Code *ATOM-3D* for 3D tomographic inversion based on active refraction seismic data

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

Novosibirsk, Russia,

Potsdam, Germany

August 2009

Table of contents:

A tomographic algorithm, *ATOM-3D* (Active TOMography in 3D) is designed for investigating 3D velocity structure based on travel times of first arrival refracted seismic rays from active sources. Both marine and land observation schemes can be considered. The calculations are performed in Cartesian coordinates (XYZ). The *ATOM-3D* code 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. *ATOM-3D* code is freely available online at www.ivan-art.com/science/ATOM_3D. Any help with installation and running the code can be obtained from the author, Ivan Koulakov (KoulakovIY@ipgg.nsc.ru).

Note!!! All the results presented in the manual are derived from processing of synthetic datasets which were simulated based on a realistic experiment configuration in Canaries.

1. General structure of files and folders in *ATOM-3D*

1.1. List of folders in the root directory

The recommended file structure in the root directory with short descriptions is presented in the Figure 1.1.

PROG	- folder with all programs for iterative, nonlinear tomographic inversion of first arrival times
DATA	- folder which contains all the data and models
PICS	- folder which contains the bitmap PNG pictures for previewing (initially empty)
FIG_files	- folder for files generated at the step of visualization, which can subsequently be viewed in Surfer or similar visualization programs
VISUAL	- folder with programs for visualization of intermediate and final results
subr	- folder which contains all the subroutines. It is necessary only if re-compiling of the programs will be performed
tmp	- folder for temporary files. Initially empty
CREATE_PICS	- folder with visualization program used for previewing the results
model_all.dat	- file which defines areas and models to be processed for real data inversion (defined by user)
model.dat	- file with current information about synthetic model (updated automatically)
preview_key.dat	- If this file contains any nonzero number, the results are previewed as PNG files
START_REAL.BAT	-BATCH file for execution of real data inversion
START_SYN.BAT	-BATCH file for execution of synthetic modeling

Figure 1.1 Folders (pink boxes) and files (white boxes) in the root directory of *ATOM-3D*.

1.2. Structure of the DATA folder

The general structure of the DATA folder is shown in Figure 1.2. The DATA folder has a two-step hierarchy structure. The DATA contains the Area folders (e. g. "CANARES_", "DATASET1", "DATASET2" etc). The name of the AREA folder should consist of any 8 characters.

Each "AREA" folder contains a mandatory subfolder "inidata" with initial data and several folders for observed data inversion and synthetic modeling (e.g. "MODEL_01" or "BOARD_01").

In addition, the "AREA" folder contains configuration files which contains parameters for visualization of the results.

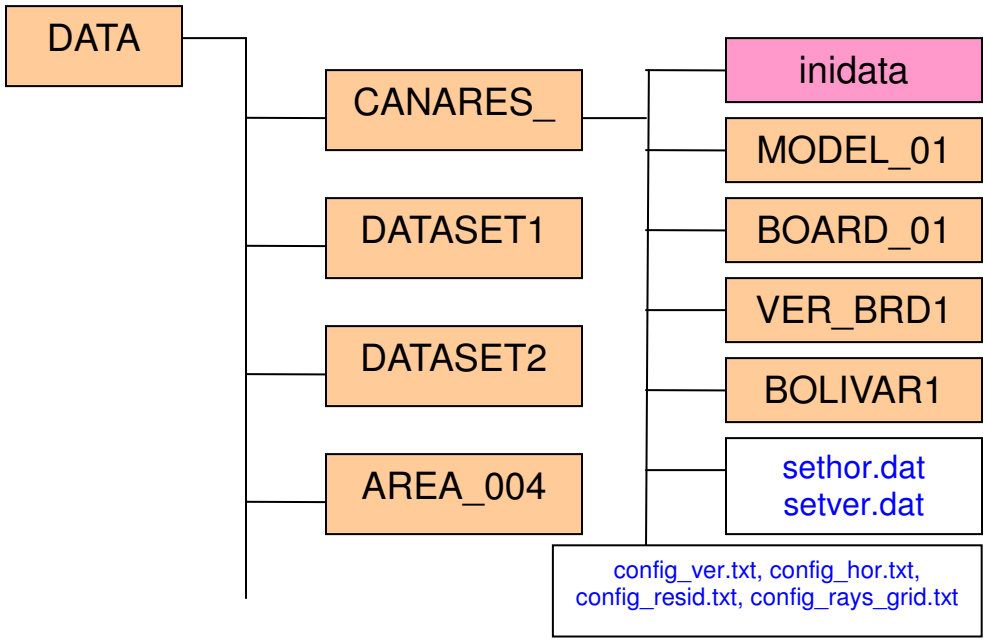


Figure 1.2. Structure of folders (orange and pink boxes) and files (white boxes) in the DATA directory.

1.3. Organization of the input data in the "inidata" folder

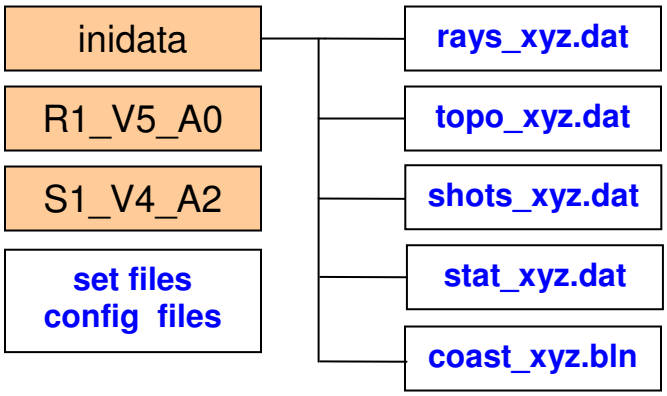


Figure 1.3. Structure of files (white boxes) and folders (orange boxes) in the "inidata" folder.

The input data are contained in the "inidata" folder, as shown in Figure 1.3 and include only one mandatory file, "rays_xyz.dat", list of all travel times

Each line of "rays_xyz.dat" contains 7 numbers which correspond to one ray:

- X-Y coordinates (in km) and depth of the sea below the source (km, positive – below sea level), or elevations of land shots (in the case of negative value).
- X-Y coordinates (in km) and depth (km, positive – below sea level) of receivers
- observed travel time (in seconds)

Format of this file is free (length of numbers is not fixed)

/DATA/DATASET1/inidata/rays_xyz.dat

```
  52.25700   81.33400    1.61664   33.18100   52.97800   -0.68600    8.87097
  52.23500   80.42700    1.50182   33.18100   52.97800   -0.68600    8.69111
  52.22400   79.66600    1.40506   33.18100   52.97800   -0.68600    8.53539
  52.21400   78.90500    1.37118   33.18100   52.97800   -0.68600    8.37678
  52.20900   77.82300    1.27471   33.18100   52.97800   -0.68600    8.12255
  52.18900   77.04400    1.22336   33.18100   52.97800   -0.68600    7.96412
  52.17800   76.25200    1.05892   33.18100   52.97800   -0.68600    7.73564
  52.16300   75.14700    0.81705   33.18100   52.97800   -0.68600    7.38806
  52.15600   73.86000    0.60956   33.18100   52.97800   -0.68600    7.07576
  41.61100   67.98300    0.26453   33.18100   52.97800   -0.68600    4.16952
  39.35500   67.87200    0.23819   33.18100   52.97800   -0.68600    3.81352
  37.77600   67.79600    0.19389   33.18100   52.97800   -0.68600    3.67196
  35.18700   67.67100    0.02780   33.18100   52.97800   -0.68600    3.36276
  52.16500   75.24300    0.91478   33.18100   52.97800   -0.68600    7.46241
  52.17700   76.13900    1.07950   33.18100   52.97800   -0.68600    7.72659
  52.18700   76.88200    1.17064   33.18100   52.97800   -0.68600    7.89918
  52.20700   77.64100    1.28924   33.18100   52.97800   -0.68600    8.09983
  52.20700   78.39400    1.32312   33.18100   52.97800   -0.68600    8.25485
  52.21700   79.14100    1.40037   33.18100   52.97800   -0.68600    8.43073
  24.37000   67.16000    0.24629   34.53000   52.74400   -0.96700    3.99896
  22.86900   67.08600    0.19587   34.53000   52.74400   -0.96700    4.13491
  21.36800   67.02100    0.46264   34.53000   52.74400   -0.96700    4.48088
  19.85800   66.95300    0.59317   34.53000   52.74400   -0.96700    4.76081
  20.34700   62.70100    0.14475   34.53000   52.74400   -0.96700    3.85565
  20.88300   61.28000    0.14008   34.53000   52.74400   -0.96700    3.59609
  21.76500   59.96800    0.12911   34.53000   52.74400   -0.96700    3.29563
  22.70700   58.81900    0.15493   34.53000   52.74400   -0.96700    3.02408
  23.65100   57.68900    0.15042   34.53000   52.74400   -0.96700    2.75250
  82.12600   51.86300    0.50376   34.53000   52.74400   -0.96700    8.89241
```

All other files are only necessary for visualization. The calculations may work without these files, but they are strongly recommended. Without then, the resulting images will not contain useful information (coast, stations and shots).

