



HAL
open science

PyRefra – Refraction seismic data treatment and inversion

Hermann Zeyen, Emmanuel Léger

► **To cite this version:**

Hermann Zeyen, Emmanuel Léger. PyRefra – Refraction seismic data treatment and inversion. Computers & Geosciences, 2024, 185, pp.105556. 10.1016/j.cageo.2024.105556 . hal-04468318

HAL Id: hal-04468318

<https://hal.science/hal-04468318>

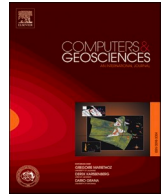
Submitted on 21 Feb 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Open licence - etalab



PyRefra – Refraction seismic data treatment and inversion

Hermann Zeyen^{*}, Emmanuel Léger

Laoratoire GEOPS, Université Paris-Saclay, CNRS, Bât. 504, 91405, Orsay cedex, France

ARTICLE INFO

Keywords:

Refraction seismics
Near-surface
Data processing
Open-source software
Python

ABSTRACT

Open-source software in the geophysical community has been increasingly taking importance since more than a decade. Following this spirit, this study presents an open-source Python software for display, processing, picking of near-surface refraction seismic data and tomographic modeling using some of the main packages already developed for geosciences. The PyRefra package allows the display of different types of gathers, application of different filters and offers manual and several types of automatic picking. We integrated the pyGIMLi manager for tomographic inversion of travel-times. This study describes the code core structure and presents a full field data example in the aim of describing the whole set of tools available in the PyRefra package.

1. Introduction

Since the 50–60s the geophysical methods have been used on various domains, far from the oil-industry aim originally developed for. This is especially the case for seismic refraction (Green, 1974), sensitive to the elastic moduli and density of the sensed media. As these techniques have been developed by the oil-industry sector, they imply large investment for the instruments (geophone and acquisition system) as well as for the maintenance and use of commercial processing softwares. Since these softwares are usually very expensive especially for teaching in public universities, open-source software and codes are good alternatives as they promote sharing between communities. In the last years several open-source Python programs have been published for seismological (Beyreuther et al., 2010; Guedes et al., 2022) and general-purpose geophysical modeling (Rücker et al., 2017). With the exception of Guedes et al. (2022), a versatile, platform-independent program of near-surface refraction seismic data pre- and post-processing is, however, missing. We present here a new multi-platform program that, based on the experience of several years of intense use in teaching and hydro-geophysical applications, integrates a large variety of multi-source multi-receiver refraction seismic data displays, processing and tomographic inversion. Much effort has been put on the ergonomics, giving the possibility to call most functions by mouse or keyboard controls. Main advantages with respect to the program refrapy (Guedes et al., 2022) are that we integrated different filtering options (frequency and velocity), allowing to measure first arrival times even at low signal-to-noise ratios and we integrated several options for quality control of first-arrival time measurements. In addition, the program can

handle three-component data. Also the possibility to correct common field errors has been integrated. In this publication, we present only the most important aspects of the program. The user will find more options and more detailed descriptions in the user manual on our github page (<https://github.com/HZeyen/PyRefra>).

2. Program structure

The code was written in Python version 3.10. However, if tomography inversion is to be done, using pyGIMLi, it is necessary to install it on Python version 3.8. The program was successfully tested and used on Windows10 and Ubuntu 20.04. PyRefra is based on a Graphical User Interface (GUI), supported by the GT5 open source package.

PyRefra depends on classical open-source packages used in scientific development such as Numpy (Van Der Walt et al., 2011), Matplotlib (Hunter, 2007), and Scikit-learn. In addition, part of the program core structure (reading and frequency filtering) is developed on top of the Obspy package (Beyreuther et al., 2010). At last the inversion scheme for tomography uses pyGIMLi library (Rücker et al., 2017). The installation procedure and all necessary dependencies can be found on the github page and can be done through, e.g., the Anaconda environment.

The core structure of the program is depicted in Fig. 1 and can be summarized as follows: (i) field seismic file reading in different formats, (ii) display of gather types such as file, shot, receiver and distance gathers, (iii) first arrival picking and (iv) tomographic inversion using pyGIMLi library (Rücker et al., 2017). In the following, we describe the principal menus and steps being the backbone of the PyRefra workflow and show a real-data example.

^{*} Corresponding author.

E-mail address: hermann.zeyen@universite-paris-saclay.fr (H. Zeyen).

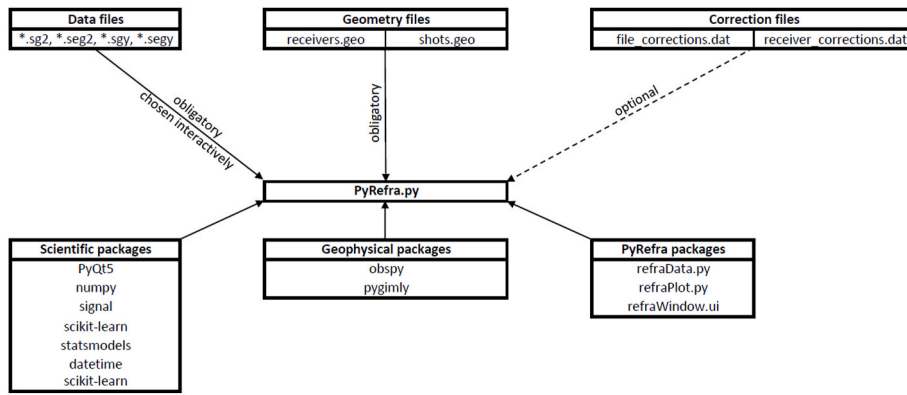


Fig. 1. PyRefra structure. Files and packages necessary for running PyRefra.

