

# The Common Reflection Surface Stack – Part II: Application

R. Jäger, J. Mann, G. Höcht<sup>1</sup>

*keywords: imaging, zero-offset simulation*

## ABSTRACT

*In this part we apply the CRS stack to a synthetic dataset. We show that the CRS stack not only improves the resulting simulated zero-offset(ZO) section but also provides important wavefield attributes, which can be used to derive a macro-velocity model. We compare the data-derived attributes determined by the CRS stack with the model-derived (forward calculated) attributes and thus confirm the correctness of these parameters.*

*The CRS stacking surface depends on three parameters, which are the emergence angle  $\alpha$  of the ZO ray, as well as two wavefront curvatures  $R_N$  and  $R_{NIP}$  associated with the ZO ray, namely the normal wave and the normal-incidence-point (NIP) wave. We also address the problem of determining an optimal parameter triple for each ZO sample to be simulated. This triple is expected to fit a CRS traveltime surface best to the reflection events. To make the CRS stack more attractive regarding computational costs much effort has to be put on a strategy to find these optimal parameter triples for all ZO samples. This leads us to the concept of global optimization.*

*For the implementation of the CRS procedure we make use of the hyperbolic second order Taylor expansion of the CRS stacking surface since this representation is more suitable to handle irregular acquisition geometries. In addition, the hyperbolic traveltime expansion formula can be simplified in specific gathers and thereby enables different search strategies.*

## INTRODUCTION

Although we have already successfully applied the CRS stack to real datasets we decided to show here an application of the CRS stack to a synthetic dataset only. The reason for this is that we want to compare the data-derived attributes,  $\alpha$ ,  $R_{NIP}$  and  $R_N$ , determined by the CRS stack, with the model-derived (forward calculated) attributes of the used synthetic model. For real datasets we could show you the experimental

---

<sup>1</sup>**email:** rjaeger@gpiwap4.physik.uni-karlsruhe.de

attribute sections but we would not be able to convince you of the correctness and excellent quality of these attribute sections as this can only be done by a comparison with model-derived attributes.

For the implementation of the CRS stack we make use of the hyperbolic second order Taylor expansion of the CRS stacking surface. This is due to the fact that in real datasets the seismic traces are usually not located on a regular grid. Therefore, the parametric CRS travelttime formula given in part I would require trace interpolation. On the contrary, using the hyperbolic travelttime approximation saves computational time, since no trace interpolation is needed at all.

## MODEL

To test the CRS stack we created a synthetic common-offset data set for the inhomogeneous model shown in fig. 1 consisting of five dome-like interfaces. We simulated multi-coverage primary reflection data from all interfaces using the ray method for half-offsets  $h = 0$  m to  $h = 1250$  m in increments of  $\Delta h = 25$  m and at midpoint intervals of  $\Delta x_m = 25$  m in the range  $-1000$  m  $\leq x_m \leq 11000$  m. As seismic signal we used a Ricker wavelet of 30 Hz peak-frequency and a sampling interval of 4 ms. Finally, random noise was added to the synthetic reflection data. It is important to remark that, for the above synthetic noisy data example, we chose the signal(peak amplitude)/noise(variance) ratio of 4. This implies that reflection events of deeper interfaces can hardly be visually identified in neither the CO sections nor the CMP gathers. A subset of the synthetic ZO section is shown in fig. 2.

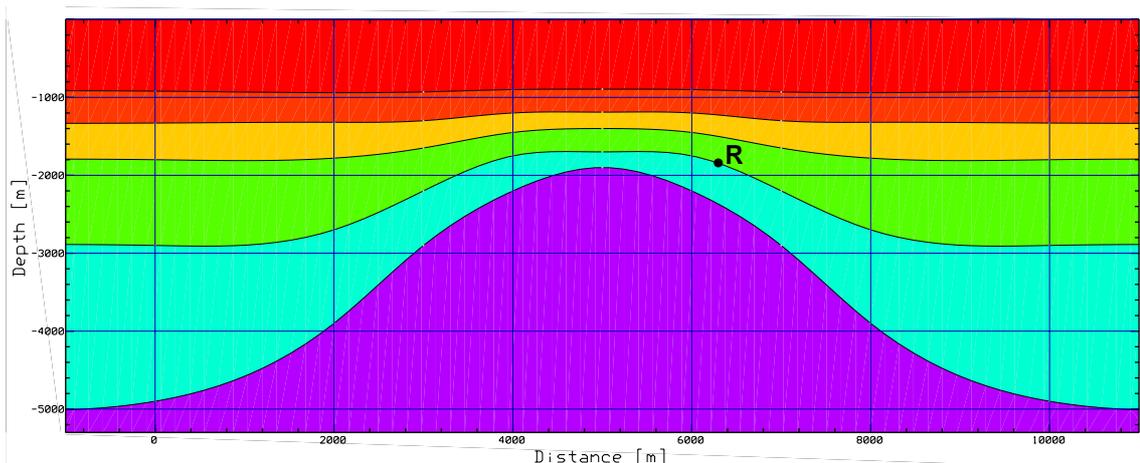


Figure 1: Model consisting of five dome-like interfaces.

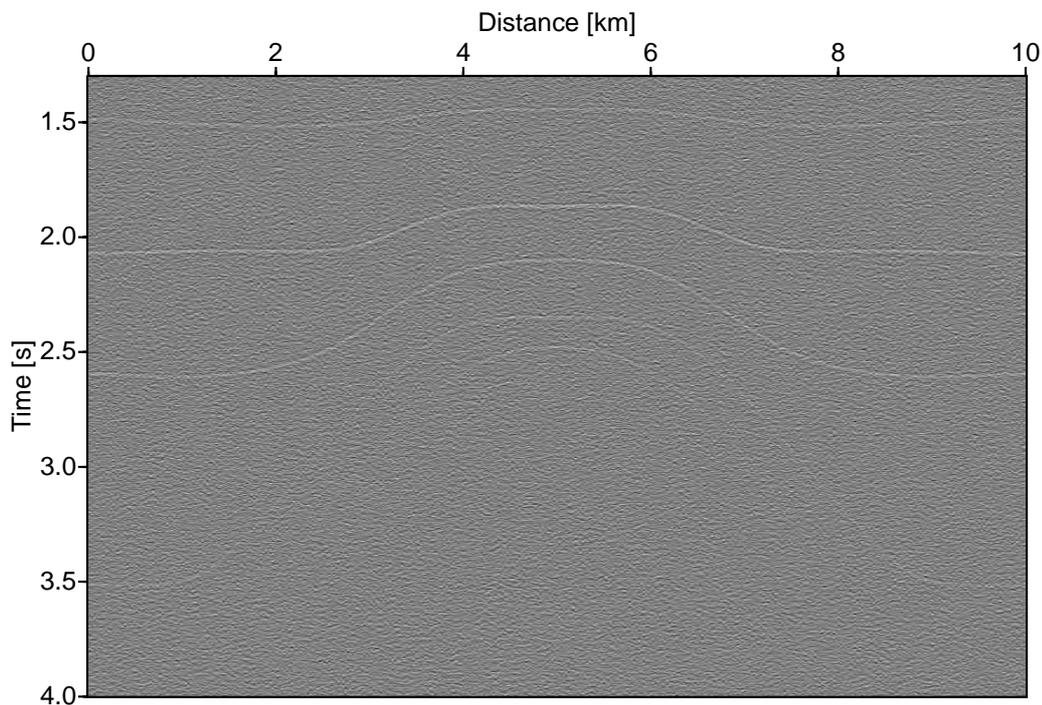


Figure 2: ZO section of synthetic dataset.

## GLOBAL OPTIMIZATION

As shown in Part I the hyperbolic traveltimes approximation of the CRS surface in midpoint  $x_m$  and half-offset  $h$  coordinates is given by

$$t^2(x_m) = \left( t_0 + \frac{2 \sin \alpha}{v_0} (x_m - x_0) \right)^2 + \frac{2 t_0 \cos^2 \alpha}{v_0} \left( \frac{(x_m - x_0)^2}{R_N} + \frac{h^2}{R_{NIP}} \right). \quad (1)$$

Another approach to obtain this equation can be found in Schleicher et al. (1993). There, it is derived by means of the paraxial ray theory. For each point  $P_0 = (x_0, t_0)$  in the ZO section we have to determine the optimum parameter triple  $\alpha$ ,  $R_{NIP}$ ,  $R_N$ , i.e., the triple for which the above traveltimes surface fits best to the reflection events. One possible approach to find these parameters would be to test all parameter triples and select the triple which leads to the highest coherency value. Of course this infinity number of test triples must be restricted to a finite three dimensional grid. However, from a practical point of view this would still be computationally much too expensive. Therefore, we have to look for a more efficient method to find the optimum parameter triple.

The problem states as follows:

Find the absolute maximum of the coherency value which is a function of the three independent variables  $\alpha$ ,  $R_{NIP}$  and  $R_N$  in a minimum computation time.