"topo_xyz.grd": XYZ representation of the relief which may include seafloor bathymetry. This file is the standard grid file used for presenting contours in SURFER software. Actually, this file is used only for presenting the results and is not necessarily required. During calculations this file is not used. The format can be described with following program fragment:

```
open(1,file='../../data/'//ar//'/inidata/topo_xyz.grd',status='old',err=234)
read(1,*)
read(1,*)nxmap,nymap
read(1,*)xmap1,xmap2
read(1,*)ymap1,ymap2
read(1,*)zmin,zmax
do iy=1,nymap
       read(1,*)(topo(ix,iy),ix=1,nxmap)
end do
close(1)
```

Topography is given in km in respect to the sea level (negative values are below sea

level). Below is an example of a starting part of the topography file:

/DATA/DATASET1/inidata/topo_xyz.grd

```
DSAA
         401          368
  0.0000000E+00   120.0000
  0.0000000E+00   110.0000
  -8.625999       3.499000
  -3.620000      -3.496000      -3.382000      -3.281000      -3.197000
  -3.131000      -3.082000      -3.049000      -3.028000      -3.016000
  -3.011000      -3.009000      -3.010000      -3.011000      -3.012000
  -3.013000      -3.014000      -3.017000      -3.021000      -3.026000
  -3.032000      -3.037000      -3.039000      -3.037000      -3.031000
  -3.020000      -3.006000      -2.990000      -2.976000      -2.964000
  -2.958000      -2.959000      -2.969000      -2.987000      -3.012000
  -3.044000      -3.078000      -3.114000      -3.147000      -3.176000
  -3.199000      -3.215000      -3.225000      -3.230000      -3.229000
  -3.225000      -3.220000      -3.216000      -3.212000      -3.210000
  -3.209000      -3.210000      -3.211000      -3.212000      -3.213000
  -3.214000      -3.215000      -3.215000      -3.215000      -3.214000
  -3.214000      -3.213000      -3.212000      -3.211000      -3.209000
  -3.209000      -3.210000      -3.212000      -3.216000      -3.221000
  -3.226000      -3.233000      -3.239000      -3.246000      -3.253000
  -3.259000      -3.264000      -3.267000      -3.267000      -3.263000
- - - - - - -
- - - - - - -
```

"stat_xyz.dat" contains the XYZ coordinates of the receivers (in km; for Z, negative value means location above the sea level)

/DATA/DATASET1/inidata/stat_xyz.dat

```
   33.18100       52.97800      -0.6860001
   34.53000       52.74400      -0.9670000
   36.16400       52.67000       -1.178000
   41.61100       56.16600       -1.574000
   35.80600       62.67000      -0.4850001
   38.65500       58.94600       -1.128000
   40.85500       59.46200       -1.119000
   42.06400       41.49200       -1.105000
   51.93300       57.86100       -1.667000
   56.33300       60.35900       -1.260000
   53.69000       57.53300       -1.944000
   63.99400       60.80900       -1.439000
   67.00700       60.93400       -1.803000
   64.61400       52.36100       -1.773000
   65.25400       51.26000       -1.469000
   65.11500       46.04600      -0.8710001
   52.37100       41.26500       -1.663000
- - - - - - -
- - - - - - -
```

"shots_xyz.dat" contains the XYZ coordinates of shots. (in km; for Z value means depth of the sea bottom below sea level; values are negative)

/DATA/DATASET1/inidata/shots_xyz.dat

```
   33.82500       37.52400      -0.2061720
   33.95800       37.23100      -0.1766308
   34.09200       36.93500      -0.2125775
   34.22500       36.63800      -0.1609435
   52.25700       81.33400       -1.616636
   52.24300       81.03300       -1.573622
   52.24100       80.88400       -1.514393
   52.23900       80.73200       -1.533813
   52.23700       80.57900       -1.484645
   52.23500       80.42700       -1.501820
   52.23300       80.27700       -1.456270
   52.24000       80.12600       -1.472121
   52.23800       79.97300       -1.431334
   52.22700       79.81900       -1.445354
   52.22400       79.66600       -1.405055
```

```
   52.23200       79.51400      -1.419026
   52.23000       79.36200      -1.382288
   52.22800       79.20900      -1.394996
   52.22600       79.05600      -1.359312
   52.21400       78.90500      -1.371176
   52.21200       78.75200      -1.337068
- - - - - - -
- - - - - - -
```

"`coast_xyz.bln`" contains the coastal line in Cartesian coordinates in BLN format used in SURFER for drawing polygons.

`/DATA/DATASET1/inidata/coast_xyz.bln`

```
4961
  33.813   64.850
  33.758   64.945
  33.666   64.947
  33.621   64.915
  33.537   64.926
  33.511   64.962
  33.528   65.057
  33.505   65.083
  33.270   65.225
  33.217   65.208
  33.177   65.229
  33.151   65.274
  33.139   65.333
  33.144   65.406
  33.103   65.403
  33.090   65.422
- - - - - - -
- - - - - - -
```

1.4. Organization of the MODEL folder.

The MODEL folder is created either for real data or synthetic tomographic models. The name of the MODEL folder should contain 8 characters (e.g. "***BOLIVAR1***", "***MODEL_01***", "***BOARD_01***"). The structure of the MODEL folder for performing the inversion of the real data with brief description of the main files and folders is shown in Figure 1.4.

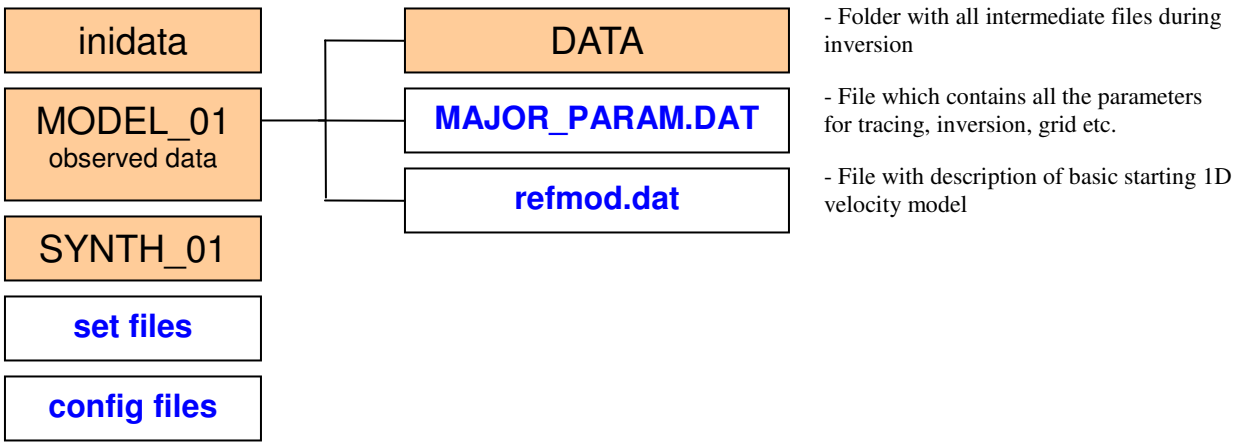


Figure 1.4. Structure of files and folders in the folder corresponding to observed data inversion

In the case of synthetic modeling (Figure 1.5), the structure of files and folders remains the same, except for one folder "`forms`" and two additional files (`anomaly.dat` and `refsyn.dat`) which determine the synthetic velocity model.

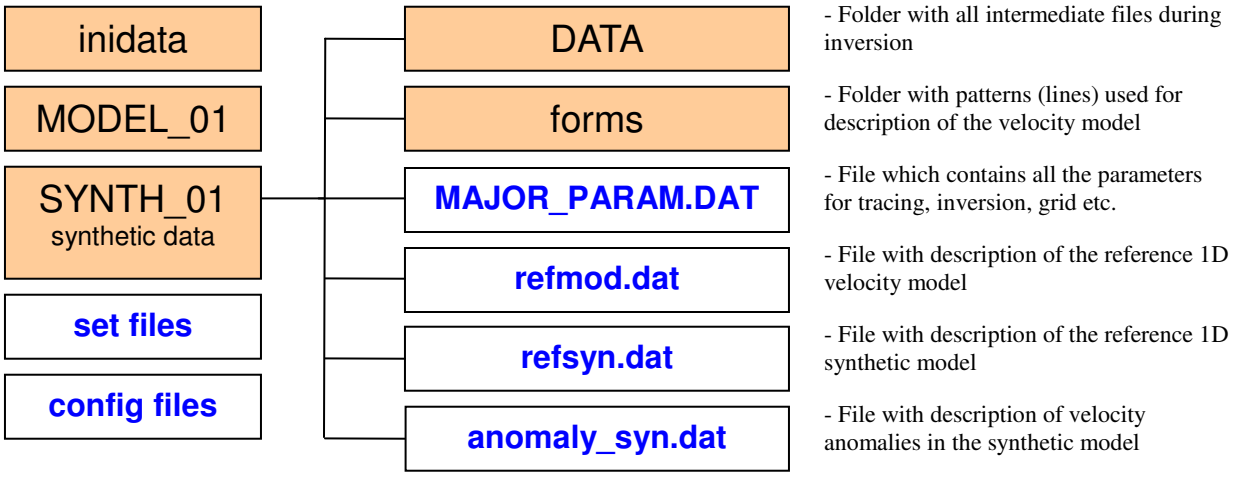


Figure 1.5. Structure of files and folders in the folder corresponding to synthetic modeling.

1.5. Description of some of the main initial parameters.

Most of the parameters for ray tracing, parameterization and inversion are defined in file '`MAJOR_PARAM.DAT`'. The content of this file is organized by rubrics. Each rubric starts with a key line. For example:

```
TRACING PARAMETERS:
GRID_PARAMETERS:
INVERSION PARAMETERS :
3D_MODEL PARAMETERS:
```
etc.

