

# **Code *LOTOS-12* for 3D tomographic inversion based on passive seismic data from local and regional events**

**Ivan Koulakov**

Head of Laboratory for Forward and Inverse Seismic Modeling  
Institute of Petroleum Geology and Geophysics, SB RAS,  
Prospekt Akademika Koptuga, 3, Novosibirsk, 630090, Russia

e-mail: KoulakovIY@ipgg.nsc.ru

Phone: +7 383 3309201

Mobil: +7 913 453 8987

Internet: [www.ivan-art.com/science/LOTOS](http://www.ivan-art.com/science/LOTOS)

Novosibirsk, Russia,

December, 2012

## Table of content:

0. The main changes with respect to the previous version, LOTOS-07 and -09 .....	4
1. Brief description of algorithms used in the LOTOS code .....	5
1.1. General information .....	5
1.2. Algorithm for 1D velocity optimization and preliminary source location .....	7
1.3. Bending algorithm for ray tracing in a 3D velocity model .....	9
1.4. Iterative tomographic inversion .....	10
1.4.1. Source locations in a 3D velocity model .....	10
1.4.2. Parameterization with nodes .....	10
1.4.3. Matrix calculation and inversion for the case of Vp-Vs scheme .....	11
1.4.4. Iterative cycling .....	12
2.1. List of folders in the root directory .....	13
2.2. Batch files in the root folder .....	13
2.3. Structure of the DATA folder .....	14
2.3.1. Format of the input data in the "inidata" folder .....	16
2.3.2. Organization of a MODEL folder .....	17
2.3.3. Major free parameters, file "MAJOR_PARAM.DAT" .....	18
2.3.4. 1D starting velocity model, file "ref_start.dat" .....	21
3. Step-by-step calculations with the LOTOS code .....	22
3.1. Preliminary location in the 1D velocity model using reference tables .....	22
3.1.1. Calculation of the reference table .....	22
3.1.2. Source location in a 1D model using the reference table .....	23
3.1.3. Preliminary location in the 1D velocity model using straight line approximation for the rays .....	25
3.2. Algorithm for 1D velocity optimization .....	26
3.2.1. Step 1. Data selection for the optimization .....	27
3.2.2. Step 2. Calculation of travel time table in a current 1D model .....	27
3.2.3. Step 3. Source location in the 1D model .....	27
3.2.4. Step 4. Calculation of the first derivative matrix .....	28
3.2.5. Step 5. Matrix inversion .....	28
3.2.6. Example of practical realization of the algorithm for 1D model optimization .....	29
3.3. Iterative 3D tomographic inversion and source relocation .....	30
3.3.1. General structure of programs .....	30
3.3.2. Source location in a 3D velocity model .....	31
3.3.3. Construction of the parameterization grid: .....	33
3.3.5. Calculation of the first derivative matrix .....	34
3.3.6. Inversion .....	35
3.3.7. Calculation of 3D model in a regular grid .....	36
3.3.8. Practical realization of LOTOS code .....	36
4. Presentation of the results .....	37
4.1. Visualization tool for previewing .....	37
4.2. Visualizing the ray paths and nodes .....	40
4.3. Horizontal sections of the resulting anomalies (NODES and CELLS) .....	43
4.4. Vertical sections of the resulting anomalies .....	47
5. Synthetic modeling .....	50
5.1. General remarks .....	50
5.2. Definition of noise .....	50
5.3. Visualization of the initial synthetic model in horizontal and vertical sections .....	51
5.4. Definition of the checkerboard anomalies (key 1) .....	52
5.5. Definition of free horizontal anomalies (key 2) .....	53
5.6. Definition of free vertical anomalies (key 3) .....	55
5.7. Definition of vertical checkerboard anomalies (key 4) .....	57
Conclusion remarks: .....	58
References: .....	58

**When using the code please refer to the following paper:**

Koulakov I., 2009, LOTOS code for local earthquake tomographic inversion.  
Benchmarks for testing tomographic algorithms, Bulletin of the Seismological Society of  
America, Vol. 99, No. 1, pp. 194-214, doi: 10.1785/0120080013

## **0. The main changes with respect to the previous version, LOTOS-07 and -09**

- Besides the inversion for  $V_p$  and  $V_s$ , we included a possibility for  $V_p$ - $V_p/V_s$  inversion.
- The results in horizontal sections can be previewed as PNG bitmap images without using SURFER or any other commercial graphical tools.
- Structure of files and programs has become more simple and convenient
- In the current version we present a set of different examples which can be used as templates to construct new models
- We propose a series of lessons which will be helpful for learning using the code.
- We provide a tool for simulating artificial datasets which can be used for planning network deploying.
- The topography can be included; the sources can be located above sea level.
- The structure of files has become more simple.
- For local cases, as an option, we propose doing the preliminary source locations based on the straight line approximation.

In this manual we describe mostly the Windows version of the code. LINUX version with convenient GMT scripts is also available. Both Windows and LINUX versions of the LOTOS-12 code are available in the Internet site:  
[www.ivan-science.com/science/LOTOS](http://www.ivan-science.com/science/LOTOS)

# 1. Brief description of algorithms used in the LOTOS code

## 1.1. General information

A tomographic algorithm, *LOTOS* (Local Tomography Software) is designed for simultaneous inversion for P and S velocity structures and source coordinates. The LOTOS algorithm can be directly applied to very different data sets without complicated tuning of parameters. It has a quite wide range of possibilities for performing different test and is quite easy to operate. *LOTOS* code can be freely provided to any interested person by Ivan Koulakov ([ivan.science@gmail.com](mailto:ivan.science@gmail.com)).

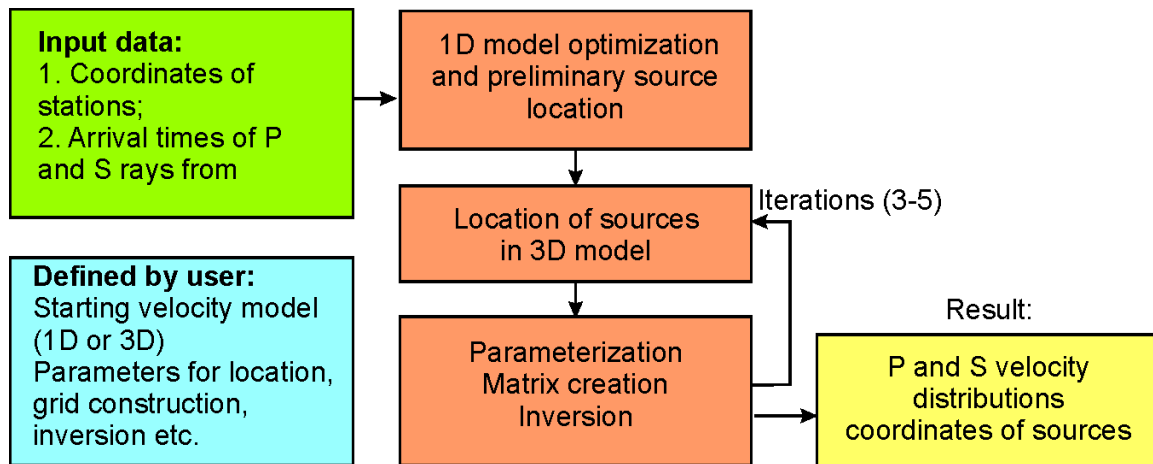


Figure 1.1. General structure of the LOTOS code. Pink blocks indicate the main program steps. Green block is the main input data; blue block contains free parameters defined by user; yellow block is the output data.

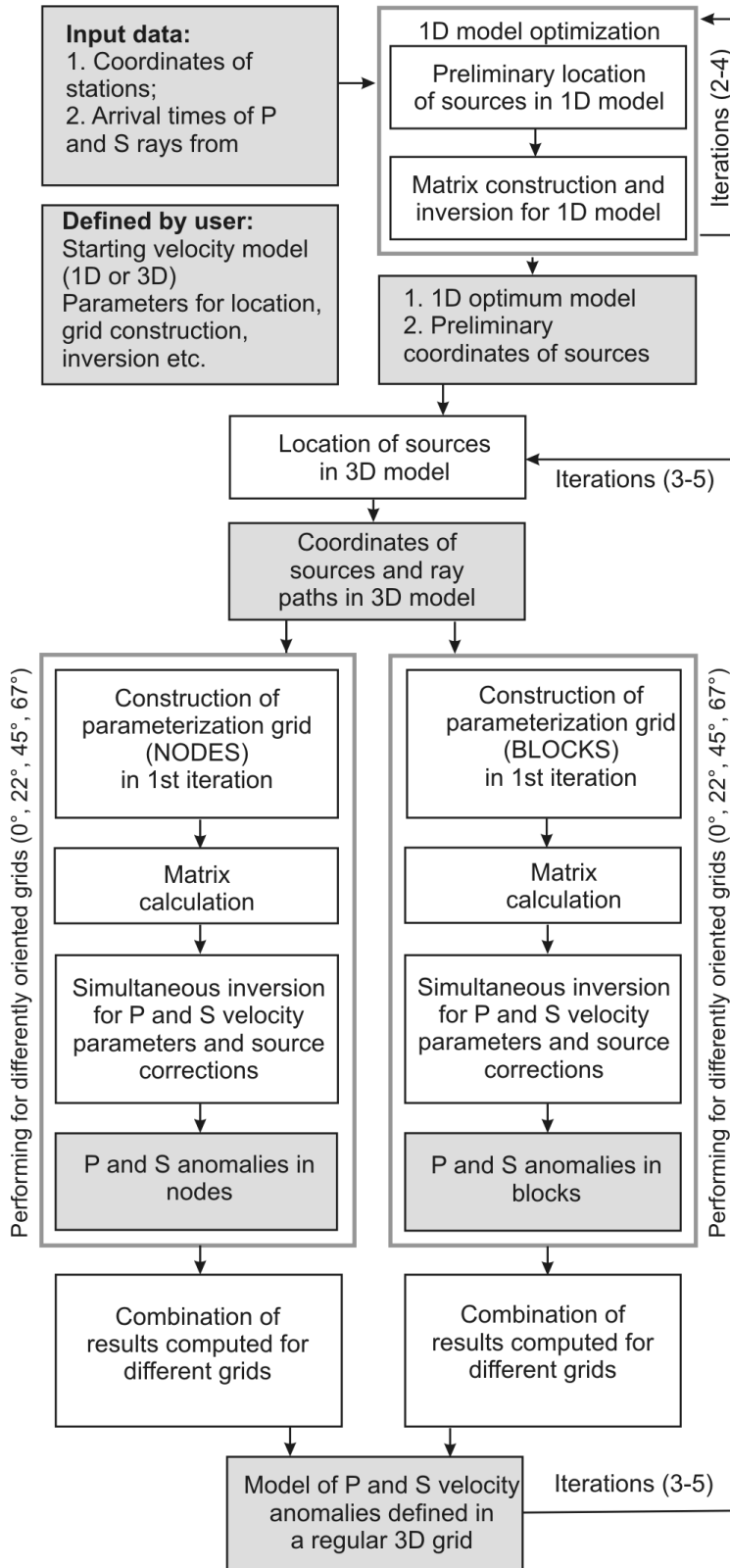


Figure 1.2. More detailed than in Figure 1.1. structure of the main program steps and data blocks in the LOTOS code for the case of Vp-Vs inversion scheme. For Vp-Vp/Vs inversion the structure of files will be the same (but only with node parameterization).

Here we present a short description of the main steps of this code. The general structure of the main stages and data blocks is presented in [Figure 1.1](#). In more details the inversion scheme for Vp-Vs inversion is presented in [Figure 1.2](#). The Vp-Vp/Vs inversion scheme has the same structure of programs, however only for node parameterization. The calculations start with two data files (green block): coordinates of the stations and arrival times of P and S seismic rays from local earthquakes to these stations. Also, additional information such as starting velocity model, parameters of grid and inversion and others is defined in a separate file (blue block). It is possible to use preliminary locations and origin times provided by picking tools or/and catalogues, but this information is not strictly required. In the case of absence of any information about sources, *LOTOS* starts searching for the source hypocenter either from the center of the network or from the station with minimal arrival times. The algorithm contains the following general steps:

1. Simultaneous optimization for the best 1D velocity model and preliminary location of sources;
2. Location of sources in the 3D velocity model;
3. Simultaneous inversion for the source parameters and velocity model using several parameterization grids.

Steps 2 and 3 are repeated in turn one after another in several iterations. Now let us describe some features of these steps.

## 1.2. Algorithm for 1D velocity optimization and preliminary source location

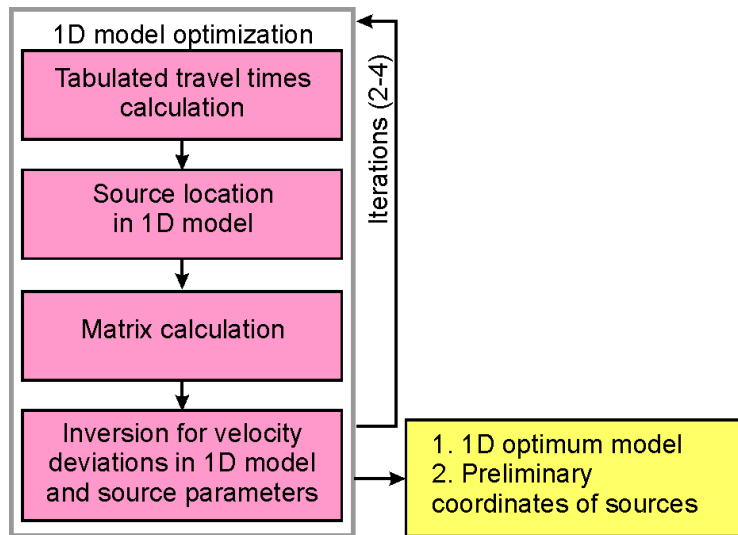


Figure 1.3. The main steps for the 1D velocity optimization and preliminary source locations.

The general structure of the algorithm for 1D velocity optimization and preliminary source locations is presented in [Figure 1.3](#). It includes the following steps:

**Step 0.** Data selection for optimization. From the entire data catalogue, we select events that should be distributed as uniformly with depth as possible. To do this, we

select for each depth interval the events with the maximum number of recorded phases. The total number of events in each depth interval should be less than a predefined value (e.g., 4 events).

**Step 1.** Calculation of a travel time table in a current 1D model. In the first iteration, the model is defined manually with the use of possible a priori information. The travel times between sources at different depths to the receivers at different epicentral distances are computed in a 1D model using analytical formulae (Nolet, 1981). The algorithm allows the incidence angles of the rays to be defined in order to achieve similar distances between rays at the surface.

**Step 2.** Source location in the 1D model. The travel times of the rays are computed using tabulated values obtained in Step 1. The travel times are then corrected for elevations of stations. The source location is based on calculating a goal function (GF) that reflects the probability of a source location in a current point. The form of the GF is defined in (Koulakov, Sobolev, 2006). Searching for the GF extreme is performed using a grid search method. We start from a coarse grid and finish our search in a fine grid. This step is performed relatively quickly as it uses the tabulated values of the reference travel times. This location algorithm is very stable. For example, it can find the correct source coordinates even if it is located at a distance of 400-500 km from the initial searching point.

**Steps 1 and 2** can be replaced with another algorithm of fast locations of sources in the 1D model based on the linear ray approximation. Instead of ray tracing, we compute travel times by integrating along straight lines. This works well for small areas with high topography.

**Step 3.** Calculation of the first derivative matrix along the rays computed in the previous iteration. Each element of the matrix  $A_{ij}$  is equal to the time deviation along the  $j$ -th ray caused by a unit velocity variation at the  $i$ -th depth level. The depth levels are defined uniformly and the velocity between the levels is approximated as linear.

**Step 4.** Matrix inversion is performed simultaneously for the P and S data using the matrix computed in Step 3. In addition to the velocity parameters, the matrix contains the elements to correct the source parameters ( $dx$ ,  $dy$ ,  $dz$  and  $dt$ ). The data vector contains the residuals computed after the source location (Step 2). Regularization is performed by adding a special smoothing block. Each line of this block contains two equal non-zero elements with opposite signs that correspond to neighboring depth levels. The data vector in this block is zero. Increasing the weight of this block smoothes the solution. If there is a-priori information about existence of interfaces (e.g. Moho), it can be included in the inversion. In this case, the link between the pair of nodes just above and below the interface would be skipped.

Optimum values for free parameters (smoothing coefficients and weights for the source parameters) are evaluated on the basis of synthetic modeling. The inversion of this sparse matrix is performed using the LSQR method (Page, Saunders, 1982, Van der Sluis, Van der Vorst, 1987). A sum of the obtained velocity variations and the current reference model is used as a reference model for the next iteration, which contains steps 1-4. The iterations are repeated several times. Among the results at each iteration we select one with the minimum RMS and use it for further processing.



### 1.3. Bending algorithm for ray tracing in a 3D velocity model

One of the key features of the *LOTOS* code is a ray tracing algorithm based on the Fermat principle of travel time minimization. A similar approach is used in other algorithms (e.g., [Um and Thurber \(1987\)](#)) and is called bending tracing. We present our own modification of the bending algorithm. An important feature of this algorithm is that it can use any parameterization of the velocity distribution. It is only necessary to define uniquely one positive velocity values at any point of the study area. It can be done with nodes or cells, with polygons or analytical laws, or any other ways. The current version of *LOTOS* includes many various options for velocity definition. However, if necessary, any other parameterization can be easily included.

A basic principle of our bending algorithm is shown in [Figure 1.4](#). In the presented example, we use a model with exaggerated velocity contrasts. In the vertical direction, the velocity varies from 2.5 to 9 km/s. The checkerboard anomalies have amplitudes of  $\pm 30\%$ . It is obvious that in this model, the ray path has a fairly complicated shape determined by the velocity distribution.

