

# **XMF 2**

## **Kinematic Seismic Model Fitter**

**Version 2.0.1**

**User Guide**

XGeo Ltd.  
St. Petersburg  
December 2021

XMF 2(Kinematic Seismic Model Fitter). Version 2.0.1. December 2021. User Guide.

Издатель: *XGeo Ltd.*

Автор: *Alexander Vinnik*

# Table of Contents

Introduction	1
FUNDAMENTALS	2
Input Data .....	2
Model .....	4
Waves .....	7
Interpretation Workflow .....	9
Data and Processing Control .....	11
Software .....	15
XMF and XTomo-LM .....	17
USER INTERFACE	19
Creating Project. Data Input .....	19
Viewing Seismograms .....	23
Initial Model: Creating .....	29
Initial Model: Import .....	31
Wireframe Editor .....	33
Binding Waves to Layers .....	39
Viewing Model .....	41
Color Velocity Spectrum .....	49
Model Editor .....	51
Forward Problem .....	56
Ray Picture .....	59
Studying TX-curves .....	62
PRE-STUDY	67
Problem Statement .....	67
Pre-study technology .....	68
Fitting a local section .....	71
Pre-study modules .....	74
SMM: Oviewview. Tools .....	78

SMM: Standard model creation .....	83
SMM: Velocity editor .....	87
Ray picture .....	91
APPENDICES .....	93
1. Seismic File Format .....	93
2. ASCII File Formats .....	97
3. Error Codes .....	100
4. Example of s-section .....	100

## Introduction

XMF (Kinematic Seismic Model Fitter) is a platform for "manual" fitting of a two-dimensional layered model to a set of line common shotpoint seismograms. As XTomo-LM, the first application of this product line, XMF is a means of solving the two-dimensional kinematic inverse problem. However, while XTomo-LM implements exact inversion algorithms, XMF is a workframe for empirical try-and-error approach. XMF is released after XTomo-LM, and this is explained by the fact that in some areas, where travel-time analysis is essential, mathematical approach makes excessive demands on input data, while simple well-tested empirical model fitting does work. The more so, if it is built in an appropriate environment. The developers consider both approaches as complementary: both products share the same method of model description and have a program interface which allows applying them to interpretation of same data.

The above mentioned layered model description assumes that both velocity function  $V(x, z)$  and seismic horizons can be represented by a single curvilinear grid. The special ray-tracing algorithms based immediately on the Kirchhoff and Fermat principles are developed for such grids.

The first XMF version contained the environment and basic tools for fitting but offered no technology. XMF 2 does offer such technological component, which allows approximation of a two-dimensional model by a set of one-dimensional sections, each being fitted to one seismogram using a simplified model in the vicinity of a shotpoint. The instantaneous ray-tracing tool and efficient user interface makes fitting 1D section a comparatively easy problem.

It takes two successive releases to put XMF 2 completely into operation. The present release 2.0.1 introduces the *preliminary study* (pre-study) unit into XMF; the next release 2.1.1 will be devoted to the entire system optimization.

The XMF documentation consists of three chapters. The first one contains formal or, at least, more or less strict description of main problem's components: input data, waves, model, processing outline. The second chapter is devoted to XMF programs' user interface. Finally, the third part contains the description of the pre-study unit.

The document is also used in the context help system, which can be accessed by F1 or help buttons in almost every window of any XMF program. To make the software functioning clear, some special terms are introduced in this document, which are not in common use and are only means of explanation. They are printed in italic at first appearance and are valid only in the documentation and user interface. Italic is also applied to the names of GUI components.

## FUNDAMENTALS

### Input Data

XMF input data consist of:

- a set of common source point (CSP) seismograms registered on a seismic line;
- positional data.

Seismograms are delivered as SEG-Y files adapted to x86/Windows platform (SEG-Y/PC). Exact requirements to seismic files are stated in Appendix 1. Positional data are extracted from seismogram trace headers or from ASCII files. In that case requirements to seismogram are minimal.