Example of the "`MAJOR_PARAM.DAT`" file is given below (names of rubrics are indicated with red):

`/DATA/DATASET1/MODEL_01/MAJOR_PARAM.DAT`

```
********************************************************
Parameters for tracing in 3D model using bending tracing
********************************************************
TRACING_PARAMETERS:
! Parameters for BENDING:
0.5         ds_ini: basic step along the rays
3           min step for bending
0.01        min value of bending
5           max value for bending in 1 step
2           k_reduce: frequency of data to be selected
100         nfreq_print: frequency of printing on console

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

********************************************************
INVERSION PARAMETERS :
40          LSQR iterations
0.4         level of smoothing
0.3         regularization level
2.00        minimal velocity
```

```
*********************************************************
Parameters for 3D model with regular grid
*********************************************************
3D_MODEL PARAMETERS:
10. 100. 0.7  xx1, xx2, dxx,
10. 100. 0.7  yy1, yy2, dyy,
-5. 30. 0.7 zz1, zz2, dzz
3           distance from nearest node
0           Smoothing factor1

*********************************************************
Parameters for grid construction
*********************************************************
GRID_PARAMETERS:
-120. 120. 1.0     grid for ray density calculation (X)
-120. 120. 1.0     grid for ray density calculation (Y)
-5. 30. 0.5        min and max levels for grid
1                  ! Grid type: 1: nodes, 2: blocks
0.5                !min distance between nodes in vert. direction
0.05 100.0 !plotmin, plotmax= maximal ray density, relative to average
-5.                !zupper:  Uppermost level for the nodes

0.05                       !dx= step of movement along x
0.05                       !dz= step of movement along z
```

Following the key line in red, a description of parameters for the current group with a fixed format is given. The order of groups and number of empty lines between groups are free.

The meaning of parameters will be explained in the description of the main steps.

Starting velocity is 1D distribution which is defined in "refmod.dat".

/DATA/DATASET1/MODEL_01/refmod.dat

```
-3   4.0
15.  6.6
30.  8.063499
```

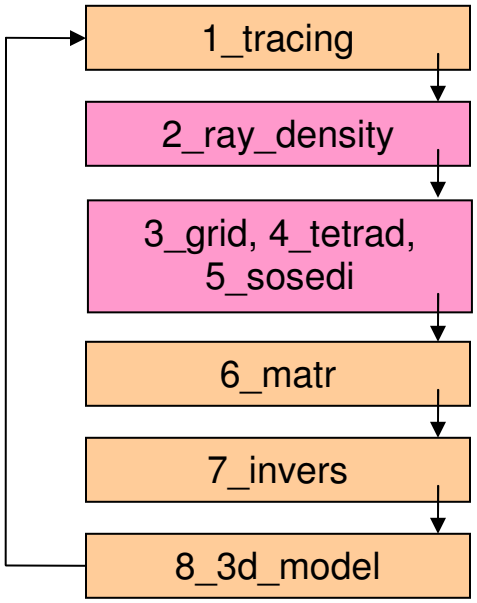
This file contains the information about starting reference velocity model. First column is depth; and the second is velocity. Velocities are linearly interpolated between the layers. Above the upper level the velocity remains constant. Any number of layers is allowed.

The other parameters will be presented in description of the *ATOM-3D* algorithm, next section.

2. Description of the ATOM-3D algorithm

2.1. Iterative tomographic inversion

Iterative inversion consists of consequent execution of the following programs:



- Program for ray tracing in a current velocity model

- Program for computing the ray density (only in 1st iteration)

- Programs for constructing the parameterization grid (only in 1st iteration)

- Program for computing the matrix of first derivatives

- Program for matrix inversion

- Program for computing the velocity model in regular 2D grid which is used as basic model in the next iteration

Programs indicated with pink are executed only during the first iteration.
We use the following indications:
'//ar//' is the AREA folder
'//md//' is the MODEL folder
'//it//' is the number of iteration

2.2. Ray tracing in the 3D velocity model

Project: \PROG\1_tracing
Input data:
in 1 iteration:
/DATA/**'//ar//'**/**'//md//'**/inidata/rays_xyz.dat
in next iterations:
/DATA/**'//ar//'**/**'//md//'**/DATA/rays**'//it-1//'**.dat

The calculations are controlled by parameters in the file:

```
 /DATA/DATASET1/MODEL_01/MAJOR_PARAM.DAT
*********************************************************
Parameters for tracing in 3D model using bending tracing
*********************************************************
TRACING_PARAMETERS:
! Parameters for BENDING:
0.5         dstep: basic step along the rays
```

```
3           bend_step: min step for bending
0.01        bend_min: min value of bending
5           bend_max: max value for bending in 1 step
2           k_reduce: frequency of data to be selected
100         nfreq_print: frequency of printing on console
```

Description of the main principle for the ray tracing

One of the key features of the *ATOM-3D* 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 *ATOM-3D* 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 2.1. 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, *B*, 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 2.1. The value of B is adjusted to obtain the curve γ(B) which provides the minimum value of the integral:

$$t = \int_{\gamma(B)} \frac{ds}{V(s)} \qquad [1]$$

where V(s) is the velocity distribution along the ray. (*ds*=**dstep**). B is varied from a maximum value, **bend_max,** to minimum value, **bend_min**.

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 iterations stop when the length of segments becomes smaller than a predefined value **bend_step**.

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 2.1, the algorithm is designed for the 3D case.

k_reduce is frequency of data which are taken into consideration (e.g. for **k_reduce**=2 only half of data is used, 3 – only a third part).

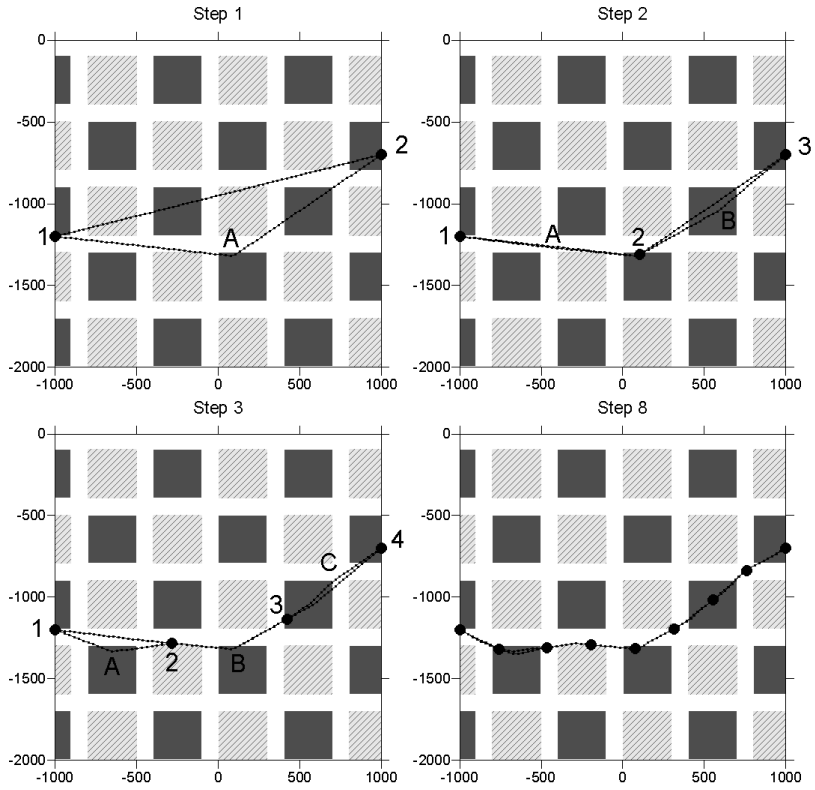


Figure 2.1. Grounds of the bending algorithm. Ray construction is demonstrated for a model with exaggerated velocity contrasts. 1D velocity varies from 2500 to 9000 m/s at 2000 m depth. Hatched light grey patterns represent negative anomalies of -30%; dark grey patterns are positive anomalies of +30%. Details of the bending algorithm are given in the text.

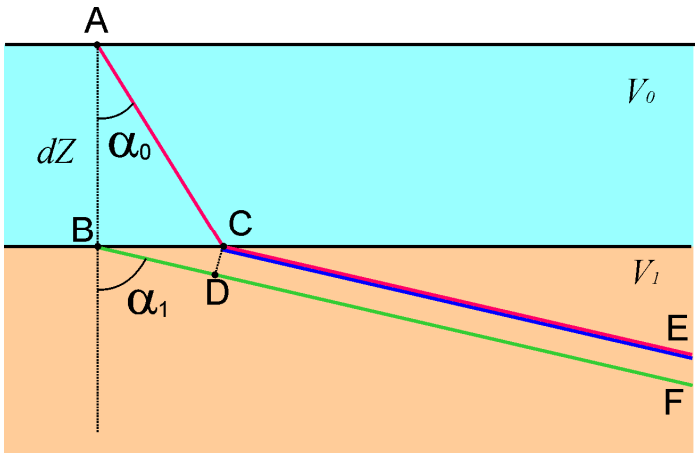


Figure 2.2. For the grounds of the sea correction introduction

In the case of marine experiments when sources are located on the sea surface, the travel time are corrected for the sea column. In this case the full tracing is not performed (in contrast to the 2D version of the algorithm, PROFIT, where tracing is performed from the source on the sea surface).

In the case of creating synthetic data (e.g. project "**b_synth_times**") tracing is performed from the point *B* located on the sea bottom directly beneath source (green line). Then the travel time of the synthetic ray passing through *A-C-E* (red line) is computed as:

$$T_{ACE} = T_{BF} - T_{BD} + T_{AC} = T_{BF} + \frac{dZ\sqrt{1 - p^2 V_0^2}}{V_0} \qquad [2]$$

where p is the ray parameter: $p = \frac{\sin \alpha_1}{V_1}$, T_{BF} is the computed travel time of the ray BF, dZ is the depth of the sea, V_0 and V_1 are velocities in the sea and in just beneath the sea bottom, respectively.

