Introducing Novices to Active-Source Seismology via SeismicUnixGui

Juan M. Lorenzo*[1], Mike Hubenthal[2], Nathan W. Benton[3], and Emilio E. Vera[4]

Abstract

Since its origin in 1998, the Incorporated Research Institutions for Seismology Undergraduate Research Internship Program, which has now been integrated into EarthScope, has benefited 262 students, helping them to initiate their careers in research and industry. From 2006 onward, interns have participated in a weeklong introductory program in active-source and earthquake seismology that emphasized field data collection, analysis, and interpretation in the classroom. Here, we describe an active-source seismology experience in which we adopted the well-known 2D, seismic reflection processing software package, Seismic Unix (SU). To encourage critical understanding of the seismic data, within the limited time frame of the instructional program, we created a new graphical user interface (GUI): SeismicUnixGui (SUG). SUG facilitates interactive testing of parameters on visual outputs, with single, mouse-driven clicks. SUG software is free, open-source with cross-platform capability. SUG runs on Linux-based systems such as Ubuntu or the Microsoft Windows Subsystem for Linux. SUG for Windows Desktop and older Apple PC platforms that use Intel processors is distributed as a Docker image. Development of SUG for more recent Apple silicon-based platforms is currently underway. Perl-based software wrappers give access to 290 SU and third-party programs that are used for a range of fundamental signal analysis functions including data format manipulation, integral transforms, seismic-waveform modeling, poststack time migration, and various image displays and manual data picking. SUG enforces a manageable, predefined directory structure to generate shareable bash and Perl scripts, which work independently of the GUI itself. The free, open, and downloadable code is accompanied by an installation manual, a demonstration field data set collected by student interns, and a tutorial with examples of the working scripts employed by the students to process the data during orientation week.

Cite this article as Lorenzo, J. M., M. Hubenthal, N. W. Benton, and E. E. Vera (2025). Introducing Novices to Active-Source Seismology via SeismicUnixGui, *Seismol. Res. Lett.* **XX**, 1–7, doi: 10.1785/0220250080.

Supplemental Material

Introduction

During the week-long orientation of the Incorporated Research Institutions for Seismology (IRIS) Undergraduate Research Internship Program, now integrated into EarthScope's "Research Experiences in Solid Earth Sciences for Students" (RESESS; Research Experiences in Solid Earth Sciences for Students [RESESS], 2024), participants keen to gain experience in seismology participated in more than 30 hr of instruction focused on both passive-(~17 hr) and active-source (~15 hr) seismology. Broadband sensors, on loan from the National Science Foundation (NSF) Seismological Facility for the Advancement of Geoscience (SAGE) EarthScope Primary Instrument Center (EPIC) Facility (EarthScope Primary Instrument Center [EPIC], 2024; formerly the Program for Array Seismic Studies of the Continental Lithosphere [PASSCAL] Instrument Center) were deployed by the participants (~3 hr) and the data were recovered several days later. Analysis of the data collected, along with data from the SAGE Facility, used Linux-based commercial software, Jupyter Notebooks, and Python-based libraries (e.g., ObsPy; Beyreuther *et al.*, 2010) to explore regional and whole-Earth structures through a mixture of lecture and hands-on learning opportunities. The active-source instructional component also involved data collection, digital signal analysis, as well as experiment design, and led students to consider the role of shallow structures in the data they had acquired. Over roughly 5 hr, students laid out a static linear array of geophones and conducted a

1. Department of Geology and Geophysics, Louisiana State University, Baton Rouge, Louisiana, U.S.A.; 2. EarthScope Corporate Headquarters, Washington, D.C., U.S.A., https://orcid.org/0000-0002-2790-7054 (MH); 3. Department of Geology and Geophysics, Louisiana State University, Baton Rouge, Louisiana, U.S.A.; 4. Facultad de Ciencias Físicas y Matemáticas, Universidad de Chile, Santiago, Chile

*Corresponding author: gllore@lsu.edu

TABLE 1
Between 2017 and 2023 Collection of Student Responses Guided and Drove Development of the SeismicUnixGui (SUG)

Questionnaire Results on Usage of SUG

Question and Sample Size (*n*)	Yes	A Little (%)	A Lot (%)	The Same (%)
The active-source experiment is one of the best parts of the orientation week (*n*-54)	56%			
My understanding of seismic data collection (station install and active-source collection) has improved (*n*-52)		33	62	5
My understanding of reflection and refraction theory (reflection coefficient, attenuation, resolution, etc.) has improved (*n*-52)		56	27	17
My ability to analyze reflection data in Seismic Unix (SU) has improved (*n*-51)		49	31	20

Years 2021 and 2022 were not held because of COVID.

simplified, horizontal shear-wave (S_H), common-midpoint seismic reflection survey.