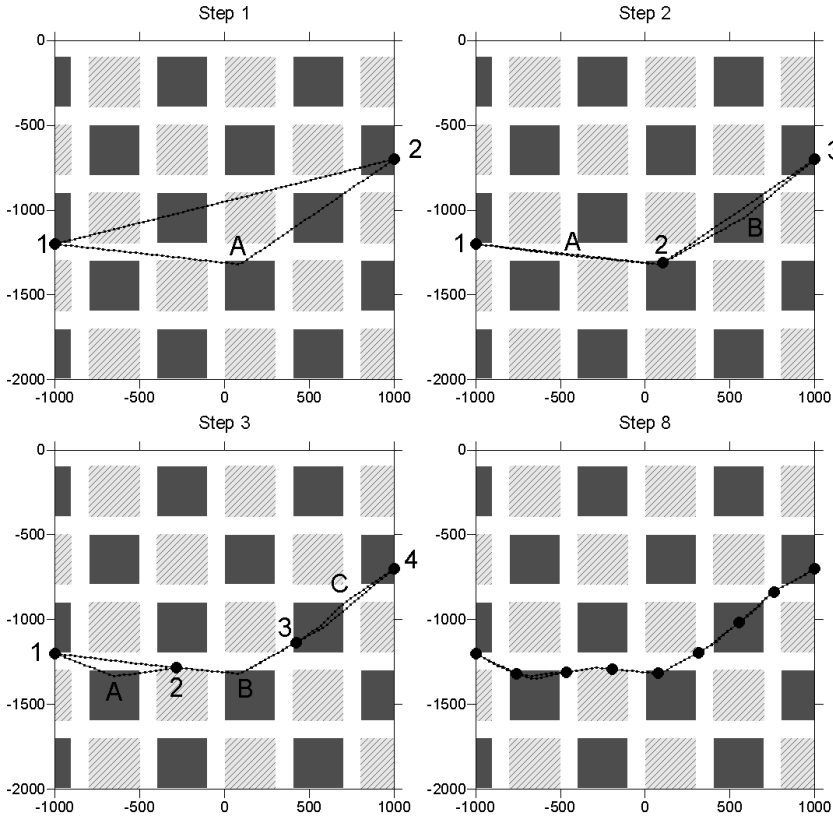


Figure 1.4. Grounds of the bending algorithm. Ray construction is demonstrated for a model with exaggerated velocity contrasts. 1D velocity varies from 2500 to 9000 m/s at 2000 m depth. Hatched light grey patterns represent negative anomalies of -30%; dark grey patterns are positive anomalies of +30%. Details of the bending algorithm are given in the text.

Searching a path with minimum travel time is performed in several steps. The starting ray path is a straight line. In the first step (Plot A), the ends of the rays are fixed

(points 1 and 2), and point *A* in the center of the ray is used for bending. Deformation of the ray path is performed perpendicular to the ray path in two directions: in and across the plane of the ray. The values of shift of the new path with respect to the previous one depend linearly on the distance from *A* to the ends of the segment, as shown in Figure 1.4. In the second step (Plot B), three points are fixed (points 1, 2, and 3), and deformation of the ray path is performed in two segments (at points *A* and *B*). In a third step (Plot C), four points are fixed and three segments are deformed. In Plot D, the results of bending are shown for eight segments. The ray constructed in this way tends to travel through high-velocity anomalies and avoids low velocity patterns. It should be noted that although a 2D model is shown in Figure 3, the algorithm is designed for the 3D case.

## **1.4. Iterative tomographic inversion**

### **1.4.1. Source locations in a 3D velocity model**

The starting 1D velocity model and initial locations of sources are obtained in the step of 1D model optimization ([Section 1.2](#)). The sources are then relocated using a code based on 3D ray tracing (bending). As for 1D modification, the location algorithm is based on finding an extreme of a goal function. The description of the goal function is the same as in the 1D case. However, the grid search method, which is very efficient for 1D models, seems to be too time consuming when 3D ray tracing is applied. We therefore use a gradient method ([Koulakov et al., 2006](#)) to locate sources in 3D models, which is not as robust as the grid search method, but is much faster.

### **1.4.2. Parameterization with nodes**

The parameterization method uses a mesh of nodes that are installed in the study volume using the algorithm described in [Koulakov et al. \(2006\)](#). The nodes are fixed on vertical lines distributed regularly in map view (e.g., with steps of 5x5 km). In each vertical line, the nodes are installed according to the ray distribution. In the absence of rays, no nodes are installed. The spacing between the nodes is chosen to be smaller in areas of higher ray density. However, to avoid excessive concentration of nodes, a minimum spacing is defined (e.g., 5 km). Between the nodes, the velocity distribution is approximated linearly. Examples of node distributions in the depth interval of 10-20 km for the P model are shown in the upper row of [Figure 1.5](#).

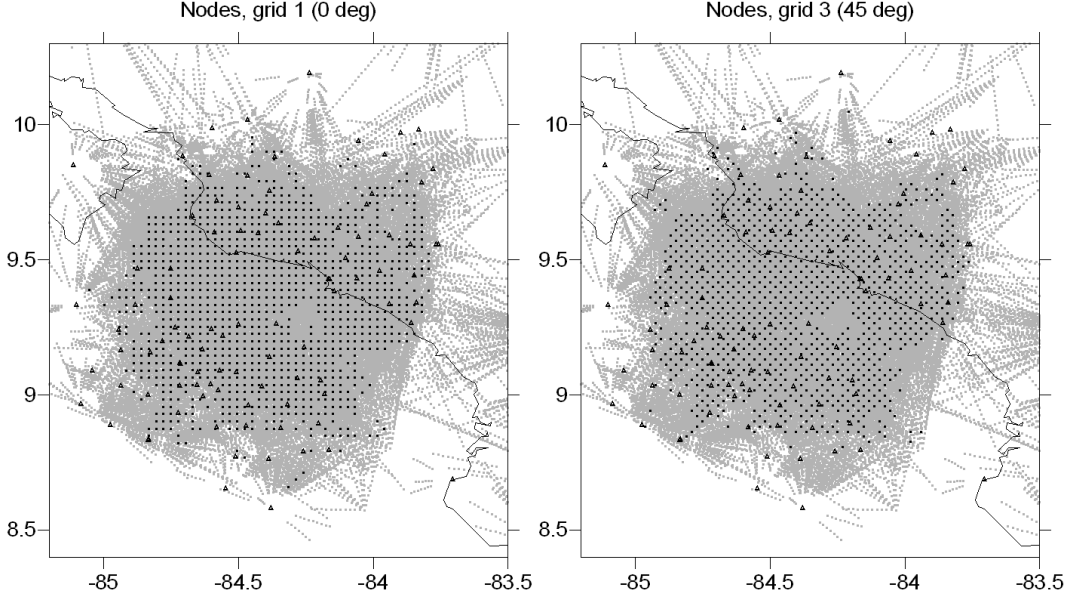


Figure 1.5. Node parameterization provided by LOTOS code. For this case, two orientations of grids are shown ( $0^\circ$  and  $45^\circ$  in left and right columns, respectively). Grey points show the paths of P rays in the depth interval 10-20 km. The grids are used to compute the P model and presented for a depth of 20 km. Triangles represent seismic stations. This model correspond to a real dataset in Costa-Rica.

In order to reduce the effect of node distributions on the results, we perform the inversion using several grids with different basic orientations (e.g.,  $0^\circ$ ,  $22^\circ$ ,  $45^\circ$ , and  $67^\circ$ ). Examples of two different grids for node parameterizations with the basic orientations of  $0^\circ$  and  $45^\circ$  are demonstrated in Figure 1.5 (left and right columns, respectively). After computing the results for grids with different orientations, they are averaged into one summary model, reducing any artifacts related to grid orientation.

It is important to note that the total number of nodes can be larger than the ray number. This does not cause any obstacles for performing the inversion, because in our case, the unknown parameters associated with the parameterization nodes are not independent, but are linked through a smoothing block that will be described later for the step of inversion. If the parameterization spacing is significantly smaller than the sizes of the expected anomalies, results of the inversion are almost independent of the distribution of nodes. In this sense, our parameterization can be considered quasi continuous. The construction of the parameterization grids is performed only in the first iteration. In the next iterations, the algorithm uses the same node configurations.

### 1.4.3. Matrix calculation and inversion for the case of Vp-Vs scheme

The first derivative matrix is calculated using the ray paths computed after the source locations in the 3D model. Each element of the matrix,  $A_{ij} = \partial t_i / \partial v_j$ , is equal to the time deviation along the  $i$ -th ray due to a unit velocity perturbation in the  $j$ -th node/block.

Inversion of the entire sparse  $A$  matrix is performed using an iterative *LSQR* code (Page, Saunders, 1982, Van der Sluis, van der Vorst, 1987). In addition to P and S velocity parameters, the matrix contains the elements responsible for the source ( $dx$ ,  $dy$ ,

$dz$ , and  $dt$ ), and station corrections. Amplitude and smoothness of the solution is controlled by two additional blocks. The first block is a diagonal matrix with only one element in each line and zero in the data vector. Increasing the weight of this block reduces the amplitude of the derived P or S velocity anomalies. The second block controls the smoothing of the solution. Each line of this block contains two equal nonzero elements of opposite sign that correspond to all combinations of neighboring node/cells in the parameterization grid. The data vector in this block is also zero. Increasing the weight of this block reduces the difference between solutions in neighboring nodes, resulting in smoothing of the computed velocity fields.

#### **1.4.4. Iterative cycling**

The steps of grid construction, matrix calculation and inversion are performed for several grids with different basic orientations. The resulting velocity anomalies derived for all grids are combined and computed in a regular grid. This model is added to the absolute velocity distributions used in a previous iteration. New iterations repeat the steps of source location, matrix calculation, and inversion.

In our opinion, the most effective and unbiased way to evaluate optimal values of free inversion parameters is by performing synthetic modeling that reproduces the real situation. This also allows qualitative estimates of amplitudes of seismic anomalies in the real earth.

## 2. General file structure of files and folders in LOTOS

### 2.1. List of folders in the root directory

The file structure in the root directory with short descriptions is presented in the [Figure 2.1](#)

DATA	- folder which contains all the data and models
COMMON	- folder with files containing various information, such as files with color scales, coastal line, political boundaries, as well as the program for previewing
PROGRAMS	- folder with programs for performing tomographic inversion
<a href="#">all_areas.dat</a>	- file which defines areas and models to be processed for inversion (defined by user)
<a href="#">model.dat</a>	- file with information about currently processed model (updated automatically)
<a href="#">START.BAT</a>	-BATCH file for execution real or synthetic calculations

Figure 2.1 Folders (pink boxes) and files (white boxes) in the root directory of *LOTOS-10*

The other folders are created automatically during performing calculations. They include:

1. “[TMP\\_files](#)” with subfolders contains GRD, BLN and DAT file which represent the intermediary and final results. They can be used directly in *SURFER* for visualization.
2. “[PICS](#)” with subfolders contains PNG bitmap files with previewing the results.

In addition, one file “[preview\\_key.txt](#)” with a key for previewing should be also defined

### 2.2. Batch files in the root folder

There are several useful batch files in the root folder which facilitate working with LOTOS:

- [START.BAT](#): This file executes the entire workflow of calculations for full processing of one or several models defined in [all\\_areas.dat](#).
- [check\\_ini\\_data.bat](#): This batch file is executed in the beginning of work with a new dataset. It checks the correctness of initial data definition. If the data are correct, this command provides the information about numbers of events and rays, gives the coordinate ranges and creates a picture with events and stations in folder [PICS](#).
- [visual\\_grid\\_raypaths.bat](#): This command can be executed after termination of the 1<sup>st</sup> iteration of tomographic inversion. It produces the plots of

the ray distributions and configurations of differently oriented grids. They are plotted in vertical and horizontal sections.

- `grid_matr_inv_visual.bat`: This command is useful for tuning the parameters of grids and inversion. After terminating 1<sup>st</sup> iteration, if we wish to play with parameters without running the time consuming step of source location, we can run this command to repeat grid construction, matrix calculation, inversion and visualization for other values of parameters. Note, that in file `model.dat` there should be defined current area and model and 1<sup>st</sup> iteration (3<sup>rd</sup> line). Important that this command runs only for one grid; therefore in `MAJOR_PARAM.DAT` there should be defined only one grid orientation.
- `inv_visual.bat`: Same as in the previous case, but only for the inversion and visualization steps. This command is useful for fast tuning of inversion parameters. This helps to see immediately the effects of changing different inversion parameters. Again, this command is valid for 1<sup>st</sup> iteration, for only one grid.
- `visual_results.bat`: This command performs the visualization of the results previously computed corresponding to area, model and iteration defined in file `model.dat`. The visualization is performed for horizontal and vertical sections. Resulting pictures can be seen in folder `PICS`, or visualized in SURFER using grid files in folder `TMP_files`.
- `visual_syn_model.bat`: This command visualizes the synthetic model in horizontal and vertical sections. The area and the model should be defined in file `model.dat`.

## 2.3. Structure of the DATA folder

General structure of the `DATA` folder is shown in **Figure 2.2**. The `DATA` folder has two-steps hierarchy structure. It contains the Area folders (e. g. “`AREA_001`”, “`AREA_002`” etc). The name of the Area folder should consists of any 8 characters.

Each area folder contains several folders with real and synthetic models which correspond to the same source/receiver configurations. The names of the sub-folders with real data or synthetic models should also be composed of any 8 characters (e.g. “`MODEL_01`” or “`SYNMOD_1`”). Details of MODEL definition are given in **Sections 2.2.2** and **2.2.3**.

The AREA folder contains also one mandatory folder, “`inidata`”, with initial data (see **Section 2.2.1** for details). The optional “`map`” folder contains coastal lines “`coastal_line.blm`” and political boundaries “`polit_bound.blm`” in BLN format (**Figure 2.3**) which are used for visualization of the results. If these files do not exist, they are skipped and the maps are produced without coastal line.

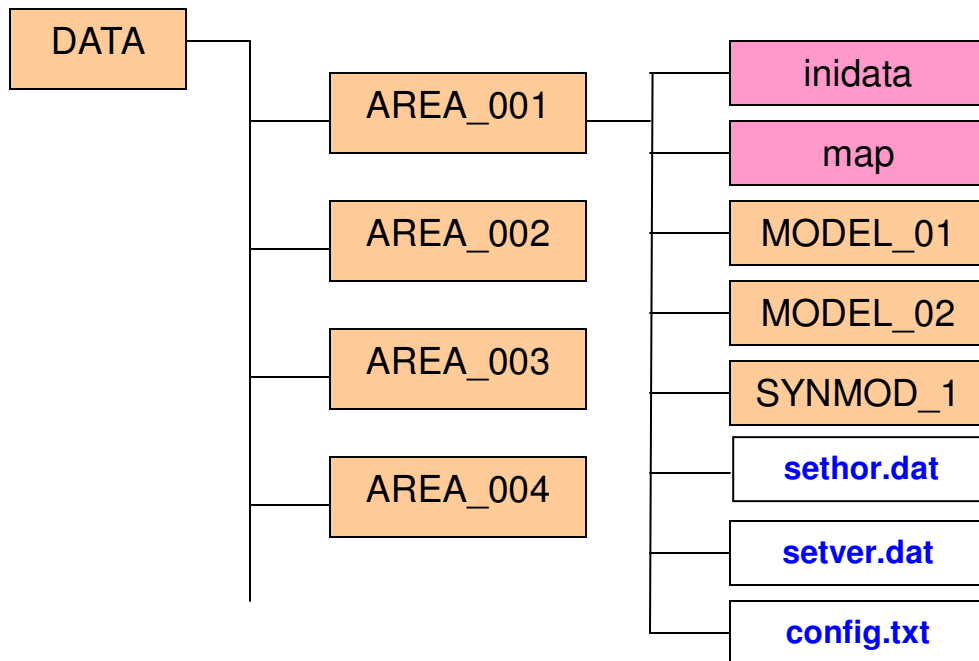


Figure 2.2. Structure of folders (pink and orange boxes) and files (white boxes) in the DATA directory

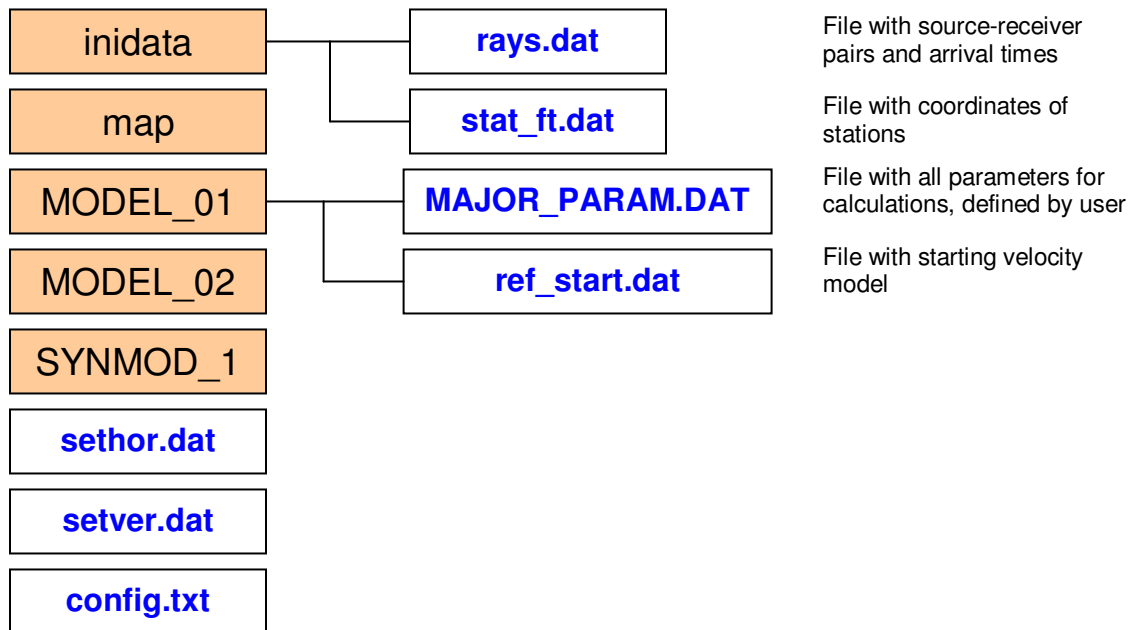


Figure 2.3. Structure of folders (colored boxes) and files (white boxes) the area directory: for “inidata” folder and for “MODEL\_01” folder which corresponds to the case of real data inversion.