In the first iteration of the inversion procedure, we replace the source from the point A to C. Doing this, we assume that α_1 is close to 90°. In this case, horizontal shift will be:

$$BC = \frac{dZ\ V_0}{\sqrt{V_1^2 - V_0^2}} \qquad [3]$$

and corrected travel time along the ray CD (blue line) is computed as:

$$T_{CE}^{obs} = T_{ACE}^{obs} - \frac{dZ\ V_1}{V_0\sqrt{V_1^2 - V_0^2}} \qquad [4]$$

In the next iterations the source is always put in the point C, and the correction is not applied anymore. The residuals are computed as:

$$dt = T_{CE}^{obs} - T_{CE}^{md} \qquad [5]$$

where T_{CE}^{md} is the travel time computed in the current 2D model along the path CE.

2.3. Construction of the parameterization grid:

Executed Projects:

\PROG\2_ray_density\
\PROG\3_grid\
\PROG\4_tetrad\
\PROG\5_sosedi\

The calculations are controlled by parameters in the file:

/DATA/DATASET1/MODEL_01/MAJOR_PARAM.DAT

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


*********************************************************
Parameters for grid construction
*********************************************************
GRID_PARAMETERS:
-40. 40. 1.0    xgr1, xgr2, dxgr: grid for ray density calculation (X)
-40. 40. 1.0    ygr1, ygr2, dygr: grid for ray density calculation (Y)
-5. 20. 0.5     zgr1, zgr2, dzgr: min and max levels for grid
1                       ! Grid type: 1: nodes (other options are not valid)
0.5             dz_min: min distance between nodes in vert. direction
0.05 100.0      dens_min, dens_max: min and max ray density, relative to average
-5.             z_upper:  Uppermost level for the nodes

0.05            dx_step: step of movement along x
0.05            dz_step: step of movement along z
```

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

The steps of grid construction are following:

1. Ray density calculation: ***\PROG\2_ray_density***
Summary ray length is computed in cells with spacing **dxgr, dygr, dzgr** and limits **xgr1, xgr2, ygr1, ygr2, zgr1, zgr2**. Then the ray density function is normalized with respect to the average value in non-empty cells. If the ray density is less than **dens_min** of the normalized density, these cells are taken off from consideration. If the ray density is more than **dens_max** of the normalized density, the ray density in this cell will be set equal to **dens_max**.

2. Installing the nodes: ***\PROG\3_grid***
The regular nodes with spacing, **dxgr, dygr**, indicate the vertical lines where the nodes are installed irregularly according to the ray density. We move along the vertical lines with the step of **dz_step** and integrate the ray density function. As soon as it becomes greater than a predefined value, we put the node, and the integration starts anew. At the same time, the spacing between the nodes in vertical direction should be not less than **dz_min**.

3. Joining the nodes into tetrahedral blocks: ***\PROG\4_tetrad***
This step is required only for the next step

4. Finding the neighboring nodes: ***\PROG\5_sosedi***
This step finds all the neighboring nodes in the grid. The pairs of nodes are used in the smoothing block during inversion

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 smoothing 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 four differently oriented grids (0˚, 22˚, 45˚ and 67˚) and then stack them. Orientations of grids are defined in "MAJOR_PARAM.DAT":

```
    /DATA/DATASET1/MODEL_01/MAJOR_PARAM.DAT
******************************************************
ORIENTATIONS OF GRIDS :
4              number of grids
0 22 45 67     orientations of grids, degrees
```

2.4. Calculation of matrix:

Project: \PROG\6_matr\

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.

2.5. Inversion:

Project \PROG\7_invers\

The parameters for the inversion are contained in the file:

/DATA/DATASET1/MODEL_01/MAJOR_PARAM.DAT

```
*******************************************************
INVERSION PARAMETERS :
40              LSQR iterations
0.4             SM, level of smoothing
0.3             AM, regularization level
2.00            Vmin, minimal velocity
```

Inversion of the entire sparse *A* matrix is performed using an iterative LSQR code (Page, Saunders, 1982, Van der Sluis, van der Vorst, 1987). Number of iterations for inversion is **LSQR**. 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, **AM**, causes a reduction of the amplitude of the derived velocity anomalies. The second block controls the smoothing of the solution. Each line of this block contains two equal nonzero elements of opposite signs, which correspond to all combinations of neighboring nodes in the parameterization grid. The data vector in this block is also zero. Increasing the weight of this block, **SM**, causes a reduction of the difference between solutions in neighboring nodes, which results at smoothing of the computed velocity fields.

Vmin is the minimal velocity which is allowed in inversion. When this value is achieved in some nodes, in the following iteration steps they are not involved anymore.

2.6. Calculation of 3D model in a regular grid:

Project \PROG\6_2dmodel\

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

/DATA/DATASET1/MODEL_01/MAJOR_PARAM.DAT

```
*******************************************************
Parameters for 3D model with regular grid
*******************************************************
3D_MODEL PARAMETERS:
10. 100. 0.7        xx1, xx2, dxx,
10. 100. 0.7        yy1, yy2, dyy,
-5. 30. 0.7         zz1, zz2, dzz
3                   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**.

2.7. Practical realization of ATOM-3D code for the real data inversion

To perform a successful run of the *ATOM-3D* code, the data structure should be created as described in Section 1. The possibility to run the Steps 2.2-2.6 presented in the previous sections manually, step by step, is also implemented. However, the *ATOM-3D* code contains a program, which performs automatic managing of all steps. The source of this program is presented below:
Program for automatic managing of the *ATOM-3D* steps:
Program: \PROG\START_real_inversion\start_real.f90
(the executable program steps are highlighted in blue)

```
USE DFPORT
character*8 ar,ar_all(10),md,md_all(10),line
character*1 rg_all(100),rg,it
integer kod_loc(10),kod_iter(10)

open(1, file='../../all_areas.dat')
do i=1,4
       read(1,*)
end do

! Read the names of all models to be computed
do i=1,10
       read(1,'(a8,1x,a8,1x,i1,1x,i1,1x,i1)',end=7)ar_all(i),md_all(i),kod_iter(i)
end do
7 close(1)
n_ar=i-1

! Start computing all the models:
do iar=1,n_ar
       ar=ar_all(iar)
       md=md_all(iar)
       niter=kod_iter(iar)

       open(11,file='../../model.dat')
       write(11,'(a8)')ar
       write(11,'(a8)')md
       write(11,'(i1)')1
       write(11,'(i1)')1
       close(11)

       open(1,file='../../data/'//ar//'/'//md//'/MAJOR_PARAM.DAT')
       do i=1,10000
              read(1,'(a8)',end=573)line
              if(line.eq.'ORIENTAT') goto 574
       end do
       573 continue
       write(*,*)' cannot find ORIENTATIONS in MAJOR_PARAM.DAT!!!'
       pause
       574 read(1,*)nornt
       close(1)

! Start executing the iterative inversion for one model:
       do iter=1,niter
              write(it,'(i1)')iter
              open(11,file='../../model.dat')
              write(11,'(a8)')ar
              write(11,'(a8)')md
              write(11,'(i1)')iter
              write(11,'(i1)')1
              close(11)
! Ray tracing:
              i=system('..\1_tracing\resid.exe')
999           continue

! Start performing the inversion in differently oriented grids
```

```
                do igr=1,nornt
                        open(11,file='../../model.dat')
                        write(11,'(a8)')ar
                        write(11,'(a8)')md
                        write(11,'(i1)')iter
                        write(11,'(i1)')igr
                        close(11)
                        if(iter.eq.1) then

! Compute the ray sampling:
                                i=system('..\2_ray_density\plotray.exe')
! Construct the parameterization grid:
                                i=system('..\3_grid\grid.exe')
                                i=system('..\4_tetrad\Tetrad.exe')
                                i=system('..\5_sosedi\add_matr.exe')
! Visualization of the ray paths and grid:
                                i=system('..\_vis_rays_grid\paths.exe')

                        end if
! Compute the matrix of first derivatives:
                        i=system('..\6_matr\matr.exe')
! Perform the inversion:
                        i=system('..\7_invers\Invbig.exe')
                end do

! Combine the results of all grids into the one 3D model in regular grid:
                i=system('..\8_3D_model\mod_3D.exe')
! Visualization of the results in horizontal and vertical sections
                i=system('..\_vis_result_hor\visual.exe')
                i=system('..\_vis_result_ver\visual.exe')
        end do ! Different iterations
! Reports about variance reduction and visualization of the residuals
        i=system('..\_var_reduct\var_red.exe')
end do

stop
end
```

This program allows running all the *ATOM-3D* steps for one or several models. The list of models is defined in file "`/model_all.dat`". An example of this file is presented below:

`/all_areas.dat`

```
1: name of the area
2: name of the model
3: number of iterations
DATASET1 MODEL_01 5
DATASET1 BOARD_01 5
CANARES_ BOLIVAR1 9
```

In the presented example, three models are defined. All of them are from two AREA folders, "`DATASET1`" and "`CANARES_`", indicated in the 1st column in lines 4, 5 and 6. For all areas, the names of the models are: **"**`MODEL_01`**",** **"**`BOARD_01`**",** and **"**`BOLIVAR1`**",** that is indicated in the 2nd column. It runs for 5, 5 and 9 iterations, respectively (indicated in the 3rd column). Any number of iterations up to 9 is allowed. It is important to define all the parameters in the file "`all_areas.dat`" according to a fixed format: (`a8,1x,a8,i3`) and they should start from line 4. Any number of different models up to 20 can be defined. They will run successively one after another.

2.8. Running the data inversion using the BATCH file

The easiest way to run the data inversion is to start the BATCH file `START_REAL.BAT,` which is located in the root directory. This file runs the `start_real.exe` described in the previous section. Before running this file it is necessary to organize the file structure as described in Section 1 and define the names of areas and models in file `model_all.dat` to be computed.

File: `\START_REAL.BAT`

```
cd PROG
cd START_real_inversion
start_real.exe
pause
```

3. Presentation of the results

3.1. Express visualization tool for previewing

The *ATOM-3D* code contains a tool for automatic Express visualization of the results after each iteration. 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 simple program which is written in *C-sharp*. The executable file is located in **`\CREATE_PICS\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 theses 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.

