MOD 3.2 3D reflection tomography designed for complex structures

Fabrice Jurado, Delphine Sinoquet, Andreas Ehinger

3D reflection tomography designed for complex structures

Fabrice Jurado, Delphine Sinoquet and Andreas Ehinger, Institut Francais du Petrole.

Summary

A 3D reflection tomography that can determine correct subsurface velocity structures is of strategic importance for an effective use of 3D prestack depth migration. We have developed a robust and fast 3D reflection tomography that is designed to handle complex models.

We use a B-spline representation for interface geometries and for the lateral velocity distribution within a layer and we restrict the vertical velocity variation to have a constant gradient.

We solve the ray tracing problem by use of a bending method with a circular ray approximation within layers. For the inversion we use a regularized formulation of reflection tomography which penalizes the roughness of the model. The optimization is based on a quadratic programming formulation and constraints on the model are treated by the augmented Lagrangian technique.

We show results of ray tracing and inversion on a rather complex synthetic model.

Introduction

Ehinger and Lailly, 1995, have shown the interest of reflection tomography for computing velocity models adequate for the seismic imaging of complex geologic structures. In 2D, reflection tomography has proved its effectiveness in this context (Jacobs et al., 1995).

In 3D, Guiziou et al., 199 1, have developed a ray tracing based on a straight line ray approximation within a layer and an inversion of poststack data. But it suffers of a non derivability of its traveltime formula due to the Gocad interface representation.

We describe a 3D tomography that handles models with the necessary derivability and allows inversion of more complex kinematics by the use of a more accurate traveltime calculation.

Model description

We choose a blocky model representation of the subsurface, each layer being associated with a geological macrosequence. A velocity law has to be associated with each layer (Figure 1). The form of the velocity law is $v(x, y, z) = v_0(x, y) + k.z$ where v_0 is the lateral velocity distribution (described by cubic B-spline functions) and k is the vertical velocity gradient.

Using blocky models can lead to difficulties associated with the possible non-definition of the forward problem (situations where there is no ray joining a source to a receiver) and more generally to all kind of difficulties involved in discontinuous kinematics. The blocky model representation allows velocity discontinuities as they exist in the earth and thus to straightforwardly integrate a priori information on velocities (see Lailly and Sinoquet, 1996, for a general discussion on blocky versus smooth model s for seismic imaging of complex geologic structures).

We use a cubic B-spline representation for interface geometries. This gives the (at least) C^1 regularity required for proper ray tracing calculations. Following Clarke et al., 1994, the B-spline interfaces are defined in the whole domain and can cross each other. The part of an interface where the velocity is the same on both sides is called a virtual part (Figure 1); this part exists only for convenience, it has thus no physical meaning.

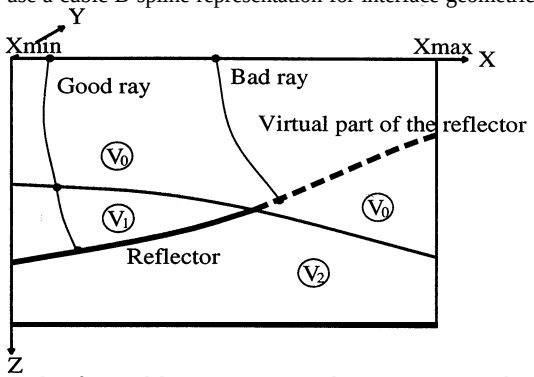


Fig. 1 Example of a model presenting a pinch-out. Continuous lines are the reflecting parts of interfaces, the dashed line is a virtual part that has no physical meaning. Specifying a ray signature allows to sort good and bad rays.

Ray tracing method

We compute normal (zero-offset) rays as well as multi-offset rays. We have to define the ray paths in which we are interested in. This consists in specifying the *ray signature* which is the ordered sequence of interfaces crossed by the ray before and after reflection. In other words, specifying the ray signature means specifying the nature of the seismic event we are interested in. In particular it can be used to remove artificially reflected rays i.e. rays that are reflected on the virtual part of an interface (Figure 1).

