

# GÊBR: A FREE SEISMIC PROCESSING INTERFACE

R. Biloti

email: [biloti@ime.unicamp.br](mailto:biloti@ime.unicamp.br)

keywords: *GêBR, seismic processing interface*

## ABSTRACT

*This article presents GêBR, a free graphical interface created for processing seismic data, with emphasis on teaching and research in this area. Besides allowing the creation and management of processing streams from predefined modules, GêBR allows adding new modules, directly by the user. This feature turns GêBR an interface to gather algorithms and programs produced by innovative research.*

## INTRODUCTION

There are many programs for processing seismic data freely available and widely spread, like those provided by Seismic Un\*x, Madagascar, FreeUSP, SEPlib, among others packages. What all those packages have in common is that they cluster command-line oriented programs, designed to be used in sequence, for processing data which flow through them. Moreover, most of that programs are controlled by command-line options, employed to set up their processing parameters.

To take full advantage of such programs, the user must have some technical skills, like shell scripting, submission tools, batch processing, process management, to name a few. All these skills, despite useful, are not a must-have knowledge for seismic data processing. A suitable graphical user interface should take care of most of the computational details, allowing the user to focus on the central problem of processing seismic data. This is particular important during training courses, where the limited duration of the course imposes a criterious selection of the material to be discussed.

A graphical user interface for seismic processing can also boost the usage of a program, making it more accessible to users and allowing its integration with other programs that are also available at the interface.

These principles have guided the development of GêBR, a graphical user interface to control command-line programs for seismic processing. It permits to build complex processing flows, from predefined modules, known as *menus*. Menus describing new programs can be added easily to the interface, extending its capabilities. GêBR was also designed to be simple, in the sense that a couple of hours is enough to introduce the core features of the interface, and allow to the user to start working with the seismic data.

## HISTORY OF THE GÊBR PROJECT

In mid-1990, Eduardo Filpo, a geophysicist of the Brazilian oil company, Petrobras, was a graduate student in the doctorate program in Geophysics of Federal University of Bahia (Brazil). He was the first to realized the need of a free interface for seismic processing programs, which were able to integrate scientific output within a platform such that it would be accessible beyond the limits of the research group (Filpo, 2009a,b).

In 2003 a first prototype interface was created by Filpo to assist in training new professionals who would work in seismic data processing. The prototype, named CIGEF, in allusion to the Introductory Course in Geophysics (*Curso Introdutório em Geofísica*, in Portuguese), taught to new geophysicists hired by Petrobras, was a proof of concept. It demonstrated how beneficial would be to have a graphical interface for processing, so that students could concentrate exclusively on the issue of seismic processing itself, and

no longer allocate time to the technicalities involved in creation, edition, submission and control of shell scripts, as was usual until then.

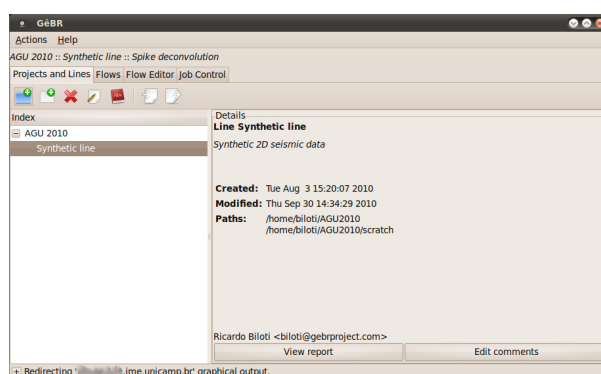
Another prototype, called GêBR, was launched in 2005. With that new interface it was possible to complete the processing of two seismic lines of real data. In mid 2007, aiming to build a solid foundation for the development of GêBR interface, a new version was written completely from scratch, by Ricardo Biloti. The current interface GêBR and other related programs are all based on that version.