Near-surface seismology has numerous applications, including studying the relationship between regional weathering patterns and stress fields (St. Clair *et al.* 2015), and geotechnical and engineering problems (Miller *et al.* 2010), such as seismic analysis of dams (Butler *et al.* 2019) and levees (Lorenzo *et al.* 2014, Wodajo *et al.* 2019), and seismic site characterization (e.g., Parolai *et al.*, 2022). Because the active seismology component of the instruction targeted the upper 40 m of the subsurface, postacquisition data processing and interpretation commonly led to discussions of near-surface processes in the upper-most portion of the critical zone (Slater *et al.* 2006).

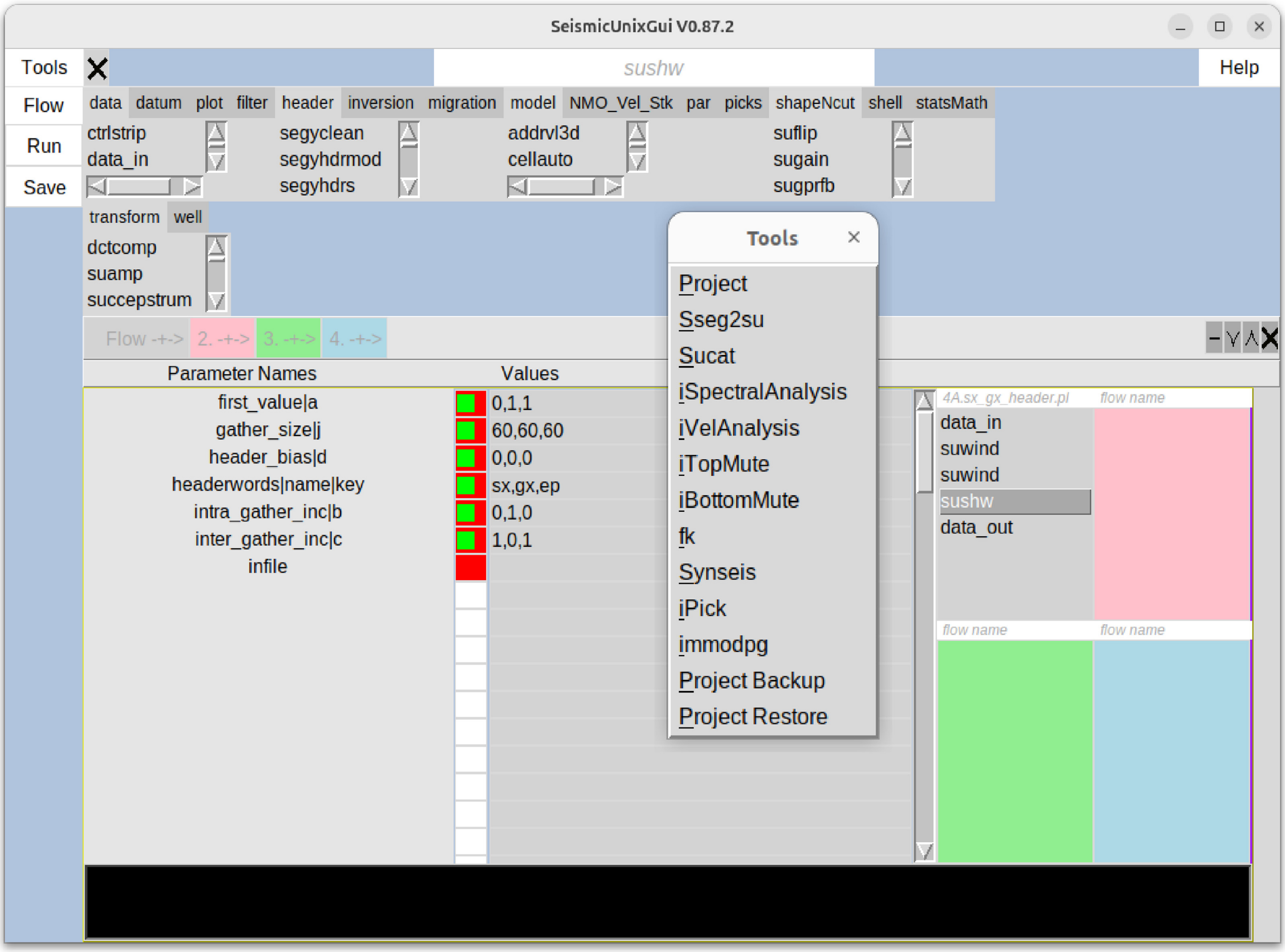
Since inception, instruction on the analysis of active-source seismic data has transitioned from early use of commercial software toward the adoption of free and open-source software. Commercial packages, although fully featured, included content that far exceeded the instructional needs of this orientation week. As a result, in 2014 we began to use Seismic Unix (SU; Stockwell, 1999), the well-known 2D, seismic reflection processing software package that runs under a Linux-family operating system (OS). The highly modular and reliable nature of the SU package requires writing shell scripts (e.g., using the bash language) to run small independent programs and exchange output via standard input and output line commands from any Linux OS. Although extensive documentation and instructional materials exist for SU (Forel *et al.*, 2005), shell-scripted workflows remain difficult for novice students, generally with limited-to-no prior coding experience, to learn in the instructional sessions available during the orientation week. As a result, instruction often had to prioritize programming skills over fundamental geophysical concepts or science discussions to interpret the subsurface data. Complex interactive processing stages, for example, selection of optimal stacking velocities, removal of bad data traces, or spectral analysis also required advanced coding beyond the scope of such introductory classes.