Two-point ray tracing

The two-point ray tracing method (zero or multi-offset) is inspired from the classical bending method. It consists of:

- a ray initialization, i.e. we choose a trajectory that starts at the source, hits the reflector and ends at the receiver satisfying a given signature. This trajectory is defined by the successive impact points on the crossed interfaces;
- a formula for calculating the traveltime between two successive impact points;
- the bending itself, i.e. the optimization technique that moves impact points until the total traveltime is stationary (Fermat's principle).

Ray initialization

For a zero-offset ray, the initial trajectory for the bending method is obtained by finding the successive intersection points of all interfaces with a circle (why a circle? see below...) starting from the source, with a given tangent vector (i.e. shooting angle), so as to reach the nearest interface. We apply Snell's law at the impact point to simulate the transmission at the interface, and iterate the circle-interface intersection procedure until the reflector is reached. The ray initialization with a given tangent vector can be very useful in the case of multiple arrivals in order to guide the bending method so as to find the desired arrival. If the tangent vector direction is not specified then by default the vertical direction is used.

In the case of multi-offset CMP acquisition, an offset continuation strategy is applied. A new ray is initialized by copying, from the previously calculated ray, the sequence of impact point coordinates except for the source and receiver locations which are updated according to the considered offset.

Traveltime formula

For the calculation of the traveltime between two successive impact points, Guiziou et al., 1991, have proposed a formula based on a straight line ray approximation. To improve the accuracy of the traveltime calculation for velocity distribution with constant vertical gradient (see ray tracing results, Figure 5), we use a formula suited for such a velocity. In the case of constant lateral velocity, it is well known that the ray trajectory is a circle that lies in the vertical plane defined by the two successive impact points P_{n-1} and P_n. The traveltime t_n is calculated by (see Virieux, 1990):

$$t_n = \left| \frac{1}{2k} \ln \left(\frac{1-\cos\alpha_2}{1+\cos\alpha_2} \frac{1+\cos\alpha_1}{1-\cos\alpha_1} \right) \right|$$

where angles α_1 and α_2 are explained on Figure 2. Note that we have translated the coordinate system origin to P_{n-1} and rotated the *(x,y)* plane to be in the vertical plane in which the ray lies (i.e. we deal with a 2D problem). Then the circle center depth is $z_C = -\frac{v_0}{k}$ and the x_C coordinate is obtained by finding the intersection between this plane and the median of segment $[P_{n-1}, P_n]$.

In the case of lateral velocity variations, v_0 is approximated by the mean of the velocity values at $P, P_n, n-1$, and M (Figure 2). In the

case of strong lateral velocity variations, this approximation can be poor. In order to improve the accuracy of the computed traveltime, we can then introduce ghost interfaces within the layer. The ghost interface has no physical meaning: its role is just to allow an extra impact point, i.e. one more degree of freedom, so as to make the approximation more accurate.

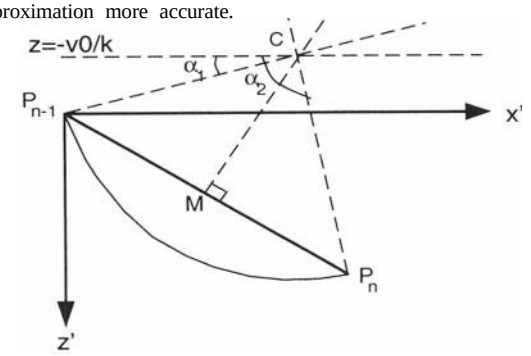


Fig. 2 Description of the traveltime calculation for circular ray path. C is the center of the circle, P_{n-1} and P_n are successive impact points. v_0 is the mean of the velocity values at P_{n-1}, P_n, M. k is the vertical velocity gradient.