This is a typical problem of global optimization and quite a lot of algorithms exist to handle such a problem. But finding the global extremum is, in general, a very difficult problem and the success of all these optimization algorithms strongly depends on the behaviour of the considered function. To give you a better idea of the problem we have calculated the coherency values along the traveltimes surfaces defined by eq. (1) for a cube of parameter triples  $(\alpha, R_{NIP}, R_N)$  for subsurface point  $R = (x=6300\text{m}, z=-1850\text{m})$  located on the 4<sup>th</sup> layer of the synthetic model shown in fig. 1. The cube of coherency values for point  $R$  is shown in fig. 3. We have excavated a brick in this cube to allow you to view the point where the coherency function has its maximum. The point of maximum coherency is located at  $\alpha = 8.9^\circ$ ,  $R_N = 4043$  m and  $R_{NIP} = 2161$  m. In fig. 4 we show cross-sections of the cube in fig. 3, orthogonal to the coordinate

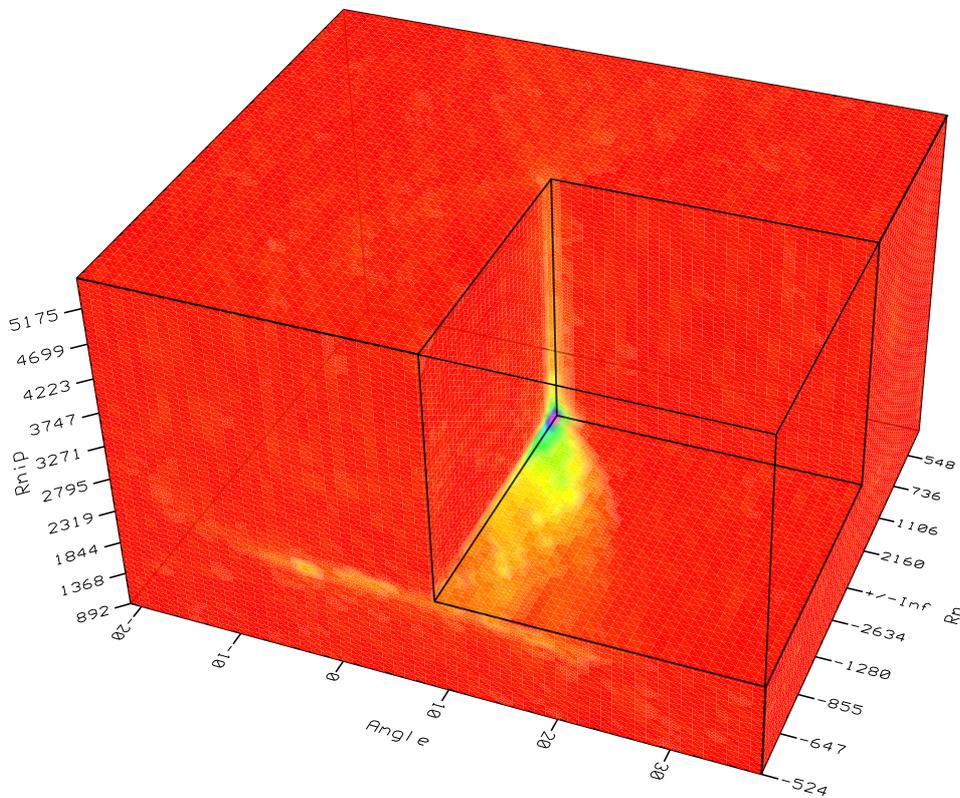


Figure 3: Cube of coherency values. The maximum coherency value is located at  $\alpha = 8.9^\circ$ ,  $R_N = 4043$  m and  $R_{NIP} = 2161$  m.

axes through the point of maximum coherency.

What we need is a multidimensional maximization algorithm which is capable to find the global maximum of the coherency function in such a cube. In multidimensional maximization algorithms, the best one can do is to give the algorithm a starting guess, that is, in this particular case, a specific parameter triple as a first point to try.

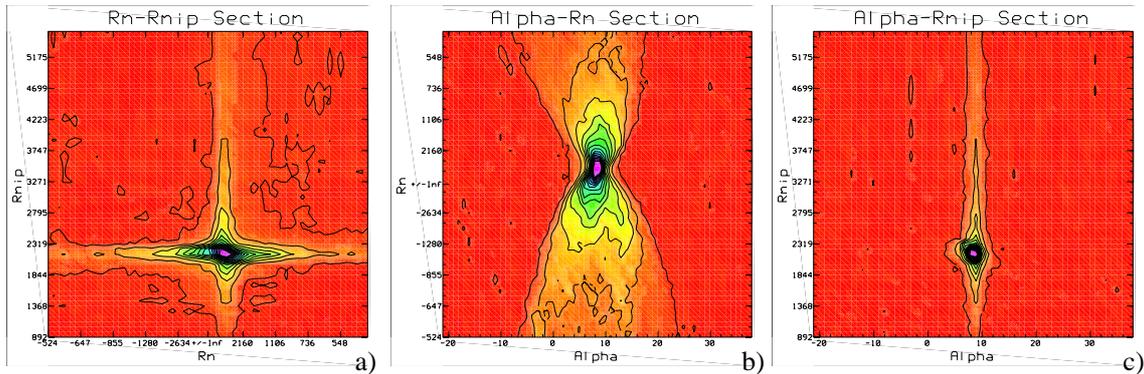


Figure 4: Sectional planes through the maximum coherency value of the coherency-cube: a)  $R_N$ - $R_{NIP}$  section for  $\alpha = 8.9^\circ$ , b)  $\alpha$ - $R_N$  section for  $R_{NIP} = 2161$  m and c)  $\alpha$ - $R_{NIP}$  section for  $R_N = 4043$  m.

The algorithm is then supposed to make its own way until it encounters a maximum. The main problem with all efficient multidimensional maximization algorithms is that one can never be sure that the found maximum is the global maximum rather than a local one. The probability to determine the global maximum is the higher the less local extrema exist and the smoother the functional behaviour is.

At a first glance, the coherency function seems to possess a quite good behaviour with respect to the optimization algorithms, i.e., the function looks smooth and has a broad, distinct maximum. Especially, if we look at the  $(\alpha, R_N)$  section in fig. 4b) one gets the impression that it should not be too difficult for an optimization algorithm to find the global maximum.

However, clipping coherency values exceeding 30 % of the maximum coherency value reveals that a lot of local extrema are present in this cube. Each of these local extrema serves as a potential trap for any optimization algorithm. The clipped version of the coherency cube is shown in fig. 5, the clipped slices are shown in fig. 6.

From these figures we see that, in general, an optimization will only be successful if we start with a parameter triple that is sufficiently close to the optimal parameter triple. Otherwise the optimization algorithm will probably run into a local maximum. This will become even clearer if you have a look at fig. 7: there we show cross sections orthogonal to the coordinate axes that do not intersect the point of maximum coherency but are shifted slightly along the coordinate axes. In this off-maximum sections things become even worse.

We can conclude that to use any optimization algorithm we need to determine an initial triple  $(\alpha^0, R_{NIP}^0, R_N^0)$  as starting point for the optimization.

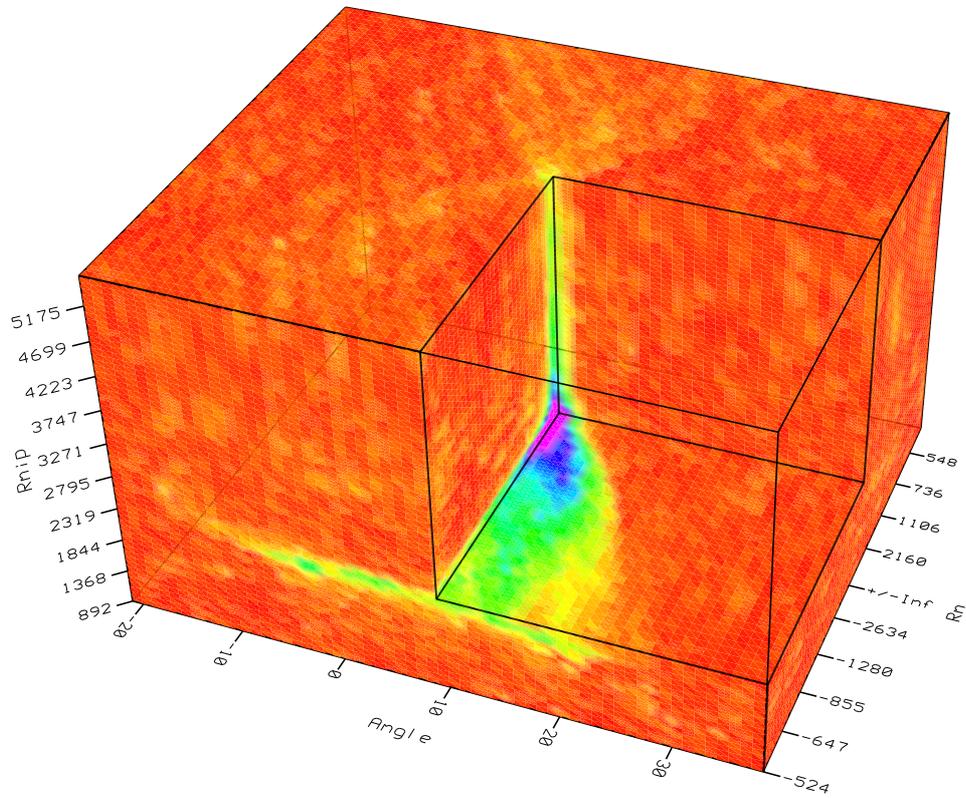


Figure 5: Same cube as in fig. 3 clipped to reveal local maxima.

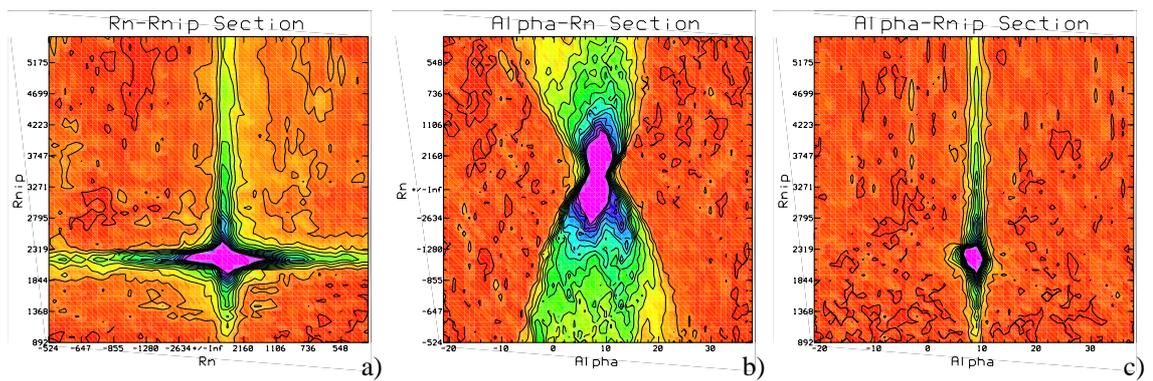


Figure 6: Sectional planes through the maximum coherency value of the cube in fig. 5: a)  $R_N$ - $R_{NIP}$  section for  $\alpha = 8.9^\circ$ , b)  $\alpha$ - $R_N$  section for  $R_{NIP} = 2161$  m and c)  $\alpha$ - $R_{NIP}$  section for  $R_N = 4043$  m.