To compensate for this limitation and to encourage critical understanding of the seismic data, within the limited time frame of the instructional program, we opted to start development of a graphical user interface (GUI) for SU (Lorenzo, 2017). During the following 6 yr (Table 1), IRIS interns used the software during instruction. The opportunity to see sets of beginners interact with the interface provided invaluable feedback and drove the continuous development and improvement of the interface (Lorenzo, 2018, 2019). Here, we present SeismicUnixGui (SUG) that addresses the challenges novices face by removing the immediate need for shell scripting and facilitate interactive testing of parameters on visual outputs, with single, mouse-driven clicks. The latest SUG is open-source and free software, available for Ubuntu, a popular and free Linux OS and the Windows OS Linux subsystem via the Docker Desktop application (see Data and Resources). Docker software provides both the frontend user interface and backend, pre-compiled SU software in the same installation step.

SU, the SUG Interface, and Open-Source Software

SUG (see Data and Resources) consists of a primary interactive window used to assemble sequences of independent programs from the SU software package and assign their appropriate geophysical parameters (Fig. 1). SU (see Data and Resources) is a seismic reflection software package that permits the processing of moderate sized 2D seismic or ground-penetrating radar data sets (Rousset and the Seismic Unix Users Group, 2020). The SUG interface (Fig. 1) currently accesses 290 SU programs that are used for a range of fundamental signal analysis functions including data format manipulation, integral transforms, seismic-waveform modeling, poststack time migration, and various image displays and manual data picking.

SU pioneered the open-software concept for seismic reflection processing software. SUG follows using an open-source approach and in the development of its graphical interface using open-source packages. TkSu (Equipment News, 2003) and Botoseis (Lima *et al*, 2009) are early examples of graphically driven interactive software, both employing Tcl/Tk (Ousterhout, 1993) and using SeismicUnix modules as their engine. Other examples



include OpenSeaSeis (Olofsson, 2012), based on a Java interface (Table 2) or Visual_SUNT (2024), a commercial package under the Windows OS. Additional open-source seismic reflection codes (Table 2) exist that have been developed in either academic, for example, Madagascar (Fomel *et al.*, 2013), or commercial settings such as FreeUSP (BP-Amoco, 2024), or Conoco Philips Seismic processing software (CPSEIS; Conoco Philips Seismic processing software [CPSEIS], 2024), but these do not rely on SU software modules.

Design Strategy for SUG and Project Management

SUG primarily uses extensible modules written in Perl (Simmons and Rézic, 2024) to "wrap" (Fig. 2) themselves around modular, independent units of compiled C or Fortran code. The wrapper code provides an additional layer of abstraction between the user and the running code. This layer helps simplify code usability and adds tailored functionality that is not directly available from some versions of SU code. For example, one standard SU module ("sukill") is used for deleting unwanted data traces within a seismic data file. However, if the unwanted traces are irregularly spaced the task becomes repetitive and cumbersome for the user.

Figure 1. A brief tour of the SeismicUnixGui (SUG) graphical user interface (GUI). One main user interface handles input parameters for most programs. Four colored windows (right side) are used to compose and run four independent sets of sequential instructions. A pull-down menu ("Tools," top-left menu, and also shown detached) contains pre-existing tools built upon complex combinations of Seismic Unix (SU) programs, including management tools ("Project," "Project Backup," and "Project Restore"), third-party format conversion ("Sseg2su"), simple synthetic seismograms ("Synseis"), raytracing ("immodpg"), and interactive spectral, velocity analysis and data removal ("mute"), frequency–wavenumber (*f-k*) analysis, and data digitization ("iPick"). Traditional SU help installation manuals are available (via click of the third mouse button), as well as access to tutorial and installation manuals ("Help"). SU modules are classified within 16 scrollable windows (upper rows of interface, e.g., "data," "datum," "plot," "filter," "header," etc.). Behind the "Tools" menu, the main interface displays an example of "parameter values" for one specific SU module ("sushw"), belonging to the "header" category. Software wrappers facilitate the addition of descriptive parameter names as well as the original variable names from SU (e.g., "first_value or a"). The color version of this figure is available only in the electronic edition.

