Traveltime tomography using irregular parameterized grids: Solving the inverse problem

IMMO TRINKS

trinks@esc.cam.ac.uk

BULLARD LABORATORIES
LITHOS
UNIVERSITY OF CAMBRIDGE

Introduction

In the previous edition of the LITHOS report a new approach to travel time tomography was described (Trinks, 2001). It is based on triangulated models and allows a spatially irregular parameterization. The idea for this project arose from the desire for an inversion technique that is capable of inverting very densely sampled long offset data. The advantages of the new method over traditional travel time tomography algorithms are the efficient formulation of the forward and inverse problem and the possibility of adapting the model to the data. The forward problem is solved using initial value ray tracing with an interpolation of the travel times at the receiver locations. A comparable traveltime inversion based on triangulated models is described by Cox & Verschuur (2001). Their model parameterization is identical to ours but they invert for Common Focusing Points using one-way tomography while we follow the approach of McCaughey & Singh (1997) and plan to invert for velocity and interface depth simultaneously using two-way travel times and ray paths.

In 2001 the ray tracing algorithm was completed and optimized. The computation of the effect of small velocity perturbations onto the travel time led to the formulation of the linearized inverse problem. In this report the development of the algorithm and the implementation of the inversion will be described in detail. A description of the realisation of the forward problem may be found in Trinks, 2001.

Completion of the forward problem

The forward problem is solved using initial value ray tracing in an isotropic medium, as formulated by Farra (1990). The square of the slowness is interpolated linearly over triangular cells. The two-point problem is avoided by shooting an equidistant fan of rays and interpolating the travel times at the receiver locations. After initial tracing of a coarse fan of rays, the range of shooting angles of those rays that intersect a predefined receiver array is determined. Within this range additional rays are inserted until a satisfactory ray density is obtained wherever this is physically possible. At least one ray should emerge between each pair of receivers. **Figure 1** displays an example in which rays are traced from the source at 9.5 km inline distance towards

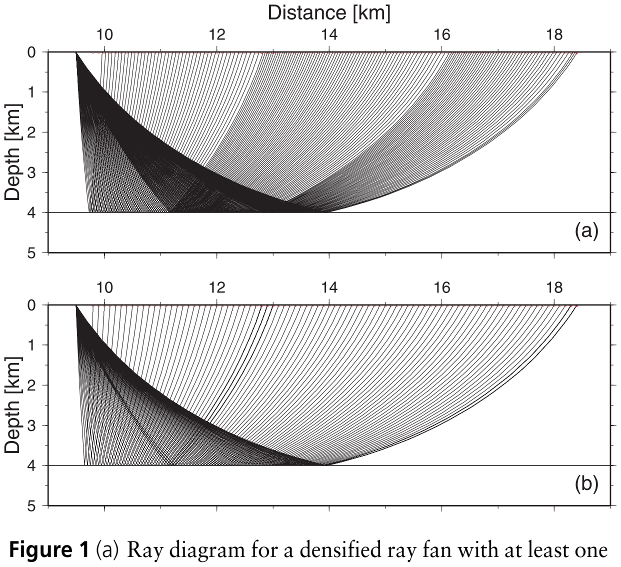

Figure 1 (a) Ray diagram for a densified ray fan with at least one ray between each pair of receivers, and (b) the corresponding ray fan with the two-point problem solved for each ray.

receivers that are located in 100 m intervals between 9.8 and 18.4 km. **Figure 1(a)** displays the ray fan for the interpolation scheme as described above. The solution of the two-point problem, where the exact ray path between the source and the receiver has to be found through variation of the shooting angle, is shown in **Figure 1(b)**. Depending on the required accuracy of the solution of the two-point problem, up to ten times more rays have to be shot than in the case of simple ray fan densification and travel time interpolation. The variable density of the ray fan in **Figure 1(a)** is due to the densification process, which halves the shooting angle if it is found that there is less than one ray between each pair of receivers.