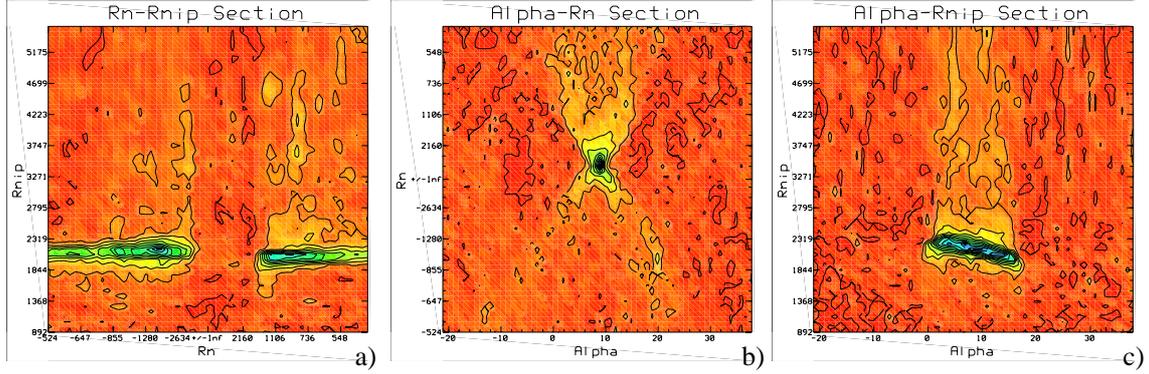


Figure 7: Slices through the cube in fig. 5 that do not intersect the point of maximum but are slightly shifted along the coordinate axes: a)  $R_N$ - $R_{NIP}$  section for  $\alpha = 14.9^\circ$ , b)  $\alpha$ - $R_N$  section for  $R_{NIP} = 2637$  m and c)  $\alpha$ - $R_{NIP}$  section for  $R_N = 1022$  m.

## STRATEGIES

In this section we introduce two methods to determine initial attributes  $\alpha^0$ ,  $R_{NIP}^0$ , and  $R_N^0$  suitable as starting points for the optimization algorithm.

One can easily verify that eq. (1) simplifies in the CMP gather for  $x_0$  and in the ZO section. In the CMP gather ( $x_m = x_0$ ) it reduces to

$$t^2(x_m) = t_0^2 + \frac{2t_0 \cos^2 \alpha}{v_0} \frac{h^2}{R_{NIP}}, \quad (2)$$

and in the ZO section ( $h = 0$ ) we get

$$t^2(x_m) = \left( t_0 + \frac{2 \sin \alpha}{v_0} (x_m - x_0) \right)^2 + \frac{2t_0 \cos^2 \alpha}{v_0} \frac{(x_m - x_0)^2}{R_N}. \quad (3)$$

This allows us to split the CRS stack into several steps. In this way we can quickly find a reasonable parameter triple as starting point for the optimization algorithm.

### Method A

#### Step I : CMP Stack

The first step common to both search strategies described below is to use eq. (2) in the CMP gathers. There the traveltimes only depend on the combined parameter

$$q = \frac{\cos^2 \alpha}{R_{NIP}}. \quad (4)$$

Hence, for each point  $P_0$  in the ZO section to be simulated we have to perform a one-parametric search for the combined parameter  $q$  in the CMP gather, i.e., for each point  $P_0$  we have to calculate the coherency value along the traveltimes given by eq. (2) for a whole range of  $q$  values and select the one which provides the highest coherency value. For each point  $P_0$  we subsequently perform a stack along the traveltime curve by means of the respective  $q$  parameter. In the following we refer to this first step as the CMP stack as we only use traveltime curves which are defined in the CMP gathers. As a result from the CMP stack we get three sections, namely: a) simulated ZO section, b) coherency section and c)  $q$ -parameter section.

## Step II : ZO Stack

In the second step we use the simulated ZO section of the CMP stack together with eq. (3). In the ZO section the traveltime curves only depend on  $\alpha$  and  $R_N$ . Thus, for each point  $(x_0, t_0)$  we have to find optimal parameters  $\alpha$  and  $R_N$  in the ZO section. This can be done in two ways.

We can perform a two-dimensional search for the parameters by calculating the coherency values along the traveltime curve given by eq. (3) on a two-dimensional grid of parameter pairs  $(\alpha, R_N)$ . Thereafter we select the parameter pair which yields the highest coherency value.

Alternatively, we can perform two one-dimensional searches by first setting  $R_N = \infty$  in eq. (3) and determining  $\alpha$  in

$$t(x_m) = t_0 + \frac{2 \sin \alpha}{v_0} (x_m - x_0) . \quad (5)$$

Having found  $\alpha$  we can substitute it into eq. (3) and perform a second one-dimensional search for  $R_N$ .

With the knowledge of attribute  $\alpha$  we can then invert the parameter  $q$  gained by the CMP stack in step I in order to determine  $R_{NIP}$ . For each point  $P_0$  we can subsequently perform a stack along the ZO traveltime curve for the found attributes  $\alpha$  and  $R_N$  in the same way as for the CMP stack. But in contrary to the CMP stack the simulated ZO stack of step II is not really needed but can be considered as a quality check. In the following we refer to this second step as ZO stack because we use traveltime curves defined in the ZO gather. As another intermediate result from the ZO stack we obtain five sections, namely: a) simulated ZO section, b) coherency section, c)  $\alpha$ -section, d)  $R_{NIP}$ -section and e)  $R_N$ -section.

### Step III : CRS Stack

In the final step we make use of the traveltime surface given by eq. (1) and apply it to the original dataset using the attributes found in step II as starting points for the optimization algorithm.

In the following we refer to this third step as CRS stack since we make use of the entire CRS stacking surface in the (x-h-t) space. As results from the CRS stack we obtain five sections: a) final simulated ZO section, b) coherency section, c) optimized  $\alpha$ -section, d) optimized  $R_{NIP}$ -section and e) optimized  $R_N$ -section.

### Comments on method A

Please note that the motivation of step I (CMP stack) and step II (ZO stack) is solely to provide suitable initial parameter triples for the optimization algorithm in step III (CRS stack). Yet, to exploit the full advantage of the CRS stack we have to use not only traveltime curves confined to the CMP gather and the ZO section but traveltime surfaces in the (x-h-t) space. The whole sequence of steps is visualized in fig. 8 as flowchart.

However, the search-strategy presented above has a disadvantage: in the second step we do not use the original dataset but the simulated ZO section of the CMP stack as input. The CMP stack is, of course, far from being optimal and using this stack as the input section for the next step may be dangerous as errors made in the first step may accumulate in the following steps. At least, by using the simulated ZO section in step II instead of the original dataset, we do not make use of all the information given by a multi-coverage reflection dataset. Nevertheless, with a minor modification in step II, presented above, we can easily overcome this drawback:

### Method B

#### Step I : CMP Stack

This step is the same as in method A described above. For the sake of clearness and to make the next step more comprehensible we briefly explain the connection of the  $q$ -parameter found for a specific point  $P_0$  and the coherency cube for this point. For every point  $P_0$  in the ZO section to be simulated, we could calculate a coherency-cube similar to the one depicted in fig. 3. With the knowledge of  $q$ ,  $R_{NIP}$  is given by

