REFRAMODEL

Program for trial-and-error modelling and inversion of refraction seismic data

User manual

Introduction

This program was mainly written for educational (teaching) purposes, although it may be also used in research. The main idea behind is that students should have the possibility to model manually refraction seismic data building and modifying a geological model interactively on the screen until a satisfactorily good fit is attained in a (mostly short) imposed time. In this way, they may get a grip on how arrival times vary with varying velocity distribution and thus varying geology, which would not be the case if they would just run a black-box inversion. In addition, standard inversions of seismic travel times result in rather smooth gridded models where it is mostly difficult to determine the position of sharp geological interfaces. Using Pygimli's option of building regionalized models and regularize independently each region, RefraModel allows running inversions that create models with smooth velocity variations inside a region (a geological body), but sharp velocity jumps across body boundaries. Many inversion parameters are controlled by the user, giving students also the possibility to better understand inversion procedures and the influence of controlling parameters. In research, it may be a good procedure to run first an unconstrained, gradient-based inversion, build geological bodies on top of the resulting smooth model, optimize them manually and run finally a region-based inversion to get models with sharp velocity contrasts.

Travel times should be first measured using another program, e.g., Pyrefra (https://github.com/HZeyen/pyrefra; Zeyen and Léger, 2024) and be stored in Gimli *.sgt format.

Installation

Files may be found at https://github.com/HZeyen/reframodel

1. Download and install **Anaconda Individual Edition or Miniconda**
2. Push the Windows key and search **Anaconda Prompt.** This opens a command window. Then type the following commands:

   ```
   conda config --add channels gimli --add channels conda-forge
   ```

   « channels » is the place where to find the source code, here « gimli » and "conda-forge"

   Pygimli is often not up to date concerning the last Python version. In this case, you are obliged to install a specific environment to be able to run `reframodel`. To simplify the installation manual, we suppose that this is the case. Create first the environment "pg" (pg is any name you want to give to the environment. If you want to call it, e.g., refra, change pg to refra in the following two commands):

   ```
   conda create -n pg pygimli
   ```

   Change focus to this new environment:

   ```
   conda activate pg
   ```

   Install reframodel together with the necessary libraries:

   ```
   pip install reframodel
   ```

   Under Linux, it may be necessary to use (at least in the Linux emulation of Windows11):

   ```
   pipx install reframodel
   ```

   For manual install, you need to install the following libraries in addition to pygimli which should all be installed automatically by Anaconda but not necessarily in Miniconda (spyder being optional):

   ```
   conda install os sys shutil pathlib pyqt5 numpy matplotlib datetime spyder
   ```

Starting the program

There are two ways to run the program:

- If you installed it via pip, you may open a command window (in Windows, use Anaconda Prompt) change directory (cd) to the one you want as working directory (in general, where your model file and picked data file are – if they exist – or should be stored) and type reframodel followed by ENTER.
- You may also pull it from https://github.com/HZeyen/reframodel and start it from Visual Studio Code or Spyder. In this case, you must define in Visual Studio Code or Spyder the environment "pg". Then open file start_reframodel.py, find the code line starting with "dir0 =" near the beginning of the file (while I am writing this manual, it is line 12) and define there your working directory. The definition of the working directory is not obligatory, since the user will be asked to choose an existing model file and the working directory will be set to the folder of this file, however, it is useful if you work several days on the same data set or if you start a completely new model.

Usually, you will work with existing first arrival times picked using another program. These picks should be stored in a file called "picks.sgt" using the gimli format (https://www.pygimli.org/gimliapi/classGIMLI_1_1DataContainer.html#a47be16cf891e88f92684d9cbab122b60). This file may be produced within program pyrefra. If the program does not find this file, it will ask the user whether it should be loaded from somewhere else (another file name and/or another folder). If then the answer is "No", the program supposes that you want to calculate a synthetic model and ask you interactively for the positions of shot and receiver points (see later). Then, the program asks for an existing model that should be read in. If a model is given, the user is asked for two parameters (Fig. 1a):

Threshold for point equality:

When modifying interactively the model, i.e. moving edges or creating new bodies, two points are considered to be equal if they are nearer than (X_max-X_min)*Threshold/100 in x direction and (Y_max-Y_min)*Threshold/10 in y direction (X_min, X_max, Y_min, Y_max are the model limits in horizontal and vertical direction). In this case, they are joined into one single point. This is also true for the limits. If the x coordinate of a point is closer to X_min or to X_max than this threshold, the point is moved to the corresponding vertical limit, if it is nearer to Y_max, it is moved to the horizontal lower model limit. Similar, if it is nearer than this threshold to the topography, it will be moved to the interpolated topography at the x coordinate of the point. On the plot (see below), the body nodes are plotted as ellipse having a dimension of +/- the threshold in both directions. The ellipse has thus the same aspect ratio as the plotted model.