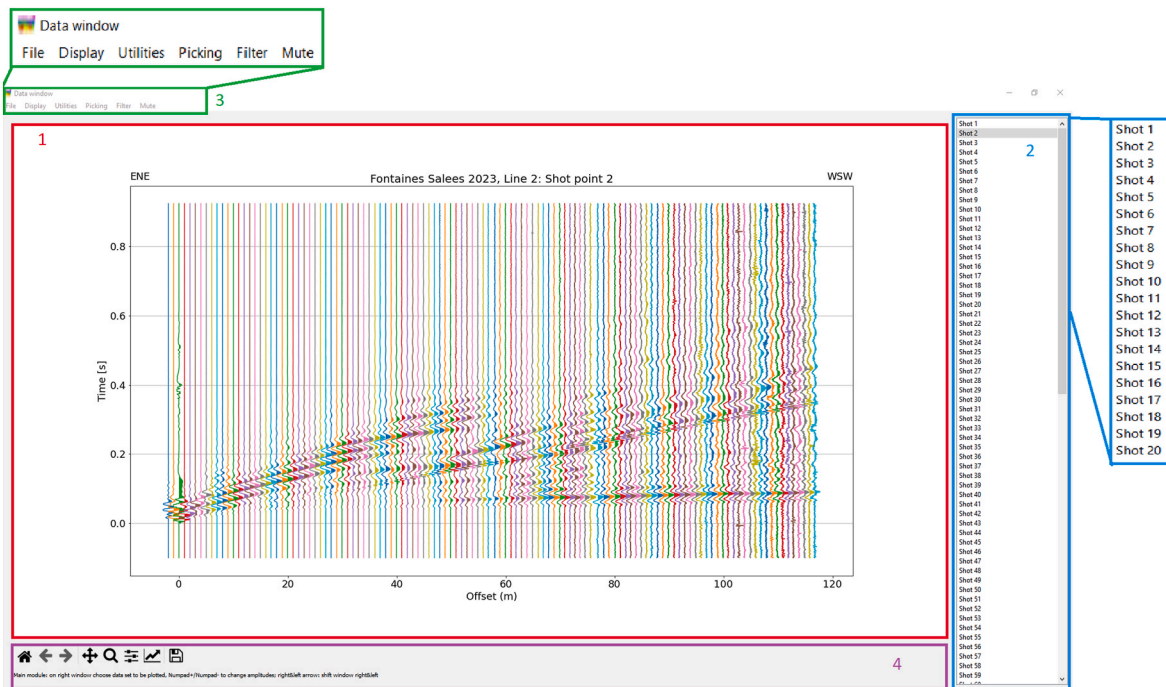


Fig. 2. Example record section shown after reading the data. The window 1 (red frame) displays the time as a function of the offset while the right panel (window 2, blue frame) allows shot selection. The window menus at the top left of the window (window 3, green frame) are detailed in Fig. 3. At the bottom (window 4, violet frame) a context-dependent help text is shown. Panel 2 and 3 are magnified for reading purposes.

2.1. Data preparation

The program allows input of SEG2 and SEG-Y data formats (Northwood et al., 1967), the most common formats used in applied seismics. In order to allow data recorded without geometry information, external geometry files in ASCII format are needed (Fig. 1). The files “shots.geo” and “receivers.geo” contain the shot point numbers and the three space coordinates. The data are supposed to be acquired along a single, roughly straight line, since the program is designed for 2D data processing. The absolute shot point and receiver point coordinates (e.g., UTM) must be transformed and, if necessary, rotated into a local coordinate system such that they follow a line in X or Y direction before storing them in the geometry files. Also the topography values may have to be transformed such that the vertical axis points downwards since pyGIMLI, which is used for tomographic inversion, uses this orientation.

As the code has been built on years of experience we denoted that typical errors which occur during field acquisition are storage of wrong shot point or receiver point numbers in the trace headers or erroneous triggering. We allow the users to correct these errors using file

“file_corrections.dat”, indicating the file number, correct shot point and receiver point numbers and a time shift for the trigger time. It is also possible to give corrections for certain receiver points in file “receiver_corrections.dat”, changing geophone polarity or muting the data of a geophone that was all the time noisy or was only recording electronic noise.

2.2. Working with the program

Once PyRefra is started, the user is asked to choose the data files to be treated. For seismic tomography, usually many shots are executed and eventually, in order to obtain long lines with few geophones, shots are repeated after geophones have been replaced to new positions, resulting in twice (or more) as many files than shot points. The files must have names of the following type: prefixNNN.ext where prefix is any alphanumeric text, NNN a unique integer number (arbitrary number of ciphers) identifying the file, which is used, e.g., in file “file_corrections.dat” and ext is the file extension (sg2, seg2, sgy or segy). All these files may be opened at once. Fig. 2 depicts the GUI once data have been read.

