

Automated three-dimensional reflection traveltime modelling to detect dipping layer geometries from active seismic data – manual to the matlab code (Samuel Zappalá – 04/08/2025)

Contact: Samuel Zappalá, Uppsala University, Department of Earth Sciences. ORCID 0000-0003-3402-6850, e-mail: samuelzappala@hotmail.it

Date: 04/08/2025

DOI: 10.57804/gqnv-0y39

Available at: <https://doi.org/10.57804/gqnv-0y39>

The code is licensed under the terms of the GNU AGPLv3.

Peer reviewed publication doi: 10.1111/1365-2478.70165

Introduction

The code presented in this manual determines the best-fitting geometries (strike and dip angles) of a dipping reflector with a known surface intersection from pre-stack seismic data with an automated approach. The modeling is first run on a single gather and then iterated on all available gathers. The theory behind the code and how it is implemented is largely described in the open source paper “Automated three-dimensional reflection traveltime modelling to detect dipping layer geometries from active seismic data” published on Computers and Geosciences journal (Zappalá et al. 2025). To use the code, open ReflectionModelling.m on Matlab© and modify the first part of the code with the desired input and variables as described later on in this manual, then, just run the code and enjoy!

Repository elements:

Main matlab code

- ReflectionModelling.m, actual code to update and use for every needed case
- Shotgathers_ex.segy, 2 shot gathers with visible dipping reflections intersecting the first breaks
- pickref_ex.txt, picks in the selected shot gathers from one of the strong dipping reflections
- pickfb_ex.txt, picks in the selected shot gathers from the first breaks
- MigratedStack_ex.segy, migrated stacked section
- CMPcoor_ex.txt, coordinates of the migrated stacked section (CMP, x, y, z)
- RMoutput*.fig, generated figure outputs from the ReflectionModelling.m code

Synthetic matlab example code

- ReflectionModelling_3Dsyntax.m, main code modified for an example on synthetic data
- Shotgathers_ex_synt.mat, 2 synthetic 3D shot gathers with a dipping reflection
- pickref_ex_synt.txt, picks in the selected shot gathers from the dipping reflection
- pickfb_ex_synt.txt, picks in the selected shot gathers from the first breaks
- RMsoutput*.fig, generated figure outputs from the ReflectionModelling_3Dsyntax.m code

Used external functions

- readHD1sgy.m, used function to read segy data, the original form is available at <https://seismic-lab.physics.ualberta.ca/>

The main code example is from a seismic acquisition in hardrock environment in Brunswick, Canada (Malehmir and Bellefleur, 2009). The used acquisition geometry are mainly 2D with minimum available azimuth differences. Still, the area of matching geometries can be largely reduced. The

code uses two shot gathers and the migrated stacked section. In a standard desktop (RAM 16 GB, processor 8 cores 3,00 GHz) the code run in around 90 seconds.

The synthetic code example consists of two modelled shot gathers in a 3D receiver grid, organized in four parallel inline. The modelled subsurface had a 2000 m/s medium velocity and a dipping reflector of 150° strike and 55° dip towards southwest. The needed coordinates for the matlab shot gather variables are added at the beginning of the code to compensate for the not SEG-Y format. The 3D acquisition geometry result in a strong RMS error minimum at the correct dip and strike geometry. The stacked section part is not run since the stacked section was not available for the synthetic dataset. The code run in approximately 70 seconds on the same standard desktop as before.

Code main structure

- Load data and define variables
- Estimate 1st layer velocity from first break picks
- Model hyperbolas: compute time for each channel, each strike and each dip
- Compute RMS error between model and picked reflection hyperbola (point by point)
- Define the best geometries with the lowest RMS error
- Iterate on multiple shot gathers
- Model the selected geometries on migrated stack section

Code input

- Shot gathers in SEG-Y format. The code is pre-set for shot gathers but it can run on any kind of pre-stack gathers after correction of the related headers
- First break picks as three columns ASCII file with receiver point, picked time in ms, and shot point.
- Reflection picks as three columns ASCII file with receiver point, picked time in ms, and shot point
Optional:
- Migrated stacked section in SEG-Y format. Better in depth domain but time can also be used
- CMP coordinates as four columns ASCII file with CMP, x coordinates, y coordinates, and z coordinates in meters (i.e., UTM or relative coordinates)

Code output

- Reflector dip, strike and dip azimuth
 - RMS error map for each dip-strike pair respect to the picked reflection hyperbola.
 - Azimuth coverage considered for the modelling
 - Modelled reflection overlaying the uploaded gather and picks and the eventual uploaded migrated stacked section
- All results are shown in overall and per each uploaded gather.