Save model to file:

Each time the model is modified, the actual model will be stored in the file given, inside the working directory. By default, the name of this file is equal to the one of the read-in model so that the modifications are stored in the same file and the initial model is overwritten. However, at the start of the program, the read-in model is saved in file "initial_model_including_extension".bak. The format of the file is as follows (all fields are coma separated):

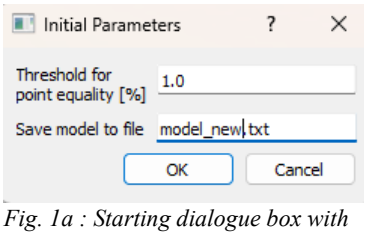

Fig. 1a : Starting dialogue box with

One file header line with the following information:

Number of bodies, names of the physical properties stored for every body (see below)

For every body:

One header line containing the number of corner points, for every physical property its value for this body (velocities in m/s) and the name of the body.

Followed by one line for every corner point with x and y coordinate in meters. The last point is equal to the first one.

If no model is chosen (click Cancel when asked to choose a model file), in addition to the above-mentioned parameters, the limits of a new model must be given, as well as its properties (Fig. 1b).

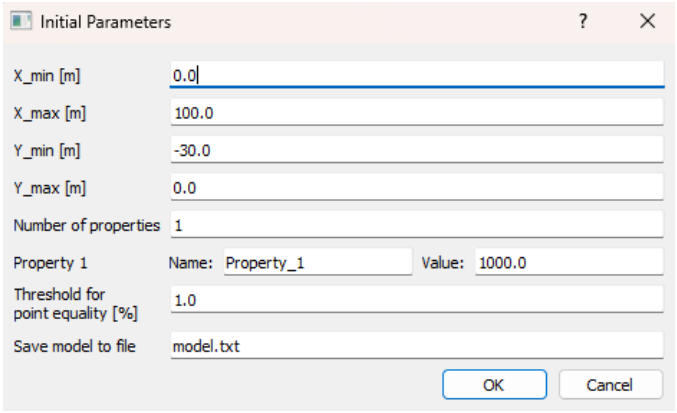

Fig. 1b : Starting dialogue box without existing model

X_min, X_max, Y_min, Y_max:

Dimensions of the model. Y coordinates are positive upwards, i.e., in general, the model y coordinates will be negative. If a pick file has been found, the values proposed for the minimum and maximum X coordinates correspond to the extrema of the shot and receiver coordinates and in Y direction, the maximum coordinate is equal to the rounded maximum topography of the receiver and shot points. The proposed minimum y coordinate (bottom of the model) is calculated as the negative of 30% of the length of the line. If no picks file is given, the user should give these coordinates manually (by default 0 and 100m in x, -30 and 0 m in y.

If a picks file has been read but no model file is given, the program checks whether all receivers/shots are at the same height, in which case Y_max is by default 0. If however, the geometry has variable topography ("z" in *.sgt file), first all z values are reduced such that the highest topography is at 0. Y_max is then again by default 0, whereas the proposed Y_min is calculated with respect to the lowest topography point. The upper limit of the model is the defined by the reduced topography of the shots/receivers.

Number of properties:

In the actual version of the program, the number of properties should be one, i.e. the velocity of a body in m/s. In the future it may become more, adding, e.g., the vertical gradient or P- and S-wave velocities. For each property its name (user-defined text) and initial value must be entered.

If no picks file was given, the program asks then for a virtual **acquisition geometry**. For receiver points and for shot points, the following values are needed (Fig. 2):

X-coordinate of first receiver/shot [m]:
Position of the first point. Y-coordinate (topography) is supposed to be zero everywhere.

Distance between receivers/shots [m]:
Station spacing

Number of receivers/shot points:
By default, this value is proposed as int(length_of_the_model/spacing)+1 but may be modified.

Store calculated travel times to file picks.sgt:
If this box is checked, calculated arrival times of the forward model are stored automatically in Gimli's *.sgt format and may later be used as synthetic arrival times. This will be done until the program is stopped, i.e. the travel times of the last calculated forward model will be stored in file picks.sgt. If not, only the geometry is used (stored in a file tmp.sgt) and travel times are all set to zero with an uncertainty ("err") of 0.5 ms.

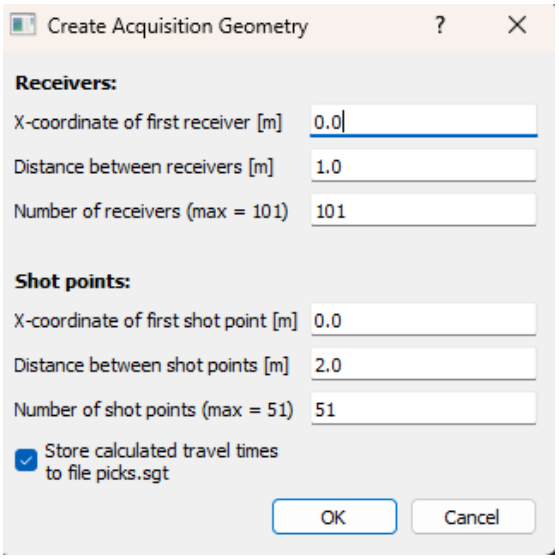

Fig. 2 : Dialogue box for definition of physical

If the user is working with two screens, the main window is then opened on the second screen which is supposed to be in general larger than a laptop screen. If only one screen is present, this is evidently no question. The read or newly defined model is then plotted into the lower part of the window and read arrival times or, if no picks file has been given, the times calculated from the starting model are plotted into the upper part of the window so that the picks of one shot are plotted with the same color, different colors for different shots (Fig. 3).