File	Display	Utilities	Picking	Filter	Mute
Choose data files	Original data this screen	P-model	Manual picks	Frequency	Mute/recover trace
Save SEG Y	Original data all shots	Tomography	Amplitude picks	Frequency filter for single trace	Mute air wave
save SU	Shot gather	Envelopes	STA-LTA	Air wave fk	Mute before line
save SEG2	Plot data of one file	False colour	Correlation picks	Velocity filter	Mute after line
save Binary	Receiver gather	Change trace sign	Move picks		
save ASCII	Distance gather	Change sign	Unvertainty change		
save Headers	Zoom	Change colors tomo	Erase all picks		
save plot	Zoom out	Wave animation	Plot all picks		
Quit	Zoom in	Attenuation	Plot calculated times		
	Zoom initial		Store picks		
	Trace normalize		Store Gimli format		
	Time gain				
	Distance gain				
	AGC				

Fig. 3. Full contents of the menus in the menu toolbar (3) represented in a table for clarity.

We added color frames highlighting the four areas of the GUI: the main large graphic window (1), the shot selection panel at the right (2), (3) the menu toolbar at the top and (4) a context-dependent help text integrated in the QT5 icon bar generated by default.

On the main large graphic window (1), each seismogram is plotted with a different color at the position of the offset between receiver and shot point in the x-axis while time is increasing upwards on the y-axis. The first record section shown on the screen corresponds to all traces recorded from the first shot point. The user may then choose on the right panel (2) to plot the data of another shot point or via the menu switch to other gather types such as receiver, distance or file gather. In simple acquisition schemes shot gather and file gather are the same, but they are different for several deployments with repeated shots.

The menu toolbar (3) contains six different sub-sections which are presented in Fig. 3: File, Display, Utilities, Picking, Filter and Mute menus.

The *File* menu allows saving the data in different formats (SEG Y, SU, SEG2, Binary or a specific in-house ASCII format), which is especially useful if one wants to store processed data (correction of field errors, filtered, muted etc.) which then should be used with another library, especially for reflection seismic applications. In addition, it is possible to save header information and plots shown on the screen in png format and finishing the program. The *Display* menu gives the possibility to choose different gather types, to zoom in and out and to use different gains (trace normalizing, time and distance amplitude recovery and Automatic Gain Control). If three-component geophones were used and all components were stored in a common file, the different components

may be displayed independently or altogether. The *Utility* menu contains simple 1D calculation of a seismic P and S velocity model, tracing interactively straight lines for the different phases and tomographic inversion. This option, though not used further in the program, is very useful for teaching purposes, allowing the students to obtain a fast, rough estimate of the velocity variations in depth and along the profile. If arrival times have been measured, it is possible to do a tomographic inversion of the picked travel times (see below, chapter 3.2, for more information). In addition, evolution of maximum amplitudes for chosen phases may be calculated and its logarithm fitted by two straight lines, resulting in a simple two-layer attenuation model. Finally, the program may plot pseudo-profiles of average velocities for each pick as function of offset and midpoint position between source and receiver points, as well as local slownesses calculated as time-difference between neighbouring picks. This option may reveal single wrong picks or problems with single shots or receivers. For teaching purposes, the propagation of the waves in time and space may be visualized in the form of a film, plotting at each sampled time the amplitudes of the traces as a function of offset. The *Picking* menu allows to pick arrival times manually, or by one of the different automatic or semiautomatic methods offered, including correlation with a master trace, STA-LTA (Short Time Average over Long Time Average (Withers et al., 1998),) or an experimental method based on evolution of extreme values. Picks may be shifted manually or deleted and their uncertainties may be modified. The *Filter* menu allows application of different filters. For frequency filtering one may choose frequencies interactively on a plotted spectrum and Obspy routines are used for execution. In addition, fk-filtering is possible (an

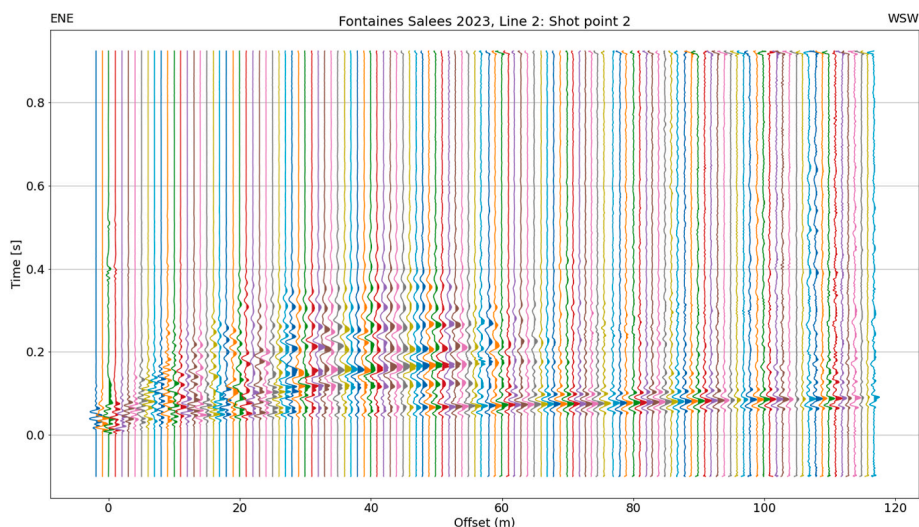


Fig. 4. Data of Fig. 2 filtered with a low-pass of cut-off frequency of 170 Hz and an fk filter eliminating velocities below 800 m/s.

