Geol 483.3

Lab project #4

Analysis of a 2D refraction dataset from 2022 Geophysics field school

In this lab, you will analyze first-arrival travel times from one of the lines recorded along the upper trail along the Saskatchewan Crescent we collected during the Geophysics field school in 2022. This was a small 2-D survey (about 200 m long) using a rolling 96-channel receiver spread.

I performed the initial editing of the dataset. The complete dataset can be found in directory `/data/morozov/Riverbank_2022_Refr/` on Linux computer named `sura` (`sura.usask.ca`). This computer is located on the right side of Geology room 135, and it is also available remotely. For this lab, you will need to use the SEGY (seismic data) file `data_line02.sgy` from this directory. A copy of this file is here and some Matlab/Octave codes are included in this archive file. I may update this file as we progress with the lab project.

Refraction data analysis will consist of several steps:

1) Loading the SEGY file into a commercial software package called TomoPlus by GeoTomo. This software is designed for analysis and inversion of near-surface seismic records. In this software, you will display the seismic line, examine its parameters, evaluate data quality, and try various types of filtering and display.
2) Picking first arrivals in TomoPlus and exporting them into ASCII tables for Matlab.
3) Plotting the travel times in several forms in Matlab;
4) Inverting the first-arrival travel times using Matlab programs which we will develop during this lab. Methods of this inversion are outlined in section "Methods" below.

Because of limited time remaining in this term, you may not achieve the complete inversion but you will still obtain intermediate results which will be useful for deriving a model of the shallow subsurface beneath the University Drive.

Prior to starting this lab, you will need to set up your Linux account as on any of the Linux computers in room 135. The procedure is explained under link "Linux setup for labs" on the web site. You will only need to do this once. After this setup is complete, you should be able to login onto any of our machines and use it to run GeoTomo and other programs. Computer `sura.usask.ca` can be used remotely from on and off campus after performing the appropriate VPN setup.

During and after this setup, you will need to learn about basic Unix shell commands:

- `pwd` (print work directory),
- `cd` (change directory),
- `ls` (list file names),
- `mkdir` (create directory),
- `cp` (copy files), `mv` (move or rename files), `rm` (delete files)
- `man` (view manual about any command, with many options),

- `scp` (secure copy of files or directories from any machine over the network)
- `more` or less (display contents of text files),

and other.

Methods

Inversion of first-arrival travel times consist in finding a model of the subsurface which would predict travel times for head waves close to those picked from the dataset. Below, I define the various components of this inversion.

Model

We will use a layered model of the subsurface in a delay time (sometimes also called "time term") form. In this form, head wave delay times are used instead of the depths of the refracting boundaries. Between the boundaries, model velocities are constant vertically and smoothly variable horizontally. Thus, the *n*-th boundary ($n = 1..N_b$) is described by two smooth functions of the horizontal coordinate x:

delay time $\delta t_n(x)$ and slowness below the refractor $p_n(x)$. (1)

The refractor depths are related to delay times as

$$z_n = z_{n-1} + \frac{V_{n-1}}{\cos\theta_n}\left(\delta t_n - \sum_{k=1}^{n-1}\frac{\sqrt{1-\left(p_n V_k\right)^2}}{V_k}\right), \tag{2}$$

where $V_n(x) = 1/p_n(x)$ is the velocity *above* the refractor, and $\theta_n = \arcsin\left(p_n V_{n-1}\right)$ is the critical angle for the refractor (this formula is simply an inverse of the expression for the delay time for a stack of layers we saw in class). These depths should be measured relative to some smooth datum surface. As the datum, we will select a smooth line below the minimum surface elevation.

The uppermost near-surface layer ($n = 0$) also contains model parameters given in eq. (1). For this layer, function $\delta t_0(x)$ has the meaning of delay time within a very thin near-surface layer, and $p_0(x)$ give the variations of the direct-wave velocity.