Bending method

We define the total traveltime between the source P_0 and the receiver P_N by $t_{P_0 \to P_N} = \sum_{n=1}^{N} t_n$ which is a differentiable function of impact point coordinates $(x_n, y_n)_{n=1,N-1}$ (thanks to C^1).

We look for impact point locations that make the total traveltime minimum. This is done using a quasi-Newton method which requires only calculation of the first derivative of the traveltime with respect to impact point coordinates. This method has asymptotically the same convergence rate as a Newton method but is much cheaper and also much more stable.

Since interfaces can cross each other, the ray signature may change from one iteration of the bending to another (whenever an impact point moves across another interface) and the bending algorithm can find rays that violate the specified signature. We allow the bending to find such rays but we discard them by checking the signature a posteriori. Thus the bending algorithm is not slowed down by extensive signature management and at the same time we are sure to discard erratic rays. Note that finding no ray for a given source-receiver pair is quite acceptable since we cannot be sure that there exists such ray when dealing with a blocky model.

Also, as in all bending methods, there is no guarantee, in the case of multiple arrivals, to find the arrival of interest: the solution depends on the ray initialization, An a posteriori check of the shooting angle of the ray allows to detect a change of the traveltime branch.

Despite this minor limitation our ray tracing has been designed to be fast (thanks to the quite simple traveltime formula) and reliable (thanks to the B-spline representation of surfaces leading to the differentiable definition of the total traveltime). The bending algorithm converges towards a trajectory with zero traveltime derivatives with respect to impact point positions. This is an important condition to be fulfilled for correct calculation of the Jacobian.

Tomographic inversion method

The tomographic inverse problem consists in finding the model - namely, the B-spline parameters of the lateral velocity $v_0(x, y)$ and the interface depths $Z(x, y)$ for each layer (the constant velocity gradient has a given and fixed value) - that yields traveltimes that match the data. The data of the inversion are the prestack traveltimes. These data could be obtained for instance from zero-offset (poststack) traveltimes together with offset move-out information. We are able to invert multivalued traveltimes if different branches of a traveltime curve are defined by different shooting angles of the zero-offset ray. This a priori information can be read from the zero-offset data (first derivatives of the traveltime surface). Inversion of prestack traveltimes gives the possibility to invert more complex kinematics as compared to the direct inversion of stacking velocities. We also expect better stability since prestack traveltimes yield a bigger amount of information than the associated stacking velocity.

As shown in Delprat-Jannaud and Lailly, 1993, the inverse problem has to be regularized. Indeed, some parts of the model may not be illuminated by rays and thus are not determined by the data. We have chosen to regularize by penalizing the second derivatives of the difference between the actual model and an a priori model. We thus solve the following minimization problem. where *NT* is the number of traveltime data and the model *m* is composed of *NV* velocity laws $(v_0^{iv})_{iv=1,NV}$ and of NZ interface depths $(z_{iz})_{iz=1,NZ}$:

$$\min_m \sum_{it=1}^{NT} \varepsilon_{it}^2 (t_{it}(m) - t_{it}^{obs})^2 + \sum_{iv=1}^{NV} \int D^2 (z_{iz} - (z_{iz})^{a\ priori}) dxdy + \sum_{iz=1}^{NZ} \int D^2 \left(v_0^{iv} - (v_0^{iv})^{a\ priori}\right) dxdy$$

with $D^2 f = \varepsilon_x^2 \left(\frac{\partial^2 f}{\partial x^2}\right) + \varepsilon_y^2 \left(\frac{\partial^2 f}{\partial y^2}\right) + \varepsilon_{xy}^2 \left(\frac{\partial^2 f}{\partial x \partial y}\right)$.

The first term in the objective function measures the misfits between the traveltimes computed by the ray tracing in the current model *m* and the traveltime data. The regularization term means that we are looking for a model with a roughness close to the a priori model or, if there is no a priori model, a small roughness.