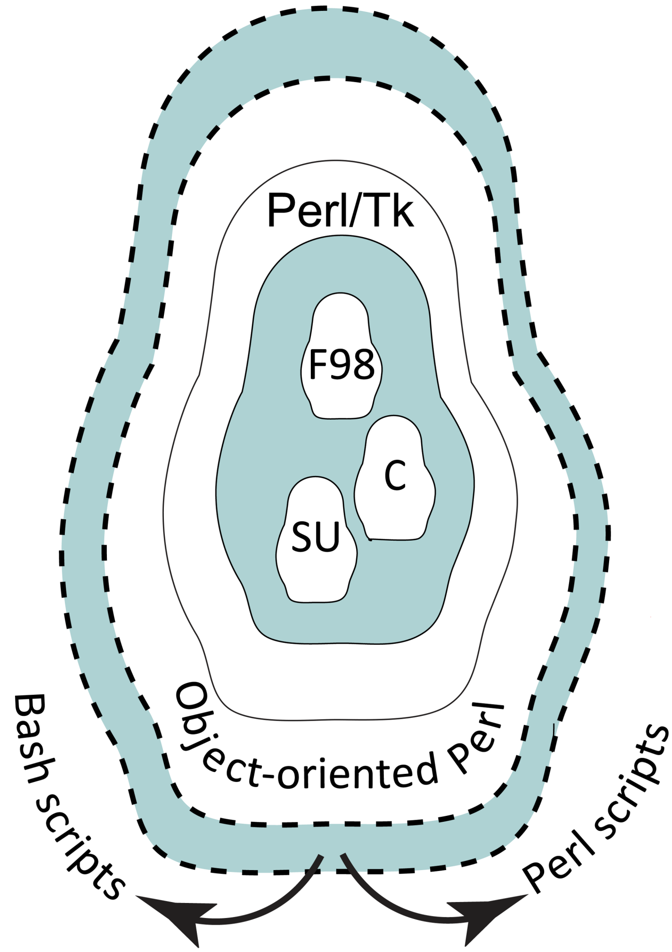


Figure 2. Schematic outline (inspired by nested Russian folk dolls), of the SUG architecture. SUG shows that both Perl and bash scripts are generated to run sequences of SU modules from the interface or command line. Perl integrates (Simmons and Rézic, 2024) the Tk toolkit, with background third-party software written in C (SU) and Fortran (F98); see the supplemental material. The color version of this figure is available only in the electronic edition.

Instead, the wrapper introduces a new option for the program that allows unwanted data traces to be specified more simply in a text-based list. The wrapper handles the process of data removal in a manner that is seamless to the user.

Software wrappers permit integration of varied and tested software into SUG. Although SU already has a SEG2 (Pullan, 1990)-to-SEG-Y (Levin and New, 2024) seismic format conversion module, one SUG wrapper incorporates data format conversion from another open-seismic processing software, Scripps Institute of Oceanography Seismic processing system (SIOSEIS; Scripps Institute of Oceanography Seismic processing system [SIOSEIS], 2011). Students can improve and develop their understanding of geological models via forward modeling of reflected and refracted seismic arrival data (e.g., Song and Ten Brink, 2004). For this purpose, SUG tool uses Perl wrappers to run an interactive raytracing program for a 1D layered earth model that is written entirely in Fortran 77 and 95. Perl scripts communicate asynchronously with the Fortran code via text messaging, and the Fortran dynamically updates data and calculations in response to button clicks and data input within the GUI. Perl can also readily incorporate Linux shell programs into its interface such as "evince," an open-source document viewer.

SUG forces data and generated Perl scripts into a prescribed directory structure. Directories and minimal files needed by SUG are created automatically at the start of each newly named Project using a "Project Selector" tool. Because Linux commands allow free navigation throughout the directories, the user can add new subdirectories within the existing file structure, without affecting SUG behavior. The default SUG Project directory structure allows multiple students to share one Project directory structure, while keeping independent copies of their scripts and data. This centralized location of student work on a hard drive has proven convenient for managing and supervising class exercises.