For numerical inversion, the continuous functions and $\delta t_n(x)$ and $p_n(x)$ need to be discretized. They can be discretized by selecting several control points (these points may be different for the depth-related functions $\delta t_n(x)$ and for velocity-related functions. Between these points, the values of functions will be determined by interpolation. Interpolation of discrete points can be represented by summations:

$$\delta t_n(x) = \sum_{i=1}^{N_{\delta t}}\varphi_i(x)\,\delta t_{ni}\,,\quad p_n(x) = \sum_{i=1}^{N_p}\varphi_i(x)\,p_{ni}\,, \tag{3}$$

where δt_{ni} and p_{ni} are the model parameters at the discrete points, and $\varphi_i(x)$ is a "sawtooth"-shape basis function centered on the *i*th control point x_i. The numbers of control points $N_{\delta t}$ and N_p can be different. Usually, N_p (number of points at which the layer velocities are defined) is small (2 to 5), although in the midpoint method described below, N_p can be large. The number $N_{\delta t}$ controls the detail of depth variation of the layers, and this number would usually be larger than N_p.

Thus, matrices δt_{ni} and p_{ni} contain all parameters of the model we will need to invert for. If these parameters are known, the model can be plotted and all travel times can be predicted.

Travel time prediction

The complete predicted travel-time model that we will use for matching the observed travel times is

$$t_n^{\text{pred}}\left(S,R\right)=t_S^{\text{rec}}+t_{Sn}^{\text{elev}}+t_{Rn}^{\text{elev}}+t_0\left(x_S\right)+t_0\left(x_R\right)+t_n^{\text{model}}\left(S,R\right)+t_S+t_R\,, \tag{4}$$

where S denotes the source, R denotes the receiver, and n is branch of the wave (direct or refracted on the nth boundary). In eq. (4):

1) t_S^{rec} is the reciprocal-time correction applied to each source (explained in the next subsection);
2) Elevation-related terms t_{Sn}^{elev} and t_{Rn}^{elev} representing additional delay times of the source and receiver locations due to their elevations relative to the datum: $t_n^{\text{elev}}=\frac{\sqrt{1-\left(p_nV_0\right)^2}}{V_0}\left(z-z_{\text{datum}}\right)$.
3) Zero-offset time terms $t_0\left(x_S\right)$ and $t_0\left(x_R\right)$ due to a possible very low velocity, thin near-surface layers. These terms are "surface consistent", which means that they relate to the surface locations x only and are equal for source and receiver located at the same point.
4) Term $t_n^{\text{model}}\left(S,R\right)$ is the travel time predicted by a layered subsurface model, <u>with the source and receiver located on the datum</u>;
5) The last terms t_S and t_R account for small source- and receiver related travel-time variations which are not accounted for by the 2-D model. These terms are also described in the next subsections.

For a given layer number $n \geq 0$, the travel time $t_n^{\text{model}}\left(S,R\right)$ from source S to receiver R is predicted by the delay-time relation:

$$\begin{aligned} t_n^{\text{model}}\left(S,R\right)=t_S^{\text{rec}}+t_S^{\text{elev}}+t_R^{\text{elev}}+t_0\left(x_S\right)+t_R\left(x_S\right)+ \\ +\left[\delta t_n\left(x_S\right)+\delta t_n\left(x_R\right)+\int_{x_S}^{x_R}p_n\left(x\right)dx\right]+ \\ +t_S^{\text{NSC}}+t_R^{\text{NSC}}. \end{aligned} \tag{4}$$

With $n = 0$, this equation gives the direct-wave travel times and with $n > 0$ – head wave travel times. Using eq. (3), the integral in this expression is transformed into a sum:

$$t_n\left(S,R\right)=\sum_{i=1}^{N_{\delta t}}g_i\left(S,R\right)\delta t_{ni}+\sum_{i=1}^{N_p}p_{ni}\,f_i\left(S,R\right), \tag{5}$$

where $g_i\left(S,R\right)=\varphi_i\left(x_S\right)+\varphi_i\left(x_R\right)$ and $f_i\left(S,R\right)=\int_{x_S}^{x_R}\varphi_i\left(x\right)dx$ is an integral of the ith basis function along the source-receiver path. These integrals are easily calculated analytically using the known piecewise-linear functions $\varphi_i(x)$, and so eq. (4) represents a matrix product and summation which can be easily evaluated in Matlab.