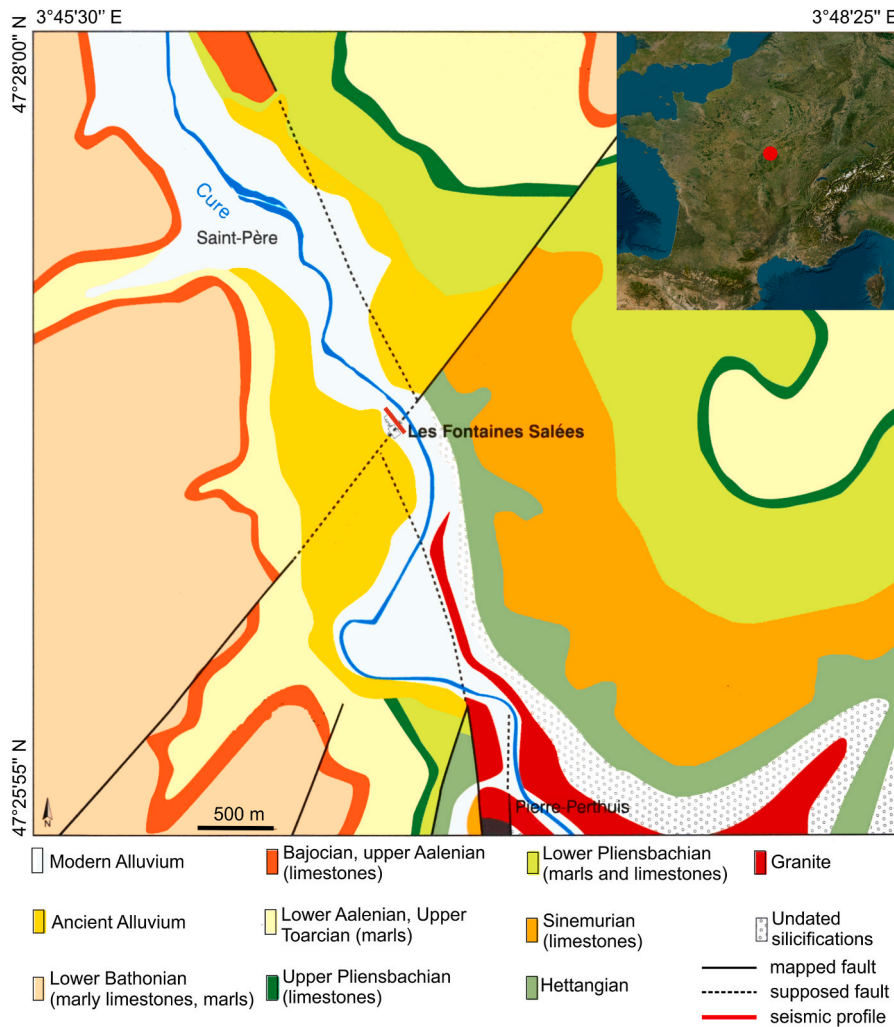


Fig. 5. Simplified geological map of the area. Red line in the center of the map, within the red circle, represents the seismic profile. Modified from <https://commons.wikimedia.org/w/index.php?curid=62546488>.

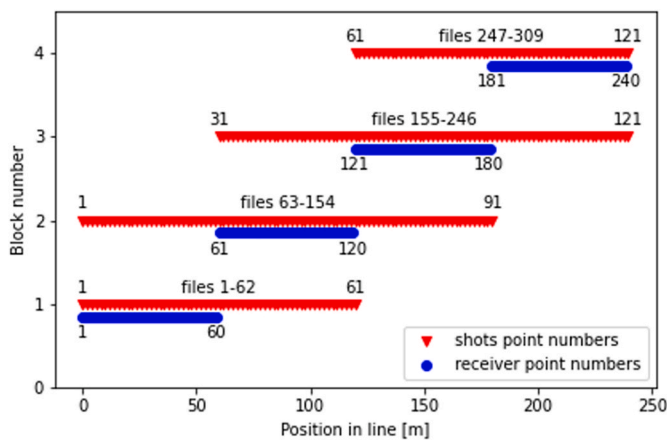


Fig. 6. Acquisition scheme of presented data. Each line is associated with a block number. Blue dots represent receiver positions while red triangles represent shot point positions. Acquisition file numbers are represented above each line, one for each recorded shot.

example is presented in Fig. 4 where data from Fig. 2 are f-k filtered). At last, the *Mute* menu is offering single trace muting or, if muted, recovering, as well as surgical muting of air-waves. Specific phases may be

highlighted by muting the signal before or after a crooked line.

After having described the main core menu and core capabilities of PyRefra, we present now a real-data field example where the picking and tomography capabilities of PyRefra will be illustrated.

3. Example of real field case

A seismic profile has been acquired near the French town of Vézelay to image a supposed normal fault, the main boundary fault between the Morvan massif, which is part of the French Massif Central, and the Paris Basin (Fig. 5) (e.g. Brigaud et al., 2021; Horon et al., 1959). Along this fault, Hercynian granites are juxtaposed with Jurassic marls and limestones. A 240 m long profile with receivers every 1 m and shot points every 2 m was shot and processed with PyRefra. Since only sixty 10-Hz geophones were available, the profile was shot in 4 blocks, shifting the geophones by 60 m and shooting with an overlap of 60 m to both sides (Fig. 6). Using this acquisition scheme, e.g., files 1 and 63, both contain data from shot point 1 but recorded at different geophone positions.