Shareable SUG Scripts

With SUG, a user populates a parameter sheet and runs a sequence of programs. Beneficially, SUG also provides

TABLE 2

List of Open-Source Seismic Reflection Processing Programs with Graphical User Interfaces That Interface with Seismic Unix (SU) Modules

Additional SU-Related Software

Name	Prerequisites	Source	License	Reference Author
BotoSeis	Java SE Runtime >6.0, Ubuntu 16.04	https://github.com/botoseis/BotoSeis	GNU General Public License v3.0. (GPL)	Lima *et al.* (2009)
GeBR	Gtk+, Java	https://www.ime.unicamp.br/~biloti/gebrproject/	GPLv2.0	Biloti (2012)
TkSU	C, Motif, Linux 2.2.14	http://www.henrythorson.com/software.htm	GPL	Equipment News (2003)
OpenSeaSeis	Java	https://github.com/JohnWStockwellJr/OpenSeaSeis	GPL	Olofsson (2012)

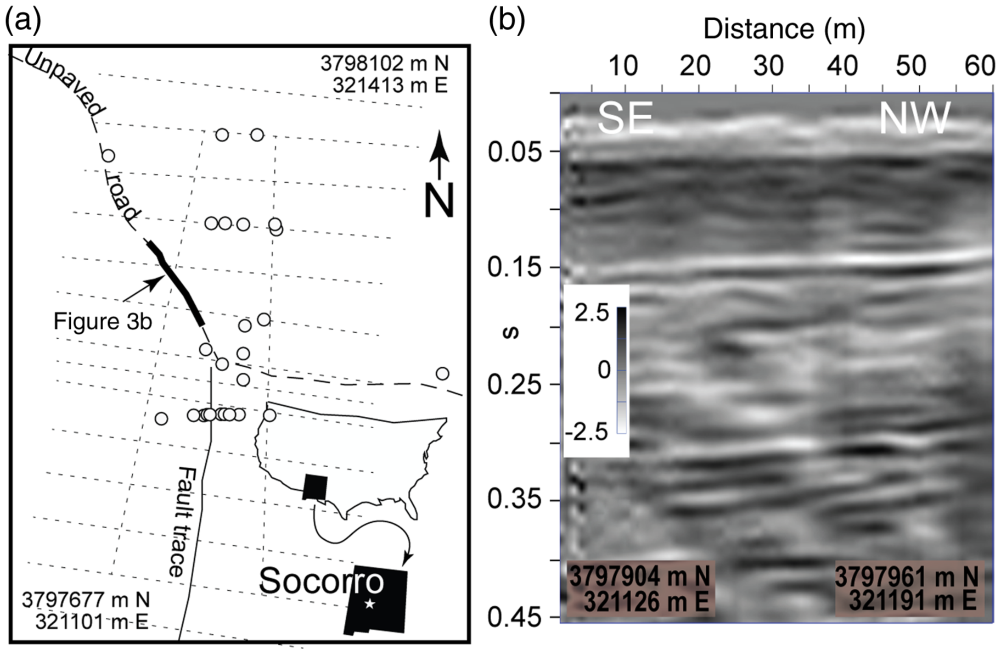


Figure 3. (a) Seismic line transect (thick-black line) along unpaved road in Servilleta National Wildlife Refuge, ~30 km north of Socorro, New Mexico, United States (UTM Z = 13, see inset). Straight dashed lines locate direct current resistivity profiles and open circles mark well locations also from prior study by Barnes *et al.*, (2021). (b) Output example from tutorial (step 9, flow 8A; see Data and Resources) displays a final stacked section that is produced by students. A linear stacking velocity gradient model ranging from 100 to 250 m/s is applied together with a low-pass filter (below 60 Hz) and automatically scaled amplitudes for the display. The reflector at approximately 0.14 s of two-way travel time can be interpreted to represent an erosional surface between sediments of the Rio Grande (below) and Rio Salado (above). The color version of this figure is available only in the electronic edition.

executable bash and Perl script versions of the command sequences, self-documented with a simple-to-use Perl markup language. Every bash script run from SUG is logged to a text file, which includes a time stamp and version number of the SUG that was used. Once generated, bash scripts can be recovered from the logging file and executed on the command line. Similarly, the generated Perl script can later be run from the command line directly or reapplied from within the GUI. Given the predefined directory structure, unmodified scripts also serve as a useful reproducible record of the processing steps that can be readily shared between students.

