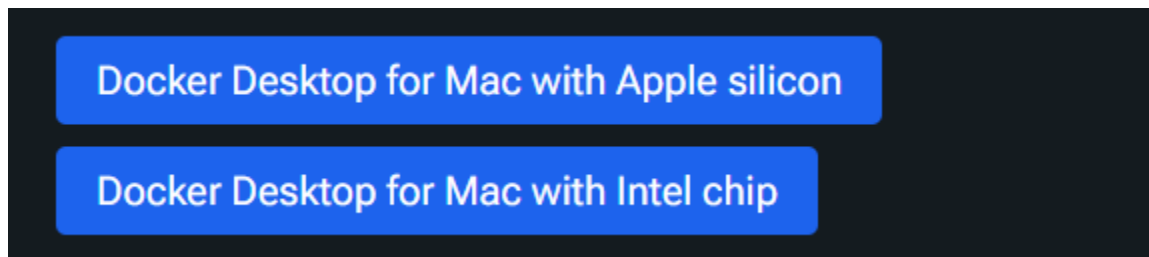
For rendering graphics, download the software package: <https://www.xquartz.org/>

This software is equivalent to “X-Server” on the Windows OS. Xquartz will capture and display images that are broadcast from the Docker Container.

To set up Docker Desktop for macOS, read the official documentation: <https://docs.docker.com/desktop/setup/install/mac-install/> . Start by downloading and installing required “Docker.dmg” file for the Docker Desktop (required).

The following instructions highlight those found online and may be helpful:

1. Proceed to select the Intel option (as shown below) and choose *Intel chip*:



2. After the file finishes downloading, “double-click” the “Docker.img” or install from the command line, to invoke the startup and configuration process for Docker Desktop.
3. Complete the installation instructions for the Docker Desktop as well as for X-Quartz. Once you are assured that both software packages are running, download the AMD-based Docker image by running the following command:

```
docker pull nathanbenton/sug
```

4. Beyond this point, all other instructions are similar to those used in the Windows OS.

1.3 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: SeismicUnixGui)

We have used the following instructions to install SeismicUnixGui 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 and 8
Debian	10 (buster)
Ubuntu	18.04.3, 20.x, 22.x, 24.x
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 load correctly. The following installation guide deals with the popular Ubuntu operating system.

Special NOTE: Do not simply cut and paste from this document into your linux environment without paying special attention not to include invisible characters (e.g., spaces) which will generate execution errors.

1.4 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_44R22"`

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.

If you clone SeismicUnixGui from the github, place Perl programs under “**/usr/local/pl**”. Then, in order for all the Perl scripts and other programs to run, you will need to add several new lines of instructions within my file “**.bashrc**” file, which is located in my home directory. These environment variables will be different from an automated installation (Section 2) from the CPAN and are as follows:

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

```
# for Seismic Unix
export CWPROOT=$LOCAL/cwp_su_all_44R22
export PATH=$PATH:$CWPROOT/bin
export PATH=$PATH:$CWPROOT/src/Sfio/bin
```

```
# for SeismicUnixGui
export SeismicUnixGui=/usr/local/pl/App-SeismicUnixGui/lib/App/SeismicUnixGui
export SeismicUnixGui_script=$SeismicUnixGui/script
export PATH=$PATH:$SeismicUnixGui_script
```

```
# for SeismicUnixGui extend PATH declaration to OS
export APP_LIB=$PL/App-SeismicUnixGui/lib
export PERL5LIB=$PERL5LIB:$APP_LIB
export PATH=$PATH:$SeismicUnixGui/fortran/bin
export PATH=$PATH:$SeismicUnixGui/c/bin
```

```
# for PGPLOT (fortran plotting libraries)
export PGPLOT_DIR=$LOCAL/pgplot
# default display device for pgplot
export PGPLOT_DEV=/XWINDOW
```

```
# for SIOSEIS binaries
Export SIOSEIS=$LOCAL/sioseis
Export PATH=$PATH:$SIOSEIS
```

If you use the Automated installation of SeismicUnixGui by cpan (Section 2), then examples of the environment variables you will need should be provided by the installation process.