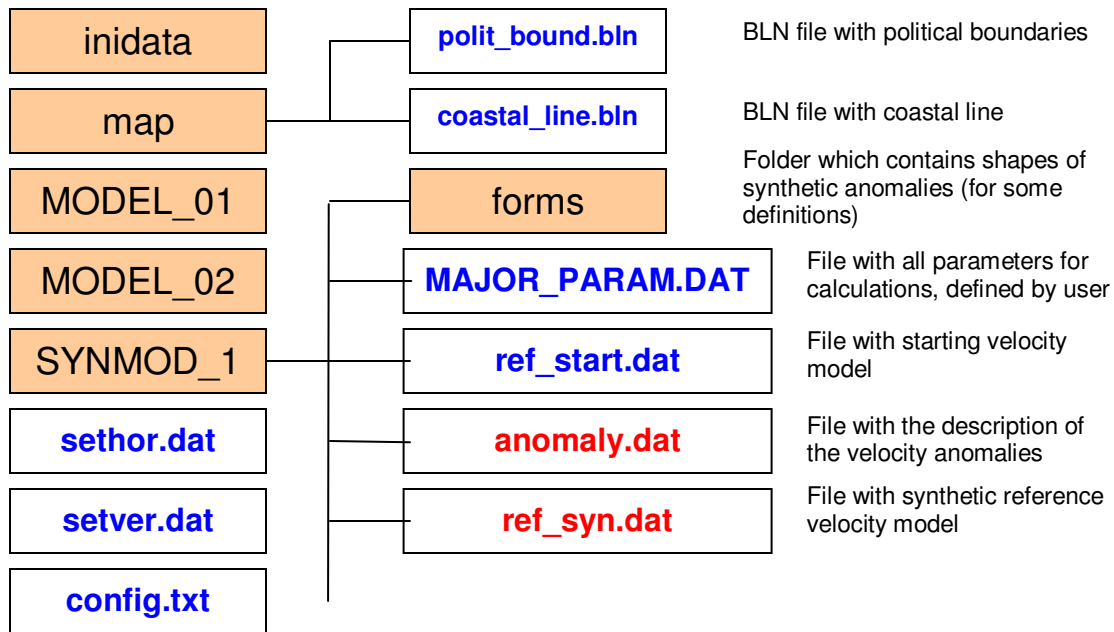


Figure 2.4. Structure of folders (colored boxes) and files (white boxes) the area directory: for “map” folder and for “SYNMOD\_1” folder which corresponds to the case of synthetic modeling. Files indicated by red are related to definition of the synthetic model. Files indicated by blue are the same as in the case of real data inversion (Figure 2.3)

### 2.3.1. Format of the input data in the "inidata" folder

The input data are contained in the “inidata” folder (Figure 2.3) which includes two mandatory files:

1. “rays.dat”: list of all travel times
2. “stat\_ft”: list of stations

Examples of the file “rays.dat” is shown in the examples below (only for two events).

/DATA/DATASET1/inidata/rays.dat

-70.18384	-20.90750	33.93000	16
1	11	25.35048	
2	11	43.90528	
1	8	13.87840	
2	8	24.04150	
1	3	19.67791	
2	3	34.08391	
1	9	8.231387	
2	9	14.26471	
1	4	15.32037	
2	4	26.55076	
1	5	14.31970	
1	2	27.74261	
1	6	20.04849	
1	10	31.38969	
1	12	21.47189	
1	7	22.69749	
-70.29102	-20.40950	25.36000	18
1	10	25.71727	



2	10	44.54014
1	3	14.10538
2	3	24.42953
1	11	22.75151
2	11	39.39725
1	1	30.41859
2	1	52.68230
1	2	21.69525
2	2	37.57591
1	5	18.71232
2	5	32.39684
1	7	15.83892
2	7	27.43102
1	12	23.01083
2	12	39.85330
1	8	7.472907
1	4	13.51934

First line of “`rays.dat`” is a description of event which includes geographical coordinates: longitude (degrees), latitude (degrees, S-negative) and depth (km, down-positive), and number of recorded phases, **NPhase**. If information about source is not available, any coordinate within the study area can be indicated. In any case, the source will be relocated. After the line of source description, **NPhase** lines follow. First column is phase indicator (1:P, 2:S), second column is number of station according to the list in “`stat_ft.dat`”. Third column is travel time, in seconds.

Examples of the file “`stat_ft.dat`” is shown in the examples below

`/DATA/DATASET2/inidata/stat_ft.dat`

-71.12700	-18.63300	1.000000
-71.09300	-19.31700	1.000000
-71.02700	-20.00000	1.000000
-70.97700	-20.66700	1.000000
-70.91000	-21.31700	1.000000
-70.84300	-21.93300	1.000000
-70.71700	-19.63300	1.000000
-70.63300	-20.28300	1.000000
-70.55000	-20.96700	1.000000
-71.80000	-19.63300	1.000000
-71.70000	-20.28300	1.000000
-71.61700	-20.96700	1.000000

File “`stat_ft.dat`” contains geographical coordinates of stations: longitude, latitude (S-negative) and elevation (km, above sea level is negative). Fourth column with station names is optional. Line number should correspond to the station numbers in the “`rays.dat`” file.

### 2.3.2. Organization of a MODEL folder

A MODEL folder is created either for observed data or synthetic tomographic models. The name of the MODEL folder should contain any 8 characters (e.g. “`BRD_mod2`”, “`MODEL_01`”, “`VBRDmod2`”). The structures of the MODEL folder for the cases of real and synthetic data inversion are shown in **Figures 2.3 - 2.4**.

For real data inversion MODEL folder initially should contain two files indicated in **Figure 2.3:**

```
MAJOR_PARAM.DAT
ref_start.dat
```

For synthetic modeling two other files which define the synthetic model are added (See **Figure 2.4:**

```
MAJOR_PARAM.DAT
ref_start.dat
anomaly.dat
ref_syn.dat
```

For the case of synthetic modeling, one additional folder, “forms” might be required. It contain shapes of synthetic patterns which are used for definition of the synthetic model (See details in **Section 5**)

### 2.3.3. Major free parameters, file “MAJOR\_PARAM.DAT”

The main files with the initial parameters are shown in **Figures 2.3 - 2.4**. Most of the parameters for source location and inversion are defined in ‘MAJOR\_PARAM.DAT’. The content of this file is organized by rubrics. Each rubric starts with a key line. For example:

```
GENERAL INFORMATION :
AREA_CENTER :
ORIENTATIONS OF GRIDS :
INVERSION PARAMETERS :
etc.
```

Example of the “MAJOR\_PARAM.DAT” file is given below (names of rubrics are indicated with red):

```
/DATA/DATASET2/MODEL_01/ini_param/MAJOR_PARAM.DAT
*****
GENERAL INFORMATION :
1 KEY 1: REAL; KEY 2: SYNTHETIC
1 KEY 1: Vp and Vs; KEY 2: Vp and Vp/Vs
0 KEY 0: all data, KEY 1: odd events, KEY 2: even events
1 Ref. model optimization (0-no; 1-yes)

*****
AREA_CENTER :
-71 -20.5      Center of conversion to XY

*****
ORIENTATIONS OF GRIDS :
4              number of grids
0 22 45 67     orientations

*****
1D MODEL PARAMETERS :
2              Iterations for 1D inversions
```

```

-10 3. 5          zmin, dzstep depth step for finding the best event
1 1 300          dsmin, dzlay,zgrmax : parameters for 1D tracing
5.              dz_par, step for parameterization
0.2
6. 9. sm_p,sm_s
0.0 0.0 rg_p,rg_s
10 10 1 w_hor,w_ver,w_time
300             LSQR iterations
0 nsharp
27 27          z_sharp

*****
INVERSION PARAMETERS :
40 1             LSQR iterations, iter_max
1 1. Weights for P and S models in the upper part
0.7 1.2 level of smoothing (P, S and crust)
0.0 0.0 regularization level (P, S and crust)

0.0001 0.0001    weight of the station corrections (P and S)
2.0 wzt_hor
2.0 wzt_ver
1.0 wzt_time

*****
Parameters for location in 1D model using reference table
and data selection:
*****
LIN_LOC_PARAM :
9              Minimal number of records
100            km, maximum distance to nearest station
1.7            S max resid with respect to P max resid
100            dist_limit=100 : within this distance the weight is equal
1              n_pwr_dist=1    : power for decreasing of W with distance
30            ncyc_av=10

! For output:
30            bad_max=30        : maximal number of outliers
0.05          maximal dt/distance
30            distance limit

10            Frequency for output printing

3             Number of different grids

-----
10 10 10       dx,dy,dz
0.0            res_loc1=0.2    : lower limit for location (for LT residuals, W=1)
5.            res_loc2=1.5    : upper limit for location (for GT residuals, W=0)
2.            w_P_S_diff=2 (+ causes better coherency of P and S)

-----
3 3 3          dx,dy,dz
0.0            res_loc1=0.2    : lower limit for location (for LT residuals, W=1)
3.            res_loc2=1.5    : upper limit for location (for GT residuals, W=0)
2.            w_P_S_diff=2 (+ causes better coherency of P and S)

-----
0.5 0.5 0.5    dx,dy,dz
0.            res_loc1=0.2    : lower limit for location (for LT residuals, W=1)
1.5           res_loc2=1.5    : upper limit for location (for GT residuals, W=0)
2.            w_P_S_diff=2 (+ causes better coherency of P and S)

*****
Parameters for 3D model with regular grid
*****
3D_MODEL PARAMETERS:
-200. 200. 5    xx1, xx2, dxx,
-300. 300. 5    yy1, yy2, dyy,
-5. 150. 5      zz1, zz2, dzz
15             distance from nearest node
0              Smoothing factor1

*****
Parameters for grid construction

```

```

*****
GRID_PARAMETERS:
-300. 300. 5.      grid for ray density calculation (X)
-300. 300. 5.      grid for ray density calculation (Y)
-5. 150. 5.        min and max levels for grid
1                  ! Grid type: 1: nodes, 2: blocks
5.                !min distance between nodes in vert. direction
0.05 100.0 !plotmin, plotmax= maximal ray density, relative to average
-3.                !zupper: Uppermost level for the nodes

0.3                !dx= step of movement along x
0.3                !dz= step of movement along z

*****
Parameters for location in 3D model using bending tracing
*****
LOC_PARAMETERS:
! Parameters for BENDING:
4          ds_ini: basic step along the rays
10         min step for bending
0.04       min value of bending
10         max value for bending in 1 step

! Parameters for location
50         dist_limit=100 : within this distance the weight is equal
1          n_pwr_dist=1   : power for decreasing of W with distance
30         ncyc_av=10

0.         res_loc1=0.2   : lower limit for location (for LT residuals, W=1)
2.         res_loc2=1.5   : upper limit for location (for GT residuals, W=0)
2.         w_P_S_diff=2 (+ causes better coherency of P and S)
5.         stepmax
0.5        stepmin

5          Frequency for output printing

*****
Parameters for calculation of the reference table:
*****
REF_PARAM:

1.         min step
200.       max depth
300.       max distance
3          number of depth steps
0 1 depth, step
20 2       depth, step
50 5       depth, step
200        maximal depth

```

After the key line, a set of current rubric parameters follows with a fixed amount of parameters in each line. The numerical format of the parameters is free. The order of groups and number of empty lines between the rubrics are free.

For example, the first rubric in the presented file is:

```

*****
GENERAL INFORMATION :
1  KEY 1: REAL; KEY 2: SYNTHETIC
1  KEY 1: Vp and Vs; KEY 2: Vp and Vp/Vs
0  KEY 0: all data, KEY 1: odd events, KEY 2: even events
1  Ref. model optimization (0-no; 1-yes)

```

It presents the main keys for performing the task:

- 1.) Is it real or synthetic modeling?
- 2.) Is the inversion performed for Vp-Vs or Vp-Vp/Vs scheme?

- 3.) Are all data used, or only subsets with odd and even events?
- 4.) Is the optimization for the 1D model performed?

Another example is in the second rubric:

```
*****
AREA_CENTER :
-71 -20.5      Center of conversion to XY
```

This parameter group contains the geographical coordinate of the central point of the study area which is used as a reference point for conversion to Cartesian coordinates. The meaning of the key parameters will be explained during description of the main steps.

More details about MAJOR\_PARAM.DAT will be given in the text and in separate files.

### 2.3.4. 1D starting velocity model, file “ref\_start.dat”

Example of file with starting 1D velocity distribution:

[/DATA/DATASET1/MODEL\\_01/ref\\_start.dat](#)

1.7	Ratio vp/vs	
-1.000	4.3	2.22
6.000	5.5	3.26
12.000	6.7	3.67
15.000	6.8	3.96
35.000	8.1	4.42
74.000	8.3	4.54
104.000	8.4	4.60
124.000	8.45	4.62
154.000	8.5	4.66
400.000	9.0300	5.00

This file contains the information about starting reference velocity model. First line is the Vp/Vs ratio. If it is zero, the S-velocity is defined according to third column of the following lines. Otherwise, the S velocity is computed from P velocity (second column) using constant value of Vp/Vs ratio. Velocities are defined at some depth levels (first column) and linearly interpolated in between.

The other parameters will be presented in description of the LOTOS algorithm, next section.

## 3. Step-by-step calculations with the LOTOS code

### 3.1. Preliminary location in the 1D velocity model using reference tables

It is recommended to perform the preliminary location of sources using 1D velocity model. It will save a lot of time and increase the accuracy of further processing. The basic principle of the algorithm is described in Section 1.2. The algorithm starts from creation of the reference table. In paths for the files we use the indications:

'//ar//' is the AREA folder  
'//md//' is the MODEL folder  
'//it//' is the number of iteration

#### 3.1.1. Calculation of the reference table

Project: \ PROGRAMS\1\_PRELIM\_LOC\0\_1\_ref\_rays\

The reference model is taken from file:

in 1 iteration: /data/'//ar//'/'//md//'/data/refmod.DAT

The calculations are controlled by parameters in the MAJOR\_PARAM.DAT file (see example below):

/data/DATASET1/MODEL\_01/ MAJOR\_PARAM.DAT

```
*****
Parameters for calculation of the reference table:
*****
REF_PARAM:
1.      epi_step: min horizontal step between output points of rays
200.    zraymax: max depth of ray penetration
300.    distmax: max distance
3       n_interv: number of depth steps
0 1     depth, step
20 2    depth, step
50 5    depth, step
200     depth_max:      maximal depth
```

In this step, a reference table corresponding to all combination of source depths and epicentral distances, for P and S rays, is computed. The receivers are presumed to be located at  $Z=0$ . Equations for the program are taken from G.Nolet, Linearized inversion of (teleseismic) data. In: R.Cassinis, 'The inverse problem in geophysical interpretation, Plenum Press, NY, 1981. The calculation is performed by tracing of rays from the source using fixed small step of the dipping angle variation. Only rays with distance more than “**epi\_step**” from each other are included in the table. The dipping angle varies from  $180^\circ$  (vertical) to the value for which either epicentral distance is more than “**distmax**”, or depth of the deepest point of the ray is greater than “**zraymax**”. Depths of the sources for the table can be defined with variable steps. For example, in a shallow depth interval, the step can be smaller than that for the deeper layers. Number of intervals for the depth step determination is “**n\_interv**”. In the example presented below, we consider 3

intervals. From 0 to 20 km depth, the interval is 1 km; 20-50 km: 2 km; 50-500 km: 5 km.

The resulting table of reference time is written in binary format to the file:

`/data/'//ar//'/'//md//'/data/table.DAT`

### 3.1.2. Source location in a 1D model using the reference table

Project: `\PROGRAMS\1_PRELIM_LOC\0_2_loc_event\`

Input data: `/data/'//ar//'/inidata/rays.data`

The calculations are controlled by parameters in the `MAJOR_PARAM.DAT` file (see example below):

`/data/DATASET1/MODEL_01/MAJOR_PARAM.DAT`

```
*****
Parameters for location in 1D model using reference table
and data selection:
*****
LIN_LOC_PARAM :
9          krat_min: Minimal number of records
100        dist_to_stat: km, maximum distance to nearest station
1.7        P_to_S_res: max resid with respect to P max resid
100        dist_limit: within this distance the weight is equal
1          n_pwr_dist: power for decreasing of W with distance
30         ncyc_av

! For output:
30         bad_max : maximal number of outliers, in %
0.05       dt_dist_max : maximal dt/distance
30         dist_lim : distance limit

10         freq_print: Frequency for output printing on console

3          n_gridding: Number of different grids

-----
10 10 10   dx,dy,dz
0.0        res_loc1: lower limit for location (for LT residuals, W=1)
5.         res_loc2: upper limit for location (for GT residuals, W=0)
2.         w_P_S_diff: (+ causes better coherency of P and S)

-----
3 3 3     dx,dy,dz
0.0        res_loc1: lower limit for location (for LT residuals, W=1)
3.         res_loc2: upper limit for location (for GT residuals, W=0)
2.         w_P_S_diff: (+ causes better coherency of P and S)

-----
0.5 0.5 0.5 dx,dy,dz
0.         res_loc1: lower limit for location (for LT residuals, W=1)
1.5        res_loc2: upper limit for location (for GT residuals, W=0)
2.         w_P_S_diff: (+ causes better coherency of P and S)
```

Source location is based on searching for an absolute extreme of a goal function (GF) which reflects the probability of the source position being at a point in 3D space. The GF is described in [*Koulakov and Sobolev, 2006*]. For this step, the travel times are calculated based on tabulated values computed once for the rays with different epicentral distances and sources depths. The search of the GF maximum is performed starting from

the location in the initial catalogue. If no locations were performed previously, starting point for searching may coincides with center of the network or coordinate of a station with minimum arrival time.