3.1. PyRefra picking procedure

An example of a complete record section for shot point 2 has already been shown in Fig. 2. The seismograms are trace-normalized and show therefore primarily the surface waves. For first-arrival time picking, data were zoomed to the first 80 ms where trace normalization is based only

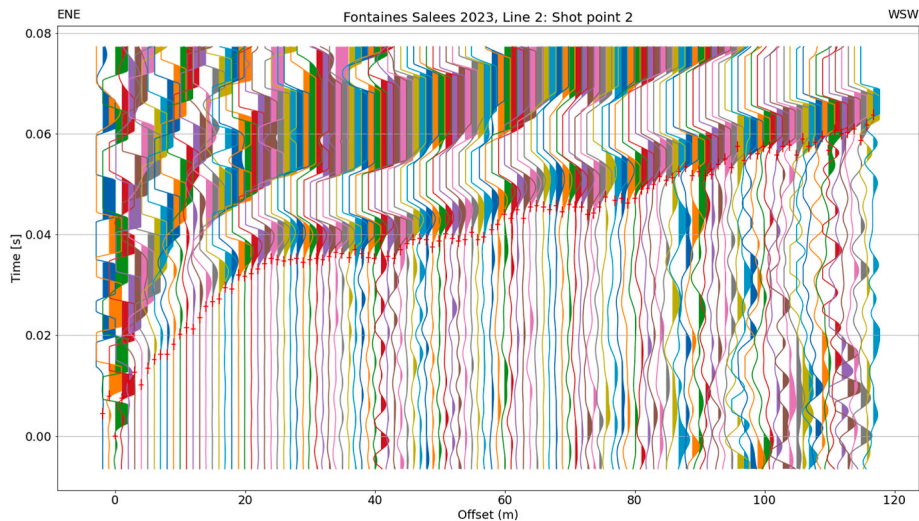


Fig. 7. Zoom on the first 80 ms showing the automatically picked arrival times with red crosses, the vertical extent of which indicate uncertainty bars. Note the variable size of the uncertainty bars depending on signal-to-noise ratio. Data were filtered using a low-pass filter with a cut-off frequency of 170 Hz.

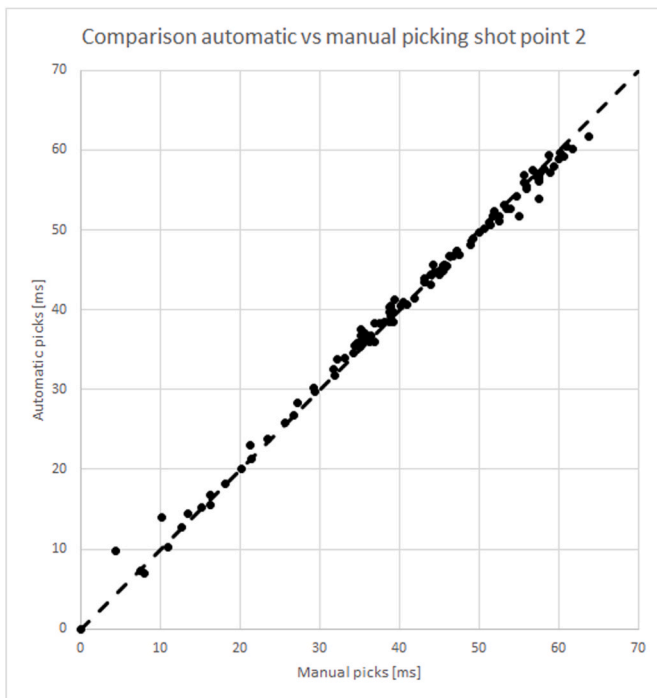


Fig. 8. Comparison of automatically picked arrival times and manually picked ones for the same shot point number 2. Dashed line is 1:1. RMS is 0.8 ms.

on the visible data and therefore amplitudes are increased (Fig. 7). Fig. 7 shows also the result of automatic picking of first arrivals based on a correlation algorithm (menu Picking). For this, first a manual pick is done on a reference trace and correlation between the signal following the pick on the reference trace and the signal of all other traces is calculated. If picks exist from other, nearby shots, the relative maximum of correlation coefficient nearest to the measured arrival time at the same offset from the nearest other shot is used for calculation of the picked arrival. After automatic picking, picks have been shifted manually to a better position wherever necessary. Fig. 8 shows a comparison between the automatically picked times and the times after manual adjustment. The standard deviation was in this case 0.8 ms. Finally, picking uncertainty, that is fixed in the automatic procedure as 2 samples, may be modified manually. These uncertainties are taken into

account by the inversion procedure in pyGIMLi. In addition, the distance gather display allows doing a quality control. Traces are plotted by CMP (Common MidPoint) such that shot receiver combinations for which the waves have traveled the same path in opposite directions are plotted at the same position, slightly displaced to distinguish them (Fig. 9). In this way, it is possible to check whether for those combinations the picked travel times are similar. In total, more than 18,000 arrival times have been picked in this way (Fig. 10).

3.2. PyRefra tomography inversion procedure

The arrival time picks were used in the travel-time tomography inversion module (menu Utilities) to create an image of P-wave velocity distribution down to 40 m depth. When calling the tomography option, the user is asked to provide control parameters for this procedure. The maximum depth of the model to be inverted is proposed as one third of the profile length. Important parameters are the smoothing parameters. The use of these parameters is described in the pyGimli manual (e.g., https://www.pygimli.org/_tutorials_auto/3_inversion/plot_5_Regularization.html). Based on our experience, the dialogue window proposes as default value a general smoothing parameter of 200. Clearly smaller values will result in quite rough models (high lateral variability of the velocities), clearly larger values will result in over-smoothed models. PyGimli allows reducing this parameter from one iteration to the next, so that in the beginning, the inversion process is stabilized by smoothing the model relatively strongly, but in later iterations, when the data adjustment becomes better, parameters are allowed changing more freely. The evolution of the smoothing parameter is controlled by the smoothing reduction per iteration. Finally, in z-direction, the smoothing may be reduced to allow velocity jumps at layer limits, which is controlled by “Smoothing in z-direction”. These smoothing parameters may depend on the measured data and the user should do several inversion runs with different values in order to find out the best combinations. This is also true for the initial model which is defined as a 1D model with a constant velocity gradient, giving the initial velocity at the top and at the bottom of the model. For the presented inversion, the default smoothing parameters were used. Once the tomographic inversion has finished, the results are shown on a screen with seven subfigures (Fig. 11): The upper half of the screen (Fig. 11A) is occupied by the resulting P-wave velocity distribution. The profile is plotted with an aspect ratio of 1:1. Color scale is user-controlled and no colors are plotted where no ray passed during the whole inversion process. For comparison, the starting model is shown in Fig. 11B with the same color

