Lindu Software: A Free Seismological Data Processing Software For Traveltime Tomography Using Python Framework

Yudha Styawan¹, Ruhul Firdaus¹*, Tedi Yudistira² and Cahli Suhendi¹

¹ Geophysical Engineering, Department of Industrial Technology, Institut Teknologi Sumatera, Lampung Selatan, Indonesia
² Geophysical Engineering, Faculty of Mining and Petroleum Engineering, Institut Teknologi Bandung, Bandung, Indonesia
* E-mail: ruhul@tg.itera.ac.id

Abstract. Earthquake data can be used to infer some physical properties for representing the subsurface condition. The 3-Dimensional (3D) seismic velocity structure as a kind of these important properties contains the information of variation in lithology change and fluid saturation. The most common method for inverting from the travel time of seismic event into 3D seismic velocity structure is travel time tomography which is based on the relation between velocity and travel time of P- and S-wave. Based on this concept, we develop a module of Lindu software to infer this seismic velocity structure from travel time data. This module is a part of seismological data processing sequences that have been integrated into Lindu software. The Lindu software uses Python framework, a kind of high-level programming languages. The pseudo-bending raytracing method is employed to calculate the travel time between the event sources and stations and also to build the kernel matrix. The resolution test that relates density of rays and resulted tomogram uses the synthetic Checkerboard Resolution Test (CRT) by using Damped-Least Squares (DLS) method for the inversion. For validating this module, it has been tested by using both synthetic and real data.

1. Introduction
Earthquake data produced by seismic waves propagation contains physical information of subsurface, so that a lot of researchers have developed the various methods and applications to extract them. Currently, some applications can be found as open-source or commercial in various operating systems. Generally, there are two kinds of application, the first is console-based and the second is the graphical user interface (GUI) based. For advanced researcher, the console-based is the main priority because the running time relatively fast and for other reasons. Meanwhile, for some researcher, student, or public, the console-based is relatively difficult to handle. Therefore, they more comportable to use GUI based.

There are some various of seismological data processing that can be utilized by seismologists on implementing seismic tomography. Traveltime tomography is a kind of tomography that use travel time data of seismic wave to obtain P- or S-velocity of the subsurface. In order to give more alternative for seismological community, we develop the Lindu software that have some modules for routine seismological data processing, the first is guided auto-picking feature which employs Akaike Information Criterion (AIC) [1], Modified Energy Ratio (MER) [2], and S/L Kurt algorithm [3]. The second is hypocenter determination which employs Geiger’s Adaptive Damping approach [4]. The third is the earthquake hypocenter relocation employing the Joint Hypocenter Determination method [5].
based on a simple modification of Fortran engine for Python [6,7]. In this paper, we will explain the developed feature on Lindu software for traveltime tomography that validated by 2-Dimensional and 3-Dimensional models using the synthetic data and the real data for the demonstration.

2. Methods

2.1. Traveltime Equation

There is some information that is gotten by earthquake data for tomography processing, a kind of them is travel time data. General equation to obtain it \( T_{ij} \) is \[ T_{ij} = \int_{L_{ij}(3D)} s(r) dl, \]

where \( s(r) \) is slowness as a function of position \( r \), \( dl \) is the length of the ray segment that integrated along line \( L_{ij}(3D) \) that connect the hypocenter \( i \) to receiver \( j \) locations that through the unknown 3D earth structure.

2.2. Pseudo-bending Ray Tracing Method

The pseudo-bending method uses the Fermat principle to deal with the track of seismic rays between source and receiver. Therefore, the time of tracking has to be the minimum value. By this consideration, the choice of segment ray location is based on the velocity gradient in the block approached of the segment ray. Because it is a non-linear solution, the segment ray location is performed as a perturbation. The equation that is used to obtain the perturbation is \[ n' = (\nabla V) - \left( (\nabla V) \cdot (X_{k+1} - X_{k-1}) \right) / \left| X_{k+1} - X_{k-1} \right|, \]