October 2007 marked the official launch of Project GêBR to the community with its public presentation at the X International Congress of the Brazilian Geophysical Society, in Rio de Janeiro. Initially created to host the development of GêBR interface, the GêBR Project ended up gaining a larger dimension. Currently, the project aims to encourage the integration of the Brazilian Geophysical community, providing a graphical interface for processing seismic data not only as a useful tool for teaching students and professionals, but also as an aggregating point and vehicle for the dissemination of research results, accelerating and enhancing their impact within the academic community and the society as a whole.

## THE GÊBR INTERFACE

The main and most visible result of the GêBR Project is its graphical seismic processing interface. Unlike other free interfaces for creating seismic processing flows, such as TksU<sup>1</sup>, GêBR is not only designed to create and submit processing flows, but also to manage the whole data processing of a 2D seismic data.

The GêBR is organized into four tabs (see Figure 1): (1) Projects and Lines, (2) Flows, (3) Flow Editor, and (4) Job Control. The first two tabs are designed to facilitate the organization of the work, allowing the management of several projects, each of which with their own seismic lines and processing flows. The third tab is where the processing flows are assembled from processing modules already available in the interface, known as *menus*. After this stage of assembly, for each execution of a flow, a job is created. The fourth tab is where the monitoring and management of these jobs takes place.



**Figure 1:** GêBR interface. On the left, it is shown the project *AGU 2010* with one line, *Synthetic line*. On the right, it is shown some detailed information about that selected line.

### Simplicity

The simplicity of the GêBR graphical interface is not a negative point, possibly indicating lack of resources. On the contrary, it is its greatest advantage, making it feasible for new users to take advantage of the interface almost immediately. In each tab there is always a toolbar with buttons for implementing key actions associated with the context. Moreover, to avoid overloading the interface and to make access to functionalities faster, context menus (accessible through the right mouse button) are available, presenting shortcuts for actions relating only the element (or group of elements) selected.

<sup>1</sup><http://www.henrythorson.com/interface.htm>

### Extensibility

Another strength of the GêBR interface is its dynamics. New processing modules can be incorporated into the interface. This allows GêBR to act as a dissemination vehicle for the new research products. New programs can be immediately incorporated into the interface and ready to use.

In principle, any program that can be executed through the command line of a shell can be integrated to GêBR. The more the program is controlled by options provided in the command line, the better the integration of it with GêBR. Currently, around 400 programs are already integrated into GêBR, mainly from CWP Seismic Unix processing package (Stockwell, 1999).

The integration of a new program to GêBR is straightforward. It is done through an auxiliary interface, also developed under the GêBR Project, known as DéBR. To complete this integration there is no need to access the source code of the program, only its usage documentation.

### Organizational resources

To help users to organize their work within the GêBR interface there is a hierarchy of information to be respected and explored. The first level of this hierarchy is inhabited by *projects*. Projects are responsible for hosting groups of *lines*, an entity that occupies the second level of the hierarchy. Both for projects as well as for lines, it is possible to define basic properties, like the owner or responsible for the project, a title and a short description. For lines it is still possible to define preferred paths. These paths are used during the construction of processing flows, to facilitate the organization of the input and output files. The third level of the hierarchy is populated by *processing flows*. Just like for projects and lines, it is also possible to set basic properties for each flow.

Beside the lists of projects and lines or flows, there is always a summary table displaying the main details about the selected element. Besides the properties of the element, for flows, for example, it is also shown when the flow was last ran, in which server, and which files are configured as input or output. GêBR also yields detailed reports for lines and flows containing the list of all the flows of the line, tables summarizing the setup of each program of each flow, the input and output files for the flow, among other information. In addition, there is also placeholder for users to add their own comments on the element.

At each execution of a processing flow, a job is created and displayed on the last tab GêBR interface. When selecting a specific job in that tab, the user has access to all information generated by the processing flow, like the current state of flow (submitted, waiting in a queue for execution, running, completed, or stopped), the command line that was actually used to run the job, error messages, run time, etc. Furthermore, users can stop a running flow or cancel the execution of a flow which is waiting in a queue to start its execution.