$$R_{NIP} = \frac{\cos^2 \alpha}{q} . \quad (6)$$

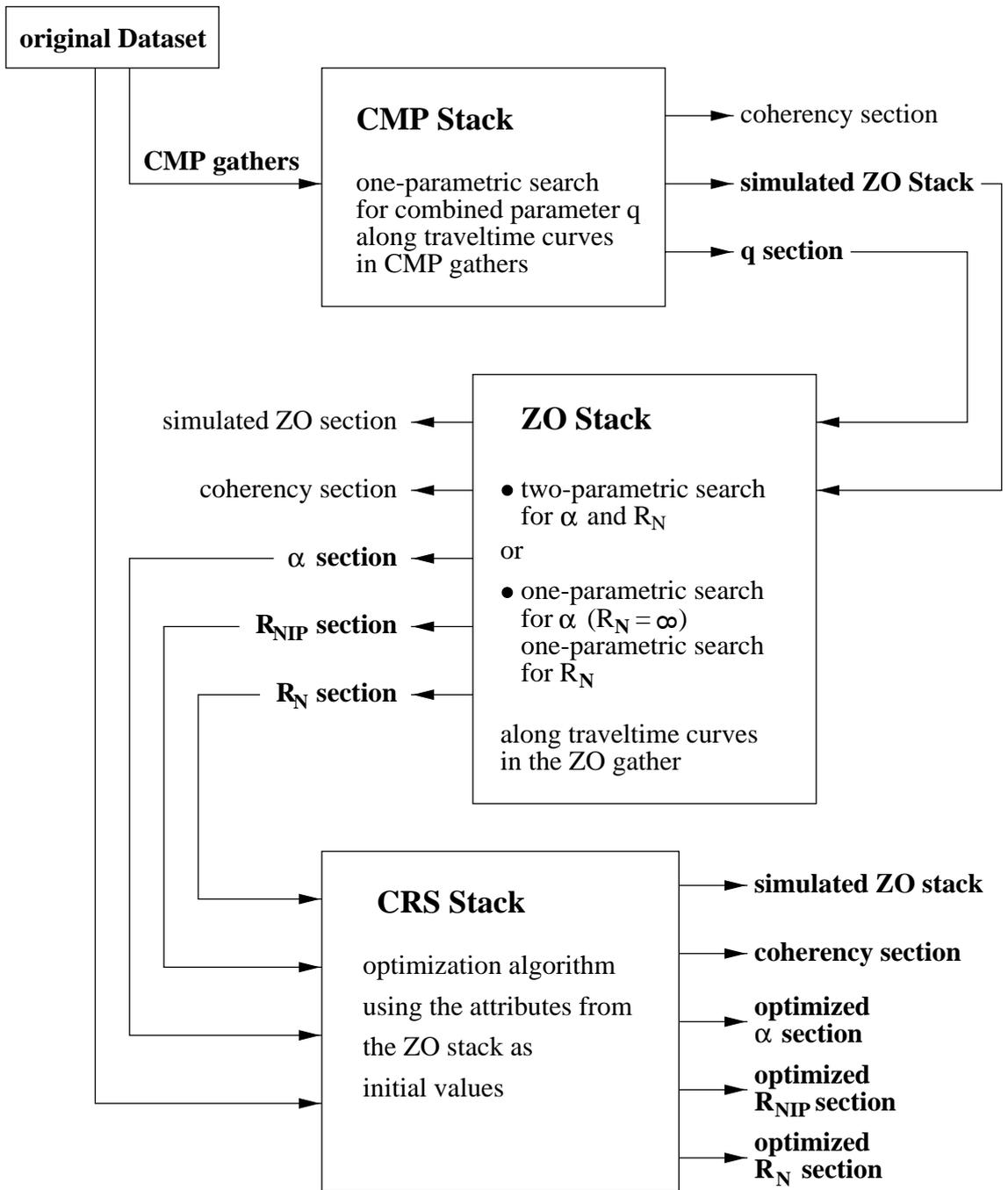


Figure 8: Flowchart for method A.

As this equation holds for all  $R_N$ , the  $q$ -parameter found in the CMP stack defines a surface in the cube of coherency values. The  $q$ -surface for point  $R$  is displayed in fig. 9. In case point  $P_0$  is located on a ZO reflection event a clear distinct maximum will be present. If we assume the  $q$ -parameter to be optimal, the surface will obviously pass through the maximum coherency value of the cube.

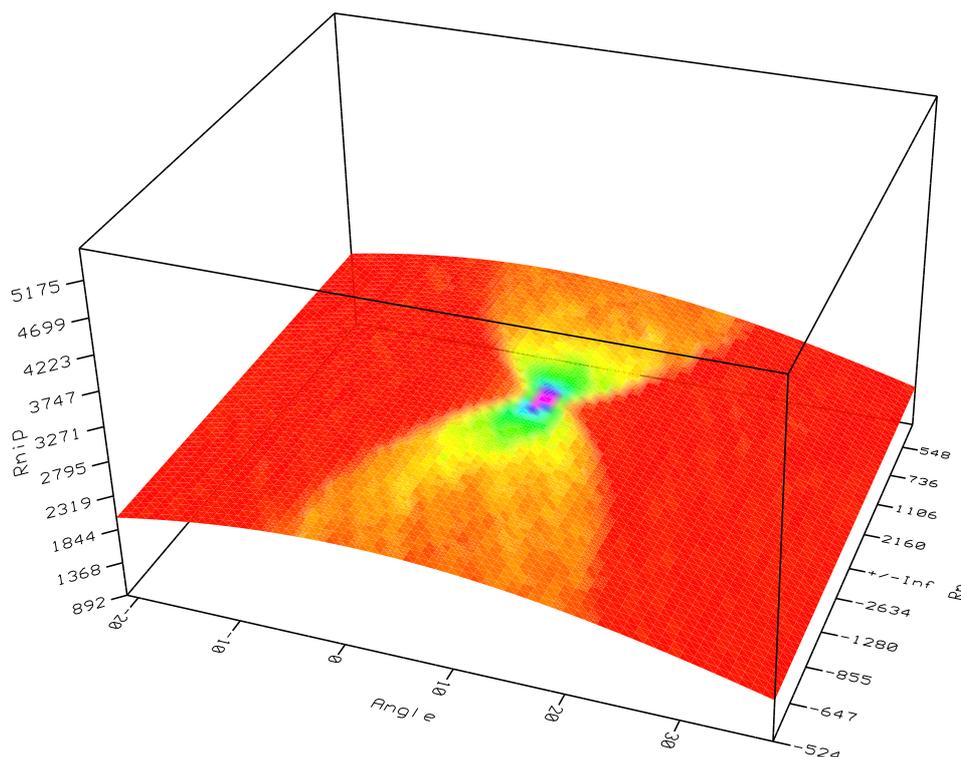


Figure 9: Surface in the coherency-cube defined by the  $q$ -parameter.

## Step II : Restricted CRS Stack

In the second step of method A we needed the  $q$ -attribute section only to invert for the last parameter  $R_{NIP}$  after having found  $\alpha$  and  $R_N$ . During the search neither the determined  $q$ -parameter nor  $R_{NIP}$  are involved because the ZO traveltime curve does not depend on  $q$  or  $R_{NIP}$ .

However, to search  $\alpha$  and  $R_N$  in the original dataset and not confine the search to the simulated ZO section of step I we can use eq. (1) even if this equation depends upon  $R_{NIP}$ . This is possible as for every parameter pair  $(\alpha, R_N)$   $R_{NIP}$  is determined by the  $q$ -surface in the  $(\alpha, R_N, R_{NIP})$  search space. Thus, for every parameter pair

$(\alpha, R_N)$  we determine  $R_{NIP}$  by means of eq. (6) and calculate the coherency value for the resulting parameter triple along the traveltimes surface given by eq. (1). Note that although we use all three parameters  $\alpha$ ,  $R_{NIP}$  and  $R_N$  in eq. (1) for the calculation of the coherency value, we still perform a two-dimensional search only, because  $R_{NIP}$  is related to  $\alpha$  by the  $q$ -attribute already determined. Again, we have two possibilities to find  $\alpha$  and  $R_N$ : on the one hand, we can make a two-dimensional search for  $\alpha$  and  $R_N$  on a grid as described in method A but using eq. (1) instead of eq. (3). On the other hand, we can perform two separate one-dimensional searches by firstly setting  $R_N = \infty$  in eq. (1) and looking for  $\alpha$  in

$$t^2(x_m) = \left( t_0 + \frac{2 \sin \alpha}{v_0} (x_m - x_0) \right)^2 + \frac{2 t_0 \cos^2 \alpha}{v_0} \frac{h^2}{R_{NIP}}. \quad (7)$$

Having found  $\alpha$  we can substitute it into eq. (1) and perform a second one-parametric search for  $R_N$ .

In the following we will refer to this process as the “restricted CRS stack” because, firstly, we use the full CRS traveltimes surface in the (x-h-t) space, and secondly, we test only for parameter triples restricted to the  $q$ -surface in the  $(\alpha, R_N, R_{NIP})$  search space.

### Step III : CRS Stack

This step is the same as in method A described above. The attribute sections of step II are used as initial points for the optimization algorithm.

### Comments on method B

The motivation of step I (CMP stack) and step II (restricted CRS stack) is to provide suitable initial parameter triples for the optimization algorithm in step III (CRS stack). This method has the advantage that all search procedures are carried out directly in the original dataset. No simulated ZO section is needed as input for step II anymore. This is achieved by already making use of the  $q$ -parameter during the search for  $\alpha$  and  $R_N$ , whereas in method A  $q$  is only used to invert for  $R_{NIP}$  after  $\alpha$  and  $R_N$  are determined. The disadvantage of this method is that step II is computationally more expensive than in the first method: for every parameter pair  $(\alpha, R_N)$  we have to calculate the coherency value along a surface in the (x-h-t) space, whereas in the first method we determined the coherency value along traveltimes curves in the ZO section only.

The whole sequence of method B is visualized in fig. 10 as flowchart.