Travel times corresponding to the synthetically shot rays are compared with the actually measured travel time data at the ray-emergence positions. Therefore the picked travel times are supplied separately, together with the positioning information of the receivers. Source and receiver positions are well defined in the formatted data and can be used to set up the geometry of the experiment. The travel time misfit is interpolated cubically at the location of the intersection between the ray and the receiver array (that is the ray-streamer intersection in the marine case). It does make more sense to interpolate the observed travel time at the ray emergence point than at the receiver location, since we are interested in the dependency of the travel time misfit from the path the ray has taken.

The accuracy of the travel time computation was benchmarked against the results of analytical travel time computations and compared with numerical results from the established ray tracing software *Jive3D*. In contrast to the new algorithm, which uses grids consisting of Delaunay triangles, *Jive3D* is based on a regular parameterization of the velocity model (Hobro & Singh, 1999). First a simple 1D velocity model with flat lying constant velocity layers was chosen and the travel times for reflected rays were computed. Additionally analytical travel times were computed for this case and compared with the results of the new algorithm. **Table 1** lists the constant velocities and the thicknesses of the layers of the 1D model. Each layer was defined by four node points. Travel times for reflections from the upper interfaces of the second, third and fourth layer were computed.

Layer	Velocity [km/s]	Thickness [km]
1	1.5	1
2	2.0	1
3	4.5	3
4	6.5	3

Table 1 Parameters for model with constant velocities.

The maximum difference between the analytical computed travel times and the corresponding ray traced values for the reflection from the first interface is 0.1 ms. Similar results were obtained for the reflections from the second and third interface.

Next a vertical velocity gradient was introduced into the first layer of the model described in **Table 1**. The velocity at the top of this layer is 1.5 km/s while the velocity at the bottom of the layer is 2.0 km/s. Two models with different node densities were used to test the new algorithm: the first model consisted of 2,000 irregular distributed velocity nodes while the second model was constructed of 10,000 velocity nodes. The *Jive3D* model was constructed of 2,448 nodes. Travel times of the reflected rays from the first interface were computed. Analytical travel times were computed by subdividing the model into a stack of very thin layers (8 mm thickness) with constant velocities that increased according to the velocity gradient. The offset of the analytical computed rays was obtained by applying Snell's law at each interface of those thin layers. Travel time deviations of the ray traced results from the two models of the new algorithm and the *Jive3D* model are shown in **Figure 2**.

Whilst the travel time curve of the new ray tracer for the model with 2,000 node points shows deviations up to 6 ms, the results for the finer parameterised model as well as the results obtained with *Jive3D* lie very close to the analytical solution. The travel time deviation of the new algorithm is not larger than 1 ms for the 10,000 node point model. An equivalent accuracy can be expected for the results of *Jive3D* in case of a similar finely parameterised model.

Even though at 10,000 the number of velocity node points seems to be high it has to be kept in mind that this is the number of node points in the whole model. Adaptive parameterisation could be

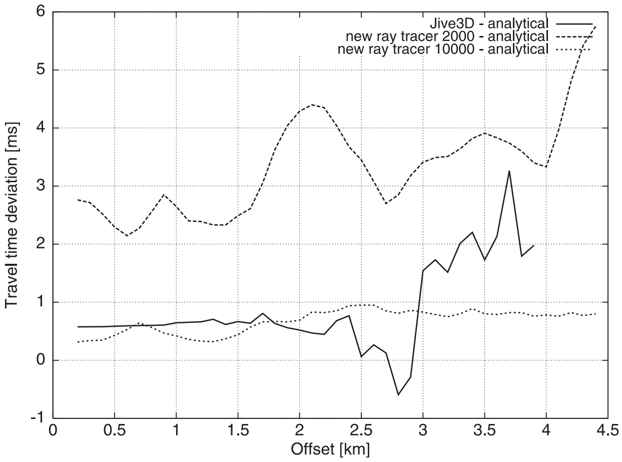

Figure 2. Travel time difference between the analytical computed travel times and Jive3D (solid line) and the new ray tracer for the model with 2,000 (thick dashed line) and 10,000 (thin dashed line) velocity node points respectively.

used to reduce the number of employed parameters drastically. For example horizontally elongated triangles could be used to represent the layered structure of the earth and the node point density could be adapted to the ray coverage in the model.