The goal function, which reflects the probability of source position plays a key role in the location algorithm. We propose a special form of the goal function which can be written as:

$$G = \frac{\sum_{i=1}^N A(\Delta t_i) B(d_i) C}{\sum_{i=1}^N B(d_i) C},$$

where  $A$  is a term which reflects the values of residuals:

$$A(\Delta t_i) = \begin{cases} 1, & \text{if } |\Delta t_i| / C_{PS} < \tau_1 \\ (\Delta t_i - \tau_2) / (\tau_1 - \tau_2), & \text{if } \tau_1 < |\Delta t_i| / C_{PS} < \tau_2 \\ 0, & \text{if } |\Delta t_i| / C_{PS} > \tau_2 \end{cases},$$

where  $N$  is the total number of records of the event, and  $\tau_1$  and  $\tau_2$  are predefined limits for the values of residuals (in the presented example **res\_loc1** and **res\_loc2**, respectively). In case if all the residuals are less than  $\tau_1$  the goal function would be 1. Values of  $\tau_1$  and  $\tau_2$  are determined from expected values of velocity anomalies.

$B$  is a term of the distance dependence.

$$B(d_i) = \begin{cases} 1, & \text{if } d_i < d_{\min} \\ (d_{\min} / d_i)^m, & \text{if } d_i > d_{\min} \end{cases},$$

Long rays accumulate more time anomalies along their path and consequently usually have greater residuals. That is why in the location algorithm they should have smaller weight than short rays.  $d_{\min}$  (**dist\_lim**=30, in the example below) is the size of a near zone, where weights of all rays are equal.  $m$  reflects rate of the weight decreasing with distance (**n\_pwr\_dist**).

$C$  is a term discriminating the phase weighting. For P phase, the weight is 1. If S phase has not pair in P phase, its weight should be smaller (in our case,  $W_s=1/\mathbf{P\_to\_S\_res}$ ). In case if there are both P and S phases for one station, we consider differential residual:

$$\Delta t_i = (t_{obs}^S - t_{ref}^S) - (t_{obs}^P - t_{ref}^P)$$

Weight of differential residual is **w\_P\_S\_diff**. Increasing this weight causes better correlation of P and S velocity models.

The time residuals for the location procedure are computed as:

$$\Delta t_i = t_{obs}^P - t_{ref}^P - \Delta t_0 \text{ for P-phase, and S-phase, without pair in P phase.}$$



Correction of the origin time  $\Delta t_0$  is obtained from the condition:

$$\sum_{i=1}^{N_p} B(d_i) (t_{obs}^P - t_{ref}^P - \Delta t_0) = 0$$

where  $t_{obs}^P$  is observed travel time,  $t_{ref}^P$  is a reference travel time computed with the use of the reference table and may be corrected for the Moho depth and topography, if available.

Moreover, each individual observation should satisfy the following condition:

$$|t_{obs}^P - t_{ref}^P - \Delta t_0| < \tau_2$$

Calculation of the goal function is performed in nodes of a regular 3D grid with center in a current point. If maximum of the Goal function is achieved in the border of the grid, we explore another grid with the center in the node with maximum value of the GF. This procedure can be performed in different steps for different grid spacing and parameters. In our case, we define three steps (“**n\_gridding**”). In the first step we use rather coarse grid with spacing of 10 km along X, Y, Z (**dx**, **dy**, **dz**). In other steps the grid becomes finer and the allowed residuals are smaller.

**krat\_min** is the minimal allowed number of picks for an events (summary P + S).

**bad\_max** is the minimal amount of bad residuals, in %, when the event is rejected.

**dt\_dist\_max** and **dist\_lim** are parameters for defining bad residuals after location. If distance to the source (dist) is less than **dist\_lim**, the maximal allowed residual is computed as **dist\*dt\_dist\_max**. Otherwise, the maximal residual is **dt\_dist\_max\*dist\_lim**.

**dist\_to\_stat** is the maximum allowed distance from the nearest station of the network. If it is too big, the source can be outside the network, the GAP (empty sector) is too big and the reliability of source location is too low.

Output data:

```
/DATA/'//ar//'/'//md//'/data/rays0.dat (binary)
/DATA/'//ar//'/'//md//'/data/ztr0.dat
```

The second file contains geographical coordinates of the located sources which can be visualized in Surfer or any other graphical editor.

### 3.1.3. Preliminary location in the 1D velocity model using straight line approximation for the rays

Project: `\PROGRAMS\1_PRELIM_LOC\loc_straight\`

Input data: `/data/'//ar//'/inidata/rays.data`

In cases of significant relief altitude and relatively small size of the study area it is recommended to perform the preliminary location of sources using linear approximation of rays. In this case, the travel times is computed as integral along the straight line between the source and receiver. The principle of location is identical to one described in section 1.3.2.

Output data:

`/DATA/'//ar/'/'//md/'/data/rays0.dat (binary)`

## 3.2. Algorithm for 1D velocity optimization

The purpose of the algorithm is estimation of 1D velocity model which can be then used as starting model for 3D tomographic inversion. We present our own version of the algorithm which is rather simple and fast.

The projects for performing this stage for the case of **the table-based** approximation:

`\PROGRAMS\1_PRELIM_LOC\1_select\  
\PROGRAMS\1_PRELIM_LOC\2_reftable\  
\PROGRAMS\1_PRELIM_LOC\3_locate\  
\PROGRAMS\1_PRELIM_LOC\4_matr\  
\PROGRAMS\1_PRELIM_LOC\5_invers\`

All these steps are directed from one program from the Project:

`\PROGRAMS\1_PRELIM_LOC\START_1D\`

The projects for performing this stage for the case of **the straight-line** approximation:

`\PROGRAMS\1_PRELIM_LOC\1_select\  
\PROGRAMS\1_PRELIM_LOC\_1_lov_line\  
\PROGRAMS\1_PRELIM_LOC\_2_matr_line\  
\PROGRAMS\1_PRELIM_LOC\_3_invers_line\`

All these steps are directed from one program from the Project:

`\PROGRAMS\1_PRELIM_LOC \_START_1D_line\`

The calculations are controlled by parameters in the `MAJOR_PARAM.DAT` file (see example below):

```
/data/DATASET1/MODEL_01/MAJOR_PARAM.DAT
*****
1D MODEL PARAMETERS :
4          iter, Iterations for 1D inversions
-10 3. 5    zmin, dzstep kratmax
1 1 300     dsmin, dzlay, zgrmax : parameters for 1D tracing
5.          dz_par, step for parameterization
```

```

0.2      ray_min, inversion is in depth where ray density > ray_min (normalized)
6. 9.    sm_p,sm_s, smoothing coefficients
0.0 0.0  rg_p,rg_s, regularization coefficients
10 10 1  w_hor,w_ver,w_time, weights for source parameters
300      N_LSQR iterations
0        nsharp, number of interfaces with possible sharp velocity jump
27 27    z_sharp, depth of the interfaces

```

The meaning of these parameters will be explained in the following description.

Below is the description of the main program steps for the 1D velocity optimization.

### 3.2.1. Step 1. Data selection for the optimization.

Project: `\PROGRAMS\1_PRELIM_LOC\1_select\`

Main input file: `"/data/'//ar//'/'//md//'/data/rays0.dat`

Main output file: `"/data/'//ar//'/'//md//'/data/rays_selected.dat`

From the entire data catalogue (file computed after preliminary location in 1D (“`rays0.dat`”) we select the events. As far as possible they should be distributed uniformly with depth. To do so, in each depth interval we select the events with maximum number of recorded phases. Total number of events in each depth interval (`dzstep=3`) should be less than a predefined value (`kratmax=5`).

### 3.2.2. Step 2. Calculation of travel time table in a current 1D model

Project: `\PROGRAMS\1_PRELIM_LOC\2_reftable\`

The input files:

in 1 iteration: `/data/'//ar//'/'//md//'/data/refmod.DAT`

in other iterations: `/data/'//ar//'/'//md//'/data/ref'//it//'.DAT`

(`'//it//'` is the number of iteration)

The output file:

Resulting table of parameters of reference rays in binary format:

`/data/'//ar//'/'//md//'/data/table.DAT`

The travel times between sources at different depths to the receivers at different epicentral distances are computed in 1D model using analytical formulas (Nolet, 1981), same as described in Section 3.1.1.

### 3.2.3. Step 3. Source location in the 1D model.

Project: `\PROG_1D_MODEL\3_locate\`

The input file: `/data/'//ar//'/'//md//'/data/rays_it'//it-1//'.dat`

The output file: `/data/'//ar//'/'//md//'/data/rays_it'//it//'.dat`

(`'//it//'` is the number of iteration)

This step is performed relatively quickly only for events selected in Step 1 (Section 3.2.1).

The algorithm is the same as described in Section 3.1.2.

### 3.2.4. Step 4. Calculation of the first derivative matrix

Project: `\PROG_1D_MODEL\4_matr\`

The input: `/data/'//ar//'/'//md//'/data/rays_it'//it//'.dat`

The output: `/tmp/matr_1D.dat`

The matrix is computed along the rays traced in 1D model derived in the previous iteration. Each element of the matrix  $A_{ij}$  is equal to time deviation along  $j$ -th ray caused by unit velocity variation in  $i$ -th depth level. The depth levels are defined uniformly (with the step of `dz_par=5`). Velocity distribution between the levels is approximated linearly.

### 3.2.5. Step 5. Matrix inversion

Project: `\PROG_1D_MODEL\5_invers\`

The input: `/tmp/matr_1D.dat`

The main output: `/data/'//ar//'/'//md//'/data/ref'//it+1//'.dat`

The inversion is performed simultaneously for P and S data using the matrix computed in Step 4. Together with velocity parameters, the matrix contains the elements for correction of source parameters ( $dx$ ,  $dy$ ,  $dz$  and  $dt$ ). The data vector contains the residuals computed after source location (Step 3). Regularization is performed by adding a special smoothing block. Each line of this block contains two equal non-zero elements with the opposite signs which correspond to neighboring depth levels. The data vector in this block is zero. Increasing the weight of this block results at smoothing the solution. Amplitude regularization is performed by adding of a diagonal matrix block with only one nonzero element in each line.

#### ***Controlling parameters:***

Smoothing of the resulting velocity variations: `sm_p=6`, `sm_s=9`

Amplitude of the resulting velocity variations: **rg\_p**=0, **rg\_s**=0

**ray\_min**: limit of the ray density normalized with respect to the average ray density. If the ray density at a certain depth is < **ray\_min**, inversion for this depth is not performed.

Weights for the source parameters (horizontal, vertical shift and origin times: **w\_hor**=10, **w\_ver**=10, **w\_time**=1)

Number of LSQR iterations: **N\_LSQR**=300

The inversion of this sparse matrix is performed using the LSQR method (Page, Saunders, 1982, Van der Sluis, van der Vorst, 1987).

The updated reference model is used for the next iteration which consists of the steps 2, 3, 4 and 5. Number of iteration (**iter**=4) is determined according to results of synthetic modeling.

### 3.2.6. Example of practical realization of the algorithm for 1D model optimization

This algorithm is fairly simple in practical realization and relatively fast. For example, performing the optimization based on 100 events using 4 iterations takes about 10-15 minutes in a regular laptop.

Results of 1D model optimization based on real data set are shown in Figure 3.1. Here we use different starting models to investigate the stability of the optimization. For example, in the cases plotted in A and B, the values of Vp/Vs in the starting 1D velocity distributions are significantly different. Nevertheless the results of optimization in these cases are similar. In case of shallower depth of the uppermost low-velocity layer (crust) in the starting model (plot C), the optimized model tends to deepen this layer. Combination of all the resulting velocity distribution in Plot E shows that the most coherent results are obtained for the depths below 50 km. It seems to us paradoxical because, as will be shown later, the vertical resolution below 50 km depth is rather poor. In the depth interval of 0-50 km, absolute velocities vary in a range of 10%. A model with the best fit (Plot E and blue line in F) is used as a reference distribution for further 3D inversion.

In order to check the reliability of the optimization results and to estimate the optimum values of free parameters, we have performed estimation of 1D velocity model in a synthetic test (Figure 3.1). The synthetic model is represented by checkerboard anomalies of  $\pm 7\%$  amplitude superimposed with 1D absolute velocity distribution. Optimization of 1D model for this case has started with a model (black line in Figure 3.1) which was knowingly very different of the “true” synthetic 1D basic model (blue line). The derived model (green line) appears to be fairly close to the synthetic “true” model. The optimum free parameters, which provided the best result, were then used for the case of real data processing.

*Examples of 1D model optimization:*

Below, two examples of 1D velocity model optimizations are presented for two different starting models:

Starting model 1:			Starting model 2:		
-1.000	4.2	2.82	-1.000	4.0	2.62
6.000	5.5	3.26	6.000	5.2	3.06
12.000	6.2	3.67	12.000	6.4	3.47
15.000	6.7	3.96	15.000	6.5	3.76
35.000	8.0	4.62	35.000	7.8	4.42
74.000	8.2	4.74	74.000	8.0	4.54
104.000	8.3	4.80	104.000	8.1	4.60
124.000	8.35	4.82	124.000	8.15	4.62
154.000	8.4	4.86	154.000	8.2	4.66
400.000	9.0300	5.00	400.000	9.0300	5.00

Result of optimization for this starting model is shown in [Figure 3.1](#)

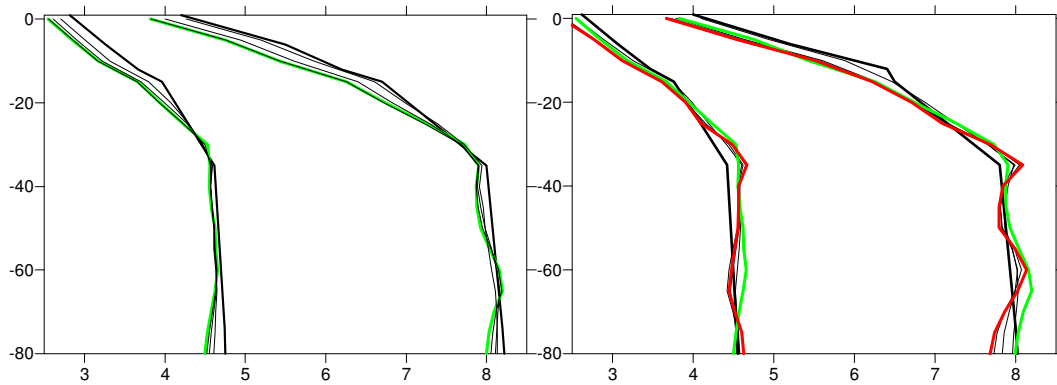


Figure 3.1. Results of 1D velocity model optimization using the starting model 1 (left) and 2 (right). Black bold line is starting model; thin lines are the results after 1-3 iterations; green line is the final result after 4-th iteration with model 1; red line is final result for the model 2.

### 3.3. Iterative 3D tomographic inversion and source relocation

#### 3.3.1. General structure of programs

We remind that in paths for the files we use the indications:

'//ar//' is the AREA folder  
 '//md//' is the MODEL folder  
 '//it//' is the number of iteration

The actual version of *LOTOS-10* allows performing inversion using two different methods of parameterization for the case of  $V_p - V_s$  inversion scheme.