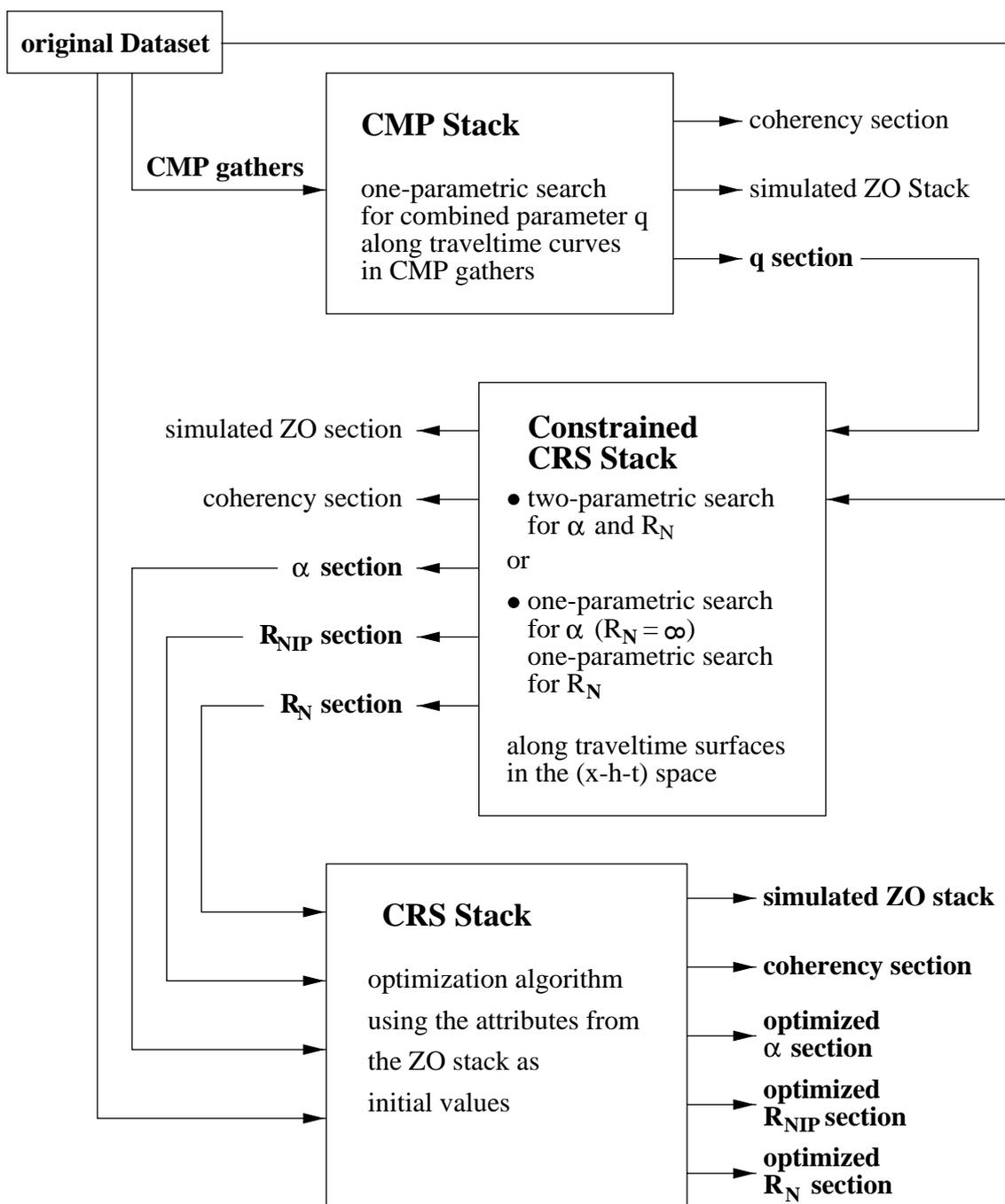


Figure 10: Flowchart for method B.

## RESULTS

To demonstrate the power of the CRS stack we apply method B to the synthetic dataset described in section . As coherency criterion in all steps we use semblance (Taner and Koehler (1969)).

As results of step I we get the simulated ZO section shown in fig. 11 together with the semblance section shown in fig. 12. The corresponding  $q$ -section is shown in fig. 13.

From step II we obtain initial attribute sections  $\alpha^0$ ,  $R_N^0$ , and  $R_{NIP}^0$ . As these sections are intermediate results we do not show them here. We determined these results by a two-parametric search for  $\alpha$  and  $R_N$  on the  $q$ -surface as described in section .

In step III we use the initial attribute sections of step II as initial points for the optimization algorithm. We use the so-called downhill simplex method in multi dimensions (Press et al. (1986)) to optimize the attributes. The CRS stack and the semblance section are shown in figs. 14 and 15. Compared to the CMP stack the CRS stack reveals a much better signal/noise ratio, since the CRS stack uses stacking surfaces, whereas the CMP stack only involves stacking curves. Therefore, more traces are used to simulate a point in the ZO section and thus noise is suppressed more successfully.

Figure 16 shows the optimized  $\alpha$ -section in a gray-scale representation. In order to compare the data-derived attributes with the model-derived attributes we switch to the 3D representation shown in fig. 17. There, we only show the  $\alpha$ -attribute for the points in the ZO section that correspond to reflection events. Each point in the ZO section is represented by a point in three dimensions, where the third dimension corresponds to the attribute's value. The data-derived  $\alpha$ -attributes are shown as thin black lines, the model-derived attributes as gray points. To further facilitate a comparison between data-derived and model-derived attributes we perform a projection along the time axis onto the (midpoint, $\alpha$ ) plane shown in fig. 18. Again, the data-derived attributes are represented by black lines, whereas the model-derived attributes are depicted in gray.

There is no need to emphasize the good accordance of the data-derived with the forward-calculated attributes.

The optimized  $R_{NIP}$ -section is shown in fig. 19. For comparison to the model-derived  $R_{NIP}$ -attributes we refer to fig. 20. The optimized  $R_N$ -section is depicted in fig. 21. For clearness the model-derived  $R_N$ -attributes are compared separately for each interface in fig. 22.

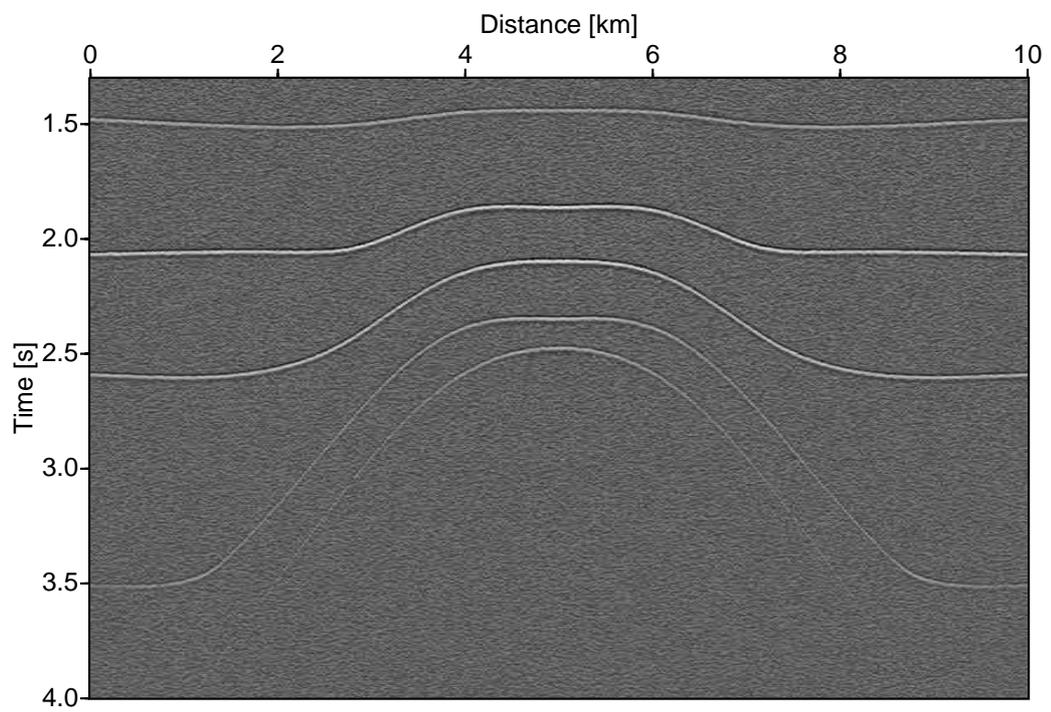


Figure 11: CMP Stack

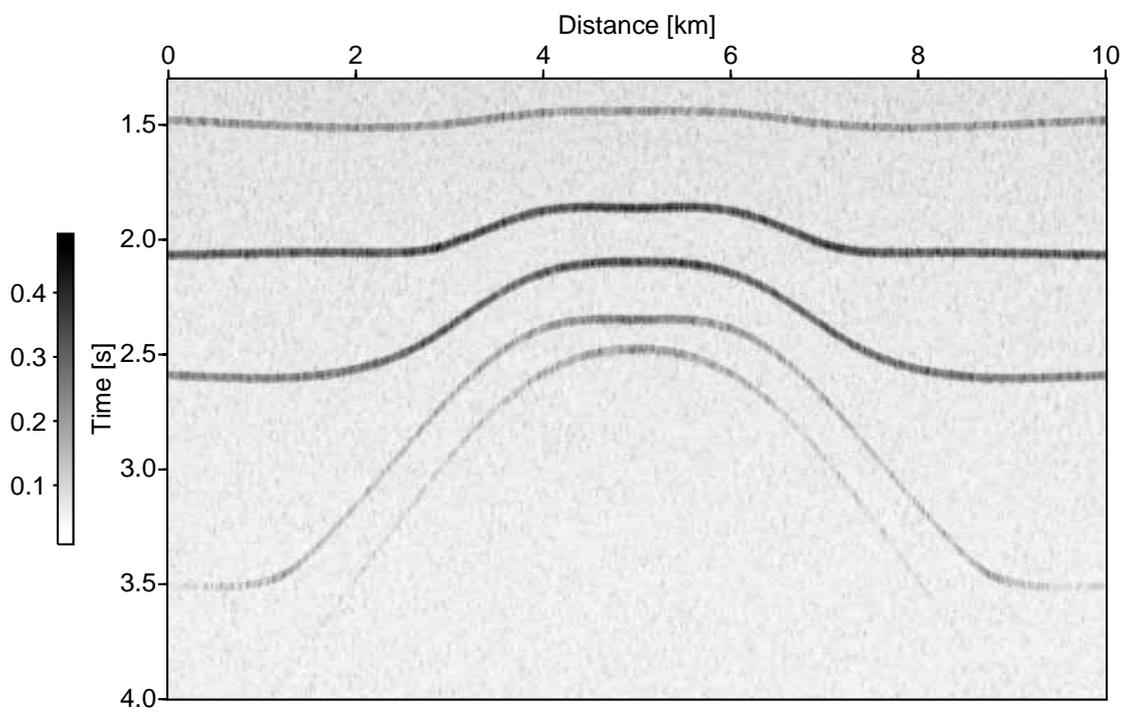


Figure 12: Semblance section for the CMP stack.