We provide a test example in a folder "`CREATE_PICKS`" which can be used as template for creating new config files and to check the correctness of the visualization tool. Example of the "`config.txt`" file is presented below:

CREATE_PICS/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.bln
_______ 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 1st group (black) 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): contour line (blue), polygons (red) and dots (green) using the GRD, BLN and DAT files respectively, located in the same folder. Running the "**CREATE_PICS/visual.exe**" will result at producing a file "**CREATE_PICS/picture.png**" which should be identical to "**CREATE_PICS/picture_correct.png**". Before using this program, we recommend to perform this test to check the correctness of the visualization tool at your system.

The derived image is presented below:

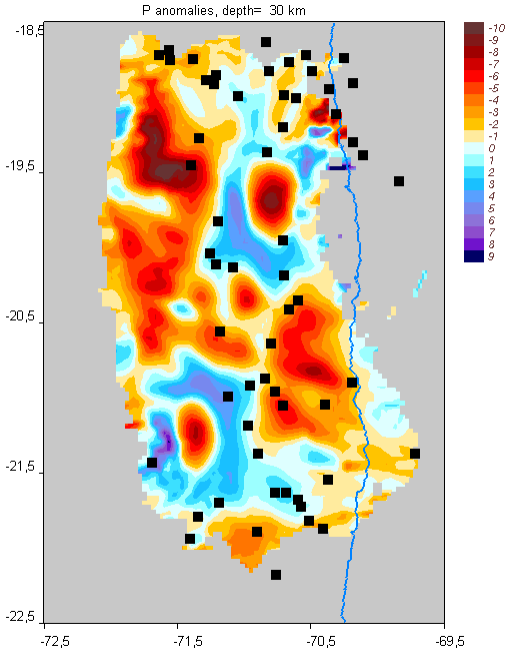


Figure 3.1. Resulting image (**\CREATE_PICS\picture.png**) obtained as a result of running the file **\CREATE_PICS\visual.exe** using the configuration from **\CREATE_PICS\config.txt.**

Drawing 2D functions (key 1) requires using the color scales with indicated path (e.g. **..\..\FIG_files\blue_red.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 **\CREATE_PICS** can be used for immediate control of the visualization tool.

To control realization of intermediary steps and to visualize the results, some special program should be run. They produce the files in the "**/FIG_file**" folder, in corresponding subfolders, which can be presented in Surfer or other similar visualization software.

3.2. Preview of the intermediate and final results as bitmap images in PNG files

Final and intermediate results are visualized automatically and stored in the folder `PICS` in corresponding subfolders. In order to activate this option, the file `preview_key.txt` in the root directory should contain only one number (1 or any other nonzero integer number). In case of absence of this file, or if it contains 0, previewing is not performed.

The parameters of the previewing are defined in the `config` files in the AREA folder, for example:
`DATASET1/config_rays_grid.txt`
`DATASET1/config_hor.txt`
`DATASET1/config_ver.txt`
`DATASET1/config_rays_grid`

The color scales should exist in the folder `/FIG_files/.`

The main steps of performing the data inversion can be seen in the PNG files produced in folder `/PICS/'//ar//'/'//md//'/`

Below is the description of the main visualization products.

3.3. Ray paths after first iteration and grid nodes

The ray paths and nodes of the parameterization grid in a defined depth interval can be visualized using the program `PROG/_vis_rays_grid/path.exe.` This program takes the currently computed model which is indicated in `model.dat`. The parameters for this program are contained in file `config_rays_grid.txt`:

`/DATA/DATASET1/config_rays_grid.txt`

```
0 3                zmin,zmax
1           frequency for ray selection
1           frequency for points on ray selection
```

The size of the map are defined in `config_hor.txt`:
The limits of the map are defined in `sethor.dat`:

An example for grid 2 with orientation of 22 degrees is presented in Figure 3.2.

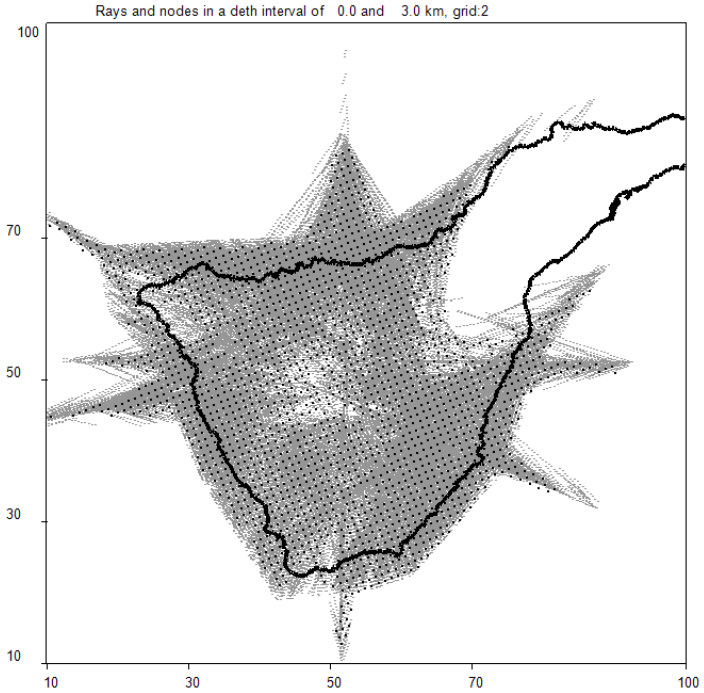


Figure 3.2. Picture in file /PICS/DATASET1/BOARD_01/rays_grid2.png, which corresponds to ray paths in the 1st iteration.

3.4. Horizontal sections of the resulting velocity model

The resulting velocity anomalies and absolute velocities after each iteration can be presented in horizontal sections. They can be visualized using the program PROG/_vis_result_hor/visual.exe. This program produces the GRD files of the resulting anomalies and absolute velocities which can be visualized using the SURFER software, or our own visualization tool.

The names of the AREA and MODEL, and iteration are defined in file "model.dat".

The parameters for visualization are defined in the file **"**sethor.dat**"** (see example below).

/DATA/DATASET1/sethor.dat

```
4
0 3 6 9            Depths of sections
10 100 0.5 10 100 0.5 x1,x2,dx,y1,y2,dy, limits of the map and steps
2            dismin, distance from nearest node
0            smooth, Smoothing factor
```

In this case, the map limits along X and Y are between 10 and 100 km. Steps along X and Y are 0.5 km. Four horizontal sections at the depth of 0, 3, 6 and 9 km will be shown.

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** (2 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.

The output of this program:
"/FIG_files/hor/dv lev it .grd": Relative anomalies, in percent.
"/FIG_files/hor/abs lev it .grd": Absolute velocities in km/s.

"lev" is a number of depth level according to the information in "sethor.dat". "it" is the number of iteration. This file can be directly visualized in the Surfer Software as a contour line plot or using our express visualization tool.

The express visualization runs automatically, if the "preview_key.txt" in the root directory is not zero. Express visualization of the results in horizontal sections require definition of some parameters in file "config_hor.txt". An example of the definition is shown below:

/DATA/DATASET1/config_hor.txt

```
600 600              size of the plot in pixels
black_white.scl      scale for anomalies
-10 10               diapason of the color scale for the anomalies, in %
rainbow_small.scl    scale for absolute velocities
4 6.7                diapason of the color scale for velocities, in km/s
20 20                ticks on axes
3 0 0 0              Line1: width, color (RGB)
3 0 0 0              Line2: width, color (RGB)
2 1 0 0 0            Dots: size, symbol (1 circle, 2 square), color (RGB)
```

An example of figure produced by the Express visualization tool is shown in Figure 3.3.

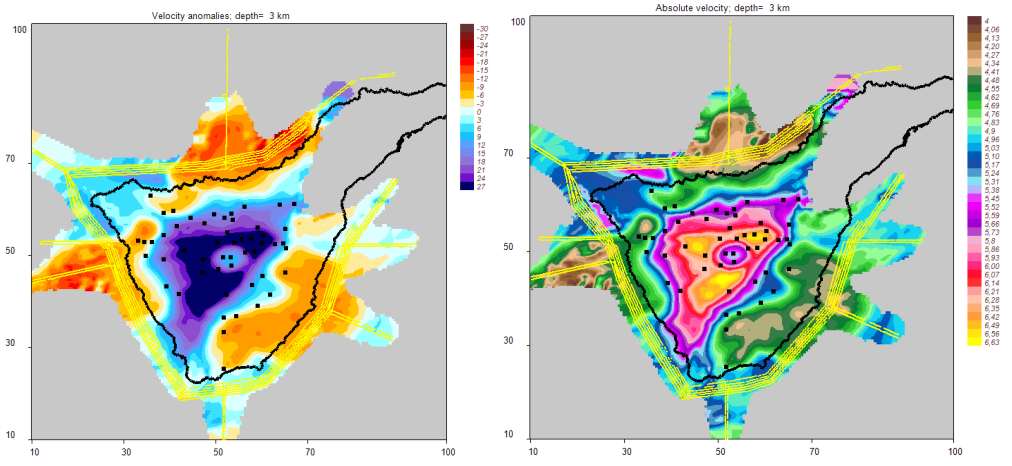


Figure 3.3. Picture in file /PICS/DATASET1/MODEL_01/dv5 2.png and /PICS/DATASET1/MODEL_01/abs5 2.png which corresponds to results of inversion after 5 iterations in the 2nd depth level. Black dots indicate the stations, yellow dots are the shots.

3.5. Vertical sections of the resulting velocity model

The resulting velocity anomalies and absolute velocities after each iteration can be presented in several vertical sections. They can be visualized using the program PROG/_vis_result_ver/visual.exe. This program produces the GRD files of the resulting anomalies and absolute velocities which can be visualized using the SURFER software, or our own visualization tool.