Inversion

Inversion of the observed travel times consists in finding the subsurface model and the additional terms in eq. (4) so that $t_n^{\text{pred}}(S,R) \approx t^{\text{obs}}(S,R)$ in the least-squares sense. This inversion can be performed in the order of terms shown in eq. (4), as described below.

Reciprocity-based corrections of source times

First, you will **invert the mismatches of all reciprocal times for the source time errors**. For each source S_1, consider all other sources S_2 such that each of them has the travel times picked in the vicinity of the other source. Any velocity structure has the reciprocity property $t^{\text{model}}(S_1,S_2) = t^{\text{model}}(S_2,S_1)$, and therefore the difference of these reciprocal travel times equals

$$t_{S_1}^{\text{rec}} - t_{S_2}^{\text{rec}} = \Delta t^{\text{reciprocal}}(S_1,S_2), \tag{6}$$

where $\Delta t^{\text{reciprocal}}(S_1,S_2) = t^{\text{obs}}(S_1,S_2) - t^{\text{obs}}(S_2,S_1)$ is the difference between the observed travel times for the two shots. This is a <u>linear inverse problem</u> for t_S^{rec}, which can be solved by the least-squares method.

When inverting eq. (6), you will notice that the inverse is nonunique because this equation allows adding an arbitrary constant to all t_S^{rec}. This problem is easily corrected by adding an additional constraint to the system of equations (6). The constraint can be setting $t_S^{\text{rec}} = 0$ for one shot or requiring that the average of all t_S^{rec} equals zero: $\sum_S t_S^{\text{rec}} = 0$.

Ideally, $\Delta t^{\text{reciprocal}}(S_1,S_2)$ should equal zero and therefore all $t_S^{\text{rec}} = 0$. However, as $\Delta t^{\text{reciprocal}}(S_1,S_2) \neq 0$ in the real data, inversion of eq. (6) gives the source times t_S^{rec} correcting for this error. These terms should then be subtracted from the data:

$$t^{\text{obs}}_{\substack{\text{corrected}\\\text{reciprocity}}}(S,R) = t^{\text{obs}}(S,R) - t_S^{\text{rec}}, \tag{7}$$

giving corrected input data for further inversion, which are free of reciprocal travel-time mismatches.

Analysis of the first-arrival Travel-Time Field (TTF)

The next two steps of inversion consist in obtaining the subsurface model (parameters $t_0(x)$ and δt_{ni} and p_{ni} in the preceding section). For this, it is useful to view the first-arrival travel times as samples of a <u>continuous "time field"</u> (TTF) function $T(x_S, x_R)$ of continuously variable source and receiver coordinates x_S and x_R. The corrected observed picks (eq. (7)) represent sampling of this function at the available source and receiver pairs. For the subsequent plotting, data analysis and inversion, it is convenient to grid this function on a <u>regular grid of midpoint</u>

coordinates $x_{mp}=\frac{x_S+x_R}{2}$ and (signed) source-receiver distances $d=x_R-x_S$. This gridding can be easily performed using function `griddata` in Matlab (which uses the Delaunay triangulation).