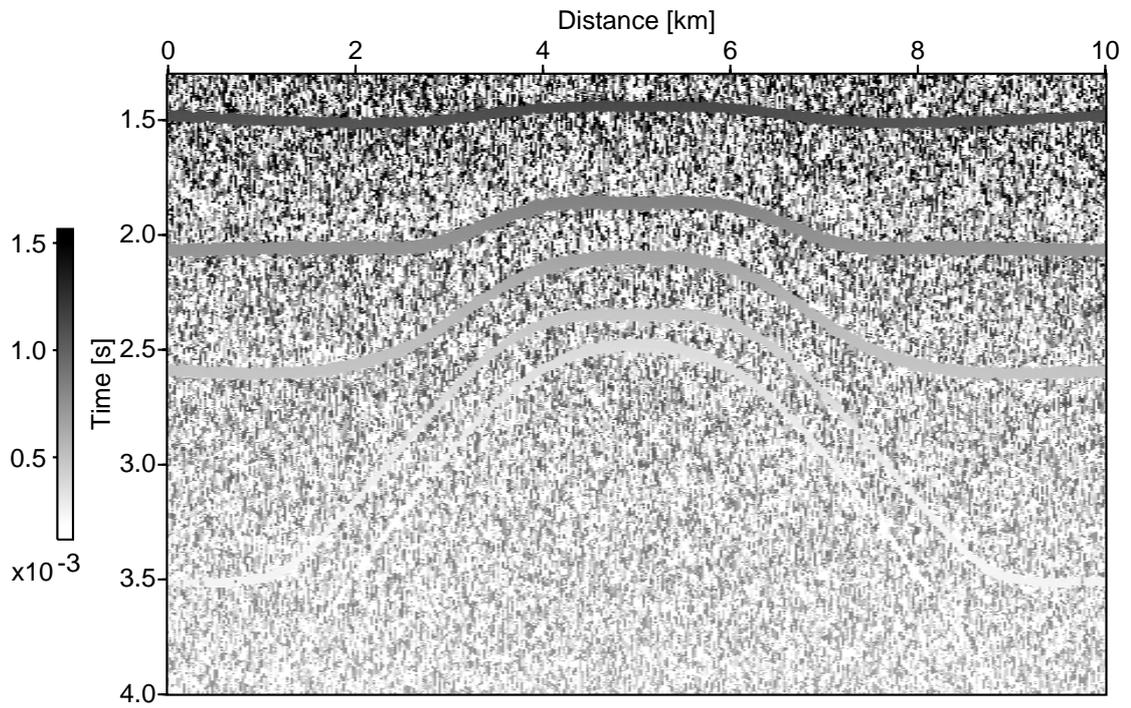


Figure 13: Combined  $q$ -parameter of CMP stack [1/m].

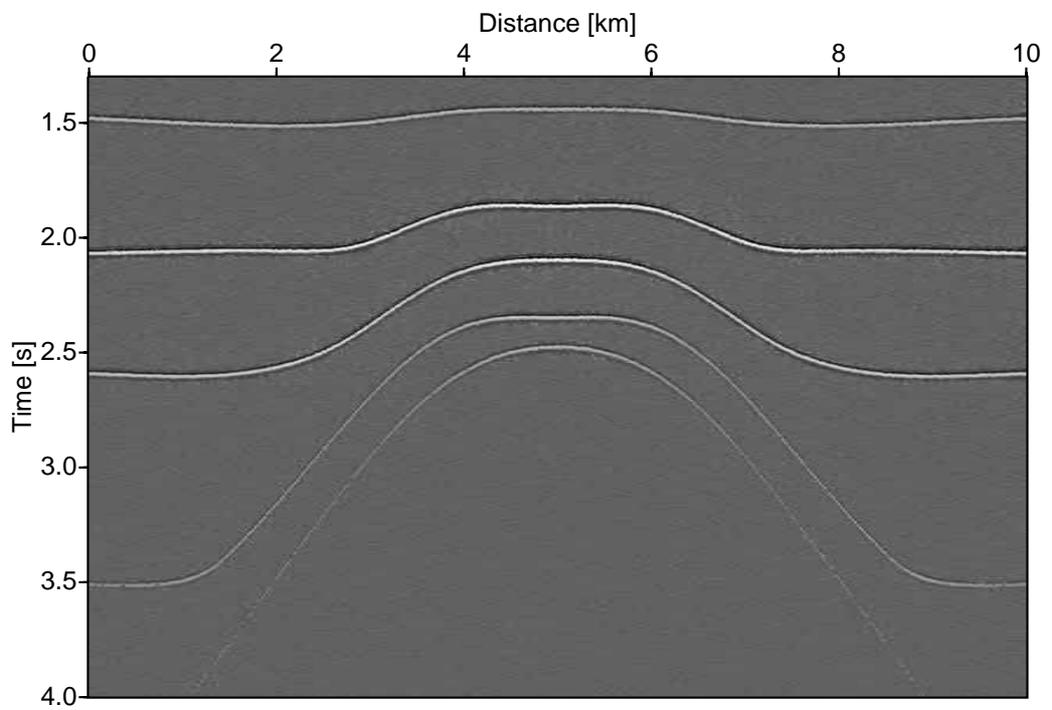


Figure 14: CRS Stack

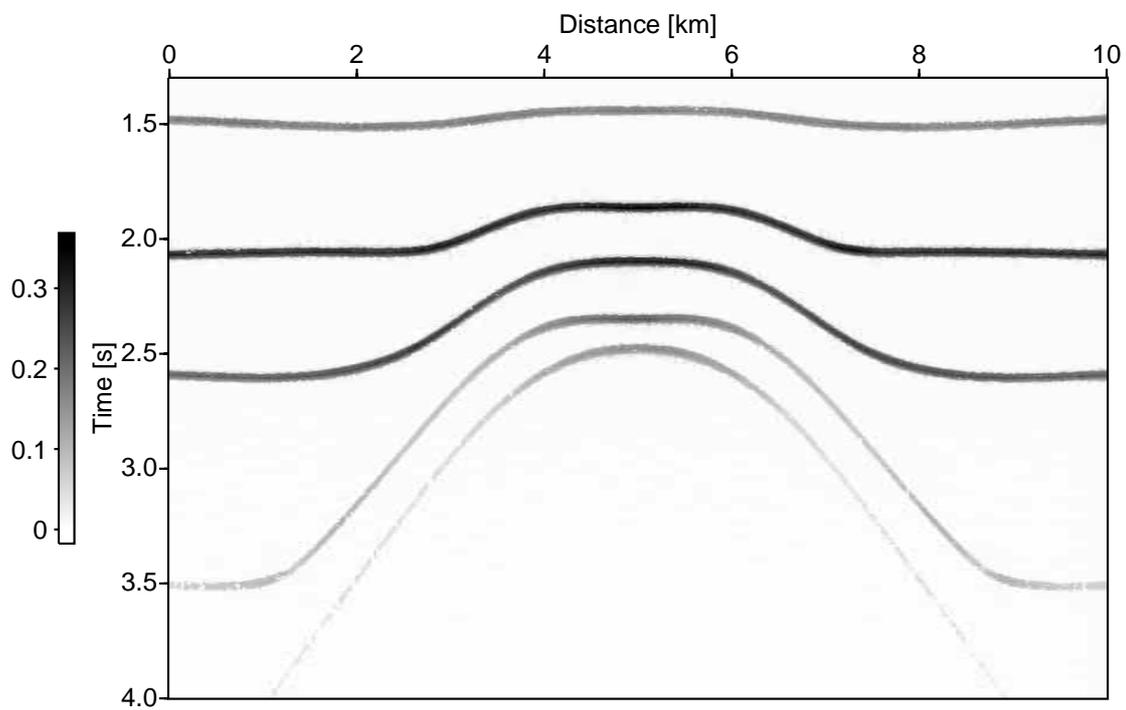
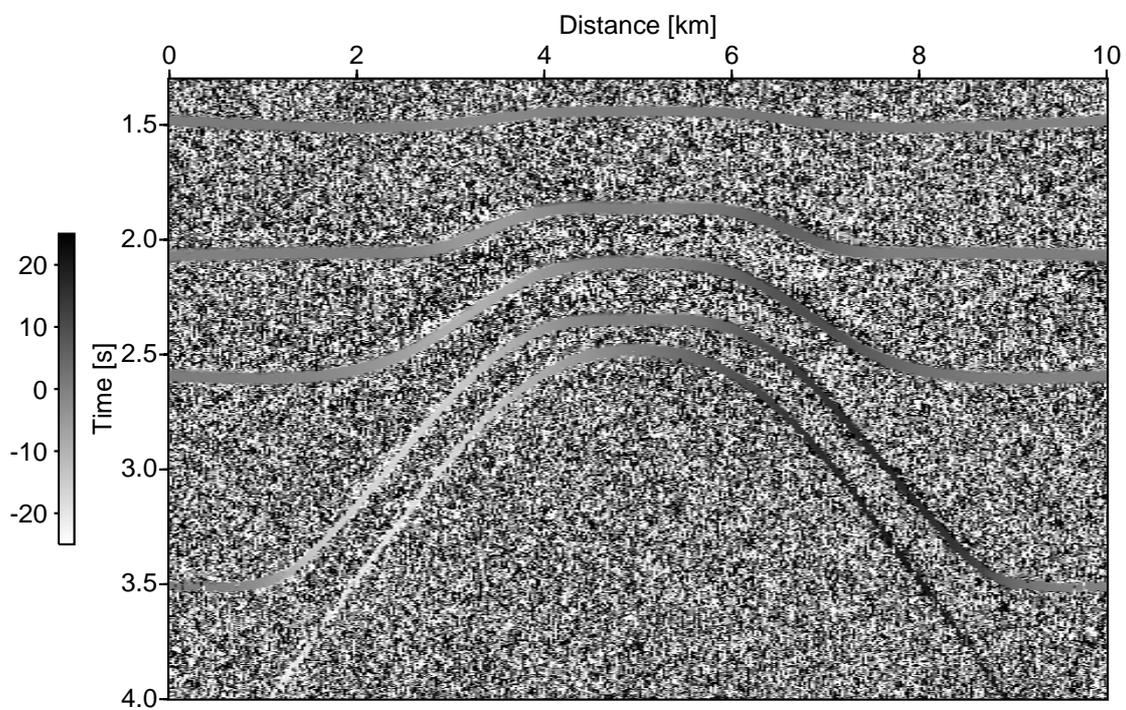


Figure 15: Semblance section for the CRS stack.