where \( \nabla V \) is gradient velocity on the block, \( X_{k+1} \) and \( X_{k-1} \) are segment location after and before the current location, respectively. So that the unit vector \( n \) is \[ n = n' / \left| n' \right|, \]

and the equation of perturbation value \( R_c \) is \[ R_c = - \left[ \frac{(cV_{mid} + 1)}{4cn(\nabla V)_{mid}} \right] + \left[ \frac{(cV_{mid} + 1)^2}{(4cn(\nabla V)_{mid})^2 + \frac{L^2}{(2cV_{mid})}} \right]^{1/2}, \]

with \[ L = \left| X_{k+1} - X_{mid} \right| \text{ dan } c = \left( 1/V_{k+1} + 1/V_{k-1} \right)/2, \]

where \( V_{mid} \) is velocity on the current block, \( X_{mid} \) is the current segment location, \( V_{k+1} \) and \( V_{k-1} \) are velocity value after and before the current location, respectively. After that, to obtain the new segment ray location, we can use a basic equation as following \[ X_{k}' = X_{mid} + n \cdot R_c. \]

2.3. Inverse Solution

The inverse solution on this module uses the damped-least squares (DLS) method that is simply formed by the Gauss-Newton method as the following \[ \delta m = \left[ G^T C_d^{-1} G + \epsilon C_m^{-1} + \eta D^T D \right]^{-1} G^T C_d^{-1} \delta d, \]

with \( \delta m \) is a model perturbation, \( G \) is kernel matrix, \( \epsilon \) is damping factor (norm damping), \( \eta \) is smoothing factor (gradient damping) with \( D \) is the second derivative operator, \( C_d^{-1} \) is the inverse of the data.
covariance matrix, $C_m^{-1}$ is the inverse of the model covariance matrix, and $\delta d$ is the discrepancy between observation and calculation data.

The matrix-vector form of the tomographic equation can be represented as follows:

$$
\begin{pmatrix}
C_d^{-1/2} G \\
\sqrt{\epsilon} C_m^{-1/2} \\
\sqrt{\eta} D
\end{pmatrix}
\delta m = 
\begin{pmatrix}
C_d^{-1/2} \delta d \\
0 \\
0
\end{pmatrix},
$$

(8)

or the other matrix equation can be simplification, by ignoring the data weighting, as the following

$$
\begin{pmatrix}
G \\
\alpha \mathbf{I} \\
\gamma D
\end{pmatrix}
\delta m = 
\begin{pmatrix}
\delta d \\
0 \\
0
\end{pmatrix},
$$

(9)

Where in the compact form of (9) can be written as $A \Delta m = \Delta d$, that can be solved by the general least squares inverse as follow

$$
\Delta m = (A^T A)^{-1} A^T \Delta d,
$$

(10)

with $\Delta m$ is a model perturbation, $A$ is kernel matrix, and $\Delta d$ is a vector discrepancy between observation and calculation data.

2.4. **Checkerboard Resolution Test**
Checkerboard resolution test (CRT) is a technique that generally is used to see resolution level in every model block with using observation data and performed in a pattern model (as chess-board) as the variety of high and low-velocity value [8]. When the inverse model sees a part that possesses a pattern same with the true model, the part of this model possesses the high resolution, the same with another way.

3. **Result and Discussions**

3.1. **Graphical User Interface (GUI)**

![Figure 1](image)

Figure 1. The interface of the Tomography tab in Lindu Software.
The interface of Lindu Software contains some main tabs which we only concern to Tomography tab being shown in fig. 1. This tab contains the menubar on the upside, the status bar on the downside, the main group box for tomography processing on the left side, and the view group tab on the right side for visualization. For 2D tomography, it can be opened on the Analysis menu and choosing 2D Analyze sub-menu. The main tabs on this part are Create Model tab to create the 2D model and locate source and receiver, Forward Modeling tab, CRT Modeling tab, and Inverse Modeling tab as shown in fig. 2.

3.2. Validation by Synthetic Data
Every validation in this paper uses fixed-parameter besides of the parameter that is tested as shown in table 1. The first validation is the pseudo-bending ray tracing algorithm. Figures 3(a)-(f) show the 2D models that are tracked by a ray between receiver and source. Every model provides different respond on ray tracing because of the different model forms.