1.5 Software required by Seismic Unix and SeismicUnixGui

1.5.1 Seismic Unix

Some users experience problems when installing Tk modules in SeismicUnixGui, or Xmotif modules in Seismic Unix. 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 update
% sudo apt install ncftp
% sudo apt install dpkg-dev
% sudo apt install libx11-dev libfreetype6-dev libxft-dev libmotif-dev
% sudo apt install aptitude libpng-dev libz-dev libjpeg-dev
% sudo apt install evince (automatically shipped with ubuntu)
% sudo apt install gfortran (e.g., 4:9.3.0-1ubuntu2)
% sudo apt install libxext-dev (e.g., 2:1.3.4-0ubuntu1)
% sudo apt install git
```

1.5.2 SeismicUnixGui

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

You will need:

- *evince*
- *ncftp*
- *plocate*

For example, evince which is a viewer for postscript files may already present in your system. If it is not, the following command (for Ubuntu) is an example of how to install the program.

```
% sudo apt install evince
```

2 Automated installation of SeismicUnixGui by cpan

Most linux-type systems come automatically with the Perl language, but you can check to see if you have Perl installed and its version at the same time, by entering the following command:

```
% perl -v
```

If you have **perl** installed, then the simplest and quickest way to install **SeismicUnixGui** is with the following instruction:

```
% sudo cpan App::SeismicUnixGui
```

There are two questions that **cpan** will ask, related to where you want to install the software. For the first question, in the general case, answer [y].

If you are going to make the software accessible to everyone on your system (“global installation”) you should use the “**sudo**” or superuser option. If your installation will be personal/local use the default option for the second question (“local::lib”).

Cpan will automatically detect if you need additional packages and install them as needed. During the installation you will see that each program is tested thoroughly before. Do not be surprised if the installation takes several minutes.

If, however, you see that some packages are still missing, then load them manually. For example if there is a failure to find the “Moose” package then enter the following on the command line:

```
% sudo cpan Moose
```

2.1.1 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.

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

2.1.2 Loading cpan

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

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

```
% sudo apt upgrade
```

```
% sudo cpan -v
```

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

2.1.3 How to load required Perl modules from the CPAN

When you install SeismicunixGui using **cpan**, cpan will automatically install the packages that you will need. But, if you install SeismicUnixGui via the github clone (see below, Section 3) likely that if you install SeismicUnixGui you will also need to install the necessary Perl packages manually. These include some, or all of the following Perl modules:

- *MIME::Base64*
- *Module::Refresh*
- *Moose*
- *PDL::Core*
- *Clone*
- *File::ShareDir*
- *File::Slurp*
- *Shell*
- *Test::Compile::Internal*
- *Time::HiRes*
- *Tk*
- *Tk::JFileDialog*
- *Tk::Pod*
- *aliased*
- *namespace::autoclean*

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

Perl module name in cpan format	Ubuntu (24.04.x)	Debian 9.9	Versions tested under CentOS7x	cygwin
Clone	0.47	0.38	0.39	0.43
File::ShareDir	1.118			
File::Slurp	9999.32			
MIME::Base64	3.16_01			3.15
Module::Refresh	0.18			
Moose	2.2207	2.187	2.2010	2.2012
PDL::Core	2.093			
Perl	5.38.2	5.24.1	5.16.3	5.26.3
Shell	0.73	0.73	0.73	0.73
Test::Compile::Internal	3.3.3			
Time::HiRes	1.9775			
Tk::JFileDialog	2.40	1.62	2.2	2.20
Tk or PerlTk	804.036	804.033	804.034	804.034
Tk::Pod	0.9943	5.41	5.41	5.41
aliased	0.34			
namespace::autoclean	0.29			

However, if any of these packages fail to install and you get errors that mention the packages by name then manually install the missing Perl package. Use the following as examples:

```
% sudo cpan PDL::Core
% sudo cpan aliased
```

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.

3 Installation of SeismicUnixGui from github

Most of the SeismicUnixGui scripts, are written in Perl, available at: www.github.com/gllore and can be installed anywhere on your machine, as long as you prescribe their location to the operating system (See 1.2.1).

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 SeismicUnixGui 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
```

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_SeismicUnixGui=/usr/local/pl/

# create installation directory
mkdir $installation_directory_for_SeismicUnixGui

# change into the installation directory
cd $installation_directory_for_SeismicUnixGui

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

# git status
git status
```

It is likely that if you install SeismicUnixGui via the github clone you will also need to install the necessary Perl packages manually. If a package is missing when running SeismicUnixGui, then the error messages will mention them by name and you will need to install the missing Perl package manually. Use the following as examples:

```
% sudo cpan PDL::Core
% sudo cpan aliased
```

Required Perl packages are listed above in Section 2 (Automated installation of SeismicUnixGui by cpan).

On occasion, when building Perl modules, the following packages may be needed.

```
% sudo cpan Module::Build      (e.g., V0.4231 installed)
% sudo cpan TAP::Harness       (e.g., V3.42 installed)
```

4 Fortran modules in SeismicUnixGui

There are “Tools” in SeismicUnixGui that communicate with programs written in Fortran (e.g., immodpg). These SeismicUnixGui 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)

*Instructions that follow work with Ubuntu, and many (not all) of the following steps are very similar in other operating systems(OS) but for all OS it is essential to have the directory paths correctly identified for your operating system.

4.1 pgplot (*Ubuntu)

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 a remote web directory to your local home directory:

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

- *Decompress the downloaded file: `pgplot5.2.tar.gz`, and place the reconstituted directories and files it contains within a general directory: **`"/usr/local/src/pgplot"`**, i.e.:*

```
% mkdir /usr/local/src
% tar -xzf 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)*

```
% 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.

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

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

(Remember that your directory address for unpacking files is at location: **/usr/local/src/pgplot** but that the final address where you will install pgplot is at: **/usr/local/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**”.

Stay in the same directory (“**/usr/local/pgplot**”) and finish up by running the following these three commands in order:

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

You are now ready to try out the demonstrations. Go to **/usr/local/pgplot**:

```
% cd /usr/local/pgplot
```

Once you are in the directory, run the following program and you will see what pgplot can do:

```
% pgdemo1
```

Modify your **.bashrc** file to include necessary variable definitions (Section **1.2.1** for Linux and *Error! Reference source not found.* for Windows)

That is, add the following line(s) to your **.bashrc** file

```
export PGPLOT_DIR=/usr/local/pgplot
```

4.2 mmodpg (*Ubuntu)

For interactive, forward modeling of the arrival times of refraction and reflections we use **mmodpg** (Vera, 1994). You have the option to install fortran libraries after you finish installing SeismicUnixGui. (See Help->About in the GUI itself)

A directory called **“fortran”** contains a raw installation script which is called **“run_me_only.sh”**. The installation script compiles, links the fortran code and places the executable version in the appropriate directory.

4.3 SioSEIS

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 inquiries. 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.

SeismicUnixGui 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:

4.3.1 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 SeismicUnixGui 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 += -fallow-argument-mismatch -O2
```

Use of this additional switch during compilation will prevent errors.

4.3.1.1 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
```

If these environment variables are already defined in your .bashrc script, do not define them again, as you may possibly hinder your ability to work with the operating system.

5 C-based modules in SeismicUnixGui

5.1 Synseis

For simple interactive, normal-incidence synthetic seismogram modeling we use Synseis (under “**Tools**” tab). When you first install SeismicUnixGui you will be asked whether you want to carry out the C installation as well. If you decide to say “no” you can go back later and rerun the **cpan** command to install the software.

A directory called “**c/synseis**” contains a raw installation script which is called “**run_me_only.sh**”. The installation script compiles, links the C code and places the executable version in the appropriate directory.

6 Seismic Unix installation

At present, for the latest versions of Seismic Unix, we recommend that you download Seismic Unix and install the program as per **<https://wiki.seismic-unix.org/doku.php>**

But, if you prefer there is also a version that you can download from the github.

6.1 Direct download of Seismic Unix from wiki.seismic-unix.org (ubuntu 20.x)

Download version 44R22 of Seismic Unix from the following site:

<https://wiki.seismic-unix.org/doku.php>

Once you have downloaded and untarred and unzipped the “**/src**” directory into your local directory you can carry out the following instructions in individually or as a set of instructions in a file called, for example, “**copy.sh**”