Variables to define

For the modelling:

- *vrange*: Velocity range and increment to model
- *dip*: Reflector dip range and increment to model (max needed 1 to 90)
- *str*: Reflector strike range and increment to model (max needed 1 to 360)
- *srec*: Receiver point where the reflector intersects the surface

Plot:

- *fp* and *lp*: First and last sample to visualize in ms
- *sh*: Eventual constant shift to apply to the imported picks
- *dipex*, *strex*, and *shot_ex*: Eventual dip, strike, and shot point to visualize in example
- *clims*: Limits for the colorbar of the RMS error map plots

Additional stack:

- *scdp*: CMP where reflector intersects the surface
- *cdp_sp*: CMP spacing in meters
- *datum*: Elevation datum used for the processing of the migrated section in meters
- *domain*: Stacked section domain, 0=time or 1=depth
- *xstack*: CMP points index to plot
- *fps* and *lps*: First and last sample to visualize in ms or m in stacked section

Variables read from the headers are already set, but they should be checked to be sure that the used bins correspond.

When no post-stack modelling is desired, lines 8 and 9 of the code should be commented.

If the known position where the reflector intersects the surface is outside the profile, a fake receiver point can be added to the SEG-Y geometries at the known coordinates to be indicated as *srec* in the variable definition.

Uncommenting lines 177-196 will enable to visualize the shot set as example and to analyze the behavior of the reflection with respect to strike and dip changes.

Notes about the code

The code uses the 3D traveltime equation for dipping layers in a homogenous media generalized by Diebold (1987)

$$t^2 = \frac{4 h^2 + x^2 - 4 h x \cos(\alpha)}{V^2}$$

and implemented as:

$$t^2 = \frac{4 (K_1 \cos(\varphi + K_2) \sin(\delta))^2 + K_3 - 4 K_1 K_3 \cos(\varphi + K_2) \sin^2(\delta) \cos(\varphi + K_4)}{K_5}$$

Where:

t = time

h = 3D perpendicular depth of the reflector at the source point

$x = K_3$ = offset with K_4 = corresponding angle to the north

α = 3D angle between the reflector depth h and the source-receiver direction

$V = K_5$ = media velocity

δ = reflector dip angle

φ = reflector strike angle

K_1 = distance between the source position and the position where the dipping reflector intersects the surface with K_2 = corresponding angle to the north

All physical distances used for the computation are positive since the direction is considered in the angles (i.e. no positive and negative offset needed).

All equations and distances used in the code are corrected also for elevation differences, useful in rough topography conditions.

The modelled time matrix is a 3-dimensional matrix with each value corresponding to the dip in the 1st dimension, the channel in the second dimension and the strike in the third dimension. The modelled time matrix and the RMS error matrix show in the first row and first column the corresponding strike and dip angle. The final RMS error matrix is an average of all shot's matrices.

The code can compute the depth of the reflector at each CMP point from the resulting geometries to overlay on the migrated stacked section. The result can be converted in time domain if needed (to set at the beginning of the code) through the homogeneous velocity used for the reflection modelling, but a comparison in depth domain is more accurate.

IMPORTANT! the code do not need a big quantity of data in input, but it is quite sensible to the quality of the uploaded reflection picks. Therefore, it is strongly recommended to limit the amount of used reflection picks to the ones picked with high confidence for the code to estimate the correct reflector geometry. If the picks are correct and from different azimuths, ten of them will be enough to correctly define the geometries.

Interpretation

For strictly 2D cases, different geometries fit the same reflection causing a semicircular area of minima in the RMS error matrix and inaccuracy in the geometry estimation. To achieve a unique solution multi-azimuthal data is required or a known intersection position of the reflector with the surface outside the source-receiver line. Crooked profiles may resolve this inaccuracy and generate a single strong minimum.

References

- Diebold, J.B., 1987. Three-dimensional traveltimes equation for dipping layers. *Geophysics* 52, 1492–1500. <https://doi.org/10.1190/1.1442267>
- Malehmir, A., Bellefleur, G., 2009. 3D seismic reflection imaging of volcanic-hosted massive sulfide deposits: Insights from reprocessing Halfmile Lake data, New Brunswick, Canada. *Geophysics* 74, B209–B219. <https://doi.org/10.1190/1.3230495>
- Zappalá, S., Westgate, M., & Malehmir, A. (2026). Automated Three-Dimensional Reflection Traveltimes Modelling to Extract 3D Dipping Layer Geometries. *Geophysical Prospecting*, 74(4). <https://doi.org/10.1111/1365-2478.70165>