An advantage of the generated scripts is that they can be readily modified by a user to develop more elaborate, complex sophisticated interactive tools. Such added-on-tools that amalgamate the functionality of several SU modules are available under a drop-down menu (“Tools”; Fig. 1). These tools also can be run from the command line for convenience but are attached to the menu options in the GUI. We find that complex interactive processing stages, for example, selection of optimal stacking velocities, killing bad data traces, or spectral analysis, requires these more prebuilt tools because complex flows are beyond the scope of introductory classes.

A Data Set and Tutorial

An accompanying data set and tutorial is available for SUG (see Data and Resources). The data set was collected over half a day of field work by student participants in the IRIS internship program with the assistance of EPIC instrument center staff (EPIC, 2024; formerly the PASSCAL Instrument Center) and processed over a period of a few hours during the summer internship orientation week. Between 2018 and 2019, we collected S_H seismic reflection data across a nearby outcropping fault (Fig. 3a; Barnes *et al*, 2021). For planning and discussion of the experimental design of the survey, the Loma Blanca fault is readily identified in satellite imagery. In the field, the fault zone appears as a weathering-resistant ridge indurated by calcite cementation. The subsurface is well characterized by core samples from 29 boreholes and 16 direct-current electrical resistivity profiles in a small area approximately 100 m × 400 m in rectangular extent (Fig. 3a). Moreover, a sharp acoustic impedance contrast that exists at shallow depths of ~7 m (0.14 s at 100 m/s; Fig. 3b) provides readily recognizable reflection hyperbolae for velocity analysis and quick stacking.

Shear-wave data were collected because they were easier to separate by muting (see the supplemental material) from accompanying surface waves than traditional higher-velocity *P* waves. Further details of the acquisition geometry are included in the tutorial. Additional processing was required to stack pairs of shotpoint gathers with opposite polarities. At each S_H shot point location we generate two data sets of opposite polarities by hitting a small, embedded I-beam on opposite sides. Differencing the data sets attempts to double the amplitude of true S_H arrivals while attenuating converted *P*-wave modes (Helbig, 1986). Students’ exercises transition from the stage of raw field data to the final stacked section in approximately eight steps, and interact with the effects of the multiple processing parameters along the way guided by the instructor.

Future of SUG

The new EarthScope program continues to evolve while incorporating prior internship programs from the former UNAVCO;

the hands-on seismic experimentation will remain an integral part of the weeklong internship program. SUG exposes students to tailored and function open-source software they can continue to use during relevant research experiences. We hope that SUG or other similar projects will help introduce near-surface seismology to future generations of geophysicists. Continuous updates to the SUG will be available at three public sites (see Data and Resources) and enquiries can be directed to the first author. Changes will be dictated based on future student needs and responses from potential users.

Data and Resources

Source code for the SeismicUnixGui (SUG) program, which is intended for use under an Ubuntu version of the Linux operating system (OS), is available from two different internet locations. (1) The complete SUG package can be downloaded and installed free of charge, under the Artistic and General Public license, from https://github.com/gllore/App/SeismicUnixGui (last accessed June 2025). (2) SUG is also publicly available from the Comprehensive Perl Archive Network (https://www.cpan.org, last accessed June 2025) and can be installed automatically using the "cpan" utility in Perl. This automatic installation will detect and install missing modules and test the code during the installation process. For Windows OS and Apple PC users, a Docker image is available for free from https://hub.docker.com/r/nathanbenton/sug (last accessed July 2025). The current SUG will run on older Apple PCs that use Intel chipsets. Development of SUG for more recent Apple silicon-based platforms is underway. The docker image contains a precompiled, fully working version of SUG, Seismic Unix (SU), and other third-party software libraries (C and Fortran) that run seamlessly within the Windows Subsytem for Linux (WSL2). Docker is an open-source containerization software analogous to Anaconda (for Python, 2024) that allows for exchange of prefabricated software systems. Both a tutorial for the accompanying data set and an extensive manual explaining how to install SUG, SU, and third-party software such as SIOSEIS are available from: https://github.com/gllore/App-SeismicUnixGui/tree/V0.87.3/lib/App/SeismicUnixGui/doc (last accessed June 2025). Once SUG is installed, this document is also available under the "Help" button. Because the tutorial scripts and data set used are quite large in size, they are included separately as a single compressed file (~364 MB) from a different Zenodo site (doi: 10.5281/zenodo.13401754). Zenodo is general-purpose open-research repository (https://zenodo.org/). These data were collected by students and analyzed during the 2018 Undergraduate Research Internships in Seismology (URISE; Undergraduate Research Internships in Seismology [URISE], 2024) orientation week. The supplemental material for this article includes an example image output from the online tutorial and technical details of the design strategy for SUG. SUG is designed to interact with modules from SU, a free, open-software package, licensed under a New Berkeley Source Distribution. For purely Linux-based OS, the SU package should be installed separately and is available from https://github.com/JohnWStockwellJr/SeisUnix (last accessed July 2025).