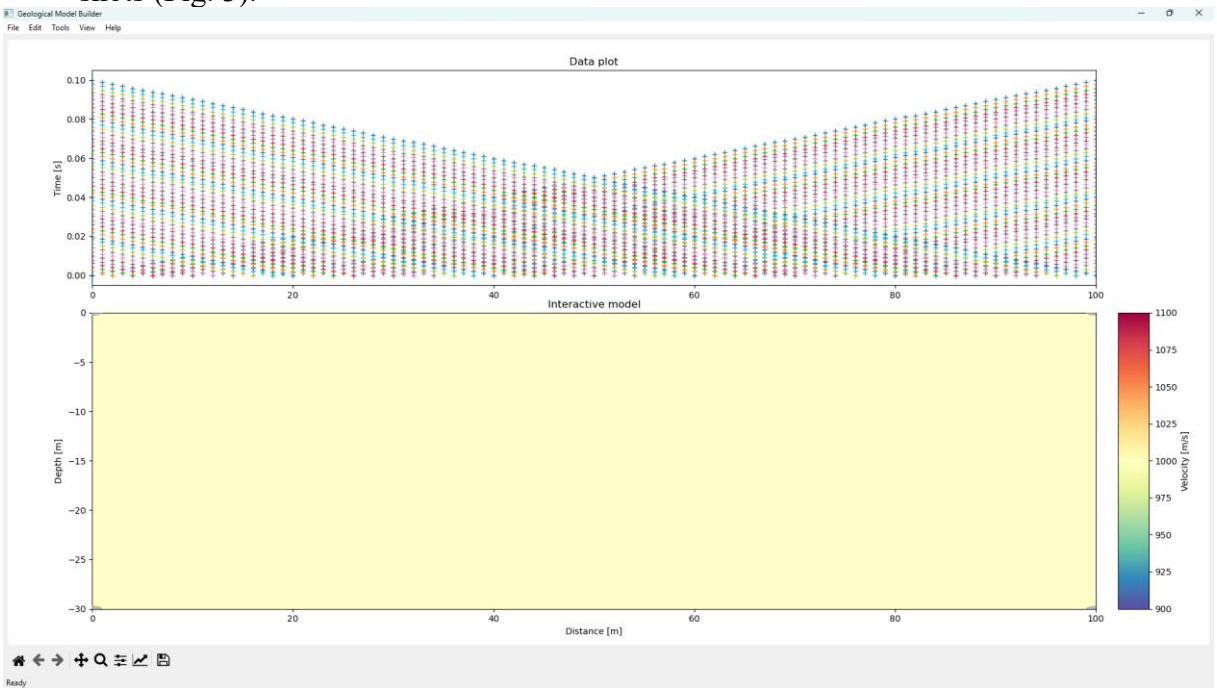

Fig. 3a: Plot showing the starting screen for the case where no picks file and no model were given. The top part shows synthetic picks calculated for a user-defined velocity of 1000 m/s and the bottom part shows the starting model.

Fig. 3b: Plot showing the starting screen if picks and model are defined: All measured arrival times (top) and the actual model (bottom).

Menu options:

File

Contains a few general menu items.

Save model as (keyboard shortcut: CTRL+S)
Allows saving the actual model to a different file than given in the starting dialogue.

Exit (keyboard shortcut: CTRL+Q)
Finishes the program asking first for confirmation. The actual model is stored in the same file as the one opened at the beginning. If no file name was given, it is stored as "model.txt".

Edit

Contains procedures for editing interactively the model. Choosing one of these items starts the corresponding module. All three allow doing several modifications of the same type and must be finished be pressing the keyboard ENTER (confirmation of all modifications) or ESC (annulation of all modifications done since calling the module). A message on the screen recalls this condition. Each time ENTER is pressed, the modified model is stored into the file defined in the dialogue box of Fig. 1.

Body Editor (keyboard shortcut: B)
Allows changing entire bodies in two ways:

- Left mouse pressed and pulling a line splits one or more bodies into two parts. Moving the mouse with the CTRL key pressed produces horizontal or vertical limits, depending on the principal movement direction. The line must cross the body entirely. If several bodies are crossed entirely, all these bodies are split along the drawn line. The two parts of a split body have the same properties, only the names are changed appending "_split". Therefore, after finishing this module

pressing ENTER, the Property Editor should be used to set properties and names of the two partial bodies to the wanted values.

- Pressing the right mouse inside one body followed by a click into a neighboring body joins both bodies. The joint body has the properties of the one clicked first. The bodies must have a common edge, not just a common node.