ε represent weights which calibrate in the objective function the different terms according to their physical meaning. ε_{it} represents the inverse of the uncertainty associated with traveltime t_{it}^{obs}. Choosing the weights $\varepsilon_{(x,y,xy)}$ correctly comes down to finding the appropriate compromise between possibly contradictory pieces of information (see Lailly and Sinoquet, 1996).

The minimization of the non quadratic objective function is carried out with the Gauss-Newton method. In fact, for better stability, it is important to put constraints on the model. Each iteration thus consists in solving a constrained least-squares problem. We make use of a quadratic programming formulation which allows the inversion of a large number of data: with this formulation we just need to store the Hessian matrix instead of the Jacobian matrix. To handle equality and inequality constraints we make use of a specific implementation of the augmented Lagrangian method. We can thus, for example, fix the velocity at the surface or at well locations, or constrain the depth or the slope of an interface, or constrain the relative position of two interfaces.

Applications of ray tracing and inversion

Synthetic model

We have created a synthetic model to test our ray tracing and tomographic inversion, The model is inspired by a real salt dome structure

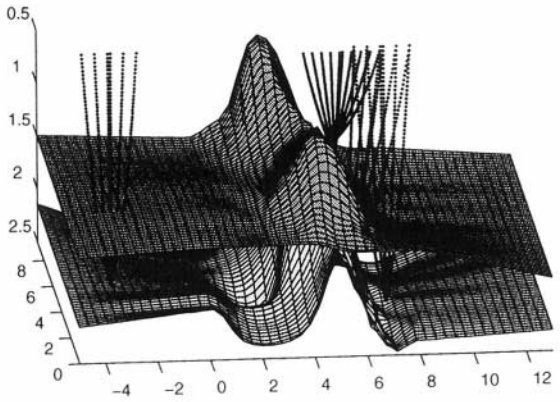


Fig. 3 Ray tracing on the synthetic model The depth of the ternary base vanes from 600m to 1800m and the depth of the chalk base varies from 600m to 2800m (see Figures 6 and 7).

in the North Sea and consists in total of 7 interfaces. The upper two interfaces (Figure 3) are the tertiary base and the chalk base. Each interface is explicit in the depth direction *(Z(x, y))* and represented by 30 x 15 B-spline coefficients on the independent axes. The model extends from - 5000m to 13000m in *x* direction and from 0m to 9400m in *y* direction.

The lateral variations of the velocity in the first layer are shown in Figure 6 (bottom right). The constant vertical velocity gradient is $0.55s^{-1}$. For the second layer, the lateral velocity is represented in Figure 7 (bottom right). The vertical gradient is $1s^{-1}$. Each lateral velocity distribution is represented by 14 x 8 B-spline coefficients.

The acquisition survey consists in zero-offset shots and CMP gathers. 360 x 188 zero-offset shots are positioned on a regular grid of 50m width. We place 37 x 24 CMPs on a regular grid of 400m width with 60 offsets varying from 50m to 3000m The zero-offset shots were first calculated by an accurate ray tracing developed by Clarke et al., 1994, in order to access the shooting angles in a simple way rather than by an interpretation. Using these shooting angles we then recomputed the zero-offset data and the multi-offset data with our ray tracing.

Ray tracing results

In Figure 5, we present the improvement of the circular ray formula with respect to a straight line ray approximation. Figure 5 (left) shows results of ray tracing using the circular ray approximation compared to a straight line ray approximation, i.e. in each layer the ray is a line and the traveltime is computed using the mean slowness value (see Guiziou et al., 1991). We see that the difference can be large especially for large offsets. In fact, the difference would be larger when dealing with more complex interfaces. The ray tracing with the circular ray approximation is faster than that with the straight line ray approximation when no ghost interfaces are used; moreover in practice one ghost interface is necessary for the straight line ray approximation to deal with vertical velocity gradient and increase the accuracy, its CPU time is then at least multiplied by 3.