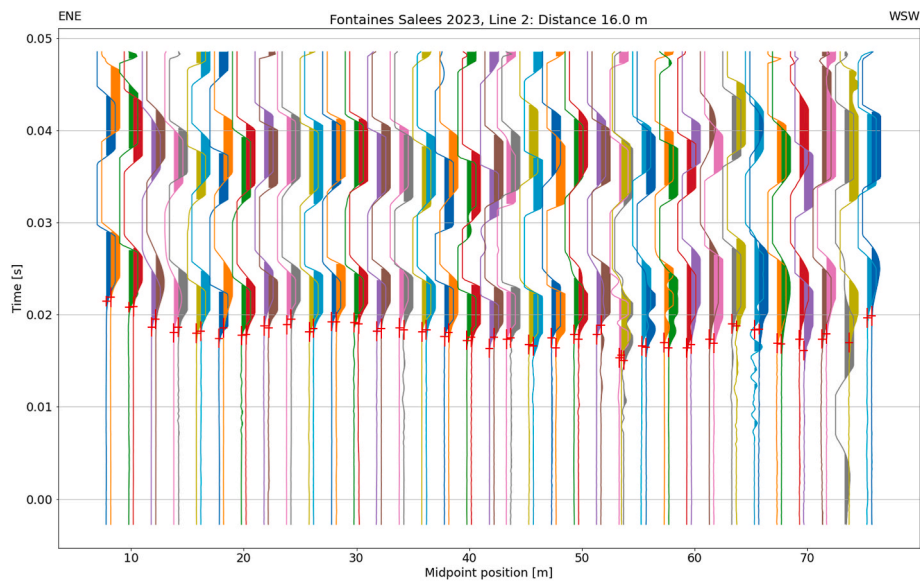


Fig. 9. Distance gather with picks for shot-receiver combinations at 16 m offset. For better visibility, common midpoints between 8 and 76 m have been zoomed in.

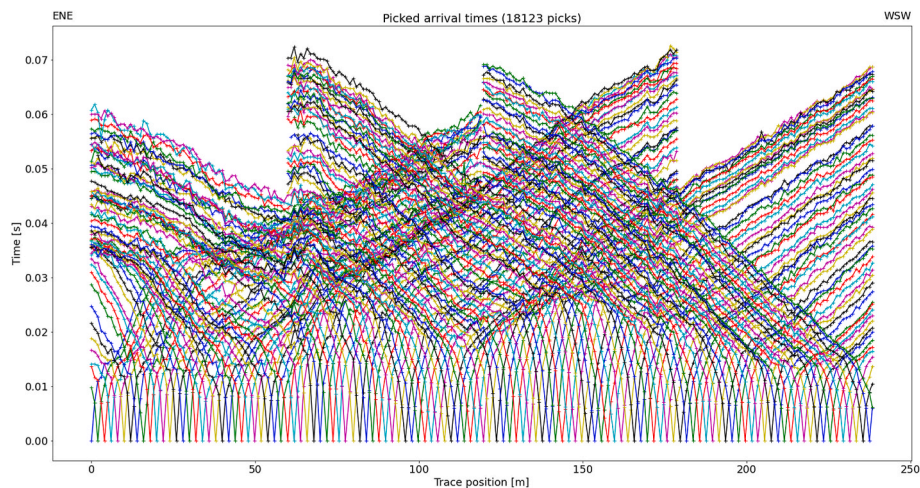


Fig. 10. Plot of all picked arrival times vs position for all shot points along the profile representing 18,123 picks.

scale as in Fig. 11A. The starting model is created on the run as a simple 1D model with user-defined velocities at the top and at the bottom of the model and a linear gradient between them. Fig. 11C shows a measure of coverage of the rays as the sum of ray lengths having crossed each cell of the model, cumulated over the different inversion steps. In addition, the rays for the final model are plotted on top of the coverage map. Figs. 11D and E shows measured travel times and the misfit for each measured combination of shot and receiver. The axes indicate positions of receiver points (x-axis) and of the shot points (y-axis). For these plots, PyGimli routines are used. Fig. 11F gives the evolution of the χ^2 weighted misfit. Usually, the χ^2 decreases drastically during the first iteration, and slowly lowers afterwards. Finally, Fig. 11G gives another measure of misfit, showing the average misfit for each receiver point and each shot point as a function of their respective positions. This graphic may be useful to see if a geophone or a shot point had problems (wrong position, wrong time break ...). Once the tomographic inversion is done PyRefra allows to display the calculated arrival times on the seismogram, illustrated, as an example, on Fig. 12, showing the same data as Fig. 7 including the calculated arrival times for comparison.