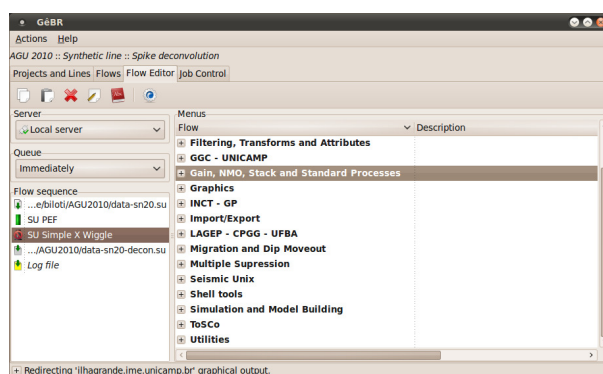
### Resources to assembly and configuration of flows

To facilitate the construction of processing flows, the menus are sorted by categories (see Figure 2). The same menu can belong to more than one category, making it easy to find, possibly using incremental search by the name of the menu. After finding the desired menu, by requesting the inclusion of it in the processing flow, all programs that make up the menu will be included in the flow, independently, allowing the user to disable or even remove some of them.

The configuration of each program of a processing flow is made through a parameters' dialog specific to the program. Figure 3 shows the parameters' dialog for program *S88 Modeling*. GêBR supports various types of parameters: numeric (integer, real, with or without limitation of range), text, multiple choice within a predefined list of options, Boolean variables and files. This specificity allows that the interface correctly validates the values filled out for each parameter, turning the task of configuring programs less error prone.

Through this dialog box of parameters, the user has access to the documentation of the program in question, as well as to an external link that refers to a page with more details about the program. By means of a button at the bottom of the dialog box, the setting of all parameters can be returned to default values.

Besides being able to fill in directly the value of a parameter, it is also possible to employ the *dictionary of parameters*, where values can be labeled to be shared among programs of the same flow, or even among



**Figure 2:** Flow Editor tab. On the left in the frame labeled *Flow sequence* it is shown the programs which compound the flow. In this case the second program (*SU Simple X Wiggle*) is disabled. On the right, it is shown the list of menu's categories, from which menus can be grabbed to assemble the flow.

programs of different flows (see Figure 4). For example, if two or more programs need to know a reference velocity, it would be better to fill this value in the dictionary and then set the appropriate parameters of those programs through the dictionary. Thus, changing the value of this property in the dictionary will have the effect to automatically propagate this change to all programs. The definition of the parameters in the dictionary is done by scope. Parameters defined for the flow have priority over parameters set for the line, overlapping them if there are collision of labels. Parameters can also be set through arithmetic expressions, employing values defined within the dictionary of parameters.

### Resources for flow execution

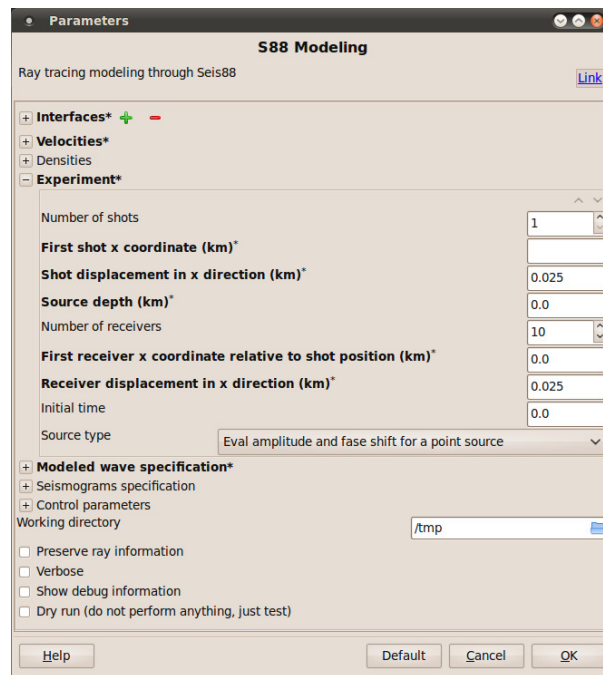
Through GêBR, it is possible to connect to multiple processing servers, allowing the user to submit flows for execution on different machines and to control the progress of each job. A flow may be subjected to immediate execution or queued, being blocked from running until a specific job has been completed. This feature, common in large machines, like clusters, is natively available on any machine through GêBR. Upon a job submission, GêBR itself makes several validations in the generated job to minimize potential problems in the flow setup, alerting the user about them and blocking the submission until the problems have been solved.