Declaration of Competing Interests

The authors acknowledge that there are no conflicts of interest recorded.

Acknowledgments

The authors are grateful to John Stockwell and the Colorado School of Mines for developing Seismic Unix (SU) and to D. Roussset (Univ. Pau) for promoting growth and distribution of the SU. The former Incorporated Research Institutions for Seismology (IRIS/URISE) program was supported by the National Science Foundation (NSF) Awards EAR-1852339 and EAR-1724509. This funding supported numerous students who, with the assistance of New Mexico Institute of Mining and Technology (NMT) faculty including S. Bilek, and G. Spinelli, and the Program for Array Seismic Studies of the Continental Lithosphere (PASSCAL) instrumentation facility staff, especially G. Chavez, collected the tutorial data set and tested early versions of the SeismicUnixGui (SUG). Many funded research activities and graduate student projects also motivated code development, including Codelco Chile, ExxonMobil Upstream Research, DOE/LEQSF (2004-7)-L, USDS-Fulbright, K. Hayashi at Geometrics Inc., Louisiana Sea Grant, Southeast Louisiana Flood Protection Authorities, American Petroleum Institute (API) Delta Chapter- New Orleans, New Orleans Geological Society (NOGS), Southeastern Geophysical Society (SGS), and above all the Louisiana State University (LSU) Department of Geology & Geophysics for its active support of graduate student research. The original article benefitted from the highly constructive reviews of A. Kafka and an anonymous reviewer.

References

Barnes, H., J. R. Hinojosa, G. A. Spinelli, P. S. Mozley, D. Koning, T. G. Sproule, and J. L. Wilson (2021). Detecting fault zone characteristics and paleovalley incision using electrical resistivity: Loma Blanca Fault, New Mexico, *Geophysics* **86,** B209–B221.

Beyreuther, M., R. Barsch, L. Krischer, T. Megies, Y. Behr, and J. Wassermann (2010). ObsPy: A python toolbox for seismology, *Seismol. Res. Lett.* **81,** 530–533.

Biloti, R. (2012). GeBR:A free seismic processing interface, *74th EAGE Conference and Exhibition - Workshops*, July 2012, cp-295-00129, EAGE.

BP-Amoco (2024). BP-Amoco software, available at https://stuartschmitt.com/FreeUSP/FreeUSP_License.html (last accessed October 2024).

Butler, K. E., D. B. McLean, C. Cosma, and N. Enescu (2019). A borehole seismic reflection survey in support of seepage surveillance at the abutment of a large embankment dam, in *Levees and Dams: Advances in Geophysical Monitoring and Characterization*, J. M. Lorenzo and W. Doll (Editors), Springer International Publishing, Cham, Switzerland, 41–67.

Conoco Philips Seismic processing software (CPSEIS) (2024). ConocoPhillips seismic processing software, available at https://sourceforge.net/projects/cpseis/ (last accessed October 2024).

EarthScope Primary Instrument Center (EPIC) (2024). EarthScope primary instrument center, available at https://www.epic.earthscope.org/ (last accessed October 2024).

Equipment News (2003). *Seismol. Res. Lett.* **74,** no. 1, 50, doi: 10.1785/gssrl.74.1.50.

Fomel, S., P. Sava, I. Vlad, Y. Liu, and V. Bashkardin (2013). Madagascar: Open-source software project for multidimensional data analysis and reproducible computational experiments, *J. Open Res. Softw.* **1,** 1.

Forel, D., T. Benz, and W. D. Pennington (2005). *Seismic Data Processing with Seismic Un*x: A 2D Seismic Data Processing Primer*, Society of Exploration Geophysicists, Tulsa, Oklahoma.

Helbig, K. (1986). Shear waves- What they are and how they can be used, in *Shear-Wave Exploration Geophysical Development Series*, S. H. Danbom and S. N. Domenico (Editors), Society of Exploration Geophysicists, Tulsa, Oklahoma, 19–36.

Levin, S. A., and S. New (Editors) (2024). *SEG-Y_r2.1: SEG-Y revision 2.1 Data Exchange format*, Society of Exploration Geophysicists, available at https://library.seg.org/pb-assets/technical-standards/seg_y_rev2_1-oct2023.pdf (last accessed October 2024).

Lima, W., G. Garabito, and J. C. R. Cruz (2009). BOTOSEIS: A new Seismic Unix based interactive platform for seismic data processing, *11th Int. Congress of the Brazilian Geophysical Society and EXPOGEF 2009*, Salvador, Bahia, Brazil, 24–28 August 2009, 1507–1510.