Due to the source-receiver reciprocity, travel time of any wave remains unchanged if the source and receiver are switched places. Therefore, the TTF is always an even function with respect to *d*: $T\left(-d,x_{mp}\right)=T\left(d,x_{mp}\right)$. This is a very useful property because it allows considering only non-negative values of *d* and transforming the TTF as

$$\bar{T}\left(d,x_{mp}\right)=\frac{1}{2}\left[T\left(-d,x_{mp}\right)+T\left(d,x_{mp}\right)\right]. \tag{7}$$

This averaging of reversed travel times improves sampling in the plane of (d, x_{mp}) and reduces noise in the travel-time data. At Further, it is useful to separate the zero-offset travel-time: $t_0\left(x_{mp}\right)=\bar{T}\left(0,x_{mp}\right)$ and denote the new TTF with zero values at $d=0$ $\bar{T}_0\left(d,x_{mp}\right)=\bar{T}\left(d,x_{mp}\right)-t_0\left(x_{mp}\right)$.

In the code, function $\bar{T}_0\left(d,x_{mp}\right)$ for $d>0$ is stored in a Matlab matrix, with columns representing the common-midpoint travel times and rows representing the common-offset travel times. The range of (d, x_{mp}) values is subdivided into zones containing the direct waves (near axis $d=0$, and head waves from the refracting boundaries (at progressively larger d). These zones are specified by giving the crossover distances for refractions at each x_{mp}. Measurement of these crossover distances is the primary goal of data analysis, which is performed in script `lab4_pass2.m` below.

If crossover distances are determined, then properties of the *n*th layer (direct-wave for $n=0$ and headwave for $n\geq 1$) can be determined by solving linear inverse problem within the *n*th offset interval of the TTF:

$$T\left(d,x_{mp}\right)\approx t_n^{\text{int}}+p_n d\ . \tag{8}$$

Parameters of this inverse problem t_n^{int} and p_n are the intercept time and slowness of the layer, respectively. These parameters can be relatively smoothly variable along the profile (i.e., smooth functions of x_{cmp}).

Subsurface model

The subsurface model is responsible for the terms enclosed in brackets […] in eq. (4). The model is defined as N_b layers, each *n*th layer characterized by a (spatially) slowly variable slowness p_n and delay times δt_n in eq. (1) defined at every midpoint x_{mp}.

Slowness p_n for the *n*th layer can be taken from the corresponding TTF parameter (eq. 8), and an initial estimate for δt_n can be simply obtained as

$$\delta t_n=\frac{t_n^{\text{int}}}{2}\ . \tag{9}$$

The intercept measured for the direct wave (with $n=0$) can be added to the zero-offset time function $t_0(x_{\text{cmp}})$, after which δt_0 is set equal zero. This function $t_0(x)$ is then used to evaluate the terms $t_0(x_S)+t_0(x_R)$ in eq. (4). Once parameters $t_0(x_{mp})$ and p_{ni} are estimated, the elevation-related terms t_{Sn}^{elev} and t_{Rn}^{elev} can be calculated and also included in the right-hand side of eq. (4).

After estimating all of the above "static" terms, the data can be corrected for them:

$$t_n^{\text{obs,corrected}}(S,R)=t_n^{\text{obs}}(S,R)-\left[t_S^{\text{rec}}+t_S^{\text{elev}}+t_R^{\text{elev}}+t_0(x_S)+t_R(x_S)\right] \tag{10}$$

This subtraction should reduce the scatter of the travel times due to elevation and near-surface velocity variations and improve the identification of slownesses p_n. Therefore, the evaluation of the elevation of $t_0(x)$, t_{Sn}^{elev} and t_{Rn}^{elev} and estimation of p_n should be iterated a couple times until these values become consistent.

Selections of the above "static" terms and δt_n give a fairly good starting model for the inversion. This starting model would represent an accurate solution a 1-D refraction experiment with layering as at location x_{mp} This model will be further refined by iterations described in the next subsection.