The behavior of the rays as shown in figs. 3(a)-(f) shows that the rays appropriate to Snell law. The selection of the rays was determined by the minimum of travel time on the range of iteration. The optimised ray tracing on model a, b, and f are in the last iteration and the model c, d, and e are 16th, 39th, and 5th iteration, respectively. By using the minimum travel time to estimate the optimised ray tracing, the algorithm has been appropriated to Fermat principle. Based on the analysis, the algorithm of ray tracing can be used for forwarded modeling. For validating the 3D model, figs. 4(a)-(b) show that rays can be adapted on the 3D complex model.

| Table 1. Fixed parameters of the synthetic model validation |
|----------------------------------------------------------|
| **Parameters** | **Value** |
| **Model**      |           |
| The distance of X/Y coordinate | 100 km    |
| The distance of Z coordinate  | 50 km     |
| Number of X/Y blocks          | 10        |
| Number of Z blocks            | 10        |
| **Ray**        |           |
| Number of Iteration           | 100       |
| Number of Ray Segment         | 20        |
| **Tomography** |           |
| Number of Iteration           | 10        |
Figure 3. the ray density of the model 100 km x 50 km (left), the model with ray tracing (center), and travel time versus iteration curve (right). The optimised ray being used is at the minimum travel time.

The second validation is the result of a tomography inversion algorithm. Figure 5 shows the real model and the result inversion model. By the figure, \( N_r \) and \( N_s \) is the number of receiver and source, respectively. Receivers are located randomly on the upside of the model and the sources are located on the downside of the model. Based on the result of fig. 5, the increase of data that be used can give the model that closes to the real model. Inversion result of CRT model on every \( N_r \) and \( N_s \) value can be shown that increasing of data that be used causes increasing resolution of the inversion result of the CRT model. RMS on every iteration value shows that increasing iteration value causes decreasing in RMS value.

The third validation is norm and gradient damping that is facilitated on this module. Figures 6(a)-(d) show the true and inversion model by varying on norm damping value and figs. 7(a)-(d) on gradient damping (smoothing) value. For analysis, we use some 2D models for easily describing. For varying norm damping, increasing norm damping value causes the result of the inversion model closing to the initial model. Closing to the initial model can be shown on the resolution of the CRT perturbation model is decreasing through increasing the norm damping value. For gradient damping (smoothing) analysis, increasing the gradient damping value causes decreasing contrast boundary of the velocity model. Based on some analysis of the above, the tomography module and the parameter of inversion appropriate to the theory and the concept, respectively.

Figure 4. a) Ray tracing index iteration on 1 (a), 30 (b), 60 (c), and 100 (d) in the 3D gradually increasing model 20 x 20 x 10 km; and b) more examples of ray tracing on complex models. The color represent the variant velocity between 10 and 250 km/s.
Figure 5. (a) the real model $100 \times 100 \times 50$ km (upside) and CRT model (downside), (b) inversion velocity perturbation model (upside), inversion result of CRT model (center side) and iteration versus RMS curve (downside) for $N_r = 10$ and $N_s = 10$, (c) $N_r = 10$ and $N_s = 30$, and (d) $N_r = 10$ and $N_s = 50$. The color of the CRT and perturbation model represent the perturbation velocity (%) of the initial model 4 km/s.

Figure 6. (a) True model $100 \times 50$ km (upside) and CRT model (downside), (b) inversion velocity perturbation model (upside) and inversion result of CRT model (downside) on norm damping value 10, (c) 20, and (d) 40. The color of the CRT and perturbation model represent the perturbation velocity (%) of the initial model 4 km/s.