The relatively large travel time deviation of up to 6 ms in the case of the 2,000 node point model can be explained by the incomplete representation of the velocity gradient through the model. In the case of the new ray tracer the velocity values are only exact at the location of the velocity nodes. Within the triangular cells that connect the node points the square of the slowness is interpolated linearly. Therefore a fine parameterisation is necessary in order to obtain a higher accuracy of the travel time values. Practical considerations show that the picking uncertainty in real long offset data is in the range of 20 ms for shallow reflections and 40 ms for deeper, low frequent turning ray events (Carpenter, 2000). With the typical sampling interval for industrial seismic surveys being 2 ms, the accuracy of 1 ms obtained with the new ray tracing algorithm is satisfactory. Thus the calculation errors obtained can be considered sufficiently small for our needs. Furthermore the use of adaptive parameterisation will still improve the performance of the new ray tracer, by making it possible to place more velocity nodes in regions of high ray density while the number of parameters can be reduced in areas that are poorly constrained.

Computation of the ray perturbations

The implementation of the inversion requires that we know the effect that a small velocity change in our model has on the travel time of each ray. To be able to relate the travel time to the model parameter it is necessary to calculate the Fréchet derivatives for each velocity node that has an effect on the path of a ray. This is done by stepping along the ray path and adding the individual Fréchet contributions of consecutive triangles for each involved node. Obviously the influence of a single node is greater, the longer the ray path is in the cells that are adjacent to that node. The calculation is done by differentiating Farra's (1990) expression for the analytical travel time in respect to the individual model parameter, i.e. slowness squared. The Fréchet derivatives are stored in the matrix **G**. This matrix has as many rows as rays that illuminate the model, and as many columns as the model has parameters. Model parameters of interest are not only the velocity nodes but also the interface nodes that describe the depths of reflecting or refracting interfaces. So far only the Fréchet derivative calculation for the velocity nodes is implemented.

The computation of these Fréchet derivatives can be tested numerically. Therefore one single ray is traced between a fixed source and receiver location in a velocity model with constant velocity gradient. Then the velocity value at one node point of a triangle that is intersected by the ray is perturbed by a small amount (0.1%, 1.0%, and 5.0%), thereby still assuming a linear relationship between the change in travel time and the change of the model. Next the two-point problem is solved for rays between the source and the receiver in the perturbed model with an accuracy of the two-point solution of ±1cm. The difference in travel times in the perturbed and the unperturbed model in relation to the difference in slowness squared gives the numerical value for the Fréchet derivative at the specific velocity node point. Our numerically determined values agree well with the analytical results.

Formulation of the inverse problem

The solution of the inverse problem involves a search over the model space for the most plausible model *m* that is able to explain the recorded data *d*, in this case the recorded travel times. Therefore we chose a starting model that is likely to be close to the real subsurface. We trace rays in this model and obtain a first set of synthetic travel times. The comparison of these computed travel times with the actually measured data results in the travel time misfit or travel time residual Δd. The number of parameters used is much too large to search the complete model space for the model that minimises this residual best. Under the assumption that small changes of the model Δm result in only small changes of the travel times Δd we can establish a linear relationship:

$$\Delta d = \mathbf{G}\Delta m \qquad (1)$$

with **G** being a matrix. We apply now small changes to the model in order to reduce the misfit. The new model remains in the region of linearity that surrounds the previous model. This procedure is repeated consecutively until a satisfactory reduction of the travel time residual is achieved.