To certify our ray calculation, Figure 5 (right) compares results between the ray tracing with circular ray approximation and the ray tracing of Clarke et al., 1994. We see the very good agreement even for large offsets. For this common mid-point, the *exact* ray tracing found three different branches but two of them are very close, so we have fed the bending method with only two really different shooting directions and have found only two branches. On Figure 4, the results of zero-offset raytracing with the shooting method and the bending method (circular ray approximation) show also that the bending method can have difficulties to find the desired arrivals when the initialization shooting directions are too close for two branches. The advantage of our ray tracing is to be 20 times faster (zero-offset shots) than the *exact* ray tracing.

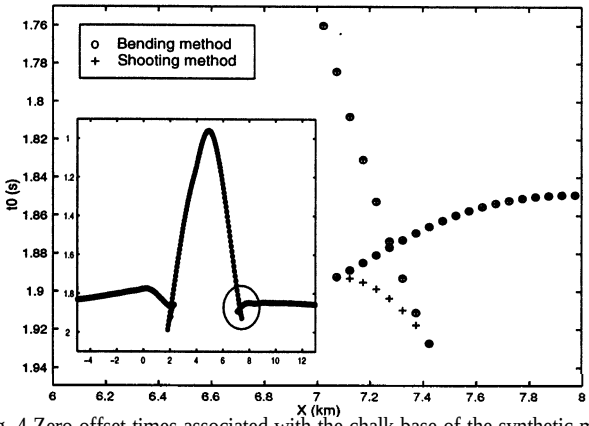


Fig. 4 Zero-offset times associated with the chalk base of the synthetic model along a line *y* = 3025m They were obtained with a shooting method (Clarke et al., 1994) and with the bending method (with the circular ray approximation). The bending method retrieved two branches of the traveltime curve but was not able to retrieve the third branch due to an initialization direction too close to that of the second branch which then was calculated twice.

Inversion results

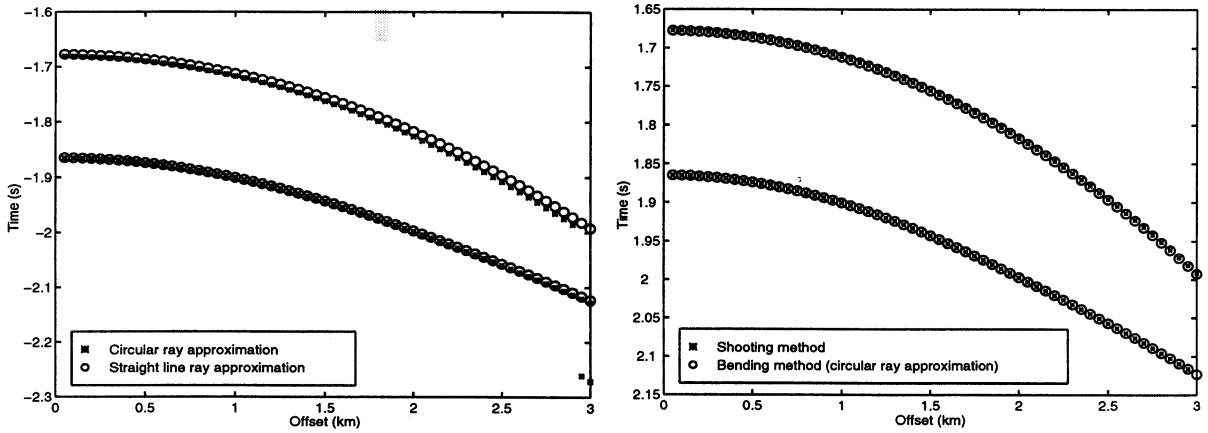
We performed inversion of the zero-offset and multi-offset traveltime data associated with the two first layers of the synthetic model by a layer-stripping approach.