Node Editor (keyboard shortcut: N)

Nodes may be modified in three ways:

- Left click near a node and pulling it, changes its position for all bodies containing this node. Moving the mouse with the CTRL key pressed moves the point horizontally or vertically, depending on the principal mouse movement direction. It is not allowed to cross with a node another edge (there is no protection, but the result is unpredictable). Also, the point should not be placed on another line segment of the same body (i.e. pass by a neighboring node). If the final position of the node is near a model border within the threshold, its position is shifted horizontally or vertically to the nearest border
- Left click followed by keyboard DEL eliminates a node. This is only allowed for nodes belonging to one or two bodies. If a node is part of three or more bodies, it is impossible to predict automatically how to reconnect the bodies after deletion. Therefore, an attempt to eliminate such a node produces a warning saying that the node cannot be eliminated. In such a case, the user may first create a dummy body by splitting one of the three bodies and join it with another of the bodies concerned such that the node to be eliminated is part of only two bodies. Then one can eliminate the node and move the modified body limit to the correct place (see below under hints)
- Right click near an edge adds a node to this edge. This node is placed onto the edge at the nearest point to the mouse position (nearest in screen coordinates, not in axis coordinates, which may be different if the vertical and horizontal scales are different). It may then be shifted with a left click to the desired position.

Property Editor (keyboard shortcut: P)

After clicking into a body, a dialogue box opens asking for new values of the body properties.

Tools

Select picks (keyboard shortcut: F4)

With this option, it is possible to reduce the number of picked arrival times shown on the screen and used for forward or inverse calculation. A dialogue box opens (Fig. 4):

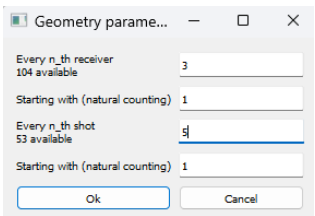

Fig. 4 : Dialogue box for definition of data to be shown

The user may choose to select one out of n receivers and shots and may choose the first receiver and shot to show.

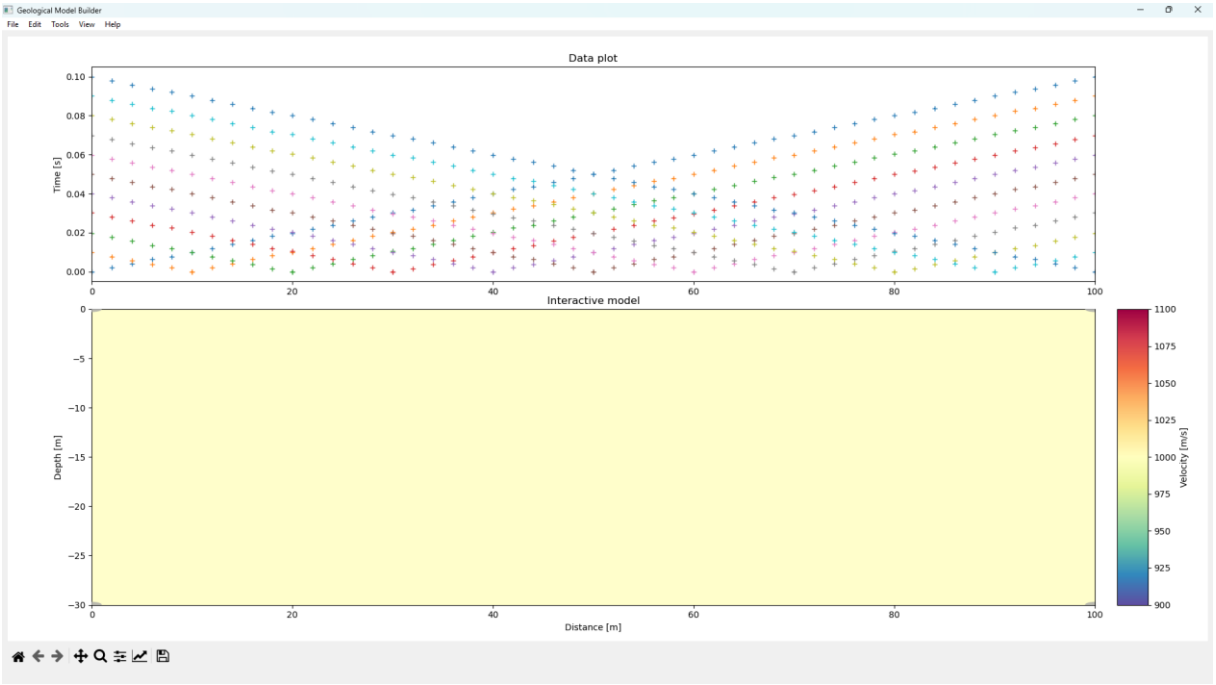

Fig. 5: The same as Fig. 3a but showing only one shot out of five and one receiver out of two, starting with the first shot and the first receiver.

Run Forward Model (keyboard shortcut: F5)

Do forward calculation to obtain the synthetic travel times of the actual model. After calculation of the forward problem, the calculated travel times are plotted as continuous lines on top of the measured ones ("+") and the misfit is written on top of the plot (Fig. 6):