Figure 16: Optimized  $\alpha$ -attribute in  $^{\circ}$

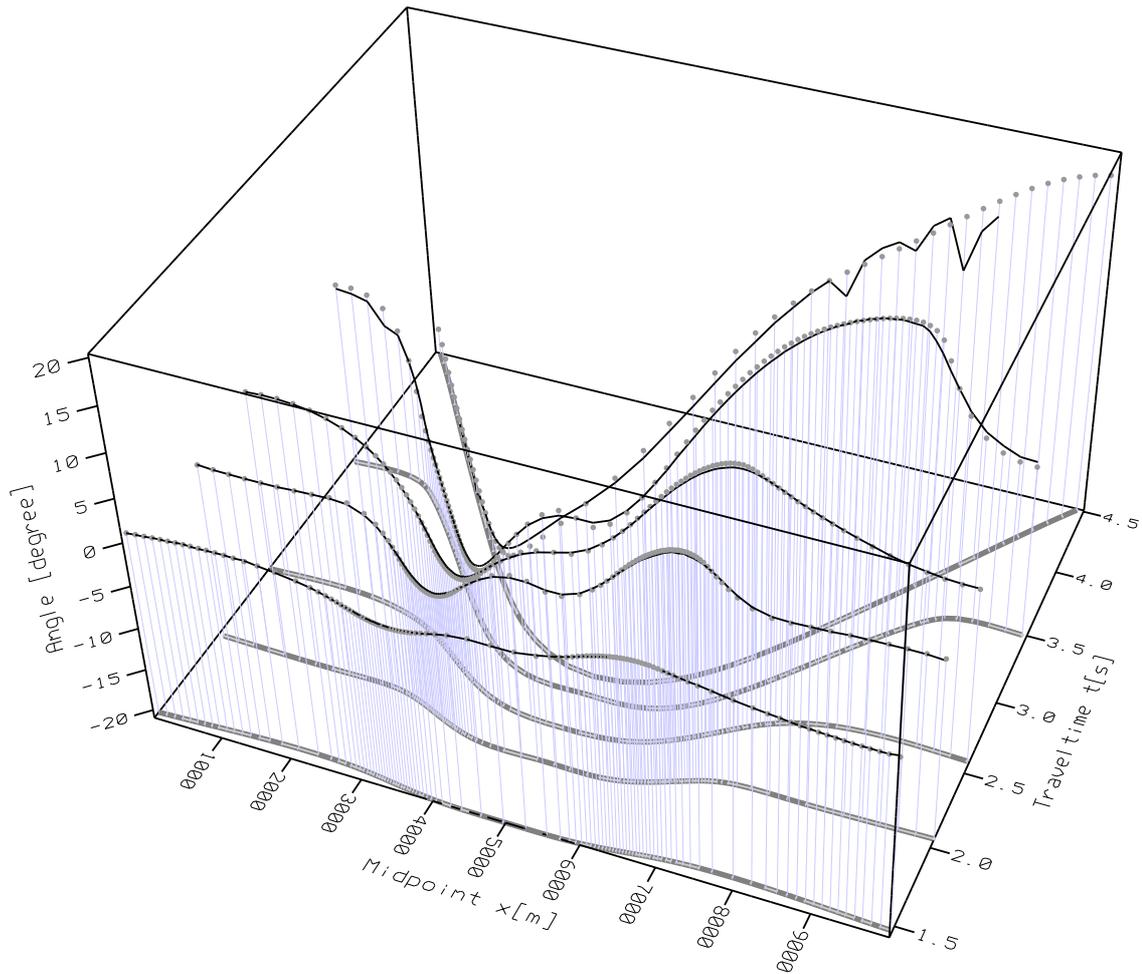


Figure 17: 3D view of data-derived (thin black lines) and model-derived (gray points)  $\alpha$ -attributes.

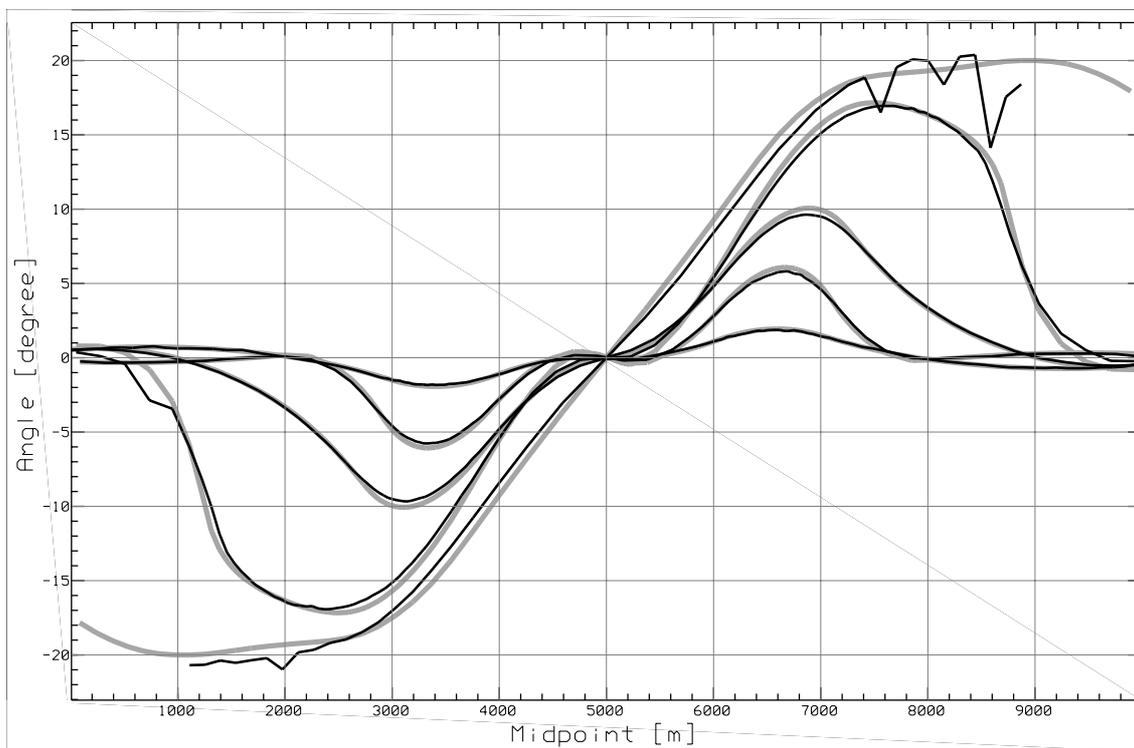


Figure 18: Comparison of data-derived (black) and model-derived (gray)  $\alpha$ -attributes for all interfaces.

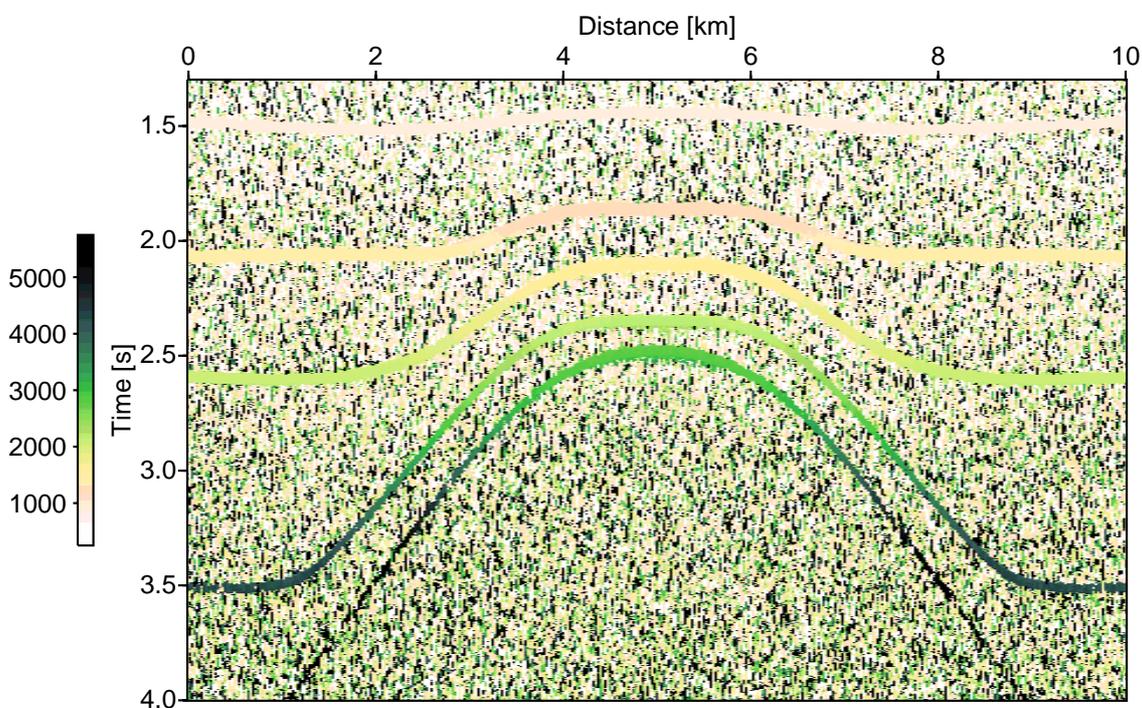


Figure 19: Optimized  $R_{NIP}$ -attribute [m].