Inversion for delay times

Using the identified layer slownesses, the delay times for refracting boundaries can be obtained by correcting the observed times for all of the above effects and solving the linear inverse problem in eq. (4):

$$t_{\text{corr}}^{\text{obs}}(S,R)\approx\sum_{i=1}^{N_{\delta t}}g_i(S,R)\delta t_{ni}\,, \tag{11}$$

where the corrected data are

$$\underset{n}{t_{\text{corr}}^{\text{obs}}}(S,R)=t^{\text{obs}}(S,R)-\left[t_S^{\text{rec}}+t_{Sn}^{\text{elev}}+t_{Rn}^{\text{elev}}+t_0(x_S)+t_0(x_R)+\sum_{i=1}^{N}p_{ni}f_i(S,R)\right]. \tag{12}$$

Equation (11) is also an overdetermined linear inverse problem for unknowns δt_{ni}, which is solved by an iterative least-squares inverse.

Residual travel-time terms

After all "surface-consistent" model-related terms in the right-hand side of eq. (4) are inverted for, the "residual" terms t_S and t_R terms can be obtained and used to improve the model (4). Such terms are called the "short-wavelength static correction" in the popular Hampson and Russell's GLI refraction statics software (also included in GeoTomo). These terms are also obtained by solving a linear inverse problem:

$$t_n^{\text{error}}(S,R)\approx t_S^{\text{NSC}}+t_R^{\text{NSC}}\,, \tag{13}$$

where $t_n^{\text{error}}=t_n^{\text{obs}}-t_n^{\text{model}}$ is the total error of the travel-time prediction by the final model (4) excluding the last to terms. Equation (13) can also be solved by the least-squares inversion. However, as its forward model (right-hand side) is very simple, it can be easily solved even in a better approximation. Let us use the median (statistical) inverse:

$$t_S^{\text{NSC}}=median\left[t_{\text{error}}(S,R)\Big|_S\right],\ \text{and}\quad t_R^{NSC}=median\left[t_{\text{error}}(S,R)\Big|_R\right], \tag{14}$$

where notation $median\left[t\big|_{S\text{ or }R}\right]$ means evaluation of the median of all values t over all travel-time picks for the given source S, or for the given receiver R. These relations give the "non-surface consistent" travel-time terms in eq. (4) due to uncorrelated near-surface velocity variations in the vicinities of the sources and receivers, respectively.

Assignments

1) **Create a work directory** under path /data/ on `sura`. Use 'cd /data/' and then `mkdir` followed by your username. Then '`cd`' to that directory. In the following, <u>place all files and work only in this directory</u>. You can create any subdirectories or files in there. In particular, when you start using GeoTomo, place its project into this directory.

2) **Start GeoTomo (TomoPlus) programs** by typing `geotomo` in a Unix shell. In GeoTomo, create a project and load file `lab4_data_line02.sgy` in it.

   Familiarize yourself with SEGY headers in the file, display geometry of the data.

   Create seismic displays and evaluate the quality of data records. There are a number of poorly recorded records, and channel 1 was disconnected. Some records are noisy because of the AC power generator being placed close to the line, and some records are contaminated by pedestrian and bicycle noise. You need to ignore the poor records or exclude them by marking as bad or 'killed' records.

   The different shots in the records will be identified by different "field file identifiers" (FFID), which are called (I think) Shot ID in GeoTomo. The individual trace records are identified by their channel numbers used in the recording instrument.

   Identify the first arrivals in the records.

3) Select display form (time range, frequency filtering, AGC, wiggle-trace or variable-intensity color style) and **perform picking of the first-arrival travel times**. This will have to be done manually, but you can also explore the available automatic picking options.

   Try picking the different FFIDs consistently, i.e, pick the zero crossings from the first negative trace swing to the large positive amplitude peak. This may not always be possible to do, so only pick the records where such identification can be made. In GeoTomo, there should be options for automatic snapping of the manual pick to the nearest zero crossing, and also for picking groups of adjacent traces.

   To help ensuring picking consistency for different FFIDs, try switching between the shot display (usually the default) and common-midpoint and common-receiver display modes. In these displays, traces from different FFIDs are shown side by side, and it is therefore easy to see whether the same wave is being picked on them.