For models with a small number of parameters it is possible to calculate the model update directly by inverting **G**:

$$\mathbf{G}^{-1} = \frac{\hat{\mathbf{G}}^t}{\det \mathbf{G}} \qquad (2)$$

with $\hat{\mathbf{G}}$ as matrix consisting of the adjoints of the corresponding elements (Bronstein & Semendjajev, 1991). The costly computation of $\hat{\mathbf{G}}$ is not feasible for models with more than a few hundred parameters.

Therefore we take the following approach to calculate the model update Δm: to reduce the misfit r between the observed data d_{obs} and the modeled data d_m we formulate an optimization function S:

$$\begin{aligned} S &= (\underbrace{d_{obs}-d_m}_{r})^t\,\mathbf{C}_d^{-1}\,(\underbrace{d_{obs}-d_m}_{r}) \\ &+ (m-m_{prior})^t\,\mathbf{C}_m^{-1}\,(m-m_{prior}) \end{aligned} \qquad (3)$$

with m as the current model and m_{prior} as the previous model, and with $\mathbf{C}_d$ and $\mathbf{C}_m$ as the data, respectively model covariance matrixes. $\mathbf{C}_d$ contains the travel time uncertainties while $\mathbf{C}_m$ contains the model uncertainties, which are divided into velocity and interface depth uncertainties. The covariance of two features measures their tendency to vary together. Hence we can use the covariance matrices to include *a priori* information that we have about the subsurface, like for example velocity information from borehole measurements. The second term in equation 3 is the regularisation term in the inversion.

Under the linearising assumption of equation 1 can we assume that the data of the $(n+1)$th iteration can be expressed as

$$d(m_{n+1}) = d(m_n) + \mathbf{G}(\underbrace{m_{n+1}-m_n}_{\Delta m_n}) \qquad (4)$$

with the n-th model update Δm_n. The optimizing function can then be written as

$$\begin{aligned} S(m_{n+1}) &= (r-\mathbf{G}\Delta m_n)^t\,\mathbf{C}_d^{-1}\,(r-\mathbf{G}\Delta m_n) \\ &+ (m_{n+1}-m_{prior})^t\,\mathbf{C}_m^{-1}\,(m_{n+1}-m_{prior}) \end{aligned} \qquad (5)$$

Hence we calculate the gradient of S to

$$\begin{aligned} \nabla S &= -2\mathbf{G}\mathbf{C}_d^{-1}(r-\mathbf{G}\Delta m_n) \\ &+ 2\mathbf{G}\mathbf{C}_m^{-1}(m_n+\Delta m_n-m_{prior}) \end{aligned} \qquad (6)$$

In order to find the model update Δm_n that minimizes the residual r we set $\nabla S = 0$ and obtain

$$\mathbf{G}^t\mathbf{C}_d^{-1}r-\mathbf{C}_m^{-1}(m_n-m_{prior}) = (\mathbf{G}^t\mathbf{C}_d^{-1}\mathbf{G}+\mathbf{C}_m^{-1})\Delta m_n \qquad (7)$$

This equation can be transformed into a formulation that does not contain $\mathbf{C}_m^{-1}$ so that no inversion of $\mathbf{C}_m$ is necessary:

$$\begin{aligned} &\underbrace{\mathbf{C}_m\mathbf{G}^t\mathbf{C}_d^{-1}(d_{obs}-d_{m_n}) + (m_{prior}-m_n)}_{b} = \\ &\qquad \underbrace{(\mathbf{C}_m\mathbf{G}^t\mathbf{C}_d^{-1}\mathbf{G}+\mathbf{I})}_{\mathbf{A}}\,\underbrace{\Delta m_n}_{x} \end{aligned} \qquad (8)$$

with the identity matrix **I**. For the first iteration $(m_{prior} - m_n) = 0$.

We can regard equation 8 as the simple linear equation

$$b = \mathbf{A}x \qquad (9)$$

with A as the curvature of the misfit function S and b as the direction of it's steepest gradient. We can then solve eqn. 9 using the biconjugate gradient algorithm as described in *Numerical Recipes* (Press *et al.*, 1992). This algorithm takes advantage of the sparseness of the matrices **G** and $\mathbf{C}_m$ and allows a very fast calculation of the model update.