The inversion workflow consists of consequent execution of the following projects:

\PROGRAMS\2\_INVERS\_3D \1\_locate\  
 \PROGRAMS\2\_INVERS\_3D \2n\_ray\_density\

```

\ PROGRAMS\2_INVERS_3D \3n_grid\
\ PROGRAMS\2_INVERS_3D \4n_tetrad\
\ PROGRAMS\2_INVERS_3D \5n_sosedi\
\ PROGRAMS\2_INVERS_3D \6n_matr\
\ PROGRAMS\2_INVERS_3D \7n_invers\
\ PROGRAMS\2_INVERS_3D \8n_3D_model\

```

### 3.3.2. Source location in a 3D velocity model

Project: \ PROGRAMS\2\_INVERS\_3D \1\_locate\

The main input file:

1. File with S/R pairs and travel times in previous iteration  
/data/'//ar//'/'//md//'/data/rays'//it-1//'.dat

The main output data:

1. File with S/R pairs and travel times  
/data/'//ar//'/'//md//'/data/rays'//it//'.dat
2. Binary file with ray paths:  
tmp/ray\_paths'//it//'.dat

The calculations are controlled by parameters in the file MAJOR\_PARAM.DAT (see example below):

/data/DATASET1/MODEL\_01/MAJOR\_PARAM.DAT

```

*****
Parameters for location in 3D model using bending tracing
*****
LOC_PARAMETERS:
! Parameters for BENDING:
4          ds_ini: basic step along the rays
10         min_segm for bending
0.04       min_value of bending
10         max_value for bending in 1 step

! Parameters for location
50         dist_limit: within this distance the weight is equal
1          n_pwr_dist: power for decreasing of W with distance
30         ncyv_av

0.         res_loc1: lower limit for location (for LT residuals, W=1)
2.         res_loc2: upper limit for location (for GT residuals, W=0)
2.         w_P_S_diff: (+ causes better coherency of P and S)
5.         stepmax
0.5        stepmin

5          Frequency for printing on console

```

Location of sources in the 3D velocity model, is performed on the basis of the rays constructed using our own version of the bending method. The parameters for bending are controlled by four major parameters:

```
! Parameters for BENDING:
```

```

4      ds_ini: basic step along the rays
10     min_segm for bending
0.04   min_value of bending
10     max_value for bending in 1 step

```

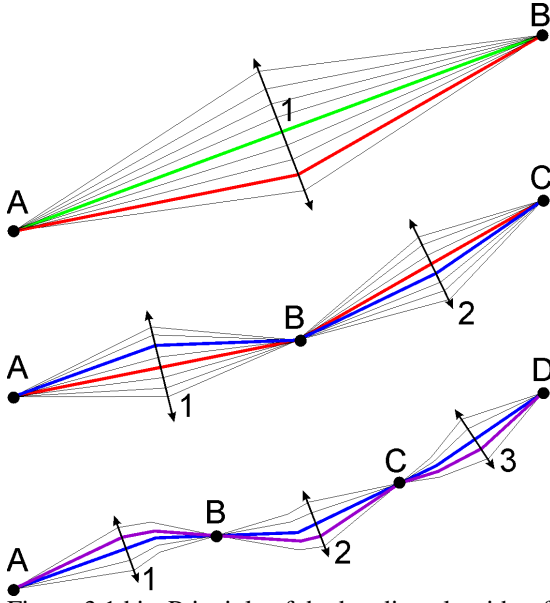


Figure 3.1.bis. Principle of the bending algorithm for the ray tracing

The principle of the bending algorithm for the ray bending is shown in [Figure 3.1.bis](#). The calculations starts from the straight line (upper plot, green line). The travel time is computed based on integration along the path with the integration step **ds\_ini**. This path is deformed in the middle point. We start with maximal bending, **max\_value**. If we achieve improvement (decreasing) of the travel time this path is used as a basic one for the next bending. Otherwise, the bending step is divided in two. Deforming of the segment ends when bending step becomes less than **min\_value**. The obtained path for the first iteration is shown in the upper plot with red line.

Then we perform the similar procedure for two segments starting from the path derived in the previous step.(red line in middle plot). Now the maximal value of bending is **max\_value/2**. The resulting path is shown with blue line. The same procedure is repeated for three (violet line in lower plot) and more segments, and it stops when the length of segments becomes less than **min\_segm**.

The source location is based on searching for a maximum gradient of the GF, similarly as in ([Koulakov, Sobolev, 2006](#)). The direction for searching the GF extreme is determined from the solution of the linear equation system:

$$P_x^i \Delta x + P_y^i \Delta y + P_z^i \Delta z + \Delta t = dt_i$$

where  $P$  is the slowness vector,  $dt_i$  are the observed residuals. If the obtained shift is greater than **stepmax** (5 km, in our case, see example below), it is reduced to the value of the **stepmax**. In the next point we compute the GF using the same formulas, as in case of location in 1D model (Section 2.1.2). If this value is less that in the previous



point, the step is divided in two. When the step becomes less than **stepmin** (0.5 km, in our case), the location procedure stops.

### 3.3.3. Construction of the parameterization grid:

Executed Projects:

```
\ PROGRAMS\2_INVERS_3D \2n_ray_density\
\ PROGRAMS\2_INVERS_3D \3n_grid\
\ PROGRAMS\2_INVERS_3D \4n_tetrad\
\ PROGRAMS\2_INVERS_3D \5n_sosedi\
```

**Main input data:**

1. Binary file with ray paths in the 1<sup>st</sup> iteration:  
tmp/ray\_paths1.dat
2. File with S/R pairs and travel times  
/data/'//ar//'/'//md//'/data/rays1.dat

**Main output data for grid "gr":**

1. File with the list of nodes:  
DATA/'//ar//'/'//md//'/data/gr'//ps//gr//'.dat
2. File with the neighboring pairs of nodes:  
tmp/otr'//ps//gr//'.dat

The calculations are controlled by parameters in the file **MAJOR\_PARAM.DAT** (see example below)

```
/data/DATASET1/MODEL_01/MAJOR_PARAM.DAT
```

```
*****
Parameters for grid construction
*****
GRID_PARAMETERS:
-300. 300. 5.    xpl1,xpl2,dxpl: grid for ray density calculation (X)
-300. 300. 5.    ypl1,ypl2,dypl: grid for ray density calculation (Y)
-5. 150. 5.      zpl1,zpl2,dzpl: grid for ray density calculation (Y)
1               K-grid_type:  Grid type: 1: nodes, 2: cells
5.             dzgrmin:  min distance between nodes in vert. direction
0.05 100.0      densmin, densmax: maximal ray density, relative to average
-3.             zupper:   Uppermost level for the nodes

0.3             dx_step:  step of movement along x
0.3             dz_step:  step of movement along z

*****
ORIENTATIONS OF GRIDS :
4               N_orient: number of different orientations for grids
0 22 45 67      ornt(i): orientation angles, degrees
```

Selected are the most important parameters which determine the vertical and horizontal spacing of the grid.

Parameterization is based on the same approach as used in (Koulakov et al., 2007). The 3D velocity anomalies are computed in nodes distributed in the study volume. Velocity distribution between the nodes is interpolated linearly using subdivision of the study volume into tetrahedral blocks. In this study, the nodes are installed in vertical planes which are spaced at **dygr** = 5 km from each other. In each vertical plane, the nodes are distributed according to ray density. In areas with small amount of rays the distance between nodes is larger. To avoid an excessive concentration of nodes in areas with high ray density, we fix the minimum spacing between nodes at **dzgrmin**=5 km, which is significantly smaller than a characteristic size of the expected anomalies. It is important to note that in our algorithm the resolution of the model does not depend on the grid spacing. It is merely controlled by flattening and regularization parameters during the matrix inversion which is described below. However, since the nodes are placed on planes having a predefined orientation, this can bring some artifacts to the result of the inversion. To reduce the effect of grid orientation we perform the inversion in **N\_orient** differently oriented grids (0°, 22°, 45° and 67°) and then average them.

After this, the nodes are joined with each other and form the tetrahedral cells (program “...**PROG/4\_tetrad**”). The data obtained in this step are used for determination of all existing pairs of neighbouring nodes (program “...**PROG/5\_sosedi**”) which is used in the step of inversion for producing the smoothing block.

### 3.3.5. Calculation of the first derivative matrix

Projects for the matrix calculation: \ **PROGRAMS\2\_INVERS\_3D\6n\_matr\**

#### Main input data:

1. Binary file with ray paths:  
**tmp/ray\_paths'//it//'.dat**
2. File with S/R pairs and travel times  
**/data/'//ar//'/'//md//'/data/rays'//it//'.dat**
- 3a. File with the list of nodes:  
**DATA/'//ar//'/'//md//'/data/gr'//ps//gr//'.dat**
- 3b. File with the list of cells:  
**DATA/'//ar//'/'//md//'/data/block'//ps//gr//'.dat**

#### Main output data for grid “gr” and iteration “it”:

1. File with the matrix:  
**tmp/matr'//gr//it//'.dat**

Matrix calculation, is performed along the rays computed by the bending method after the 2.2.1. The effect of velocity variation at each node on the travel time of each ray ( $\partial t/\partial V$ ) is computed numerically, as in (Koulakov et al., 2006). The data vector corresponding to this matrix consists of residuals obtained after the step of source location.

### 3.3.6. Inversion

Project: \PROGRAMS\7n\_invers\

**Main input data for grid “gr” and iteration “it”:**

1. File with the matrix:  
`tmp/matr'//gr//it//'.dat`
2. File with the neighboring pairs of nodes:  
`tmp/otr'//ps//gr//'.dat`

**Main output data for grid “gr” and iteration “it”:**

- 1 (for nodes, Vp-Vs scheme). File with P velocity anomalies:  
`DATA/'//ar//'/'//md//'/data/vel_p_//it//gr//'.dat`
- 2 (for nodes, Vp-Vs scheme). File with S velocity anomalies:  
`DATA/'//ar//'/'//md//'/data/vel_s_//it//gr//'.dat`
3. File with P station corrections:  
`DATA/'//ar//'/'//md//'/data/stcor_p_//it//gr//'.dat`
4. File with S station corrections:  
`DATA/'//ar//'/'//md//'/data/stcor_s_//it//gr//'.dat`
5. File with source corrections:  
`DATA/'//ar//'/'//md//'/data/ztcors_//it//gr//'.dat`

Inversion is performed simultaneously for P and S velocity anomalies, source parameters (4 parameters for each source) and P and S station corrections. The parameters for the inversion are contained in file `MAJOR_PARAM.DAT`

Example for the Vp-Vs scheme:

`/data/DATASET1/MODEL_01/MAJOR_PARAM.DAT`

```
*****
INVERSION PARAMETERS :
40          num_LSQR: number of LSQR iterations
1 1.        wg_p, wg_s: Weights for P and S velocity models
0.7 1.2     sm_p, sm_s: Smoothing of P and S velocity models
0.0 0.0     rg_p, rg_s: Amplitude damping for P and S velocity models

0.0001  0.0001 st_P, st_S: weight of the station corrections (P and S)
2.0     srce_hor: weight for horizontal shift of sources
2.0     srce_ver: weight for vertical shift of sources
1.0     srce_time: weight for origin time corrections
```

To control flattening and amplitude of the 3D velocity models, the matrix obtained in the Step 3.3.5 is supplemented with special blocks. Each line in the flattening block contains two non-zero elements with opposite signs, corresponding to neighboring parameterization nodes in the model. The data vector corresponding to this block is zero. Increasing the weight of these elements (**sm\_P** and **sm\_S**) has a flattening effect upon the resulting anomalies. The block which controls the amplitude of the model has diagonal structure with only one element in each line and zero values in data vector (controlled by **rg\_P** and **rg\_S**). Station corrections are controlled by **st\_P** and **st\_S**.

Horizontal and vertical shift of sources and correction of the origin time is adjusted by **srce\_hor**, **srce\_ver** and **srce\_time**, respectively. Determination of all the coefficients for the simultaneous inversion is a fairly crucial and delicate problem.

The resulting matrix is inverted using the LSQR method (Paige and Saunders, 1982; van der Sluis and van der Vorst, 1987). The number of LSQR iterations providing a satisfactory convergence in our case is **num\_LSQR**.

### 3.3.7. Calculation of 3D model in a regular grid

Project: `\PROGRAMS\2_INVERS_3D\8n_3D_model\`

#### Main input data for all grids **"gr"** and iteration **"it"**:

1 (for nodes, Vp-Vs scheme). File with P velocity anomalies:

`DATA/'//ar/'/'//md//'/data/vel_p_//it//gr//'.dat`

2 (for nodes, Vp-Vs scheme). File with S velocity anomalies:

`DATA/'//ar/'/'//md//'/data/vel_s_//it//gr//'.dat`

3 (nodes). File with the list of nodes:

`DATA/'//ar/'/'//md//'/data/gr//ps//gr//'.dat`

#### Main output data P or S model **"ps"** and iteration **"it"**:

1. File with a model of velocity anomalies in a regular 3D grid (binary):

`DATA/'//ar/'/'//md//'/data/dv_v//ps//it//'.dat`

After performing the inversions for several grids with different orientations, the velocity anomalies are recomputed in a 3D regular grid. Parameters of the calculation are defined in file `MAJOR_PARAM.DAT` (see example below):

`/data/DATASET1/MODEL_01/MAJOR_PARAM.DAT`

```
*****
Parameters for 3D model with regular grid
*****
3D_MODEL PARAMETERS:
-200. 200. 5   xx1, xx2, dxx: grid parameters along X
-300. 300. 5   yy1, yy2, dyy: grid parameters along Y
-5. 150. 5     zz1, zz2, dzz: grid parameters along Z
15             s_min, distance from nearest node
0              smooth, Smoothing factor1
```

Limits of the volume for interpolation and grid spacing along X, Y and Z are defined in first three lines. **S\_min** means the minimal distance to the nearest parameterization node of one of the used grids. If the distance is larger, this point is outside the resolved area and the value there is presumed 0. The algorithm allows smoothing of the velocity anomalies which is controlled by **smooth**.

### 3.3.8. Practical realization of LOTOS code

Program: `\PROGRAMS\0_START\START\start.f90`

To perform the successful run of the LOTOS, the data structure should be created, as described in [Section 2.2](#). There is possibility to run the steps described in [Section 3.3](#) manually, step by step. However, the LOTOS code contains a program which performs

automatic managing of all steps. This program performs automatically calculation steps for real data processing and synthetic modeling. Several different real or/and synthetic models can be executed at the same run.

The list of models is defined in file `“/all_areas.dat”`. Example of this file is presented below:

```
/all_areas.dat
1: name of the area (any 8 characters)
2: name of the model (any 8 characters)
3: number of iterations
*****
BOLIVAR_ NODES_01 5
BOLIVAR_ BLOCK_01 5
BOLIVAR_ Ver_BRD1 5
```

In the presented example, three models are defined. All of them are from the same AREA folder, **“BOLIVAR\_”**, indicated in 1<sup>st</sup> column. First model is **“NODES\_01”** that is indicated in 2<sup>nd</sup> column. It runs for five iterations (indicated in 3<sup>rd</sup> column). **Maximum 10 different models can be defined in one run.** They will run consequently one after another.

Scenarios of modeling are determined in file `MAJOR_PARAM.DAT` (see example below):

For example, for the presented models, the following settings are defined:

```
/data/BOLIVAR_/NODES_01/MAJOR_PARAM.DAT
*****
GENERAL INFORMATION :
1      KEY 1: REAL; KEY 2: SYNTHETIC
1      KEY 1: Vp and Vs; KEY 2: Vp and Vp/Vs
0      KEY 0: all data, KEY 1: odd events, KEY 2: even events
1      Ref. model optimization (0-no; 1-yes)
```

The first two models are the inversions of real data, while the last case is synthetic modeling.

## 4. Presentation of the results

To control realization of intermediary steps and to visualize the final results, some special program should be run. They produce the files in the `/TMP_file` folder, in corresponding subfolders, which can be presented in Surfer or other similar visualization software. Furthermore, *LOTOS* provides a simple tool for automatic previewing the results which is described in [Section 4.1](#).

### 4.1. Visualization tool for previewing

The *LOTOS* code contains a tool for automatic visualization of the results and intermediate steps. The images are created as PNG bitmap files and stored in a special

folder.

NOTE! Prompt work of the visualization tools requires installing dotNetFramework (dotnetfx.exe). In most Windows operation systems it is installed *a-priori*.

Visualization is performed using a program which is written in C-sharp. The executable file is located in `\COMMON\visual_exe\visual.exe`.

This EXE file can be moved to any location and renamed.

The program contains three major tools which are required for visualization:

- imaging 2D fields using colored contour lines (GRD format);
- drawing polylines (BLN format);
- drawing dots (DAT format) either as circles or squares.

The input files are of the same format as used for SURFER (GDR, BLN and DAT). This program can visualize any order of layers with one of these three information sources. The format of the layers is defined in file `config.txt`, which should be located in the same directory as the EXE file. Example of this file is presented below:

`COMMON\visual_exe\config.txt`

```
400 600
_____ Size of the picture in pixels (nx,ny)
-72.50000 -69.50000
_____ Physical coordinates along X (xmin,xmax)
-22.50000 -18.50000
_____ Physical coordinates along Y (ymin,ymax)
1 1
_____ Spacing of ticks on axes (dx,dy)
picture.png
_____ Path of the output picture
P anomalies, depth= 30 km
_____ Title of the plot on the upper axe
4
_____ Number of layers
*****
1
_____ Key of the layer (1: contour, 2: line, 3:dots)
grid.grd
_____ Location of the GRD file:
scale.scl
_____ Scale for visualization
-10 10
_____ scale diapason:
*****
2
_____ Key of the layer (1: contour, 2: line, 3:dots)
coastal_line.blm
_____ Location of the BLN file
2
_____ Thickness of line in pixels
0 130 255
_____ RGB color:
*****
3
_____ Key of the layer (1: contour, 2: line, 3:dots)
dots.dat
_____ Location of the DAT file
2
_____ Symbol (1: circle, 2: square)
5
```

```

Size of dots in pixels
0 0 0
RGB color:

```

This file example contains four data groups. The 1<sup>st</sup> group contains general information about the plot: size of the plot in pixels, physical coordinates, properties of axes, name of the PNG file, title of the plot.

The next four groups contain information about different layers (from back to front). In this example, the GRD, BLN and DAT files are taken from DATA subfolder. Scale is taken from SCALES subfolder. The output picture is written to PICS subfolder.

The derived image is presented below:

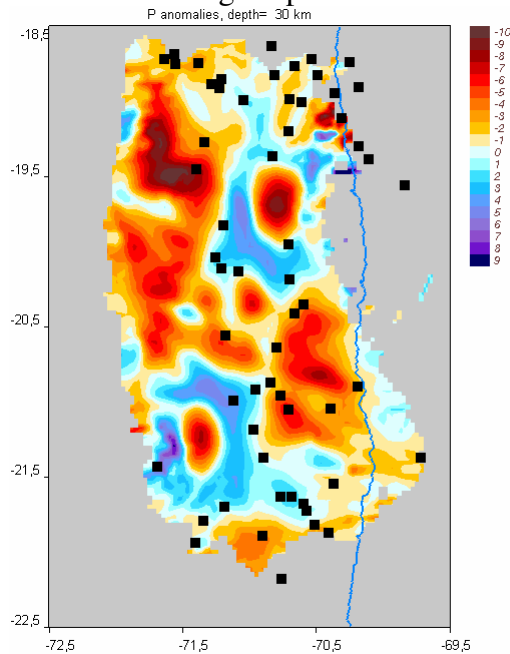


Figure 4.1. Resulting image (`\COMMON\visual_exe\dv15 3.png`) obtained as a result of running the file `\COMMON\visual_exe\visual.exe` using the configuration from `\COMMON\visual_exe\config.txt`.