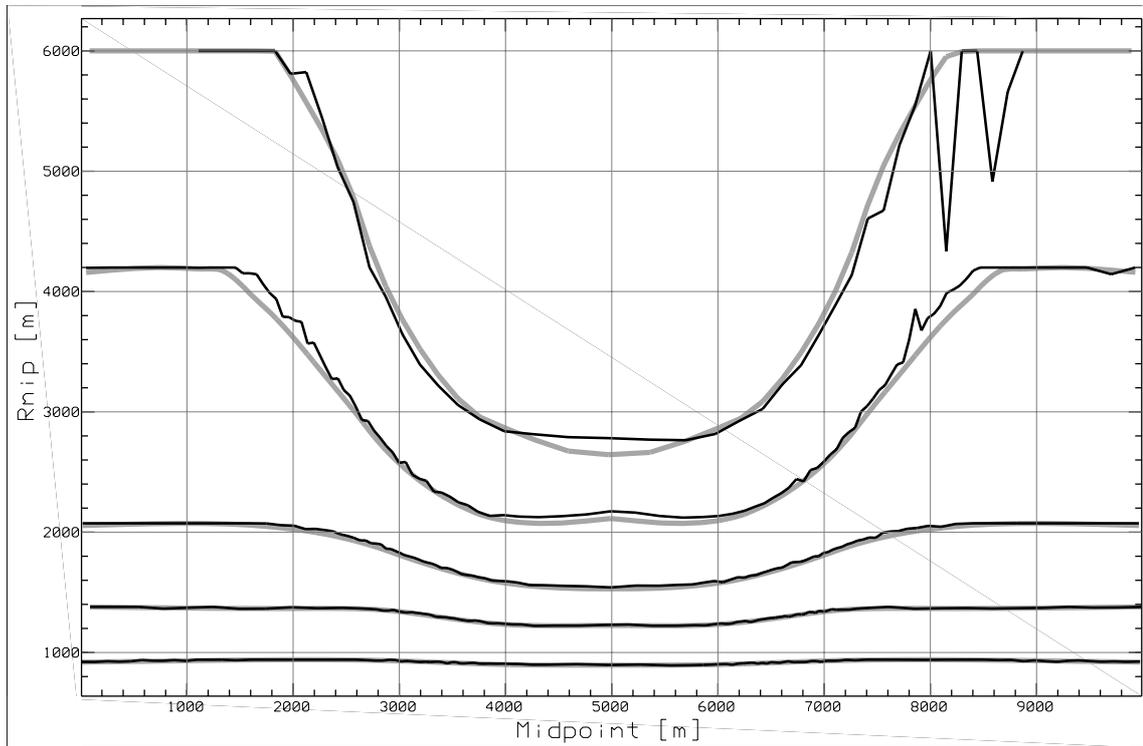


Figure 20: Comparison of data-derived (black) and model-derived (gray)  $R_{NIP}$ -attributes for all interfaces.

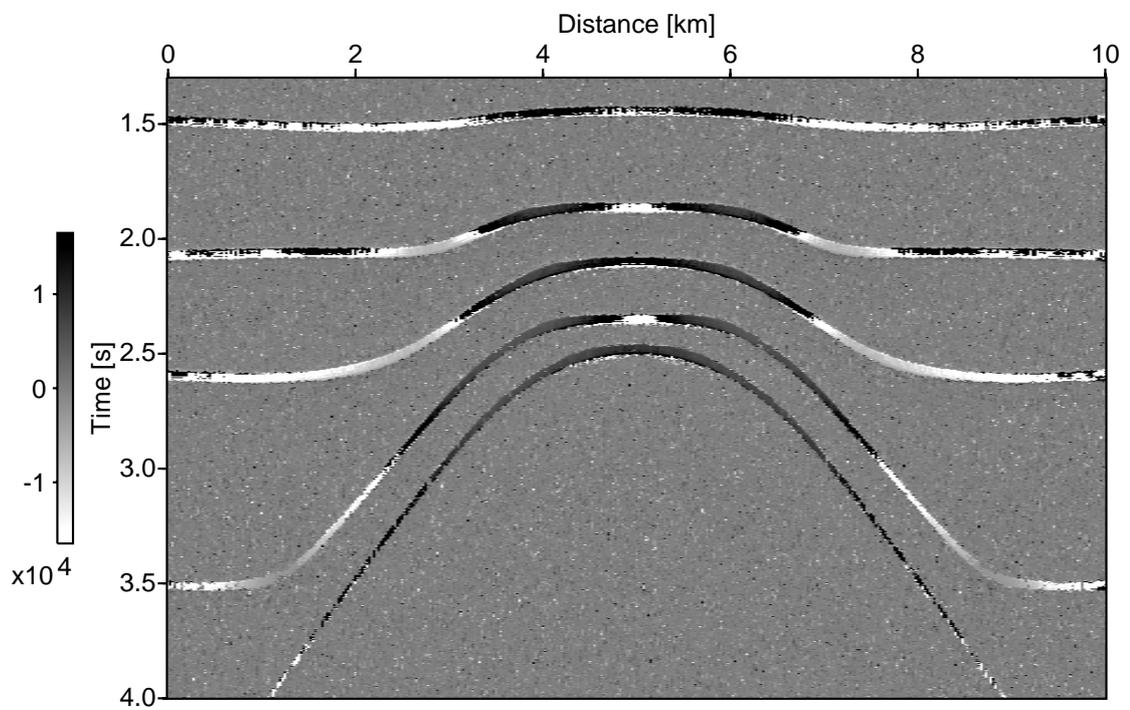


Figure 21: Optimized  $R_N$ -attribute [m].

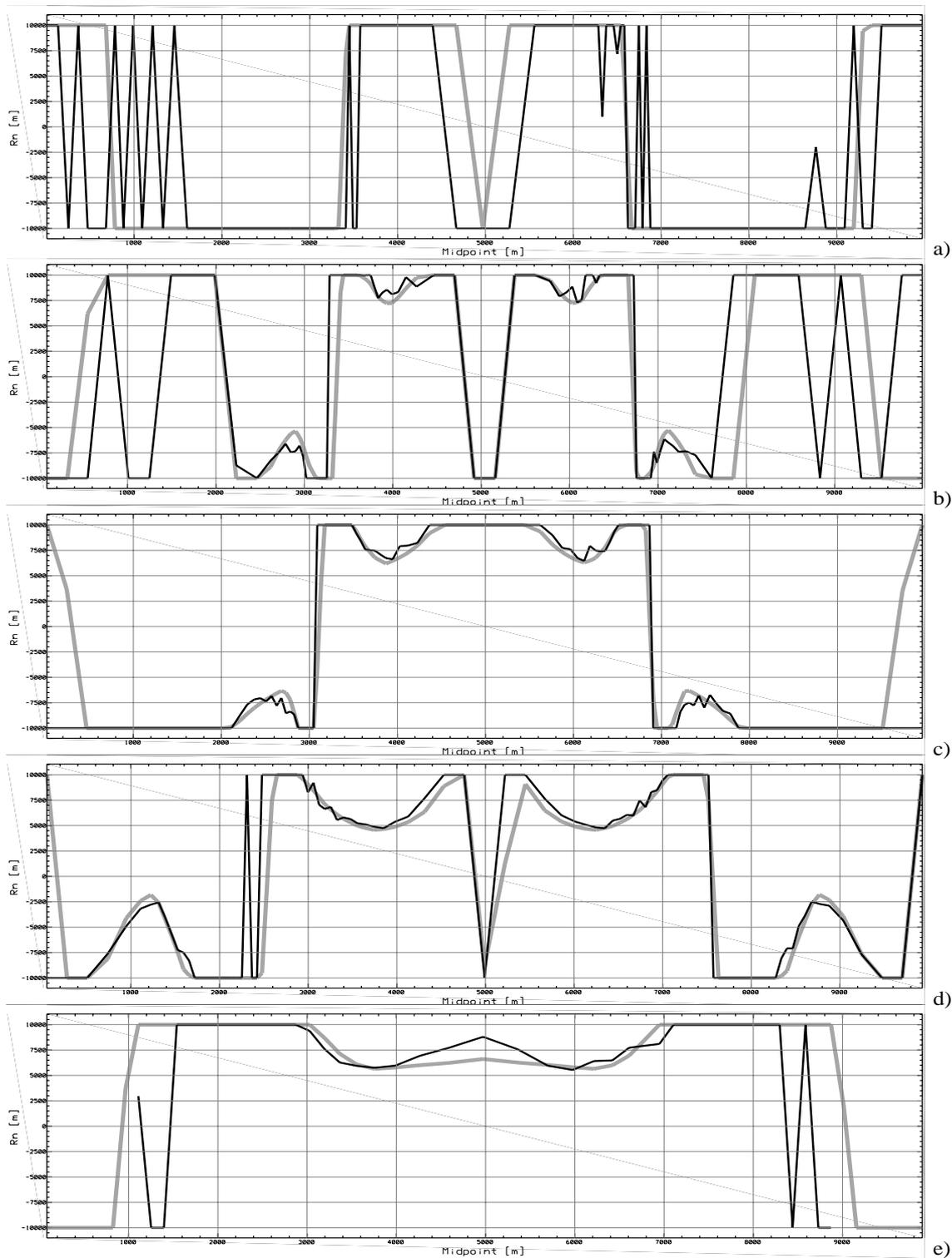


Figure 22: Comparison of data-derived (black) and model-derived (gray)  $R_N$ -attributes. The attributes for each layer are plotted separately: a) first, b) second, c) third, d) fourth and e) fifth layer.

## CONCLUSIONS

A macro velocity independent ZO simulation, which only requires the knowledge of the near surface velocity, has been proposed and applied to a synthetic dataset. Its results are beneath the stack section various attribute sections and a coherency section that can be used for a subsequent inversion.

For the determination of the required attributes different search strategies have been introduced. These strategies permit to save computation time and the use of a global optimization algorithm.

The application to a synthetic dataset showed noteworthy results with respect to the stack section and the determined attributes. In view of the authors, the proposed strategies offer an exciting approach to improve the stack section and to allow for a subsequent inversion.

The fact that the hyperbolic traveltimes approximation (Schleicher et al. (1993)) of the CRS stacking surface can be simplified in specific gathers offers a strategy to quickly determine initial attributes for an optimization algorithm.

## REFERENCES

- Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P., 1986, Numerical recipes in fortran: The art of scientific computing.: Cambridge University Press, Cambridge.
- Schleicher, J., Tygel, M., and Hubral, P., 1993, Parabolic and hyperbolic paraxial two-point traveltimes in 3D media: Geophys. Prosp., **41**, no. 4, 495–514.
- Taner, M. T., and Koehler, F., 1969, Velocity spectra – digital computer derivation and applications of velocity functions \*: Geophysics, **34**, no. 6, 859–881.