Currently the Fréchet derivative calculation is tested in order to make sure that **G** is correct. First synthetic tests of the inversion algorithm were conducted using a simple model with a vertical velocity gradient. We perturbed this model with a high velocity lens in order to obtain pseudo-real travel time data. The travel time data of the perturbed model is compared with the travel times of the unperturbed model to computed the travel time misfit and to invert for the first model update.

The above formulation of the misfit function (eqn. 3) requires the estimation of the model covariance by including *a priori* information. The subjective construction of the model covariance matrix can be avoided by applying a Tikhonov regularization as described by McCaughey & Singh (1997). Thereby we are able to regularize the problem by damping the spatial derivatives of the model. In order to obtain a smooth solution second spatial derivatives of the model parameters should be minimized.

Next Steps

To complete the inversion algorithm it is necessary to implement the final loop of iterations for the reduction of the misfit. Additionally we need to calculate the Fréchet derivatives for the interface parameters and to include them in the inversion. Tikhonov regularisation will be implemented and it's effect will be studied.

The testing of the algorithm will be carried out on simple synthetic velocity models and travel time data. Next we plan to invert a high resolution long offset seismic data set. This will be data from a two-ship seismic survey with simulated streamer length of up to 30 km from the Rockall or Faeroes region. Interesting features to resolve in this data are thick basalt flows (Faeroes data set) and thin, discontinuous basalt sills within the sediments (Rockall trough data) as well as the structure of the basement underneath.

Travel time arrivals for reflected and refracted waves will be picked using a semi-automated algorithm (*PickEd*) that was developed within the LITHOS group (Di Nicola-Carena, 1999) and recently converted from MATLAB to a portable version written in C (Oates, 2002).

Acknowledgements

I thank Satish Singh, Chris Chapman, Penny Barton, Adam Cherrett and especially Miguel Bosch for their support and good ideas. I am grateful to the LITHOS consortium for funding this studentship.

References

BRONSTEIN, I.N., & SEMENDJAJEV, K.A., 1991, *Taschenbuch der Mathemathik*, 25.th ed., B.G. Teubner Verlagsgesellschaft, Stuttgart - Leipzig.

CARPENTER, M., 2002, Seismic imaging using densely sampled, very long-offset data, M.Phil. thesis, Cambridge University.

MCCAUGHEY, M., & SINGH, C.S., 1997, Simultaneous velocity and interface tomography of normal-incidence and wide-aperture seismic travel-time data, GEOPHYS. J. INT., **131**, 87 - 99.

COX, B.E., & VERSCHUUR, D.J., 2001, Tomographic inversion of focusing operators, 63rd Annual Internat. Mtg., EAGE, *Extended Abstracts*, Session M-35.

DI NICOLA-CARENA, E., 1999, Analysis of real long offset data using a semi-automated traveltime picking procedure, *LITHOS Science Report 1999*, 55-65.

FARRA, V., 1990, Amplitude computation in heterogeneous media by ray perturbation theory: a finite element approach, *Geophys. J. Int.*, **103**, 341-354.

HOBRO, J. W. D., & SINGH, S. C., 1999, Joint interface and velocity estimation in three dimensions (Jive3D), *LITHOS Science Report 1999*, 11-20.

HOBRO, J. W. D., 2000, Seismic travel-time tomography in three dimensions: Synthetic experiments and results obtained using *Jive3D*, *LITHOS Science Report 2000*, 11-20.

OATES, P., 2002, Some preliminary results from *PickEd* version 2, *LITHOS Science Report* 2002.

PRESS, W., TEUKOLSKY, S., VETTERLING, W., & FLANNERY, B., 1992, *Numerical Recipes in C: The Art of Scientific Computing*, 2nd ed., Cambridge Univ. Press.

TRINKS, I., 2001, Traveltime tomography using irregular parameterized grids, *LITHOS Science Report 2001*, **3**, 53-57.