The names of the AREA and MODEL, and iteration are defined in file "model.dat".

The parameters for visualization are defined in the file **"**setver.dat**"** (see example below).

/DATA/DATASET1/setver.dat

```
3            Number of different sections
31.60 34.10 70.44 60.97  xa,ya,xb,yb: ends of the profile
21.67 58.78 73.21 42.86  xa,ya,xb,yb: ends of the profile
41.38 70.90 65.03 30.45  xa,ya,xb,yb: ends of the profile
5  distance from section for visualization of events
0.5               dx: step in horizontal direction
-5 15 0.5         zmin,zmax,dz: limits and step in Z direction
10               Marks for indication of position of section in map
1            dismin: Distance to the nearest node
0            smooth: Smoothing factor
```

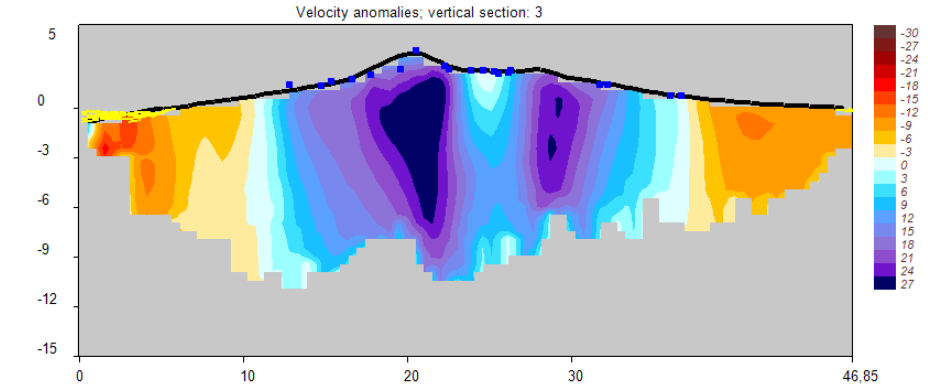
In this case, the the results will be presented in three vertical sections up to the depth of 15 km. Steps along X and Z are 0.5 km.

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** (2 km, in this example), the value of anomaly is not shown (dv=-999).

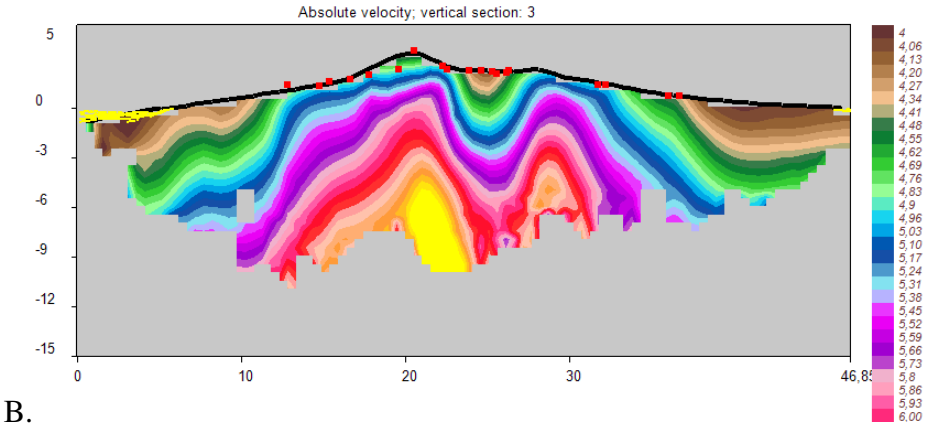
Visualization program allows smoothing of the visualized map which is controlled by the **smooth** parameter.

The output of this program:
"/FIG_files/vert/ver ver it .grd": Relative anomalies, in percent.
"/FIG_files/vert/abs ver it .grd": Absolute velocities in km/s.
"/FIG_files/vert/shots ver .dat": Projections of shots on the profile
"/FIG_files/vert/stat ver .dat": Projections of stations on the profile
"/FIG_files/vert/mark_ ver .bln": Locations of the profiles in map view
"/FIG_files/vert/topo_line_ ver .bln": Topography line in the vertical sections

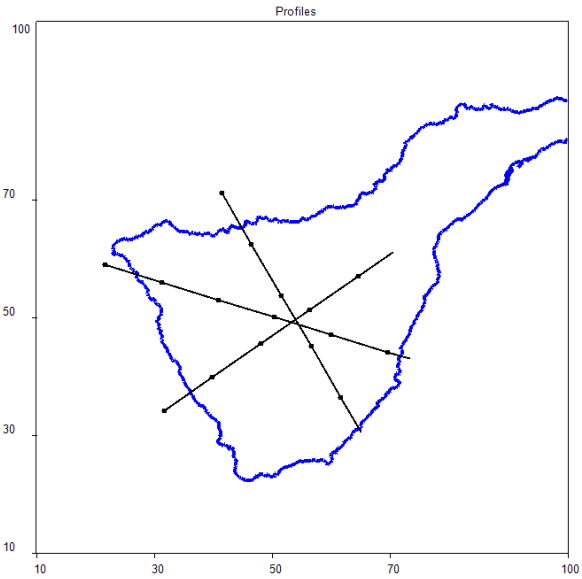
"ver" is a number of the vertical section according to the information in "setver.dat". "it" is the number of iteration. This file can be directly visualized in the Surfer Software as a contour line plot or using our express visualization tool.



A.



B.



C.

Figure 3.4. Picture in file `/PICS/DATASET1/MODEL_01/vert_dv 35.png`, `/PICS/DATASET1/MODEL_01/vert_abs 35.png`, and `/PICS/DATASET1/MODEL_01/profiles.png`. A and B correspond to results of inversion (relative anomalies and absolute velocities) after 5 iterations in the 3rd vertical section . Red dots indicate the stations, yellow dots are the shots. Plot C gives an idea about locations of the profiles.

The express visualization runs automatically, if the "preview_key.txt" in the root directory is not zero. Express visualization of the results in vertical sections requires definition of some parameters in file "`config_ver.txt`". An example of the definition is shown below:

/DATA/DATASET1/config_ver.txt

```
0 300              size in pixels
black_white.scl    scale for anomalies
-10 10             diapason of the color scale for the anomalies, in %
rainbow_small.scl  scale for absolute velocities
4 6.7              diapason of the color scale for velocities, in km/s
10 3               ticks on axes
```

In the case if the X size in the first line is zero, it is computed automatically based on 1:1 proportion of the section and length along Y.

An example of figure produced by the Express visualization tool is shown in Figure 3.4.

3.6. Report about variance reduction

After performing full tomographic inversion, the code provides full information about variance reduction and plots the residuals. This is performed using the program PROG/_var_reduct/var_red.exe.

The names of the AREA and MODEL, and iteration are defined in file "model.dat".

The parameters for visualization of residuals are defined in the file "config_resid.txt" (see example below).

/DATA/DATASET1/config_resid.txt

```
800 300                 size of the plot in pixels
0 60                    distance range
-2 2                    residual range
20 0.5                  ticks on axes
```

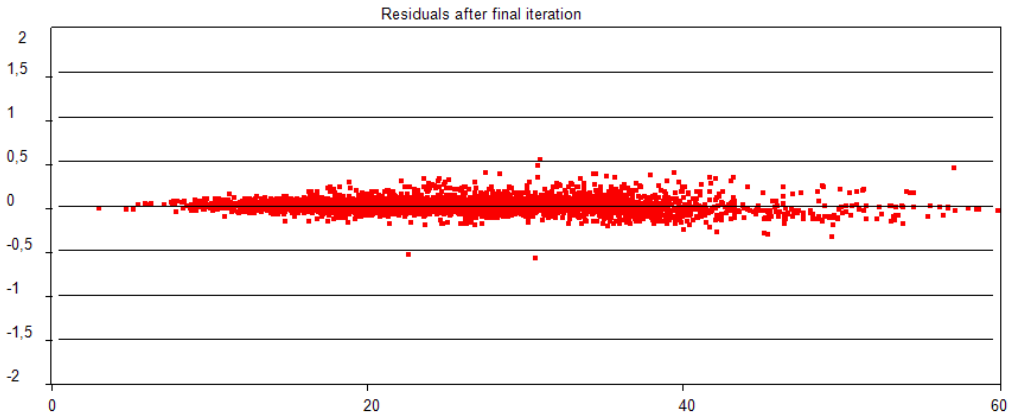


Figure 3.5. Picture in file /PICS/DATASET1/MODEL_01/resid.png which represents the residuals after 5th iteration.

In addition the program produces a report about average residuals and their reduction in iterations which can be seen on console and in file:

/PICS/DATASET1/MODEL_01/resid_norm.txt.

```
 iter=           1  dtot=  0.3754392      red=  0.0000000E+00
 iter=           2  dtot=  0.1093762      red=   70.86713
 iter=           3  dtot=  6.5221429E-02  red=   82.62797
 iter=           4  dtot=  5.1810998E-02  red=   86.19990
 iter=           5  dtot=  4.5875020E-02  red=   87.78098
```

4. Synthetic modeling

4.1. General remarks

The *ATOM-3D* software provides a wide range of possibilities for performing various synthetic tests. In the actual version there is possibility to define the synthetic model as superposition of the 1D synthetic model (file "ref_syn.dat") and velocity anomalies (file "anomaly.dat"). Velocity anomalies are defined 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. These times are the input for the entire inversion procedure.

Organization of folders for all kinds of synthetic models is shown in Figure 2.5 (page 8).

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

It is recommended to control the correctness of the definition using the programs in Projects:
\PROG\a_set_syn_hor for horizontal presentations
\PROG\a_set_syn_ver for vertical presentations

The names of area and models are defined in both cases in the file "model.dat". The depths and limits of the horizontal sections for visualization are defined in ****DATA\DATASET1\sethor.dat**,** the same as used for visualization of real data horizontal sections (Section 3.4). The coordinates of the cross-section and vertical limits for visualization are defined in \DATA\DATASET1\setver.dat, the same as used for visualization of real data vertical sections (Section 3.5).