Lorenzo, J. M. (2017). TkPl_SU: An Open-source Perl Script Builder for Seismic Unix, (263077/ NS41B-0014). Th., 14 December, *AGU Fall Meeting*, E. Morial Convention Center, 08:00-12:20, Hall D-F, New Orleans.

Lorenzo, J. M. (2018). L_SU (V 0.3.2)- a graphical user interface useful for building seismic flows from Seismic Unix, *Abstract presented at 2018 Fall Meeting, AGU*, Washington, D.C., 14 December.

Lorenzo, J. M. (2019). L_SU (V0.3.9): A graphical user interface useful for building seismic flows from Seismic Unix, *AGU Fall Meeting 2019*, San Francisco, California.

Lorenzo, J. M., J. Hicks, and E. E. Vera (2014). Integrated seismic and cone penetration test observations at a distressed earthen levee: Marrero, Louisiana, U.S.A, *Eng. Geol.* **168,** 59–68.

Miller, R. D., J. H. Bradford, and K. Holliger (Editors) (2010). *Advances in Near-Surface Seismology and Ground-Penetrating Radar*, Society of Exploration Geophysicists, American Geophysical Union, Environmental and Engineering Geophysical Society, Oklahoma; Washington, D.C.

Olofsson, B. (2012). SeaSeis: A simple open-source seismic data processing system, *74th EAGE Conference and Exhibition - Workshops*, July 2012, cp-295-00126, European Association of Geoscientists & Engineers.

Ousterhout, J. K. (1993). *Tcl and the Tk Toolkit*, Addison-Wesley Professional Computing Series, Addison-Wesley Publishing, Reading, Massachusetts.

Parolai, S., C. G. Lai, I. Dreossi, O.-J. Ktenidou, and A. Yong (2022). A review of near-surface QS estimation methods using active and passive sources, *J. Seismol.* **26,** 823–862.

Pullan, S. (1990). Recommended standard for seismic (/radar) data files in the personal computer environment, *Geophysics* **55,** 1260–1271.

Python (2024). available at https://www.python.org/ (last accessed October 2024).

Research Experiences in Solid Earth Sciences for Students (RESESS) (2024). Research experiences in solid earth sciences for students, available at https://resess.unavco.org/ (last accessed October 2024).

Rousset, D., and , and the Seismic Unix Users Group (2020). Seismic Unix, a valuable tool for seismic processing, *18th International Conf. on Ground Penetrating Radar*, Golden, Colorado, 14–19 June 2020, 251–254.

Simmons, N.-I., and S. Rézic (2024). Tk- A graphical user interface toolkit for Perl., available at https://metacpan.org/dist/Tk/view/Tk.pod (Version 804.036., last accessed October 2024).

Scripps Institute of Oceanography Seismic processing system (SIOSEIS) (2011). A computer system for enhancing and manipulating marine seismic reflection and refraction data, available at https://sioseis.com (last accessed October 2024).

Slater, L., R. Knight, K. Singha, A. Binley, and E. Atekwana (2006). Near-surface geophysics: A new focus group, *Eos Trans. AGU* **87,** 249–249.

Song, J.-L., and U. Ten Brink (2004). *RayGUI 2.0—A graphical user interface for interactive forward and inversion ray-tracing, U.S. Geol. Surv. Open-File Rept. 1426.*

St. Clair, J., S. Moon, W. Holbrook, J. Perron, C. Riebe, S. Martel, B. Carr, C. Harman, K. Singha, and D. Richter (2015). Geophysical imaging reveals topographic stress control of bedrock weathering, *Science* **350,** 534–538.

Stockwell, J. W. (1999). The CWP/SU: Seismic Unix package, *Comput. Geosci.* **25,** 415–419.

Undergraduate Research Internships in Seismology (URISE) (2024). Undergraduate research internships in seismology, available at https://www.iris.edu/hq/internship/ (last accessed October 2024).

Visual_SUNT (2024). Seismic Reflection processing, graphical interface for SEISMIC UNIX* for Windows, available at http://www.wgeosoft.ch/PDF/Visual_SUNT_Pro_Data.html (last accessed October 2024).

Wodajo, L. T., C. J. Hickey, and T. C. Brackett (2019). Application of seismic refraction and electrical resistivity cross-plot analysis: A case study at Francis levee site, levees and dams, in *Levees and Dams: Advances in Geophysical Monitoring and Characterization*, J. M. Lorenzo and W. Doll (Editors), Advances in Geophysical Monitoring and Characterization, Springer International Publishing, Cham, Switzerland, 23–40.

Manuscript received 4 March 2025
Published online 15 August 2025