#### Positional data

Positional information includes numeric identifiers of sources and receivers and their coordinates. A source identifier is its Field ID. It is also an identifier of a matching seismogram. Receiver numbers are not used at this stage. ASCII files for positional data form a triple of SRC, RCV and RAY files. File of the SRC type contains source IDs and coordinates; RCV file contains coordinate of all receivers; RAY file describes "rays", i.e. triples "source – receiver/ trace". File formats are explained in detail in Appendix 2. The files will be referred to as *SRR files*, by analogy with SSR (Shot–Station–Relations) files once used in the Focus<sup>®</sup> seismic system of Paradigm Geophysical.

Raw coordinates are integer numbers (SEG-Y scaling is not applied). One can consider the length unit as meter, though it is not, actually, used. Within XMF, one can use the same length unit or multiply it by 1000 (conditionally, kilometer). The base time unit is second.

#### Line coordinate system

Source and receiver coordinates are supposed to be defined in a local Cartesian coordinate system  $(X, Y, Z)$  with axis  $OZ$  directed vertically upward and line  $L$  lying in the  $(X, Y)$  plane. In particular, UTM coordinates can be used. If  $L$  goes along  $OX$  or  $OY$ , we have two-dimensional problem in  $(X, Z)$  or  $(Y, Z)$  coordinates. If it is not the case, coordinates must be transformed to line coordinate system  $(O_L, X_L, Z_L)$  in which the  $O_L X_L$  axis goes along  $L$  and  $O_L Z_L$  is directed vertically upward. The transformation is built in the XMF module importing input data. It creates two data stores: Raw Positional Data Cash and Positional Database. The former is an auxiliary data set containing coordinates in the same form as they are written in trace headers or SRR file. The latter is main positional

data storage. It contains both raw and line coordinates of every source and receiver. The user can view both stores as numeric tables.

We elucidate shortly how the transformation of coordinates is implemented. XMF does not use information on the exact line position, but, instead, requires that the below assumption is true:

- (P) *scattering of sources and receivers around the planned seismic line is so small that changing devices for their orthogonal projections on the line does not produce significant errors in wave kinematics.*

Keeping in mind (P), we fit a line to source point set and take it for *real seismic line*, which can slightly differ the planned one. The coordinate transformation includes orthogonal projecting of all devices onto the built line, selecting new origin  $O_L$  and computing distances from the device projections to the origin. The latter is selected programmatically in the way that for all devices  $X_L \geq 0$ . Z-coordinate does not change:  $Z_L = Z$ . Positional Database Viewer displays some details of the transformation.

[More complicated algorithm of building kinematically equivalent line coordinate system is implemented in XTomo-LM Data Preparation Unit (DPU). It does not use (P) and is designed for cases when a seismic line is laid in a rugged terrain.]

### Spatial resolution

Digital model description uses 4-byte floating-point numbers in computer's memory, which means that decimal precision is no more than  $10^{-6} - 10^{-7}$ . If distance between two objects is less than the precision, they are not distinguishable. Rounding errors produce substantial contribution in the problem. To avoid operations with too close objects, all algorithms test distances between them against the value of *resolution* – the minimal allowed distance. It is computed automatically, depends on model x-range and takes into account, very roughly, possible rounding errors. More details can be found in XTomo-LM documentation. If a test against resolution fails, XMF generates the *resolution error*. That can happen, for example, if the user makes interfaces too close or velocity grid too thick when editing the model.

*Note.* The algorithm of building line coordinate system secures that receiver line x-coordinates do not cause resolution errors. In theory, it may happen; in this case, a receiver generating resolution error, is excluded from the observation system. Only after resolution test, receivers are numerated, and each receiver acquires its number as a unique identifier.

## Model

[Overview](#) – [Model wireframe](#) – [Velocity grid](#) – [Velocity function](#) – [Grid density](#) – [Concluding notes](#).