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

results:
/FIG_files/hor/syn_dv_**M**.grd:
/FIG_files/hor/syn_abs_**M**.grd:

relative anomalies in percent and absolute velocities in horizontal sections. M is a number of a depth level according to the information in "sethor.dat".

The output of vertical presentation is written to:
/FIG_files/vert/syn_dv_**M**.grd:
/FIG_files/vert/syn_abs_**M**.grd:
Relative anomalies in percent and absolute velocities in vertical sections. M is a number of cross sections according to the information in "setver.dat".

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

They are atomatically visualized using our Express visualization tool. Some examples of synthetic model definition are presented in Figures 4.1 – 4.4.

In a case of rather complicated definition of horizontal or vertical anomalies, it is recommended to perform both Projects **a_set_syn_ver** and **a_set_syn_hor** .

4.4. 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_01/anom.dat.

```
1     1 - key for the checkerboard
_____________________________________________
7.00  anom: anomalies (e.g. +7 and -7)
-20. 120. 10. 0.0   xmin,xmax,dx1,dx2: limits and steps along X
-20. 120. 10. 0.0   ymin,ymax,dy1,dy2: limits and steps along Y
-3. 200. 200. 0.0   zmin,zmax,dz1,dz2: limits and steps along Z
```

In the presented example the anomalies are +_7% amplitude. The size of anomalies is dx1=10 km and dy1=10 for X and Y directions. There is a possibility to define empty space between anomalies (dx2=10). Along Z direction, the anomaly is the same down to 200 km depth. 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 -3 km depth.

A model corresponding to the presented example with the reconstruction results is shown in figure 4.1.

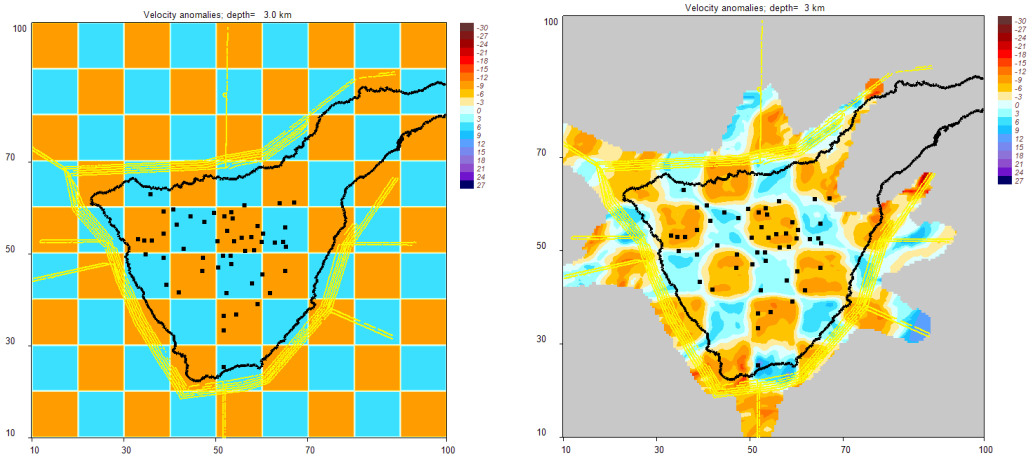


Figure 4.1. Checkerboard test. Left map represents the initial synthetic distribution (file: /PICS/DATASET1/BOARD_01/syn_dv 2.png). Right maps are the results of reconstruction at the depth of 3 km after five iterations (/PICS/DATASET1/BOARD_01/dv5 2.png).

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

4.5. 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\SYN_MOD1\forms\anom2.bln

```
12,1
54.0597682, 51.487703875
51.947018425, 51.065136775
50.679374275, 49.797492625
49.975114825, 48.529848475
50.82022045, 47.4030505
52.79212405, 46.557944875
54.200614375, 46.839637225
55.609133275, 47.4030505
56.5950565, 48.67069465
57.158469775, 50.079184975
56.172517975, 50.9242906
55.04572, 51.206011525-84.3796539533, 9.478820826
```

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

\DATA\DATASET2\SYN_MOD1\anomaly.dat

```
2       1 - board, 2 - horiz. anom, 3 - vert. anom
_____________________________________________
6       number of anomalies
*******************************
anom3           Name of the figure
0. 0. 0. 0.
15              value of anomaly, in %
-10 100         depth range
*******************************
anom1           Name of the figure
0. 0. 0. 0.
30              value of anomaly, in %
-10 100         depth range
*******************************
anom2           Name of the figure
0. 0. 0. 0.
-5              value of anomaly, in %
-10 100         depth range
*******************************
anom4           Name of the figure
0. 0. 0. 0.
-15             value of anomaly, in %
-10 100         depth range
*******************************
anom5           Name of the figure
0. 0. 0. 0.
-15             value of anomaly, in %
-10 100         depth range
*******************************
anom6           Name of the figure
0. 0. 0. 0.
-15             value of anomaly, in %
-10 100         depth range
*******************************
blac5           Name of the figure
0. 0. 0. 0.
7               value of anomaly, in %
-10 100         depth range
*******************************
```

In this example the synthetic model consists of four horizontal prisms. All of them are located in the depth interval between -10 and 100 km. All of them should be defined in folder **“forms”** in files **“anom1.bln“, “anom2.bln“, etc**. 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.

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

\DATA\DATASET1\BOLIVAR1\forms\scaling.bln

```
35 75           Xmin, Xmax on the map
20 80           Ymin, Ymax on the map
0               rotation angle, in degrees
```

A model corresponding to the presented example is shown in figure 4.2.

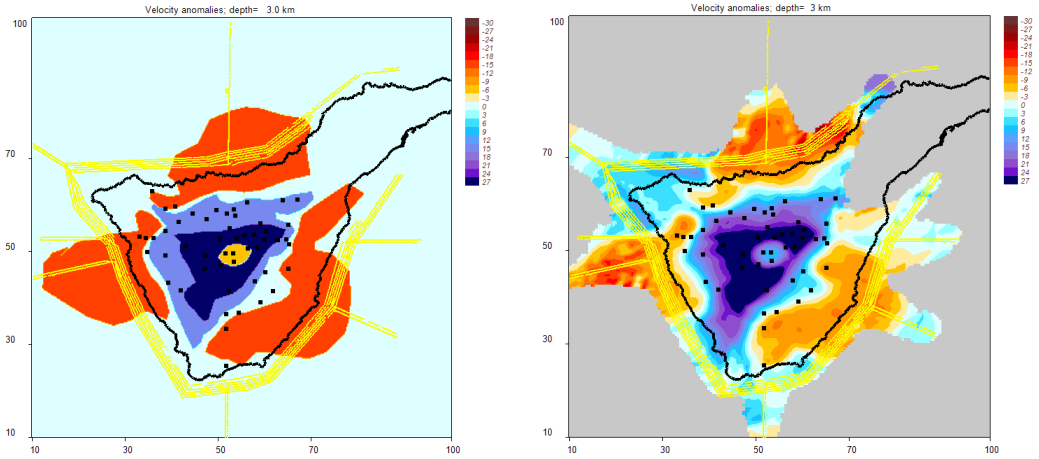


Figure 4.2. Synthetic model which was used to generate an testing dataset which is used in DATASET1 as a real dataset. Left map represents the initial synthetic distribution (file: /PICS/DATASET1/SYN_MOD1/syn_dv 2.png). Right maps are the results of reconstruction at the depth of 3 km after five iterations (/PICS/DATASET1/MODEL_01/dv5 2.png).

4.6. Definition of free vertical anomalies

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. Information about the synthetic anomaly is presented in the file "anomaly.dat**".** An example is presented below:

\DATA\DATASET2\SYN_MOD1\anomaly.dat

```
3       1 - board, 2 - horiz. anom, 3 - vert. anom
_____________________________________________
2
**********************************************
41.38 70.90 65.03 30.45  xa,ya,xb,yb: ends of the profile
anom1           Name of the figure
0 0 0 0
7               value of anomaly, in %
-20. 20         thickness of the anomaly across the profile
**********************************************
41.38 70.90 65.03 30.45  xa,ya,xb,yb: ends of the profile
anom2           Name of the figure
0 0 0 0
-7              value of anomaly, in %
-20. 20         thickness of the anomaly across the profile
```

In this example the synthetic model consists of four horizontal prisms. All of them are located in the depth interval between -10 and 100 km. All of them should be defined in

folder **"forms"** in files **"anom1.bln", "anom2.bln", etc**. 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.

In addition, the entire image can be scaled and rotated using file "../forms/scaling.dat"

A model corresponding to the presented example is shown in figure 4.3.

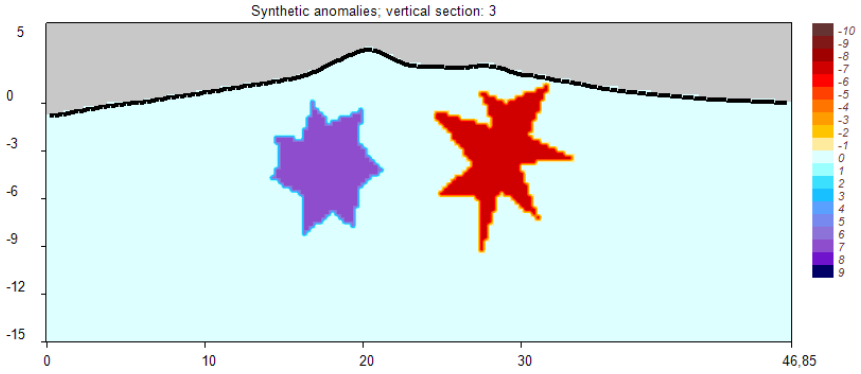


Figure 4.3. Synthetic model defined by free vertical patterns (file: /PICS/DATASET1/VERT_AN1/vert_dv_syn 3.png).