First, we invert for the tertiary base starting from an initial constant horizontal interface and an initial constant velocity; it leads to an inversion for 450 B-spline coefficients for the interface and 112 coefficients for the velocity. The model was regularized by means of second derivatives without any a priori model. Also weights on the traveltimes data were tuned to give sufficient importance of the large offset data with respect to the (huge) number of zero (or small) offset data. We invert almost 130000 traveltime data.

Figure 6 shows the exact tertiary base compared with the interface obtained after 17 iterations of the inversion and the same comparison between velocities. The RMS traveltime misfit is 0.2ms (only 5 iterations were necessary to obtain a RMS misfit smaller than lms).

We now fix the tertiary layer to the obtained result and invert the chalk base and the lateral velocity, again starting from constant values. Figure 7 shows the exact chalk base compared with the interface obtained after 14 iterations and the same comparison between velocities. The RMS traveltime misfit is 0.7ms.

Only a few traveltime misfits remain large: they correspond to a change of branch of the traveltime curve, the initialization shooting angles being not sufficient to distinguish the different arrivals (see



Fig, 5 Triplicated multi-offset times associated with the chalk base of the synthetic model for a CMP (*x* = 6775m) *y* = 4125m). Left side: comparison between our bending method with the circular ray approximation and a bending method with a straight line ray approximation. Right side: comparison between our bending method with the circular ray approximation and a shooting method (Clarke et al., 1994). Two initialization angles for the bending provide the two branches.

section on ray tracing results).

The CPU time amounts to 20 and 45 minutes for one iteration of the inversion for layer 1 and 2 (respectively) on a Silicon Graphics Indigo² workstation (including ray tracing).

Conclusions

The 3D reflection tomography described in this paper has proven to be efficient for a geologic synthetic structure. Using B-spline representation of the interfaces and of the velocity gives the necessary regularity for the ray tracing and the inversion. The ray tracing relies on a bending method which is able to compute multiple arrivals providing that we initialize the process with reliable shooting angles (first derivatives of zero-offset data). The bending method combined with the circular ray approximation allows fast calculation and gives a sufficient accuracy even with large velocity variations. The regularized inversion formulation with constraints is robust and can overcome the difficulties associated with irregular model illumination by rays which is particularly true for complex structures.

Acknowledgements

This research was carry out as part of the Kinematic Inversion Methods consortium project (KIM). The authors hereby acknowledge the support provided by the sponsors of this project. We thank Elf Aquitaine for providing us with models used to create the synthetic model. We thank Richard Clarke for the interesting comparisons done with his ray tracing, Fatmir Hoxha for interesting discussions on circular ray calculation and Patrick Lailly for his fruitful comments.

References

Clarke, R.A., Jannaud, L.R. and Peinado, M., 1994, Ray shooting in 2D and 3D complex, blocky media, PSI 1994 annual report, Institut Francais du P&role, Rueil-Malmaison, France.

Delprat-Jannaud, F. and Lailly, P., 1993, Ill-posed and well-posed formulation of reflection tomography problem, J. Geophys. Res., 98, 6589-6605.

Ehinger A. and Lailly P., 1995, Velocity model determination by the SMART method, Partl: Theory, 65th Ann. Internat. Mtg., Soc. Expl. Geophys., Expanded Abstracts, 739-742.

Guiziou, J.L., Mallet, J.L., Nobili, P., Anandappane, R. and Thisse, P., 1991, 3D Ray tracing through complex triangulated surfaces, 61st Ann. Intemat. Mtg., Soc. Expl. Geophys., Expanded Abstracts, 1497-1500.

Jacobs, J.A.C, Sinoquet, D. and Duquet, B., 1995, Velocity model determination by the SMART method, Part2: Application, 65th Ann. Internat. Mtg., Soc. Expl. Geophys., Expanded Abstracts, 1425-1428.

Lailly, P. and Sinoquet, D., 1996, Smooth velocity models in reflection tomography for imaging complex structures, Geophys. J. int., 124, 349-362 (Treitel section).

Virieux, V., 1990, Propagation in inhomogeneous media, Ray theory, Institut de Geodynamique, Universite de Nice.