### Overview

The layered model includes two components: velocity function  $V(x, z)$  и "geometry" – a set of curves representing seismic horizons. Sampling suggests definitions of two-dimensional grids for each component. Those grids are, generally speaking, different and agree with precision, required for each component. Strictly speaking, the precision depends on depth because the sounding signal loses its high-frequency components on its way to model bottom. Nevertheless, XMF uses a single curvilinear grid for all components (fig. 1), which, naturally, restricts the class of layered models under consideration.

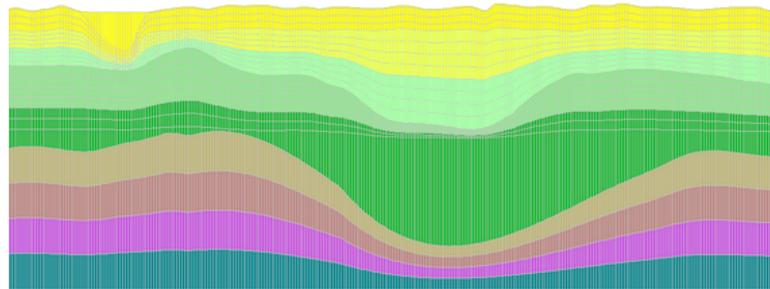


Рис. 1. Model representation by single curvilinear grid.

### Model wireframe

This is a mathematical abstraction underlying the XMF software. Model wireframe is a set of simple non-intersecting curves defined on a rectangular domain  $D$  of the  $(X, Z)$  plane and representing daylight surface line (DSL) and seismic horizons. Each curve enters  $D$  through the left side and leaves it through the right side. A curve is called *simple* if its equation has the form  $z = f(x)$ , which means that its points are unambiguously defined by their abscissas. Sampling of such curve is defined by an x-net on a segment of the  $OX$  axis, which is called a *wireframe x-net*.

It is assumed that a single x-net is used to represent all wireframe curves. This assumption is a compromise between precision of the curve representation in upper and lower model layers. At that the fact that ray-tracing time grows (non-linearly) with grid density should be taken into account.

Model wireframe is created and edited by the user in the Wireframe Editor program. However, DSL is built automatically as the lowest curve going above all sources and receivers with some tolerance. A snapshot of Wireframe Editor is shown on fig. 1.