4.7. Definition of vertical checkerboard anomalies

For investigating vertical resolution we can defines checkerboard model in 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\ini_param\anomaly.dat

```
4       1 - board, 2 - horiz. anom, 3 - vert. anom
_____________________________________________
3
***********************************************
31.603904425 34.108731775 70.44072985 60.973289425 xa,ya,xb,yb: ends of the profile
-5 5    thickness of anomalies across profile
7.00    P-anomalies
-20. 120. 7. 0.0        xmin,xmax,dx1,dx2
-3. 100. 7. 0.0         ymin,ymax,dy1,dy2
***********************************************
21.6757207 58.783244275 73.21479085 42.86894095 xa,ya,xb,yb: ends of the profile
-5 5    thickness of anomalies across profile
7.00    P-anomalies
-20. 120. 7. 0.0        xmin,xmax,dx1,dx2
-3. 100. 7. 0.0         ymin,ymax,dy1,dy2
***********************************************
41.38612705 70.901501725 65.038597525 30.458675575 xa,ya,xb,yb: ends of the profile
-5 5    thickness of anomalies across profile
7.00    P-anomalies
-20. 120. 7. 0.0        xmin,xmax,dx1,dx2
-3. 100. 7. 0.0         ymin,ymax,dy1,dy2
```

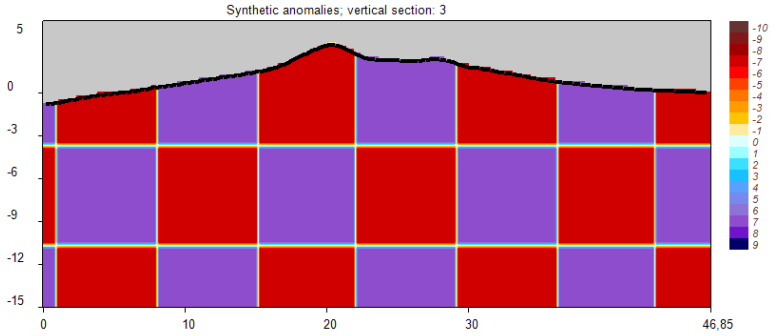
In the presented example both P and S anomalies are +_7% amplitude. The size of anomalies is 50 km along X and 30 km along Z. There is a possibility to define empty space between anomalies. These anomalies are defined across the section with the ends

defined in the first lines of each group.

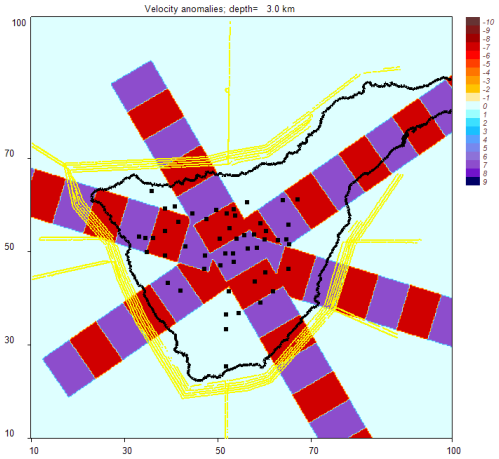
Thickness of blocks across the profile is from -5 km to 5 km.

.

A model corresponding to the presented example is shown in Figure 4.4.



A.



B.

Figure 4.4. Synthetic model defined by vertical checkerboard patterns in a vertical section (A) and in a map view (B) (depth 3 km). (files: /PICS/DATASET1/VER_BRD1/vert_dv_syn 3.png and /PICS/DATASET1/VER_BRD1/syn_dv 2.png)

4.8. Practical realization of *ATOM-3D* code for synthetic tests

To perform the successful run of the *ATOM-3D* for synthetic tests, the data structure should be created, and the anomalies should be defined, as described in section above. There is possibility to run the steps of synthetic data calculation and inversion manually, step by step. However, the *ATOM-3D* code contains a program which performs automatic managing of all steps. The source of this program is presented below:

Program for automatic managing of the *ATOM-3D* steps:
Program: \PROG\START_synth_inversion\start_syn.f90
(the executable program steps are highlighted in blue)

```
USE DFPORT
character*8 ar,ar_all(10),md,md_all(10),line
character*1 rg_all(100),rg,it
integer kod_loc(10),kod_iter(10)

open(1, file='../../all_areas.dat')
do i=1,4
        read(1,*)
end do

! Read the names of all models to be computed
do i=1,10
        read(1,'(a8,1x,a8,1x,i1,1x,i1,1x,i1)',end=7)ar_all(i),md_all(i),kod_iter(i)
end do
7 close(1)
n_ar=i-1

! Start computing all the models:
do iar=1,n_ar
        ar=ar_all(iar)
        md=md_all(iar)
        niter=kod_iter(iar)
        koe=kod_oe(iar)

        open(11,file='../../model.dat')
        write(11,'(a8)')ar
        write(11,'(a8)')md
        write(11,'(i1)')1
        write(11,'(i1)')1
        write(11,'(i1)')2
        close(11)

! Visualization of the synthetic model in horizontal sections:
        i=system('..\a_set_syn_hor\create.exe')

! Computing the synthetic travel times by ray tracing in the synthetic model:
        i=system('..\b_synth_times\rays.exe')

        !*****************************************************************
        open(1,file='../../data/'//ar//'/'//md//'/MAJOR_PARAM.DAT')
        do i=1,10000
                read(1,'(a8)',end=573)line
                if(line.eq.'ORIENTAT') goto 574
        end do
        573 continue
        write(*,*)' cannot find ORIENTATIONS in MAJOR_PARAM.DAT!!!'
        pause
        574 read(1,*)nornt
        close(1)

! Start executing the iterative inversion for one model:
        do iter=1,niter
                write(it,'(i1)')iter
                open(11,file='../../model.dat')
                write(11,'(a8)')ar
                write(11,'(a8)')md
                write(11,'(i1)')iter
                write(11,'(i1)')1
                write(11,'(i1)')2
                close(11)

! Ray tracing:
                i=system('..\1_tracing\resid.exe')

999             continue

! Start performing the inversion in differently oriented grids
                do igr=1,nornt
                        open(11,file='../../model.dat')
                        write(11,'(a8)')ar
```

```
                        write(11,'(a8)')md
                        write(11,'(i1)')iter
                        write(11,'(i1)')igr
                        close(11)
                        if(iter.eq.1) then

! Compute the ray sampling:
                                i=system('..\2_ray_density\plotray.exe')
! Construct the parameterization grid:
                                i=system('..\3_grid\grid.exe')
                                i=system('..\4_tetrad\Tetrad.exe')
                                i=system('..\5_sosedi\add_matr.exe')
! Visualization of the ray paths and grid:
                                i=system('..\_vis_rays_grid\paths.exe')
                        end if
! Compute the matrix of first derivatives:
                        i=system('..\6_matr\matr.exe')
! Perform the inversion:
                        i=system('..\7_invers\Invbig.exe')
                end do
! Combine the results of all grids into the one 3D model in regular grid:
                i=system('..\8_3D_model\mod_3D.exe')
! Visualization of the results in horizontal and vertical sections
                i=system('..\_vis_result_hor\visual.exe')
                i=system('..\_vis_result_ver\visual.exe')
        end do ! Different iterations
! Reports about variance reduction and visualization of the residuals
        i=system('..\_var_reduct\var_red.exe')
end do

stop
end
```

This program allows running all the *ATOM-3D* steps for one or several models. The list of models is defined in file "`/model_all.dat`" in the root directory. An example of this file is presented below:

```
1: name of the are
2: name of the model
2: number of iterations
DATASET1 BOARD_01 5
DATASET1 VER_SYN1 5
```

In the presented example, two models are defined. All of them are from the same AREA folders, "`DATASET1`", indicated in 1st column. The names of the models are: **"`BOARD_01`" and "`VER_SYN1`"** that is indicated in 2nd column. It runs for 5 iterations (indicated in 3rd column). It is important to define all the parameters in the file "`model_all.dat`" according to a fixed format: (`a8,1x,a8,i2`) and they should start from line 4. Any number of different models can be defined. They will run successively one after another.

4.9. Running the synthetic modeling using the BATCH file

The easiest way to run the data inversion is to start the BATCH file `START_SYN.BAT`, which is located in the root directory. This file runs the `start_syn.exe` described in the previous section. Before running this file it is necessary to organize the file structure as described in Sections 1 and 4 and define the names of areas and models in file `model_all.dat` to be computed.

File: \START_SYN.BAT

```
cd PROG
cd START_synth_inversion
start_syn.exe
pause
```

5. Closing remarks

This manual presents the *ATOM-3D* code based mainly on a real experiment with off-shore shots and onshore stations in Canaries. Further development of the *ATOM-3D* code is planned. In particular, we are working on including reflected and head waves in addition to the first arrivals. These data will be used for simultaneous inversion of velocity structures and geometry of interfaces.

I wish you a successful application of the *ATOM-3D* code and bright results. I would appreciate any help and suggestions on improving the code. In case of any inconsistencies and errors, please address the author, Ivan Koulakov (KoulakovIY@ipgg.nsc.ru). I am planning to prepare new versions of the *ATOM-3D* code with friendlier interface.

This study is supported by Russian Foundation for Basic Researches (08-05-00276-a), Heimholtz Society and RFBR joint research project 09-05-91321-SIG_a, multidisciplinary projects SB RAS 44, 21 and SB-UrO-DVO RAS 96, and project ONZ RAS 7.4.

References:

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

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, Dortrecht,*