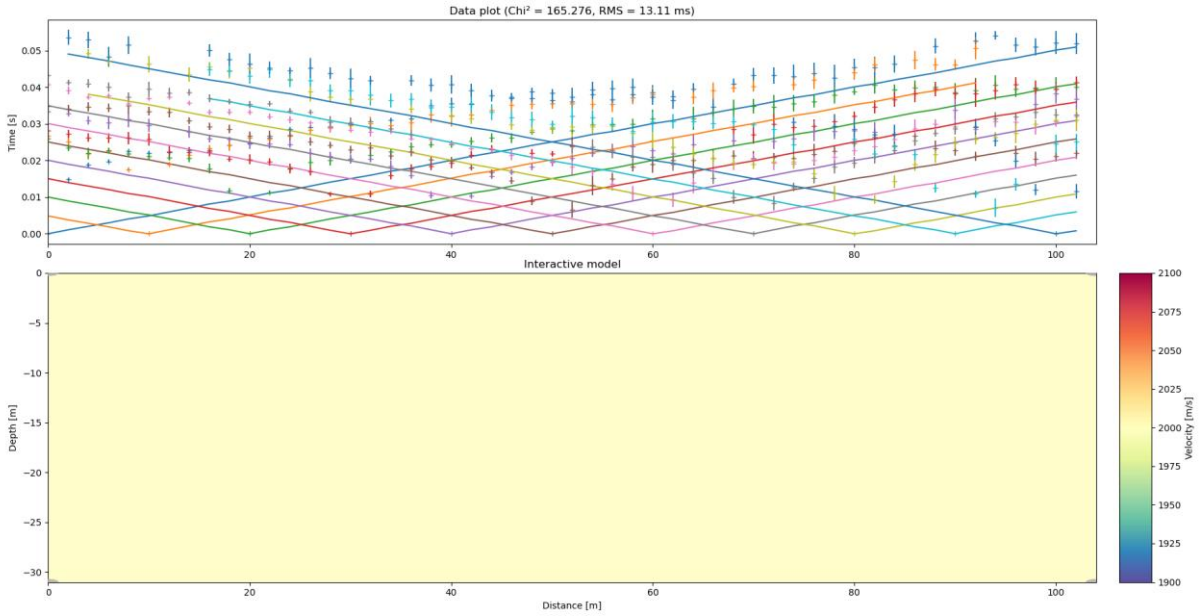

Fig. 6: Figure after calculation of synthetic travel times.

The resulting travel times are stored in the working folder in file "forward_picks.sgt" and the plot is saved as "forward_model_plot.png".

Inversion parameters (keyboard shortcut: F8)

Before doing an inversion, a number of inversion parameters must be set. The following dialogue box opens asking for general inversion parameters (Fig. 7).

One may use the actual interactively defined model or a standard pygimli model with a vertical velocity gradient. In the second case, the dialogue box is updated and asks for the velocity at the top of the model (topography) and at the bottom.
Then follow fields to define rules for stopping the inversion: Usually, the inversion stops if chi2 <= 1.0. This condition may be annulated by unhooking the corresponding box. The inversion may also stop if between two iterations, the misfit varies only a little bit. The standard value is a reduction of less than 2% of the relative rms misfit. Finally, one may restrict the number of iterations. If a value of zero is given, the iterations continue until one of the former conditions is fulfilled.

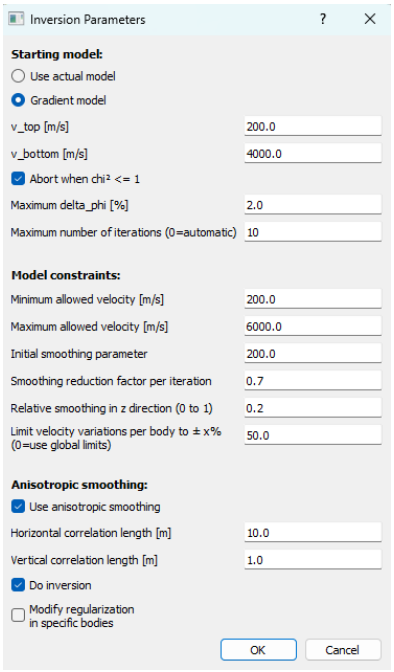

Fig. 7: Dialogue box for general inversion parameters, with all fields represented.

In addition, the velocities may be constrained between a minimum and a maximum value. If both numbers are the same, the constraint is not used.
Finally, the inversion may be regularized. Usually, one starts with a strong regularization which is reduced during the iterations, when the misfit becomes smaller. The initial smoothing value is given. This value is then multiplied by the "Smoothing reduction factor" after each iteration. The regularization factor may be different in horizontal and vertical direction. The "Relative smoothing in z direction" value determines the regularization factor in vertical with respect to the horizontal direction (zWeight = 0: no vertical smoothing, zWeight = 1: Same vertical as horizontal smoothing).
It is furthermore possible to impose structural anisotropy in the inversion giving different correlation lengths in horizontal and vertical direction.
Usually, one will want to run the inversion immediately after defining the parameters, but if this is not wanted, on an uncheck the corresponding checkbox.

If a vertical-gradient starting model is chosen, the inversion parameters are now all defined and the module is finished.

If the actual model is used as starting model, it is now possible to define specific constraints to each body if the corresponding box is checked. To define body-specific constraints, make a left click into the desired body. (TODO: You may also make a right click anywhere in the model to apply the same constraints to all bodies.) Another dialogue box opens with the following options:

- Type of constraint: The user may choose between
  - Free inversion (no specific constraint).
  - Single velocity (the optimum velocity is searched being constant in the whole body).
  - Velocity range (minimum and maximum velocity allowed for this body). If this option is chosen, the two necessary fields are activated.
  - Fixed velocity (the velocity in this body is not inverted but kept fixed and constant for the whole

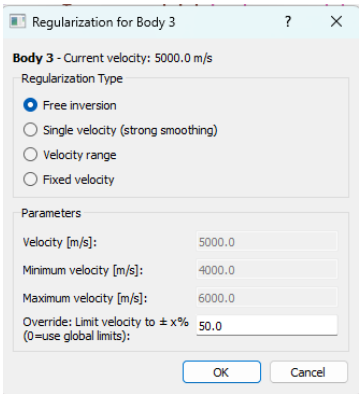

Fig. 8: Dialogue box for setting body-specific constraints

body). If this option is chosen, the corresponding field for the desired velocity is activated.