4) After the picks (or maybe a representative sample) are completed, **display them in GeoTomo** and evaluate the travel-time patterns. **Identify the direct waves, refractions** (head waves), and roughly estimate the cross-over distances.

5) **Export the first-arrival picks into an ASCII format** and prepare them for further processing in Matlab or GNU Octave.

   The preparation may require commenting out (by using symbols '`%`') certain header lines so that the resulting files can be loaded using function `load()` in Matlab. Alternatively, the pick files can be edited to include names of variables and represent parts of a Matlab program.

The following tasks are performed in Matlab using codes provided in this archive file. Try understanding what is being done and suggest modifications if needed.

The different steps of processing are split into three scripts:

- `lab4_pass1.m` performs loading the data from GeoTomo file, creating geometry, (optionally) correcting some \ errors found in the data, and performing the reciprocal-time analysis and inversion;
- `lab4_pass2.m` performs semi-interactive analysis of the travel time and obtaining estimates of the near-surface velocities and delay times (at present, velocities in the deeper layers are unavailable.)
- `lab4_pass3.m` performs inversion of the first-arrival times.

In file `lab4_pass1.m`, **replace the current file name** in command `load_picks()` with the name of the file you produced from GeoTomo. **Execute this script** and look at three figures produced by this script:

- Figure 1 shows a map of sources and receivers, with receiver station numbers printed next to them. In this dataset, I had not set the accurate geometry, and so this plot should look fairly simple.
- Figure 2 is a preliminary plot of velocity and delay-time cross-section. It will also look very simple because of the absence of elevation information.
- Figure 7 shows reciprocal-time misfits before (on the left) and after (right panel) the reciprocal-time corrections. This is the principal result of this processing step.

Next, **execute `lab4_pass2.m`** . This script will produce Figures 5 and 6 with travel times from selected sources and midpoints. The script produces the following plots:

- `Figure 201` shows all picked travel times versus profile distance. In the upper panel, the picks are shown versus the positions of receiver stations, similar to the T-X displays in GeoTomo. In the middle panel, teh same travel times are plotted vs. the source-receiver midpoints. In the bottom panel, the common-offset travel times are contoured versus profile distance.

  In these plots, look for anomalies and indications of mis-picks in the travel-time dataset.
- `Figure 211` (or similar) shows travel times of selected groups of shots vs. source-receiver offset. The lines are split into colored segments (muted near-source segments – green, direct waves – red, first and second refractions – blue and magenta, respectively). The plots are done using travel-time reduction, so that the differences in velocities are clearer.
- `Figure 212` is similar, but the travel times are plotted along common-midpoint cross-sections of the travel-time surface.

- `Figure 221` shows the interpolated travel-time field (TTF) surface plotted as a function of midpoint and offset (top panel) and the breakdown of this TTF into direct- and headwave blanches (bottom panel). In both of these plots, the cross-over lines from `model.crossover` are plotted.

  Looking at these plots, you will need to **adjust the values of columns in matrices `model.crossover`** to represent the offsets at which you estimate the crossover distances (separators between the colored segments) at the various locations within the profile. Matrix `model.mute` similarly gives the near-source offset ranges which need to be excluded from measuring direct-wave velocities (this range is shown by green color).

Repeat the travel-time plots at the end of `lab4_pass2.m` with different selections of midpoints and make more adjustments in `model.crossover`. You can make multiple plots simultaneously using different figure numbers (first parameter in `plot_ttimes`… functions).

6) After the picking of crossover distances is finished, re-run `lab4_pass1.m` and `lab4_pass2.m` once again for the cross-over picks to take effect. Then **execute `lab4_pass3.m`,** and the inversion should be complete.

7) Save in graphics formats, **review and discuss the plots** produced in the processing above steps.

   With the travel times and model loaded in the workspace, you can **try additional versions of these plots**. For example, you can try plotting the travel times and TTF with different reduction velocities.

Hand in:

Codes, plots, and report in electronic format.