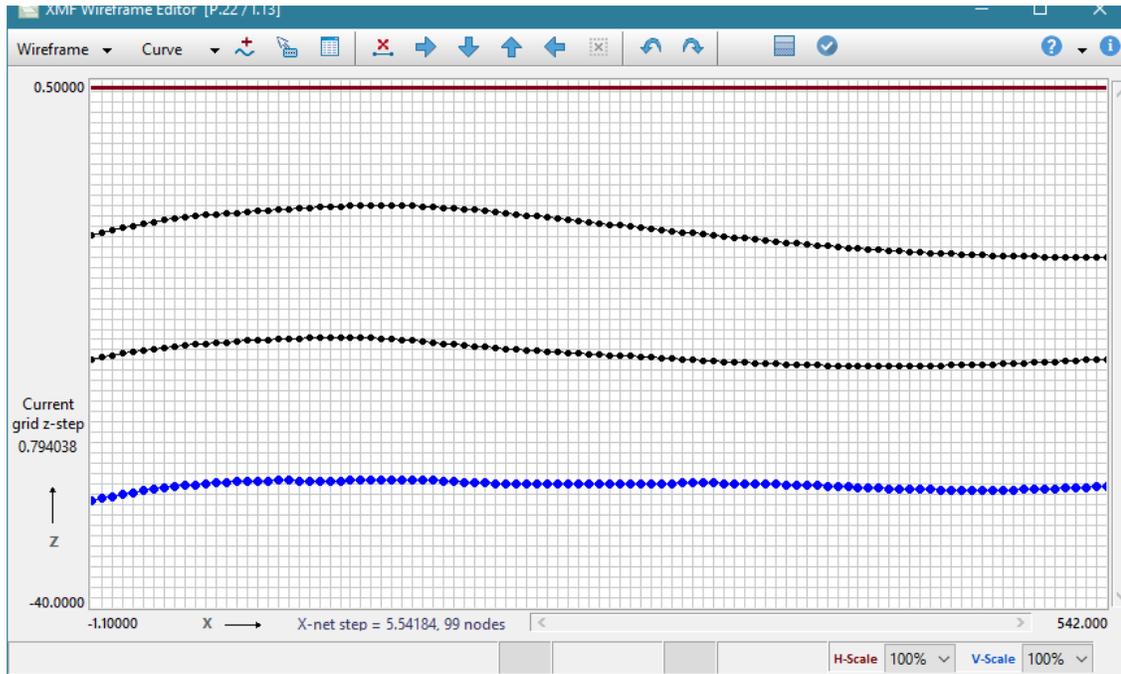


Fig. 1. Wireframe Editor's main window.

On the snapshot,  $D$  coincides with the drawing pad covered with uniform grid, whose verticals pass through the x-net nodes. The grid's horizontal lines are drawn only for convenience, so that the cells are square. Clear, that curve points belong to verticals but not necessary to horizontal. DSL is a model top border, which bounds from above the *model domain*.

A model wireframe defines *model layers*. The three-layer model on fig. 1 stands on the half-space represented by the additional fourth layer. It participates, with velocity values at its top, in modeling of a head wave on the last interface.

### Velocity grid

XMF automatically stretches two-dimensional velocity grid over the wireframe built by the user. The grid is formed by a set of verticals intersected by a family of quasi-horizontal curves called *h-lines*. They satisfy the same restrictions as wireframe curves. More grid properties are listed below.

- When the grid is built for the first time (initial model), its verticals pass through the wireframe x-nodes. Later, in the course of editing, a number of verticals can be doubled several times until the resolution error occurs. At that, the new wireframe curve points are computed by linear interpolation.
- Shape and position of h-lines are defined by XMF so that some of them coincide with wireframe curves. A number of h-lines in model, in a layer or a sublayer can be doubled several times until the resolution error occurs.

- After the model wireframe is changed, the grid is stretched over it anew, in the way that its structure is kept unaltered, if possible. The only exception is the case, when a number of wireframe curves (or model layers) is changed. In that case, the grid is built, as in initial model, from scratch.

The *standard grid subsets* are shown on fig. 3 and their names – in the underline. Additionally, the entire grid and a layer belong to standard subsets. The "subgrid" term is also used, in a broader sense, for any grid subset.

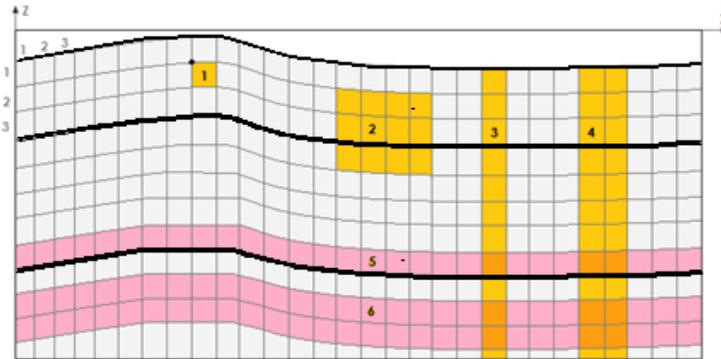


Рис. 2. Grid stretched over two-layer model wireframe.

1 – cell and its vertex; 2 – subgrid; 3 – column; 4 – vertical strip; 5 – row; 6 – horizontal strip.  
Numbers at top left corner show how rows (h-lines) and columns (verticals) are numbered.