- Finally, it is possible to change the global allowed velocity range for the clicked body. This is done by giving a non-zero value to "limit velocity to +/- x%". In this case, final velocities will not be outside the range (v_forward +/- x%)

When all desired constraints are defined, finish the inversion parameter definition pressing ENTER. If the menu item "Inversion parameters" is clicked again, all body-specific regularizations are reset to "free inversion".

If "Do Inversion" had been checked in the main dialogue box, the inversion procedure will now be automatically started.

Run Inversion (keyboard shortcut: F9)

Pressing this button starts the iterative inversion, but only if the inversion parameters have been set at least once before. When finished, the chi^2 evolution is shown on the screen (Fig. 9) so the user may decide whether more iterations should be done. **Actually, it is not possible to simply continue the iterations, the inversion must be restarted changing the number of iterations and/or other parameters.**

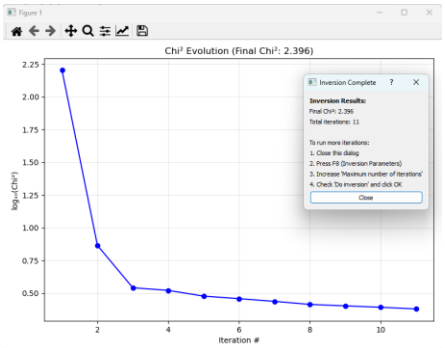

Fig. 9: Evolution of chi^2 and message concerning continuation of iterations.

In the main window, the resulting model is shown, overlain by the actual model bodies.
Figure 10 shows the result after starting with a standard vertical gradient model.

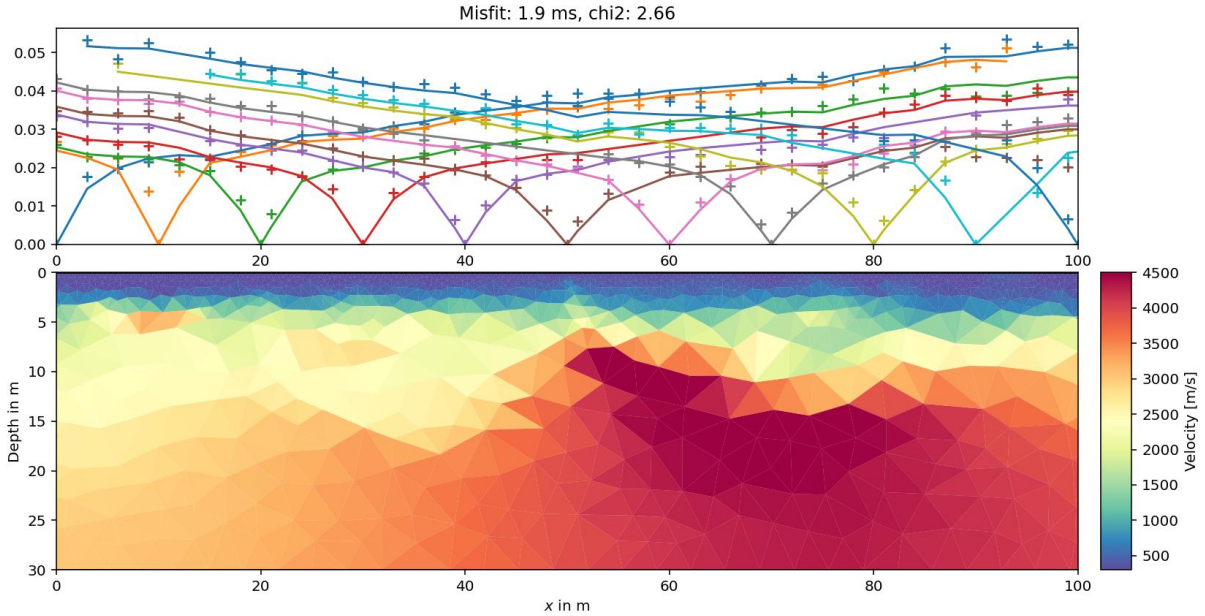

Fig. 10: Display of the inversion result after having started with a standard vertical gradient and without manually created model.

The inversion results are stored in a folder "inver_yyyy-mm-dd_hh-mm-ss" containing thus the day and time of inversion start in its name. In this folder, the following files are stored:

- inversion_parameters.txt with the parameters defined in "Prepare_inversion"

- velocity_model.txt containing the velocity model with for every mesh cell x and y coordinates of the cell center and the cell velocity
- starting_model.png, a plot of the starting model
- inverted_model.png, a plot of the final model after inversion
- travel_times.png, a plot of the measured ("+") and calculated (continuous line) arrival times.
- Travel_time_fit.txt, a file containing six columns: Consecutive number of calculated travel time, corresponding shot point and receiver numbers (natural counting), measured and calculated travel times and the residual calculated as measured minus calculated time.