```
>sudo sh copy.sh
```

```
#!/bin/bash
# this file is called copy.sh

# give a name to directory
installation_directory_for_SU= /usr/local/cwp_su_all_44R22

# create installation directory
mkdir $installation_directory_for_SU

# change into the installation directory
cd $installation_directory_for_SU

# copy from your local directory into the new directory
cp -r /your/local/directory/src .
# ... and position yourself one directory above:
cd ..

# change permissions and ownership
chmod -r your_login_name $installation_directory_for_SU
chown -r your_login_name $installation_directory_for_SU

# create a directory where your executable binary files ("bin") will be stored
mkdir bin
```

(We strongly recommend that you change the permissions and ownership of the directory that will contain the Seismic Unix files to a local user, such as yourself.)

6.1.1 Preparation of Makefile.config

Determine the location of your "**lib**"(aries) and "**include**" directories. For example, you can find these by issuing the following commands:

```
%sudo updatedb
%locate include/X11 | more
```

Please follow the instructions in all the "**Installation_Instructions**" files. Before you carry out the installation, make the following changes to the file:

"/usr/local/cwp_su_all_44R22/src/Makefile.config".

```
IX11 = /usr/include/X11
LX11 = /usr/lib/x86_64-linux-gnu
IMOTIF = /usr/include/X11
LMOTIF = /usr/lib/x86_64-linux-gnu
OPTC = -g -std=c99 -Wall -pedantic -Wno-long-long
FC = gfortran
```

```
FOPTS = -g
FFLAGS = $(FOPTS) -ffixed-line-length-none
```

(In the case above the installation is for a 64-bit operating system.)

1.1.1.1 Compilation and Installation

As per the explanation of the “Installation_Instructions”, stay in the following directory: “/usr/local/cwp_su_all_44R22/src” and run the following commands one at a time:

```
%make install
%make xtinstall
```

Beyond this point additional installations are not essential. Should you choose to continue, note that errors during installation of some of the fortran libraries can be overcome by manually deleting the pre-existing library: “*.a” and re-running:

```
%make finstall (non-essential)
%make xminstall (non-essential)
```

1.2 Download of Seismic Unix from github.com (ubuntu 20.x)

As well, a version of Seismic Unix (R19) 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).

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.

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_44R22”. 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_44R22/src”

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_44R22

# 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/JohnStockwellJr/SeisUnix.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 Administrator privileges as **sudo**, e.g.:

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

1.2.1 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_all_44R22/src/**Makefile.config**”:

1.2.1.1 Ubuntu 20.x

Determine the location of your “**lib**”(raries) and “**include**” directories. For example, you can find these by issuing the following command:

```
%sudo updated
%locate include/X11 | more
```

Please follow the instructions in all the “**Installation_Instructions**” files. Before you carry out the installation, make the following changes to the file:

“/usr/local/cwp_su_all_44R22/src/**Makefile.config**”.

```
IX11 = /usr/include/X11
LX11 = /usr/lib/x86_64-linux-gnu
```

```
IMOTIF = /usr/include/X11
LMOTIF = /usr/lib/x86_64-linux-gnu
OPTC = -g -std=c99 -Wall -pedantic -Wno-long-long
FC = gfortran
FOPTS = -g
FFLAGS = $(FOPTS) -ffixed-line-length-none
```

(In the case above the installation is for a 64-bit operating system.)

1.2.1.1.1 Compilation and Installation

Stay in the current directory:

“/usr/local/cwp_su_all_44R22/src” and run the following commands one at a time:

```
%sudo make install
%sudo make xtinstall
%sudo make xminstall
```