Drawing 2D functions (key 1) requires using the color scales with indicated path (e.g. `scale.scl`). This file contains three columns which correspond to RGB coding. The first line can be ignored. For example:

`FIG_files/blue_red.scl`

```

-1 1
102 51 51
129 24 24
159 0 0
208 0 0
255 3 0
255 64 0
255 117 0
255 157 0
255 196 0

```

```

255 235 158
222 255 255
156 255 255
60 224 255
25 192 255
89 160 255
119 136 238
141 114 216
141 77 204
112 19 204
0 0 102

```

The example in `\COMMON\visual_exe` can be used for immediate control of the visualization tool.

We recommend using these tools just for previewing. Final files which are suitable for using in publications should be created in Surfer or other commercial visualization tools. The files with corresponding format are stored to `FIG_files` folders (overwritten in a case of running a new model).

## 4.2. Visualizing the ray paths and nodes

Project: `\PROGRAMS\3_VISUAL\_vis_ray_path\`

This program provides files:  
 with ray paths within fixed depth intervals  
 with ray paths projected onto fixed vertical sections  
 with nodes within fixed depth intervals  
 with nodes projected onto fixed vertical sections

Parameters for visualization in the **map view** are defined in the file `sethor.dat` (see example below) which is a common file for all models in one dataset.

`\DATA\DATASET1\sethor.dat`

```

4      Number of sections
10 20 30 40      Depths of sections
-1.2 1.5 0.03 -2.0 2.0 0.03 fi1, fi2, dfi, tet1, tet2, dtet
8      dsmin: distance from nearest node
0      smooth: Smoothing factor1

```

Parameters for visualization in the **vertical sections** are defined in the file `setver.dat` (see example below) which is a common file for all models in one dataset.

`\DATA\DATASET1\setver.dat`

```

2      Number of different sections
-70.82 -22.55 -71.14 -18.15   fiA, tetA, fiB, tetB: Ends of the profile
-71.84 -20.45 -70.11 -19.37   fiA, tetA, fiB, tetB: Ends of the profile
40      dis_ev: distance from section for visualization of events
2      dx_step: horizontal spacing
0 80 2   zmin, zmax, dz: depth interval and spacing
50      ds_mark: Marks for indication of position of section in the map view
8      distmin: Distance to the nearest node

```



```
0      smooth: Smoothing factor
1      ind_src: If 0, no sources are visualized
```

Two additional parameters for visualizing ray paths should be in the file:

`\PROGRAMS\3_VISUAL\_vis_ray_path\SET.DAT`

```
1 : rarefication of rays
1 : rarefication of points on the rays
```

The ray paths are taken from temporary files in the TMP folder. They correspond to the most recent model computed. Any new model calculations will overwrite these files.

The computed data are written to:

`/TMP_files/rays/rays_hor//ps//lev//.dat` (horizontal projection of rays)  
`/TMP_files/rays/rays_ver//ps//ver//.dat` (vertical projection of rays)  
`/TMP_files/rays/stat_ver.dat` (vertical projection of stations)  
`/TMP_files/rays/nodes_hor//gr//ps//lev//.dat` (vertical projection of stations)  
`/TMP_files/rays/nodes_ver//gr//ps//ver//.dat` (vertical projection of stations)

They can be shown by Surfer or any other graphic tool. An example of ray distribution in the map view and in a cross section is shown in [Figure 3.2](#).

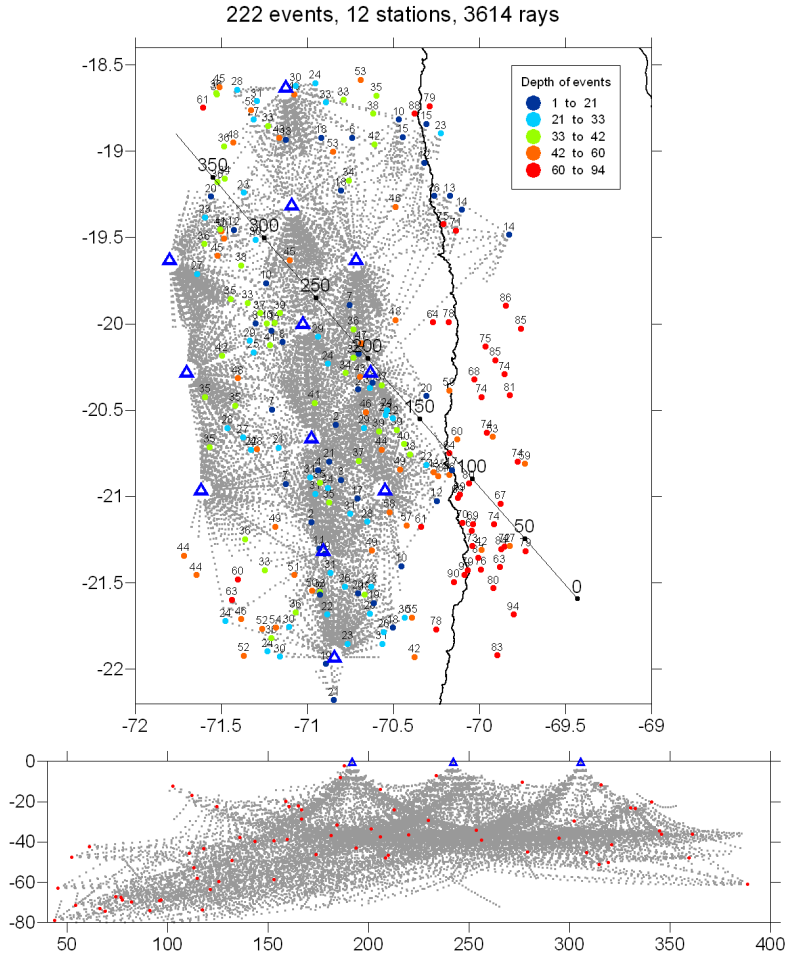


Figure 4.2. Ray paths in the map view and in a cross section shown by grey dots. Blue triangles are the stations. Red points are the relocated events. Numbers above the events indicate their depth.

The results of this step are visualized automatically using our previewing tool and stored in a folder `\PICS\ar\md\RAYS_GRIDS`.

Files with horizontal sections: `rays_hor\gr\ps\lev\`,

Files with vertical sections: `ver_rays_nodes\gr\ps\ver\`,

where:

**ar** is the name of the dataset

**md** is the name of the model

**gr** is the number of the grid

**ps** is 1 for P and 2 for S models

**lev** is the number of the depth section

**ver** is the number of the vertical section

Example of ray and node distributions in horizontal and vertical sections are shown below.

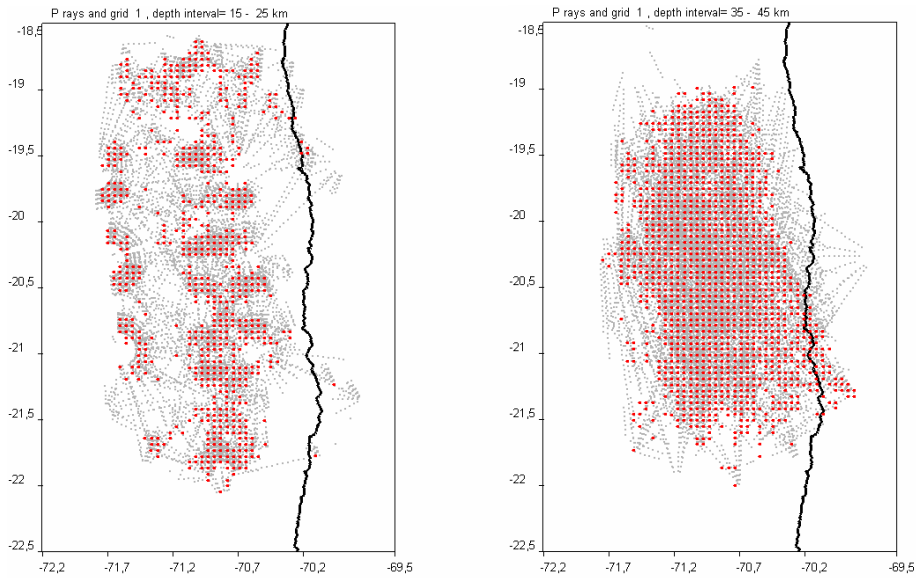


Figure 4.3. Ray paths and node distributions after 1<sup>st</sup> iteration corresponding to depth levels 2 and 4, and grid 1. Files:

[\PICS\DATASET1\MODEL\\_01\RAYS\\_GRIDS\rays\\_hor112](#)  
[\PICS\DATASET1\MODEL\\_01\RAYS\\_GRIDS\rays\\_hor114](#)

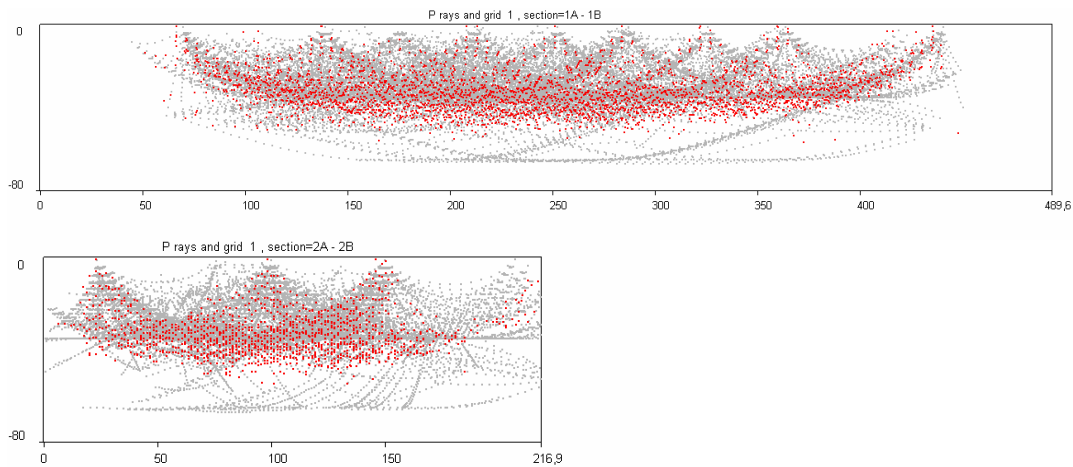


Figure 4.3. Ray paths and node distributions (grid 1) after 1<sup>st</sup> iteration in two vertical sections. Files:

[\PICS\DATASET1\MODEL\\_01\RAYS\\_GRIDS\ver\\_rays\\_nodes111](#)  
[\PICS\DATASET1\MODEL\\_01\RAYS\\_GRIDS\ver\\_rays\\_nodes112](#)

### 4.3. Horizontal sections of the resulting anomalies (NODES and CELLS).

The program which produces the GRD files of the resulting anomalies of P and S velocities based on node parameterization in horizontal sections is located in the Projects:

For NODES: / [PROGRAMS\3\\_VISUAL\vis\\_n\\_hor\\_result](#)

For CELLS: / [PROGRAMS\3\\_VISUAL\vis\\_b\\_hor\\_result](#)

The name of the model to be visualized is defined (automatically or manually, if necessary) in file [/model.dat](#)

```
DATASET1      name of the area
MODEL_01      name of the model
5             iteration
4             grids
1             Index for sources (1: the relocated sources are visualized)
```

Other parameters for visualization are defined in the file [sethor.dat](#) (see example below) which is a common file for all models in one dataset.

[\DATA\DATASET1\sethor.dat](#)

```
4           Number of sections
10 20 30 40  Depths of sections
-1.2 1.5 0.03 -2.0 2.0 0.03 fil, fi2, dfi, tet1, tet2, dtet
8           dsmin: distance from nearest node
0           smooth: Smoothing factor1
```

In this example, visualization is performed for the model **MODEL\_01** from the area **DATASET1** after **5** iterations. The results obtained separately in four different grids will be combined in one map. The results will be presented in **5** depth sections. The limits of the map are computed in respect to the center of the area defined in [AREA\\_CENTER](#) in [MAJOR\\_PARAM.DAT](#). In this case, the map longitudes are between -73° and -69°, and the latitudes are between 22.5° and 19.5°. Steps along longitude and latitude are 0.03°.

---

The resulting anomalies are only shown only in resolved areas where there are parameterization nodes. If the distance to the nearest node of one of considered grids is more than **dsmin** (8 km, in this example), the value of anomaly is not shown (dv=-999).

Visualization program allows smoothing of the visualized map which is controlled by the **smooth** parameter.

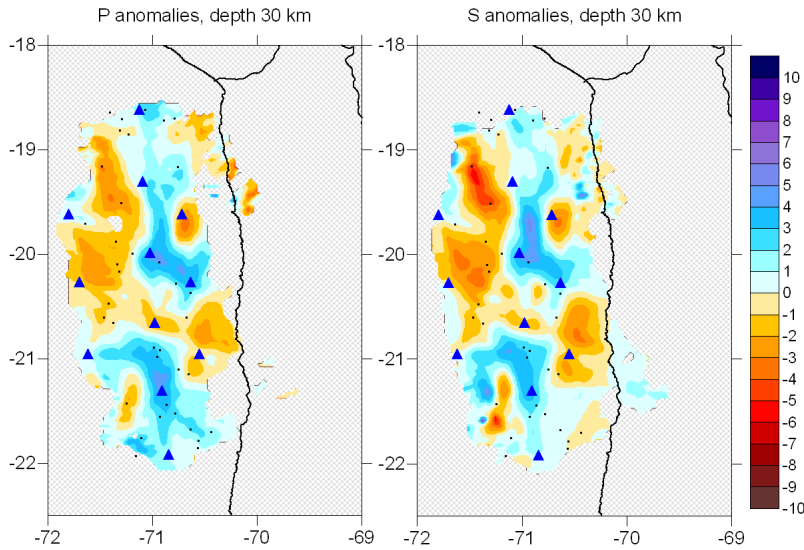


Figure 4.4. Resulting velocity anomalies at 30 km depth and relocated earthquakes in the depth interval of 25-35 km after 1 iteration. Velocity anomalies are given in percent in respect to the reference model. Black dots are the relocated sources; blue triangles are the stations.

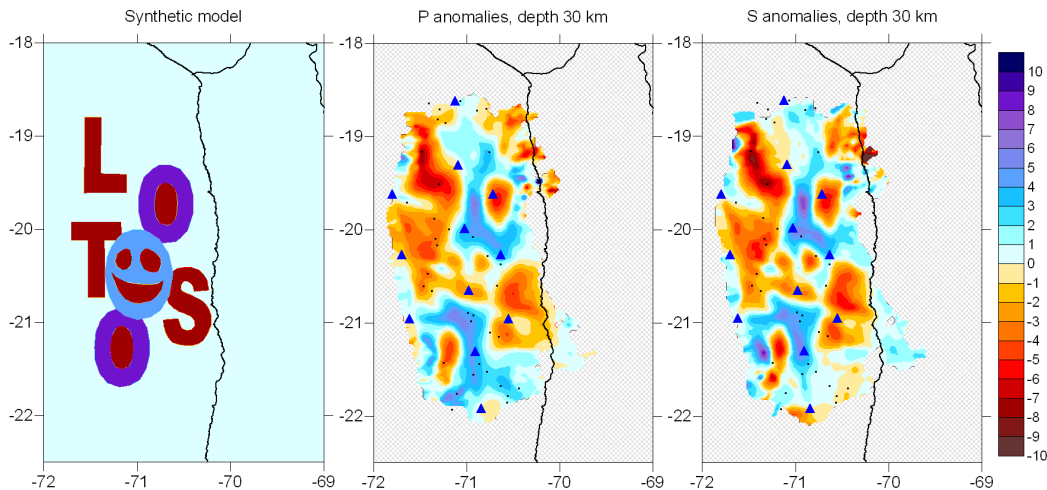


Figure 4.5. Central and right plots are the same as in Figure 3.1, but for the results after 3 iterations (best solution). Left map shows distribution of the initial data used for calculation of this data set (vertical unlimited prisms).

The output of this program:

```
\TMP_files\hor\dv//ps//it//lev//.grd :
```

relative anomalies in percent in horizontal sections. **it** is the number of iterations; **ps** = 1 for P and 2 for S model, **lev** is a number of depth level according to the information in “sethor.dat”. This file can be directly visualized in the Surfer Software as a contour line plot.

This program produces also the preview PNG files for P and S anomalies, for all depths

for all iterations as files

`PICS\\ar\\md\\IT\\it\\hor_dv\\ps\\lev\\.png`

Examples of the resulting PNG files for the **NODE** parameterization are shown in [Figure 4.6](#).