Clear Body Regularization Settings (no keyboard shortcut)
Sets back all body-specific regularizations to Free Inversion.

View

Toggle Model Plot (keyboard shortcut: CTRL+T)
This button is only active after a first inversion run. It allows toggling between plotting the inverted and the interactive model in the main window.

Toggle Ray Plot (keyboard shortcut: CTRL+R)
This button is only active after a first forward or inversion run. It allows toggling plotting or hiding the rays of the forward or final inverted model. Since PyGimli does not seem to allow simple ray-tracing for a forward model, a dummy inversion with extreme regularization and a single iteration is calculated and the resulting rays are plotted.

View Mesh (keyboard shortcut: CTRL+M)
This button is only active after a first forward run. It plots the actual mesh with on the one hand the pygimli region numbering and on the other hand the velocity distribution of the actual interactive (forward) model.

Color Scale Settings (keyboard shortcut: CTRL+L)
Allows changing the minimum and maximum values of the color scale and the color map between a few predefined types.

Help

About: no keyboard shortcut
Gives information about the program version.

Help (keyboard shortcut: F1)
Explains the main menu option to do model modification and calculations. It is, however, strongly recommended to use the manual. It is usually more up to data and easier to read.

Reference:
Zeyen and Léger (2024): PyRefra – Refraction seismic data treatment and inversion. *Computers & Geosciences* 185: 105556. doi: 10.1016/j.cageo.2024.105556

Appendix

Hints:

The upper left corner of the red body in figure A1 should be shifted upward to be located above the upper right corner of the beige body. Since it is not allowed that a corner point is moved across an existing point, use the following procedure:

1) Separate the upper right blue body through a diagonal line (Fig. A2).
2) Make a right click into the red body followed by a click into the new triangular body. Both bodies will be joined and, since the first click was located inside the red body, the combined body has the properties of the red body (Fig. A3).
3) Now, the nodes may be moved to give the red body the desired form.