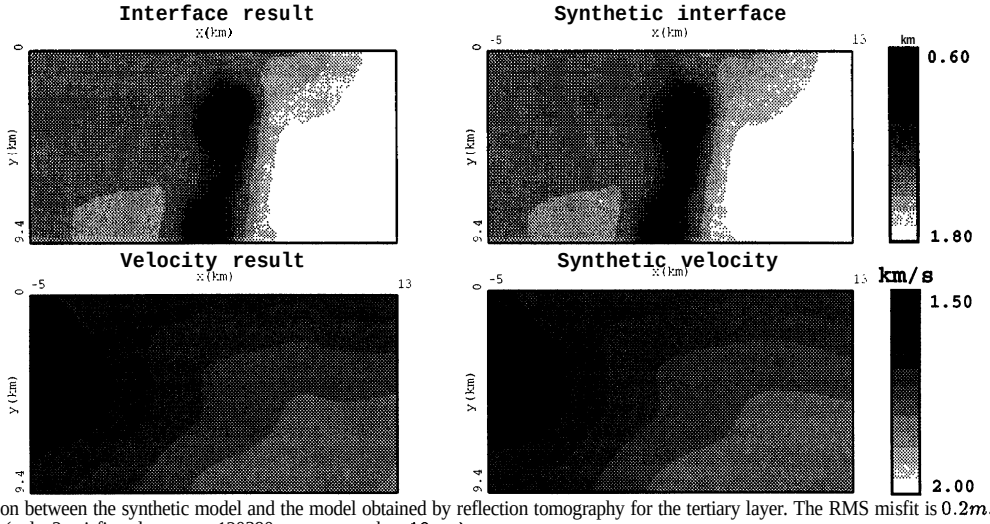


Fig. 6 Comparison between the synthetic model and the model obtained by reflection tomography for the tertiary layer. The RMS misfit is $0.2ms$, the maximum misfit is $39.7ms$ (only 2 misfit values over 120390 are greater than $10ms$).

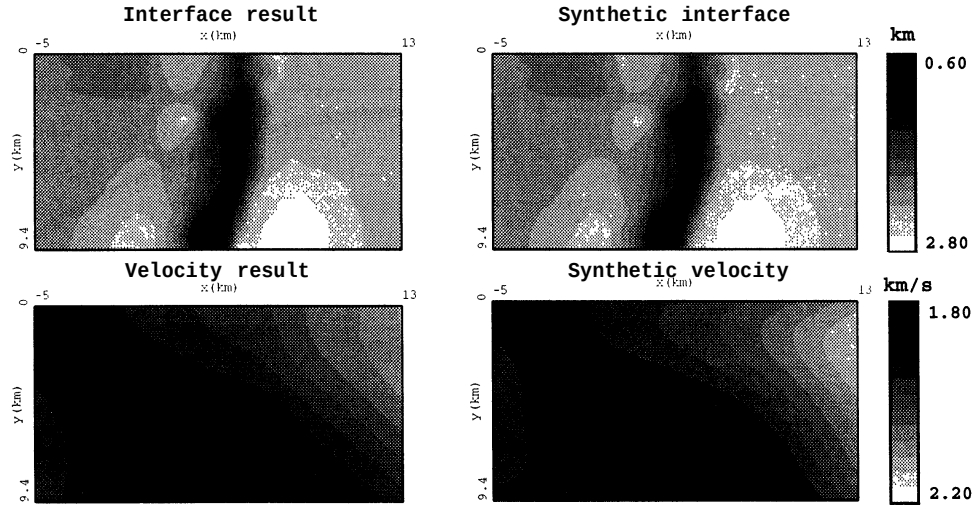


Fig. 7 Comparison between the synthetic model and the model obtained by reflection tomography for the chalk layer. The RMS misfit is $0.7ms$, the maximum misfit is $77.8ms$ (only 45 misfit values over 129300 are greater than $10ms$).