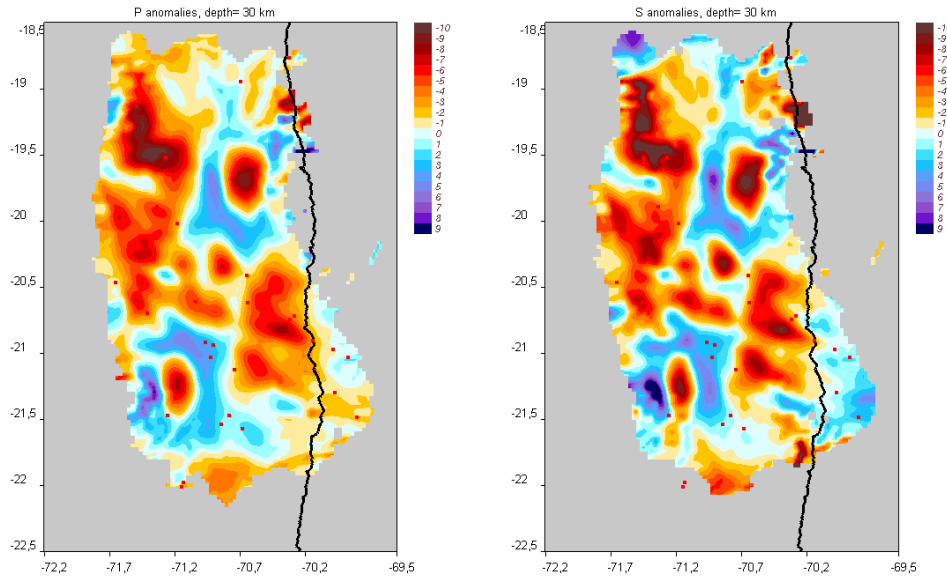


Figure 4.6. P and S velocity anomalies after 5<sup>th</sup> iteration at 30 km depth (3<sup>rd</sup> section). Files:

`\PICS\DATASET1\MODEL_01\IT5\hor_dv1 3`  
`\PICS\DATASET1\MODEL_01\IT5\hor_dv2 3`

Examples of the resulting PNG files for the **CELL** parameterization are shown in [Figure 4.7](#).

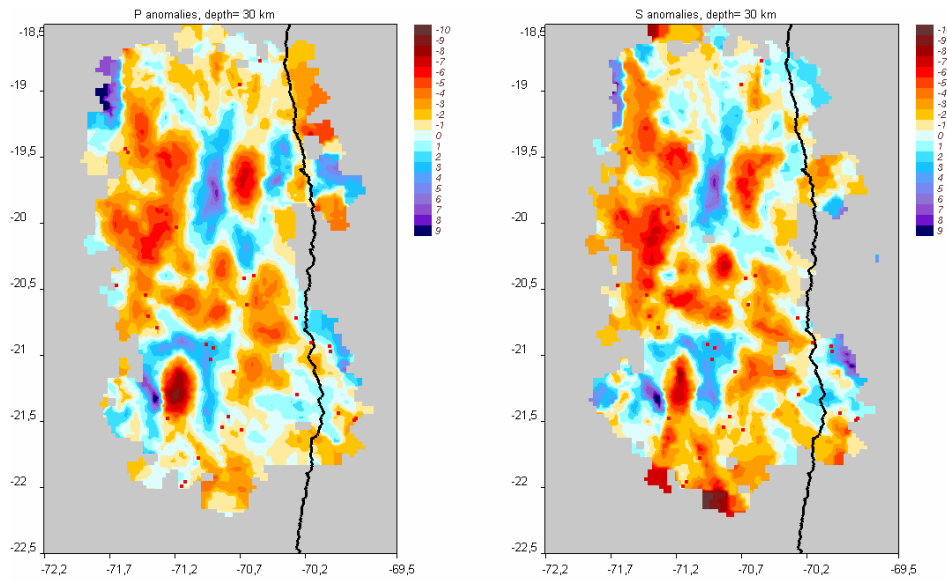


Figure 4.6. P and S velocity anomalies after 5<sup>th</sup> iteration at 30 km depth (3<sup>rd</sup> section). Files:

`\PICS\DATASET1\BLOCK_01\IT5\hor_dv1 3`  
`\PICS\DATASET1\BLOCK_01\IT5\hor_dv2 3`

## 4.4. Vertical sections of the resulting anomalies

The program which produces the GRD files of the resulting anomalies and absolute values of P and S velocities in vertical sections is in the Projects:

For NODES: / **PROGRAMS\3\_VISUAL\vis\_n\_ver\_result**

For CELLS: / **PROGRAMS\3\_VISUAL\vis\_b\_ver\_result**

The name of the model to be visualized is defined (automatically or manually, if necessary) in file **/model.dat**

```
DATASET1      name of the area
MODEL_01      name of the model
5             iteration
4             grids
1             Index for sources (1: the relocated sources are visualized)
```

Other parameters for visualization are defined in the file **.../ini\_param/setver.dat** (see example below).

**\DATA\DATASET1\setver.dat**

```
2             Number of different sections
-70.82 -22.55 -71.14 -18.15   fiA, tetA, fiB, tetB: Ends of the profile
-71.84 -20.45 -70.11 -19.37   fiA, tetA, fiB, tetB: Ends of the profile
40            dis_ev: distance from section for visualization of events
2             dx_step: horizontal spacing
0 80 2        zmin, zmax, dz: depth interval and spacing
50            ds_mark: Marks for indication of position of section in the map view
8             distmin: Distance to the nearest node
0             smooth: Smoothing factor
1             ind_src: If 0, no sources are visualized
```

In this example, visualization is performed for the model **MODEL\_01** from the area **DATASET1** after **5** iterations. The results obtained separately in four different grids will be combined in one map. The results will be presented in one cross section. The coordinates of ends of the sections are defined by **fiA, tetA, fiB, tetB**. Horizontal step along the profiles is defined by **dx\_step** (2 km, in this example). Vertical limits and step are defined by **zmin, zmax** and **dz**.

The resulting anomalies are only shown only in resolved areas where there are parameterization nodes. If the distance to the nearest node of one of considered grids is more than **dismin** (8 km, in this example), the value of anomaly is not shown (dv= -999).

Visualization program allows smoothing of the visualized map which is controlled by the **smooth** parameter.

If **ind\_src=1**, the sources located at distance less than **dis\_ev** (40 km, in this case) are projected to the cross-section.



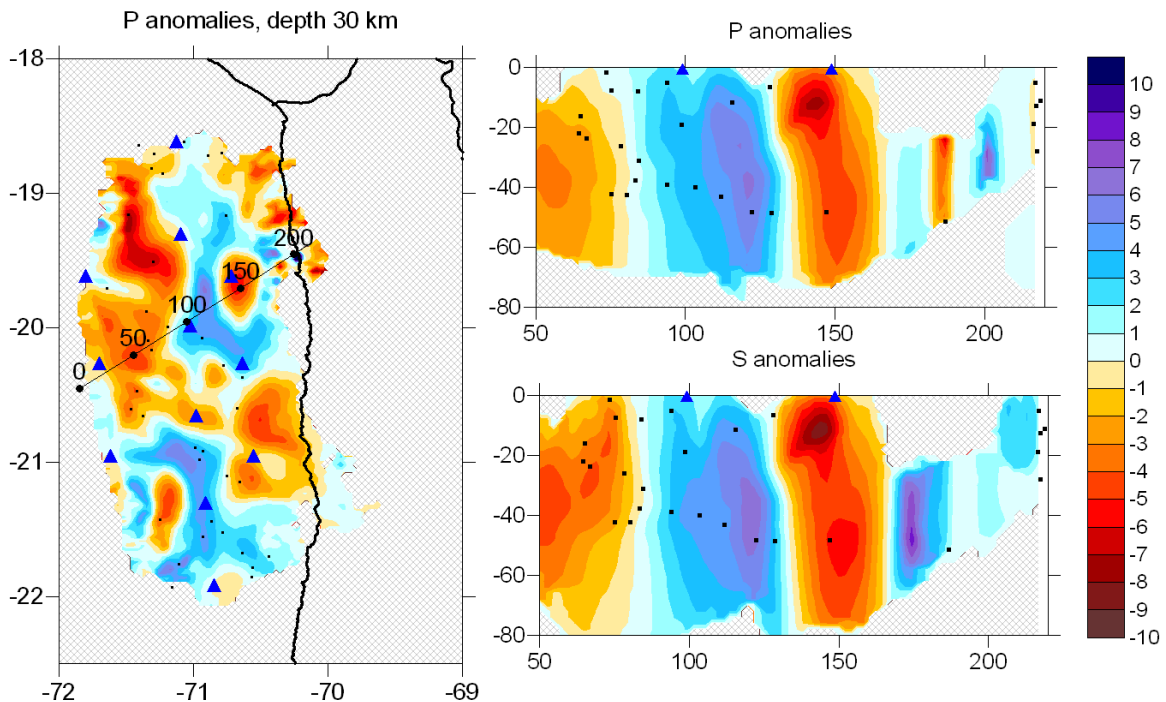


Figure 4.8. Resulting velocity anomalies and relocated earthquakes after 3 iterations in a cross section. Velocity anomalies are given in percent with respect to the reference model. Black dots are the relocated sources; blue triangles are the stations. Left map is given for showing the position of the vertical section.

### Output of this program:

1. `\TMP_files\vert\ver_//ps//it/'_//lev//.grd`: relative anomalies in percent in vertical sections. **ps** = 1 for P and 2 for S model, **lev** is a number of profile. It can be directly visualized in the Surfer Software as a contour line plot.
2. `\TMP_files\vert\abs_//ps//it/'_//lev//.grd`: absolute P and S velocities in vertical sections. **ps** = 1 for P and 2 for S model, **lev** is a number of profile. It can be directly visualized in the Surfer Software as a contour line plot.
3. `\TMP_files\vert/mark'//lev//.bln`: Position of the profile in the map view. **lev** is a number of profile. It can be directly visualized in the Surfer Software as a baseline plot. "`mark'//lev//.dat`" allows visualization of labels along the map view of the profile (e.g. 50, 100, 150...).
4. `\TMP_files\vert/ztr'//lev//.dat`: Projections of the relocated events onto the profile, **lev** is a number of profile. It can be directly visualized in the Surfer Software as a post plot.
5. `\TMP_files\vert/ztr'//lev//.dat`: Projections of the stations onto the profile, **lev** is a number of profile. It can be directly visualized in the Surfer Software as a post plot.

This program produces also the preview PNG files for P and S anomalies, for all sections for all iterations as files



PICS\\ar\\md\\IT\\it\\ver\_dv\\ps\\lev\\.png

Example of the resulting PNG files for the **NODE** parameterization is shown in [Figure 4.9](#).

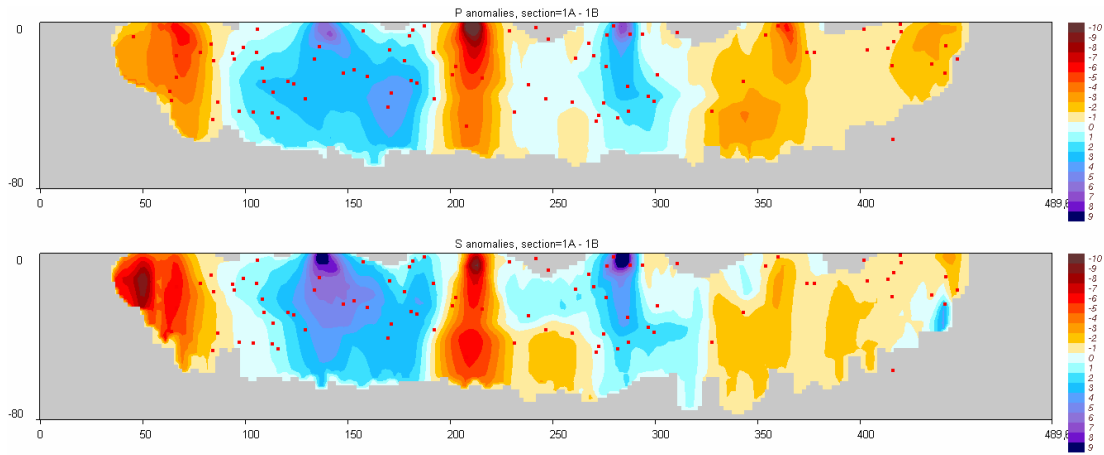


Figure 4.9. P and S velocity anomalies after 5<sup>th</sup> iteration in 1<sup>st</sup> profile. Files:

\\PICS\\DATASET1\\MODEL\_01\\IT5\\ver\_dv1 1  
\\PICS\\DATASET1\\MODEL\_01\\IT5\\ver\_dv2 1

## 5. Synthetic modeling

### 5.1. General remarks

The LOTOS software provides a wide range of possibilities for performing various synthetic tests. In the actual version there is possibility to define the initial model using four different ways:

1. checkerboard;
2. arbitrary anomalies defined in map view (horizontal anomalies);
3. arbitrary anomalies defined in some vertical sections (vertical anomalies);
4. checkerboard defined along vertical sections;

The travel times for the synthetic test are computed by tracing of rays between the sources and receivers corresponding to the real observation system. As in case of real data processing, bending algorithm is used. These times are the input for the entire inversion procedure.

Organization of folders for all kinds of synthetic models is generally the same as in case of the real data model. The Model folder name should consist of 8 characters and have the same structure of subfolders as the real data models. However, in the “*ini\_data*” folder, three additional files should be added:

1. “*anomaly*” which contains the description of the synthetic model. First line in this file is a **KEY** which sets the way of defining the model (**1**, **2**, **3**, or **4**). The following description depends on the case and will be described in details in the corresponding sections.
2. “*ref\_syn*” which contains the description of the reference synthetic model. The format is the same as in file “*ref\_start*”. These files may represent different models.

### 5.2. Definition of noise

There are two options for producing the noise. In case if in first line there is 2, the noise is defined according to remnant residuals after final inversion of real data which should be tested. An example:

`\DATA\GEO_DAT1\SMILE_01\MAJOR_PARAM`

```
*****
SYNTHETIC MODELING PARAMETERS :
MODEL_01      name of the model or "inidata ", source where S-R pairs are taken to
compute synthetic rays
1             iteration of the real data model; in case of "inidata " this line is ignored
1 type of noise 1: random generator, 2: from real data (2 cannot be for "inidata "!!!)

0.1 0.15      level of noise for P and S data. In case of real data, reduction of remnant
residuals, e.g. 0.8 0.8 (80%)
0             in percents, partition of outliers
10.          times larger than the normal noise
```

In this example, the rays after 1<sup>st</sup> iteration in model [MODEL\\_01](#) are used for defining S/R pairs to compute synthetic data. Here, the noise is produced by generator of random numbers.

In this example standard deviation of noise is 0.1 s for P data and 0.15 s for S data. In addition, the algorithm allows adding the outliers to test the robustness of the solution in case of some rough errors of phase picking. In the presented case, no outliers are added.

The basic shape of the noise distribution histogram is given in this file. In this case, the histogram is described by 1000 nodes and is distributed from -10 to 10 s. This histogram corresponds to the distribution of real residuals in the ISC catalogue. The file which determines the shape of the histogram is located in [COMMON](#).

### 5.3. Visualization of the initial synthetic model in horizontal and vertical sections

It is recommended to visualize the synthetic model with the batch file

[visual\\_syn\\_model.bat](#) which runs the Projects:

[\PROGRAMS\4\\_CREATE\\_SYN\\_DATA\a\\_set\\_syn\\_hor](#) for horizontal presentations

[\PROGRAMS\4\\_CREATE\\_SYN\\_DATA\a\\_set\\_syn\\_ver](#) for vertical presentations

The names of area and models are defined in both cases in file “[model.dat](#)”. For example:

File [/model.dat](#)

```
BOLIVAR_      name of the area
MODEL_01      name of the model
5             iteration
4             grids
1             Index for sources (1: the relocated sources are visualized)
```

The depths and limits of the horizontal sections for visualization are defined in [sethor.dat](#), the same as used for visualization of real data horizontal sections (Section 4.3). For example:

[\DATA\BOLIVAR\\_\sethor.dat](#)

```
5             Number of sections
10 20 30 40 50 Depths of sections
-2. 2. 0.03 -2.5 3.5 0.03 fil, fi2, dfi, tet1, tet2, dtet
8             dsmin: distance from nearest node
0             smooth: Smoothing factor1
```

The coordinates of the cross-section and vertical limits for visualization are defined in [setver.dat](#), the same as used for visualization of real data vertical sections (Section 4.4). For example:

[\DATA\BOLIVAR\\_\setver.dat](#)

```

1          Number of different sections
-84.97 8.73 -83.71 10.00 fiA, tetA, fiB, tetB: Ends of the profile
20          dis_ev: distance from section for visualization of events
5.          dx_step: horizontal spacing
0 130 5      zmin, zmax, dz: depth interval and spacing
50          ds_mark: Marks for indication of position of section in the map view
5          distmin: Distance to the nearest node
0          smooth: Smoothing factor
1          ind_src: If 0, no sources are visualized

```

The output of horizontal presentation is as the same as used for presenting real data results:

/TMP\_files/hor/dv'///ps//'\_///lev//'.grd :  
relative anomalies in percent in horizontal sections. **ps** = 1 for P and 2 for S model, **lev** is a number of depth level according to the information in “sethor.dat“. This file can be directly visualized in the Surfer Software as a contour line plot.

The output of vertical presentation is written to:

/TMP\_files/syn\_ini/ver\_'///ps//'\_///lev//'.grd: relative anomalies in percent in vertical sections. **ps** = 1 for P and 2 for S model, **lev** is a number of profile. It can be directly visualized in the Surfer Software as a contour line plot.

These files can be directly visualized in the Surfer Software as a contour line plot.

In a case of rather complicated definition of horizontal or vertical anomalies, it is recommended to execute programs in both Projects **a\_set\_syn\_ver** and **a\_set\_syn\_hor**.