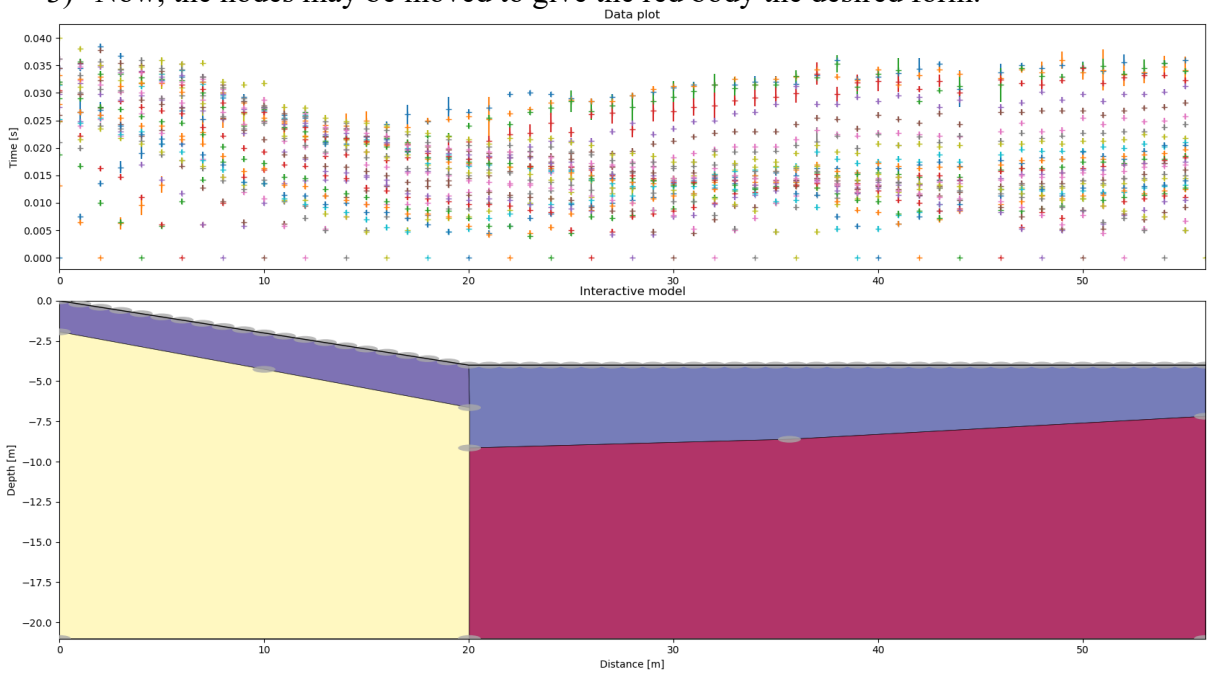

Fig. A1: Model before modification

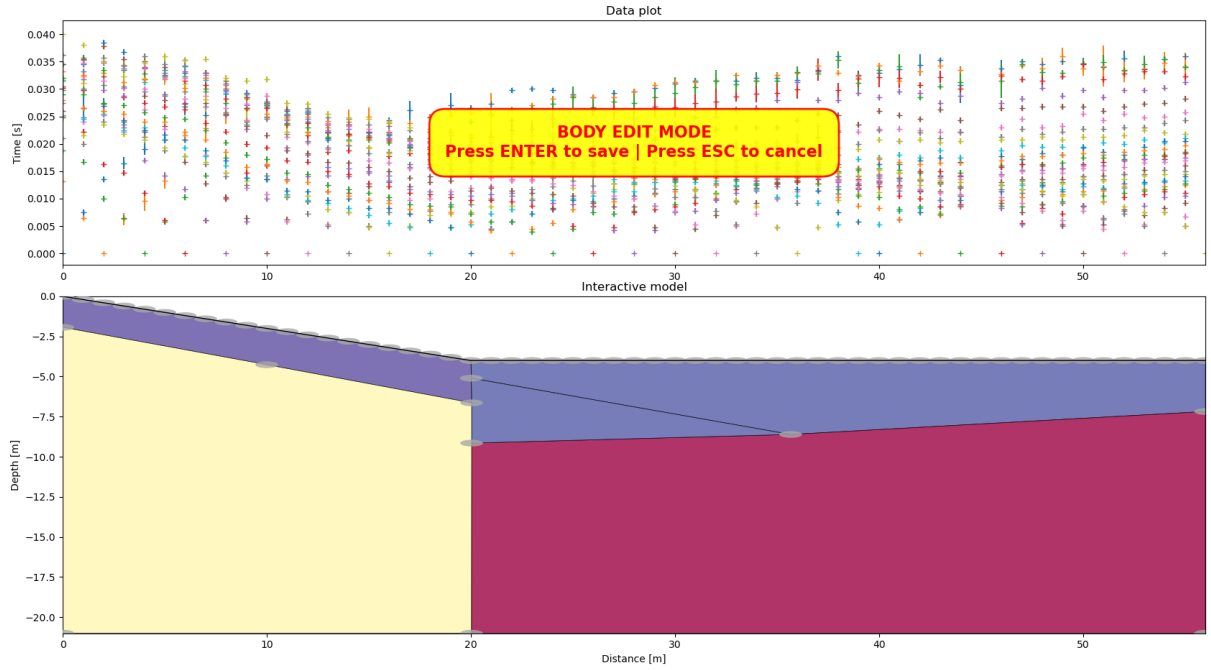

Fig. A2: Model after splitting the upper right body (the velocity of both partial bodies are the same, therefore the colors are the same).

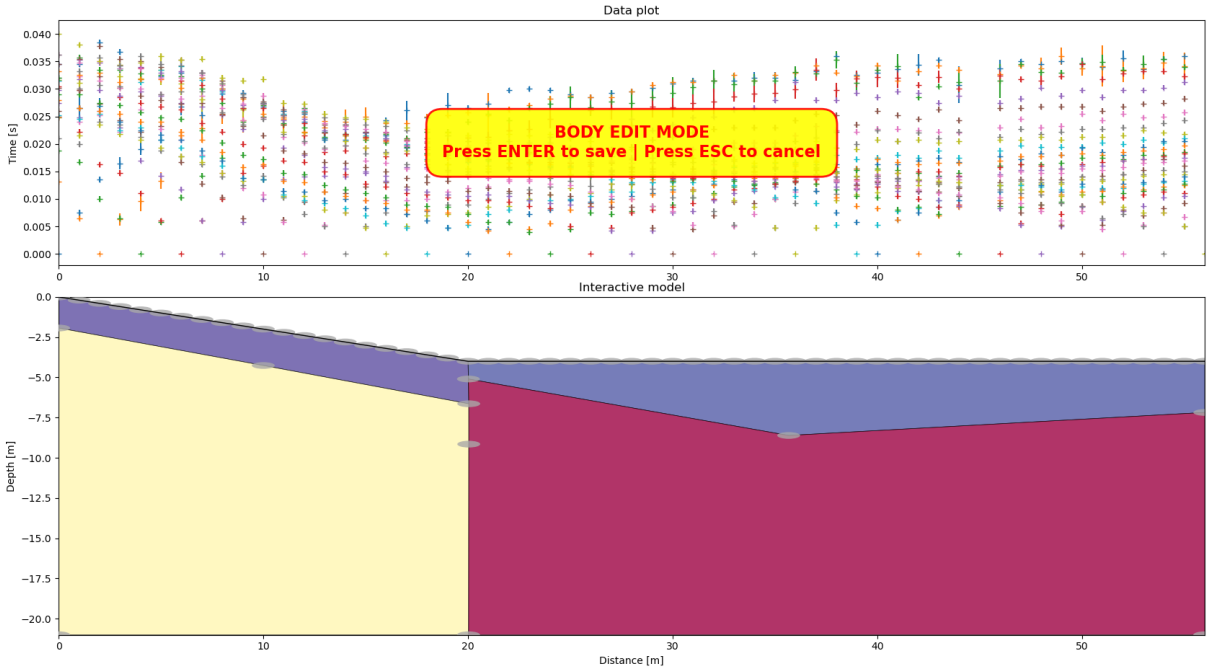

Fig. A3: Model after joining the new triangular body with the red one.