Figure 7. The similar description in fig. 6 with the variant gradient damping (smoothing) value (b) 0, (c) 5, (d) 10, and (e) 20.
3.3. Implementation on Real Data

Implementation of this paper as a demonstration for validating that the application can be used for real data. The data is the events between the southern of Sumatera island and in the west of Java island. Detail of the real data and the parameter value can be seen in Tables 2 and 3, respectively. Based on the following data and parameter, the result of inversion in fig. 8(b) shows the cross-section being located in A-B black line in fig. 8(a) and perpendicular with the subduction line on the surface. Regarding the following picture, there are the high-velocity anomalies that are predicted as subduction zone in the subsurface as illustrated in the dash-red line. The result of the CRT model as shown in fig. 8(c) also shows good resolution on the model because the checkerboard pattern in the inversion result is clear around the slab prediction.

Table 2. Information on the real data

| Information            | Value |
|------------------------|-------|
| Total of source        | 3346  |
| Total of receiver      | 33    |
| Total of traveltime data | 33261 |

Figure 8. Slice location on the map shown by a black line (a), inversion tomography result and event location (b), and inversion CRT result (c) on every slice. Strip red line on the model shows the predicted direction of subduction on this model. The distance is on kilometer (km) unit and the color is the perturbation of the initial velocity (%).
Table 3. The parameter value for tomography on the real data

| Parameter                        | Value      |
|----------------------------------|------------|
| Tomography iteration value       | 10         |
| Ray iteration value              | 100        |
| Total block on every coor. (nx,ny,nz) | (11,11,11) |
| Norm damping value               | 600        |
| Gradient damping value           | 5          |
| CRT perturbation                 | 0.3        |
| Phase type of tomography         | P          |

4. Conclusions

Based on validation test of ray tracing module shows that resulted ray appropriates to the basic concept of the seismic wave propagation so that algorithm of ray tracing is relatively stable and can be used for forwarded modeling and to build the matrix kernel. Then, the results of inversion with the chosen parameters and the implementation of real data can provide evidence that the algorithms are works appropriated with the theory so that the application can be used for real data. Even though, more comprehensive validation tests of the application still need to be performed, for example by increased by variating data, and another tomographical features of this module will be added for increasing usability and robustness of this application.

Acknowledgments

We thank BMKG, the government institution for meteorology, climatology, and geophysics in Indonesia, for the open data catalog being provided in this paper.

References

[1] Sedlak P, Hirose Y, Enoki M, and Sikula J 2008 Arrival Time Detection In Thin Multilayer Plates On The Basis Of Akaike Information Criterion J. Acoustic Emission 26 pp 182-188
[2] Han L, Wong J, and Bancroft J C 2009 Time picking and random noise reduction on microseismic data CREWES Research Report – Vol. 21 pp 1-13
[3] Li F, Rich J, Marfurt K J, and Zhou H 2014 Automatic Event Detection On Noisy Microseismograms SEG Technical Program Expanded Abstracts pp 2363-2367
[4] Herrmann R B Introduction to Earthquake Seismology Saint Louis Univ. p 16-1
[5] Thurber C H 2000 Advances in Seismic Event Location ed N Rabinowitz Kluwer Academic Publishers pp 163-204
[6] Andika P P, Styawan Y, Suhendi C and Firdaus R 2019 Lindu Software: An Open Source Seismological Data Processing Using Python Framework To Relocate Hypocenter (Preliminary Software) IOP Conf. Ser.: Earth Environ. Sci. 318 012021
[7] Styawan Y, Andika P P, Suhendi C, Firdaus R, Sudibyo M R P, Erlangga I F and Ry R V 2019 The Preliminary Results of Lindu Software: A Free Seismological Data Processing Using Python Framework IOP Conf. Ser.: Earth Environ. Sci. 311 012078
[8] Rawlinson N and Sambridge M 2003 Advances in Geophysics Vol 46: Seismic Traveltime Tomography of The Crust and Lithosphere Canberra Academic Press p 3
[9] Um J and Thurber C 1987 A Fast Algorithm for Two-point Seismic Ray Tracing Bulletin of the Seismological Society of America 77 No 3 972-986
[10] Rawlinson N, Pozgay S, and Fishwick S 2010 Seismic Tomography: A Window Into Deep Earth Physics of the Earth and Planetary Interiors 178 101-135