The velocity model (Fig. 11A) shows in the uppermost 2–8 m very low velocities corresponding the vadose zone, underlain by an

intermediate-velocity layer (2.5–3 km/s) in the W, corresponding to the Jurassic sediments, whereas in the E, high velocities (>4 km/s) indicate presence of altered basement rocks. The transition from basement to sediments is located at 70 m and seems to be about vertical.

4. Conclusion

We presented a versatile open-source multi-platform Python program to plot and treat multi-shot, multi-receiver near-surface refraction seismic data: PyRefra. Different types of gathers may be plotted (shots, receivers, distances) and different types of gains may be applied. Data treatment includes frequency and velocity filtering, different possibilities of selective muting and different ways of manual and (semi-) automatic picking. Picked data may be inverted to obtain a 2D tomographic image of the underground using the full capability of pyGIMLI for tomographic inversion. Experience shows that the program is easily manageable by students and allows fast interpretation of seismic profiles. The PyRefra software is in continuous development. As such, menus and modules may be added/modified in the near future.

Fontaines Salees 2023, Line 2

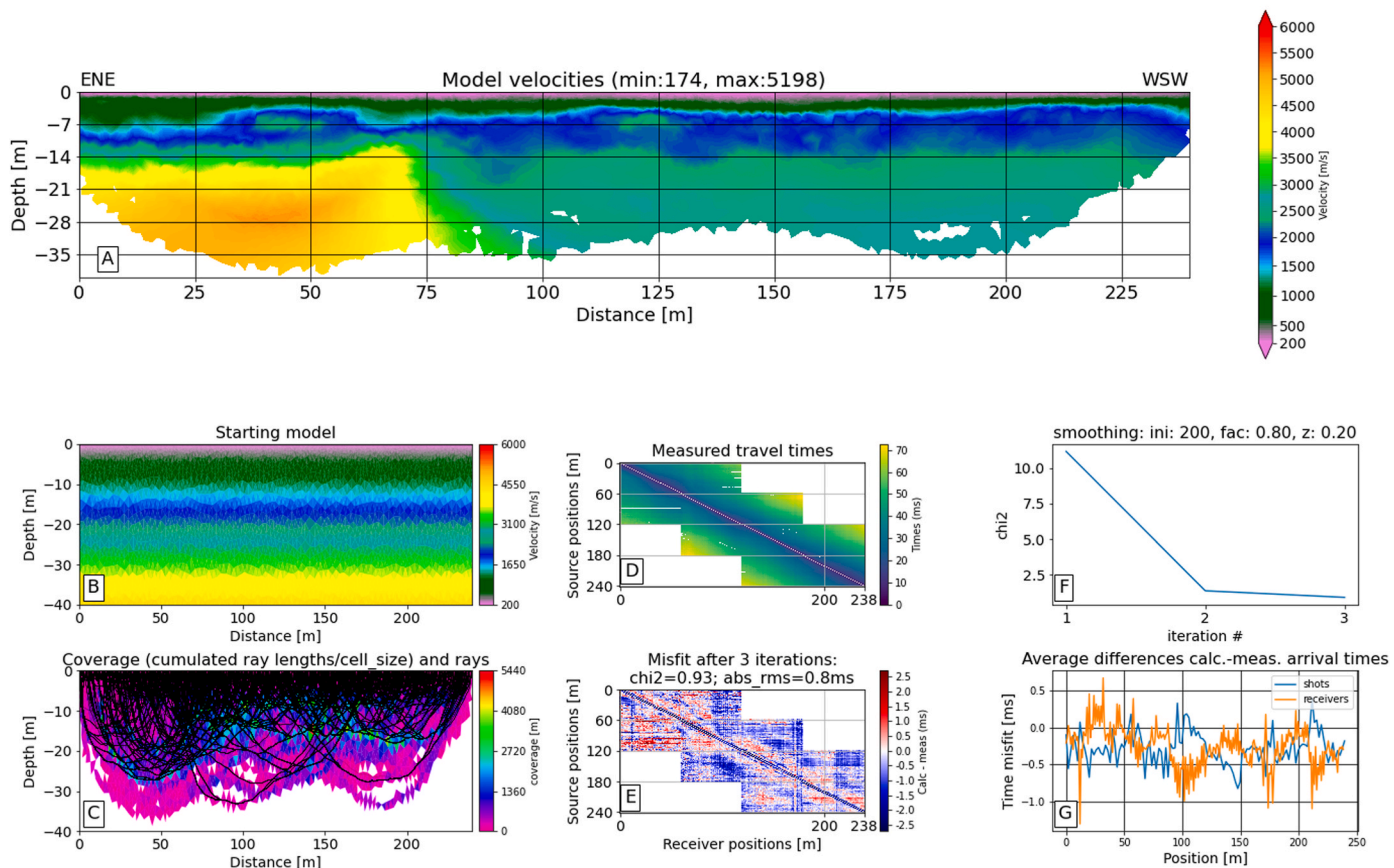


Fig. 11. Inversion result window shown once tomographic inversion is finished. A: Distribution of P-wave velocities. B: Starting model. C: Measure of data coverage (summed length of rays crossing every cell) and ray distribution of final model in black lines. D: Measured arrival times as function of receiver position (x-axis) and shot point position (y-axis). E: Misfit between calculated and measured arrival times. F: Evolution of χ^2 measure of the misfit as function of iteration number. G: Average misfit for all shots and all receivers as a function of their positions.