A flow can be executed by any of the servers connected at the time it was submitted. However, it is usual to have computational environments composed by many machines, and it is not a trivial problem to select which specific machine will conclude the job execution first. Through an heuristic based on the load and the characteristics of each machine, GêBR can automatically select the best server to execute the job. This server-auto-choose feature allows for a more rational usage of the available computational resources, balancing the load through the machines to attain better response times.

Whenever a processing flow can be splitted into a loop of smaller tasks and all tasks are disjoint, in the sense that the result of one of them does not play any role to the others, GêBR can automatically combine part or all available resources from the registered servers to concurrently execute the tasks. For example, one general migration procedure could be splitted into several common-offset migrations. GêBR can dispatch each common-offset migration to a server, selected automatically based on its load and computational power, as stated before. The response of each common-offset migration is then merged and presented at once to the user, at the job control tab. To accomplish that, the only requirement is that all servers must have access to the same input data, for example through a network-shared file system.

### Technical aspects

The GêBR interface, as well as other softwares developed by the GêBR Project, is developed mainly in C, employing the graphical toolkit GTK+, among many other common libraries, all of them being highly portable. In fact, to be able to provide an interface to a wide range of systems, it is always a concern of the



**Figure 3:** Parameters' dialog box for the program *S88 Modeling*. Note that groups of parameters are defined to encompass parameters which are related to each other. These groups can be collapsed or expanded. Required parameters are displayed in boldface.

project to choose portable libraries. Moreover, all information filled in the interface is stored as XML, a standard format whenever portability is an issue to be considered.

All the produced code by the GêBR Project is released as a free software, under the terms of the GNU General Public License, version 3. This guarantees unrestricted access to the code to everyone, who might be interested.

## DIRECTIONS

The web site of the GêBR project is <http://www.gebrproject.com/>. The project provides support for installation for the most representative linux distributions in the seismic processing field, like CentOS, Red Hat/Fedora, Debian and Ubuntu, through native binary repositories to each one. Detailed instructions are provided at the web site.

The source code of the GêBR interface, DéBR and all other software produced by the project is publicly

	Type	Keyword	Value	Comment
<b>Flow</b>				
	integer	ns	1001	Number of samples
		New		
<b>Line</b>				
	integer	nshot	251	Number of shots
	integer	nrec	69	Number of receivers
	integer	dshot (m)	50	Displacement between adjacent shots (m)
	integer	drec (m)	50	Displacement between adjacent receiver group (m)
	integer	ns	1501	Number of samples
	integer	dt (ms)	4	Sampling interval in milliseconds
	integer	offset min (m)	100	Minimum offset (m)
	real	tmax (s)	6	Maximum simulated time (s)
		New		
<b>Project</b>				

**Figure 4:** Dictionary of parameters.

available at <http://code.google.com/p/gebr/>.

#### ACKNOWLEDGMENTS

I would like to thank Brazilian oil company, Petrobras, and its Research Net in Applied Geophysics for financially supporting this work, to the National Research Council of Brazil (CNPq), and to the sponsors of the Wave Inversion Technology (WIT) Consortium.

#### REFERENCES

- Filpo, E. (2009a). GêBR – the geophysical interface from Brasil. *Boletim da SBGf*, (1):10. (in Portuguese).
- Filpo, E. (2009b). A rescue of GêBRs history. <http://www.gebrproject.com/documentation/historia-da-gebr>. (in Portuguese).
- Stockwell, J. (1999). The CWP/SU: Seismic Un\*x package,. *Computers & Geosciences*, 25(4):415–419.