## 5.4. Definition of the checkerboard anomalies (key 1)

In case of a regular checkerboard model, the key in the first line of **anomaly.dat** file should be **1**. The following lines contain description of the checkerboard. An example is presented below:

```

\DATA\DATASET1\BOARD_N1\ anomaly.dat
1          1 - board, 2 - horiz. anom, 3 - vert. anom, 4 - vert. board
7.00          anom_P: P-anomalies
-800. 800. 30. 10.0 xbrd1,xbrd2,dxanom,dxempty
-800. 800. 30. 10.0 ybrd1,ybrd2,dyanom,dyempty
-10. 100. 100. 0.0 zbrd1,zbrd2,dzanom,dzempty
7.00          anom_S: S-anomalies
-800. 800. 30. 10.0 xbrd1,xbrd2,dxanom,dxempty
-800. 800. 30. 10.0 ybrd1,ybrd2,dyanom,dyempty
-10. 100. 100. 0.0 zbrd1,zbrd2,dzanom,dzempty

```

In the presented example both P and S anomalies are +\_7% amplitude (**anom\_P**, **anom\_S**). The horizontal size of anomalies along X, Y and Z is defined by **dxanom** (30 km), **dyanom** (30 km), and **dzanom** (100 km). The empty space between

anomalies is defined by **dxempty** (10 km), **dyempty** (10 km), and **dzempty** (0 km). In this case, along Z direction, the anomaly is the same down to 100 km depth.

**Note:** It is important to define the anomaly not from a zero depth. In this case, if the anomaly is negative, the algorithm of ray tracing will try to bend the ray upward to achieve the minimum of travel time. In the presented example the anomalies are defined from -10 km depth.

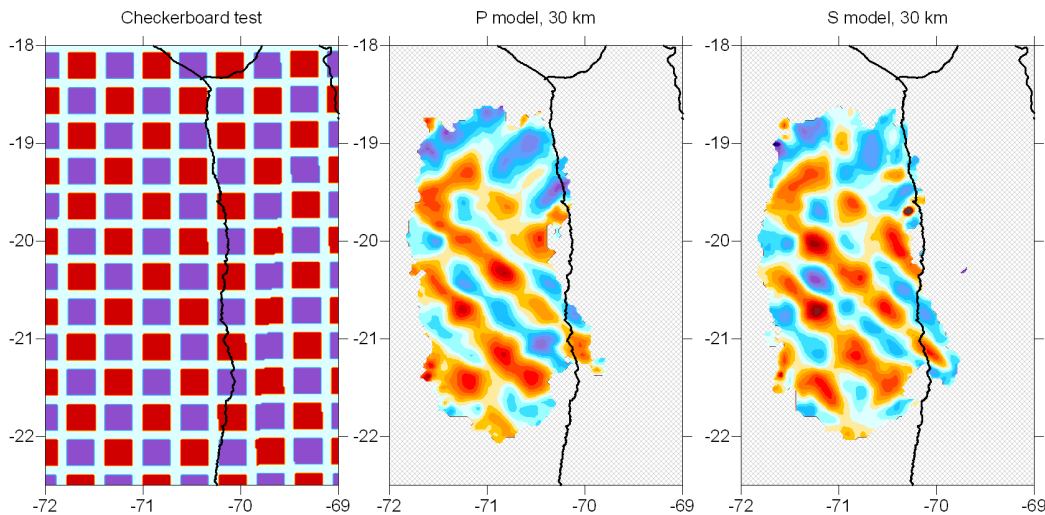


Figure 5.1. Checkerboard test. Left map represents the initial synthetic distribution. Central and right maps are the results of reconstruction at the depth of 30 km. The synthetic data are perturbed with noise of 0.15 and 0.3 s rms for P and S data, respectively.

A model corresponding to the presented example is shown in [Figure 5.1](#). In the same figure we present also the results of its reconstruction in P and S data.

This way allows also definition of horizontal blocks with unchanged values along X or Y. It is also possible to define single anomalies.

## 5.5. Definition of free horizontal anomalies (key 2)

In case of synthetic model definition using free horizontal anomalies, the indicator in the first line of `anomaly.dat` file should be **2**. In this case, a new subfolder `forms` should be created within the model folder. This folder contains the files with descriptions of the shapes which can be used for definition of the model. The names of these files should consist of any five characters and have the extension `.bln`. It can be simple forms (e.g. triangle, square, circle etc) or more complicated shapes. In practice, the synthetic anomalies can be created according to the shapes of patterns observed in the real data inversion. In this case, it is recommended to use a tool “**map/digitize**” in Surfer Software to produce the shapes in geographical coordinates which correspond to the really observed features. The curve which determines the anomaly should be not necessarily closed. Example of such a file generated in Surfer is presented below.

Example of

`\DATA\DATASET2\SMILE_01\forms\smile.blm`

```
20,1
-71.057525936, -21.128906016
-71.023617952, -21.154336496
-70.981234496, -21.171290488
-70.930373536, -21.184005728
-70.879512576, -21.18824448
-70.828651616, -21.192482216
-70.777790656, -21.192482216
-70.735406184, -21.18824448
-70.693022728, -21.171290488
-70.709975704, -21.222151448
-70.743882672, -21.25182068
-70.782028392, -21.273012408
-70.820174112, -21.27725116
-70.862558584, -21.285727648
-70.904943056, -21.281488896
-70.947326512, -21.268773656
-70.981234496, -21.247581928
-71.002426224, -21.2263902
-71.023617952, -21.196720968
-71.040571944, -21.167051736
```

Information about the synthetic anomaly is presented in the file “`anomaly.dat`”. An example is presented below:

`\DATA\DATASET2\SMILE_01\anom.dat`

```
2      1 - board, 2 - horiz. anom, 3 - vert. anom

4      number of anomalies
*****
circl  Figure
0. 0. 0. 0.
4 4
-50. 50.
*****
glaz1  Figure
0. 0. 0. 0.
-8 -8
-50. 50.
*****
glaz2  Figure
0. 0. 0. 0.
-8 -8
-50. 50.
*****
smile  Figure
0. 0. 0. 0.
-8 -8
-50. 50.
```

In this example the synthetic model consists of four horizontal prisms. All of them are located in the depth interval between -10 and 50 km. All of them should be defined in folder “`forms`” in files “`circl.blm`”, “`glaz1.blm`”, “`glaz2.blm`”, “`smile.blm`”. It is important to define the anomaly not from a zero depth. In this case, if the anomaly is negative, the algorithm of ray tracing will try to bend the ray upward to achieve the minimum of travel time. In the presented example the anomalies are defined from -10 km depth.

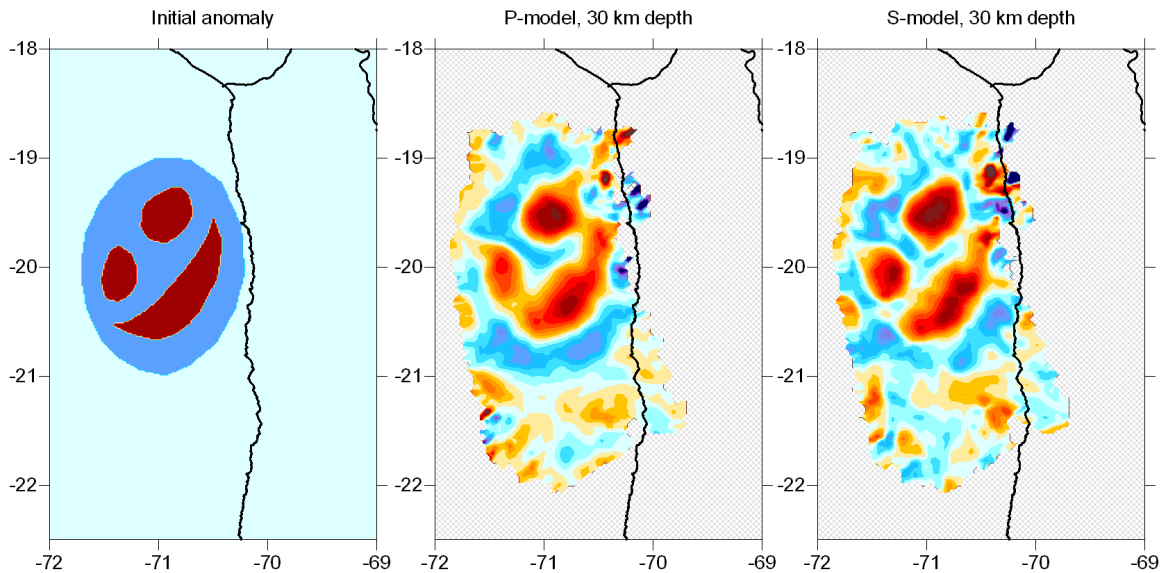


Figure 5.2. Synthetic test with free horizontal anomalies. Left map represents the initial synthetic distribution. Central and right maps are the results of reconstruction at the depth of 30 km. The synthetic data are perturbed with noise of 0.15 and 0.3 s rms for P and S data, respectively.

In addition, the entire image can be scaled and rotated. To do this, use the file:

`\DATA\DATASET2\SMILE_01\forms\scaling.dat`

```
-71.7 -70.2
-21.0 -19.0
45
```

In this example is defined in the interval of longitude of -71.7 - -70.2 and latitude -21.0 - -19.0 and is rotated to the angle of 45°

If this file is not defined, the model would be created according to coordinates defined in forms.

A model corresponding to the presented example is shown in Figure 5.2. In the same figure we present also the results of its reconstruction in P and S data.

## 5.6. Definition of free vertical anomalies (key 3)

There is a possibility to define the synthetic model as superposition of prisms which have fixed shape in some vertical sections. In this case, the indicator in the first line of “`anomaly.dat`” file should be **3**. A new subfolder “`forms`” should be created within the model folder. This folder contains the files with descriptions of the shapes which can be used for definition of the model. The names of these files should consist of any five characters and have the extension `.bln`. It can be simple forms (e.g. triangle, square, circle etc) or more complicated shapes. In practice, the synthetic anomalies can be created according to the shapes of patterns observed in the vertical sections of the real data inversion results. In this case, it is recommended to use a tool “**map/digitize**” in Surfer Software to produce the shapes in geographical coordinates which correspond to the really observed features. The curve which determines the anomaly should be not

necessarily closed. Example of such a file generated in Surfer is presented below.

Example of

`\DATA\DATASET2\VER_free\forms\smile.blm`

```
20,1
-71.057525936, -21.128906016
-71.023617952, -21.154336496
-70.981234496, -21.171290488
-70.930373536, -21.184005728
-70.879512576, -21.18824448
-70.828651616, -21.192482216
-70.777790656, -21.192482216
-70.735406184, -21.18824448
-70.693022728, -21.171290488
-70.709975704, -21.222151448
-70.743882672, -21.25182068
-70.782028392, -21.273012408
-70.820174112, -21.27725116
-70.862558584, -21.285727648
-70.904943056, -21.281488896
-70.947326512, -21.268773656
-70.981234496, -21.247581928
-71.002426224, -21.2263902
-71.023617952, -21.196720968
-71.040571944, -21.167051736
```

Information about the synthetic anomaly is presented in the file “`anomaly.dat`“. An example is presented below:

`\DATA\DATASET2\VER_free\anom.dat`

```
3      1 - board, 2 - horiz. anom, 3 - vert. anom

7      number of anomalies
*****
-70.82538797 -22.55326057 -71.14119379 -18.15829857
L1_____ Figure
0. 0. 0. 0.
-8 -8
-150. 150.
*****
-70.82538797 -22.55326057 -71.14119379 -18.15829857
Olout Figure
0. 0. 0. 0.
8 8
-150. 150.
*****
-70.82538797 -22.55326057 -71.14119379 -18.15829857
Ol_in Figure
0. 0. 0. 0.
-8 -8
-150. 150.
*****
-70.82538797 -22.55326057 -71.14119379 -18.15829857
Tl_____ Figure
0. 0. 0. 0.
-8 -8
-150. 150.
*****
-70.82538797 -22.55326057 -71.14119379 -18.15829857
O2out Figure
0. 0. 0. 0.
8 8
-150. 150.
*****
-70.82538797 -22.55326057 -71.14119379 -18.15829857
O2_in Figure
```



```

0. 0. 0. 0.
-8 -8
-150. 150.
*****
-70.82538797 -22.55326057 -71.14119379 -18.15829857
S1_____ Figure
0. 0. 0. 0.
-8 -8
-150. 150.

```

In this example the synthetic model consists of seven prisms defined in one cross section. In fact, any amount of prisms can be defined in any different cross sections at the same time. All of them should be defined in folder “**forms**” in files with corresponding names

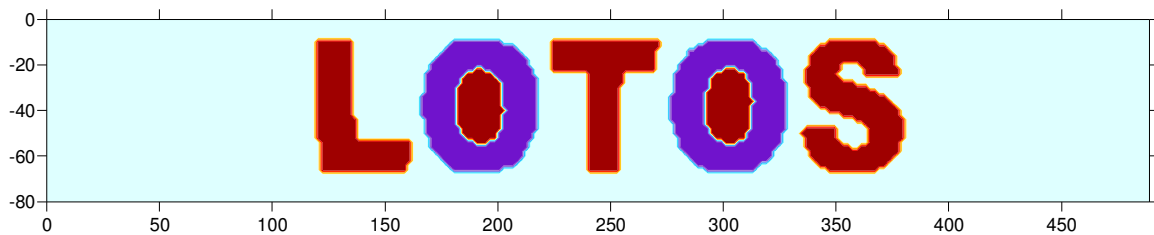


Figure 5.3. Synthetic model defined by free vertical patterns.

In addition, the entire image can be scaled and rotated. To do this, use the file:

[\DATA\DATASET2\VER\\_free\forms\scaling.dat](#)

```

120 380
10 70
0

```

In this example is defined in the interval of X of -71.7 - -70.2 and Z of -21.0 - -19.0 without rotation

If this file is not defined, the model would be created according to coordinates defined in forms.

A model corresponding to the presented example is shown in [Figure 5.3](#).

## 5.7. Definition of vertical checkerboard anomalies (key 4)

For investigating vertical resolution we can defines checkerboard model at vertical sections. In this case, the indicator in the first line of “[anomaly.dat](#)” file should be **4**. The following lines contain description of the checkerboard. An example is presented below:

[\DATA\DATASET2\VBRDmod2\anomaly.dat](#)

```

4      1 - board, 2 - horiz. anom, 3 - vert. anom
-----
1      number of profiles
-----
-70.82538797 -22.55326057 -71.14119379 -18.15829857
-90 90      y1, y2
7.00      P-anomalies
-800. 800. 50. 0.0
-5. 240. 30. 0.0
7.00      S-anomalies
-800. 800. 50. 0.0
-5. 240. 30. 0.0

```

In the presented example both P and S anomalies are  $\pm 7\%$  amplitude. The size of anomalies is 50 km along X and 30 km along Z. There is a possibility to define empty space between anomalies. These anomalies are defined across the section with the ends defined in:

-70.82538797 -22.55326057 -71.14119379 -18.15829857

Thickness of blocks across the profile is from -90 km to 90 km.

A model corresponding to the presented example is shown in Figure 5.4.

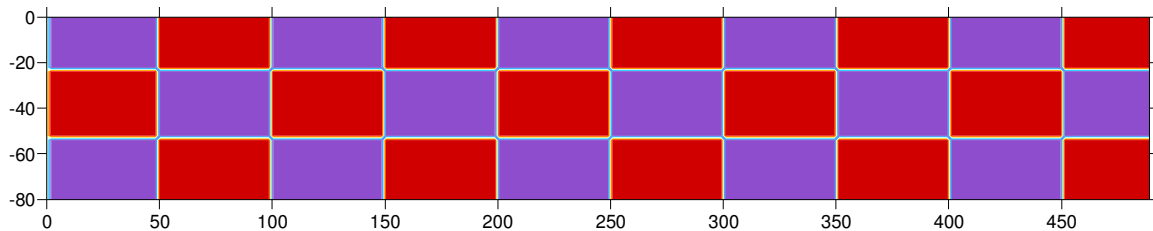


Figure 5.4. Synthetic model defined by vertical checkerboard patterns.

## Conclusion remarks:

I wish you all successful using the LOTOS code and obtaining bright results. We would appreciate any help and suggestions on improving the code. In case of any inconsistencies and errors, please address to the author, Ivan Koulakov ([KoulakovIY@ipgg.nsc.ru](mailto:KoulakovIY@ipgg.nsc.ru)). We are planning to prepare new versions of LOTOS with more friendly interface. We have also LOTOS-based versions for anisotropic tomography and for attenuation which are also presumed to be placed in open access in the nearest future.

## References:

- Koulakov I., 2009 a, LOTOS code for local earthquake tomographic inversion. Benchmarks for testing tomographic algorithms, *Bulletin of the Seismological Society of America*, Vol. 99, No. 1, pp. 194-214, doi: 10.1785/0120080013
- Koulakov I., 2009 b. Out-of-network events can be of great importance for improving results of local earthquake tomography, *Bulletin of the Seismological Society of America*, Vol. 99, No. 4, pp. 2556-2563, doi: 10.1785/0120080365
- Koulakov I. and S.V. Sobolev, (2006) A Tomographic Image of Indian Lithosphere Break-off beneath the Pamir Hindukush Region, *Geophys. Journ. Int.*, 164, p. 425-440.
- Koulakov, I., S.V. Sobolev, and G. Asch (2006). P- and S-velocity images of the lithosphere-asthenosphere system in the Central Andes from local-source tomographic inversion, *Geophys. Journ. Int.*, 167, 106-126.
- Nolet, G. (1981). Linearized inversion of (teleseismic) data. In R. Cassinis (ed.), editor, *The solution of the inverse problem in geophysical interpretation*, pages 9-37. Plenum Press.
- Paige, C.C., and M.A. Saunders (1982). LSQR: An algorithm for sparse linear equations and sparse least squares, *ACM trans. Math. Soft.*, 8, 43-71.

- Um, J., and C.H. Thurber (1987). A fast algorithm for two-point seismic ray tracing, *Bull. Seism. Soc. Am.*, 77, 972-986,
- Van der Sluis, A., and H.A. van der Vorst (1987). Numerical solution of large, sparse linear algebraic systems arising from tomographic problems, *in: Seismic tomography, edited by G.Nolet, pp. 49-83, Reidel, Dordrecht,*