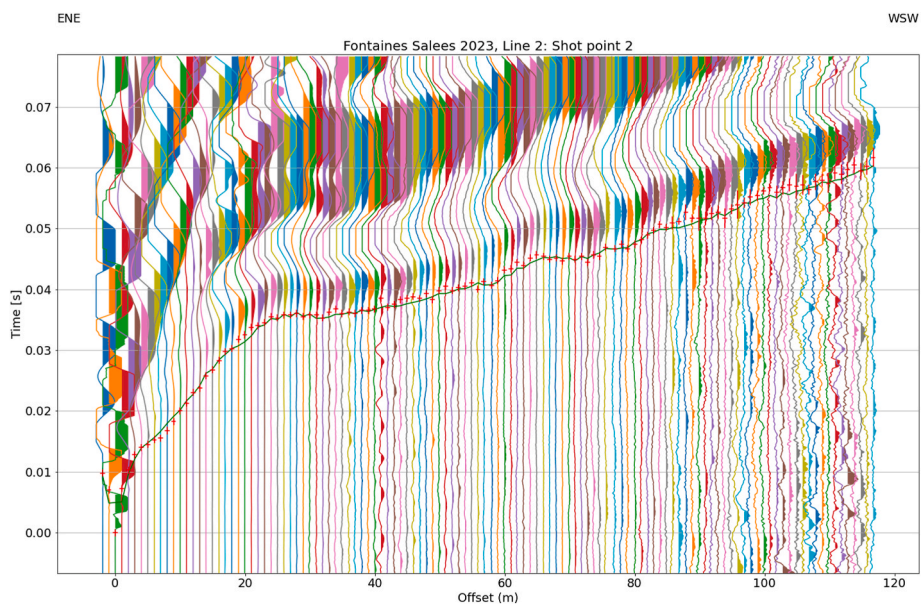


Fig. 12. Unfiltered data of shot point 2 zoomed to the first 80 ms showing the measured arrival times with red crosses, and the arrival times calculated from the final inverted model (Fig. 11) as a green thin continuous line.

5. Computer code availability

The program may be downloaded at <https://github.com/HZeyen/PyRefra>. It is distributed under the terms of the GNU General Public Licence Version 3. doi: [10.5281/zenodo.8004629](https://doi.org/10.5281/zenodo.8004629).

ORCID iD authorship contribution statement

Hermann Zeyen: Conceptualization, Software, Writing – original draft, Writing – review & editing. **Emmanuel Léger:** Software, Validation, Writing – original draft, Writing – review & editing.

Declaration of competing interest

The authors declare no conflict of interest whatsoever.

Data availability

Data will be made available on request.

Acknowledgements

We acknowledge the different bachelor and master students having discovered bugs and instabilities of the program and the three different promotions (2020–2022) of Earth science bachelor students testing the program during the multi-disciplinary field course, directed by Pr. Benjamin Brigaud. We acknowledge also the thoughtful and very helpful

remarks of an anonymous reviewer.

References

- Beyreuther, M., Barsch, R., Krischer, L., Megies, T., Behr, Y., Wassermann, J., 2010. ObsPy: a Python toolbox for seismology. *Seismol. Res. Lett.* 81 (3), 530–533.
- Brigaud, B., Barbarand, J., Battani, A., Benedicto, A., Blaise, T., Calmels, D., Durand, V., Léger, E., Lenoir, L., Pallares, C., Saintenoy, A., 2021. Origine des eaux salées, des émanations gazeuses azotées (+ Hélium) et des minéralisations de fluorine à l'extrémité nord de la Grande Faille du Bazois: Apport du couplage entre pétrologie, hydrogéologie, géochimie et géophysique. In: 27e édition de la Réunion des Sciences de la Terre. November 2021.
- Green, R., 1974. The seismic refraction method—a review. *Geoexploration* 12 (4), 259–284.
- Guedes, V.J.C.B., Maciel, S.T.R., Rocha, M.P., 2022. Refrapy: a Python program for seismic refraction data analysis. *Comput. Geosci.* 159, 105020 <https://doi.org/10.1016/j.cageo.2021.105020>.
- Horon, O., Mégnien, C., Soyer, R., 1959. Note sur les fontaines salées de Saint-Pere-sous-Vezelay (Yonne). *Bull. Soc. Geol. Fr.* S7-I (5), 461–466. <https://doi.org/10.2113/gssgfbull.S7-I.5.461>.
- Hunter, J.D., 2007. Matplotlib: a 2D graphics environment. *Comput. Sci. Eng.* 9 (3), 90–95. <https://doi.org/10.1109/MCSE.2007.55>.
- Northwood, E.J., Weisinger, R.C., Bradley, J.J., 1967. Recommended standards for digital tape formats. *Geophysics* 32 (6), 1073–1084.
- Rücker, C., Günther, T., Wagner, F.M., 2017. pyGIMLI: an open-source library for modelling and inversion in geophysics. *Comput. Geosci.* 109, 106–123. <https://doi.org/10.1016/j.cageo.2017.07.011>.
- van der Walt, S., Colbert, S.C., Varoquaux, G., 2011. The NumPy array: a structure for efficient numerical computation. *Comput. Sci. Eng.* 13 (2), 22–30. <https://doi.org/10.1109/MCSE.2011.37>.
- Withers, M., Aster, R., Young, C., Beiriger, J., Harris, M., Moore, S., Trujillo, J., 1998. A Comparison of select trigger algorithms for automated global seismic phase and event detection. *Bull. Seismol. Soc. Am.* 88 (1), 95–106.