### Velocity function

Model velocity is velocity of P-waves. In a discrete model, it is defined at grid nodes. Velocity value at a node is also its value in the cell, for which the node is its vertex. Thus, velocity can be describes as a table "row-column-V" or as cell function  $V(cell)$ . Though XMF works with the discrete model, we will often apply terms used for continuous model and write  $V(x, z)$  instead of  $V(cell)$ . The term *vertical profile* for  $v(z) = V(\cdot, z)$  is used together with *velocity column*.

It is, sometimes, more convenient to represent velocity function a set of *layer velocities*:

$$V = V(x, z) = \{ V_1(x, z), V_2(x, z), \dots, V_n(x, z) \},$$

where  $V_k$  is layer velocity in the k-th layer. In the initial model, layer velocities are constant.

After editing model wireframe, velocity also changes, because grid nodes at verticals are shifted. This dependency velocity of geometry is present in geological models as well. Velocity function is edited in the Model Editor program. Velocity values can be changed in any standard grid subset; any vertical profile can be edited numerically; velocity can be interpolated and continued in different ways.

### Grid density

Grid density must agree with model differentiation. Thus, normally, grid density is higher at model top and lower at its bottom. That is achieved by multi-doubling a number of h-lines in horizontal strips. As for verticals, their frequency is constant across the model, but can be also increased by multi-doubling. However, that operation does not increase wireframe curve precision, because new curve points are fully defined by those of initial curves. In order to make that obvious for the user, model wireframe is edited on its own x-net, independently of the current grid frequency. That means that the wireframe x-net step must be guessed at the very beginning.

The XMF settings make the grid nontrivial, even if the initial model is horizontally layered with constant layer velocity. This is done in advance minding further model elaboration.

Securing ray-tracing computational stability requires that *grid cells are not too oblong*. A ratio of the larger cell edge to the smaller one must not exceed one order, but the smaller the ratio is the better, and smoother ray paths are.

### Concluding notes

1. The restrictions on model wireframe curves define the class of layered model permissible in XMF.
2. A single-grid representation of a discrete model become feasible due to special ray-tracing algorithm that can work with curvilinear grids.
3. One should not consider a discrete model described above as a result of sampling a continuous model. Such model can be used directly for description of a geological media, the more so, that continuity is not an immanent feature of the latter.

## Waves

### Wave types

XMF solves forward kinematic problem for the following wave types:

- diving (strongly refracted wave whose rays return back to surface);
- reflection;
- head wave.

We use the term "head wave", not "refraction" to avoid ambiguity and with stress on the way it is modeled. Head wave propagates along a horizon of small curvature separating layers with velocities  $V_1$  (the upper) и  $V_2$  (the lower) such that  $V_1 < V_2$ . For the layered model, reflected and head waves have fundamental importance. The user defines which waves are formed at each layer floor. This action is called *binding waves* to model layers

and is a part of ray tracing task. The diving wave rays are traced always. If velocity in the first layer is constant, the diving wave turns into the direct wave from source to receiver.

### Converted waves

Converted wave travels partly as P-wave and partly as S-wave, changing its nature upon reflection or refraction at a seismic horizon. The kinematic aspect of conversion can be treated as change of layer velocities. If  $V(x, z) = \{ V_1(x, z), V_2(x, z), \dots, V_n(x, z) \}$  is model velocity function and  $V_k$  is layer velocity (of P-wave) in  $k$ -th layer, the converted wave propagates with velocity

$$(\#) \quad V(x, z) = \{ C_1 \cdot V_1(x, z), C_2 \cdot V_2(x, z), \dots, C_n \cdot V_n(x, z) \}, \quad 0 < C_k \leq 1,$$

where *conversion coefficient* (CC)  $C_k$  pertains to  $k$ -th layer and

- = 1, if a wave propagates in  $k$ -th layer as P-wave;
- < 1, if a wave propagates in  $k$ -th layer as S-wave.

(The term "conversion coefficient" in the above sense relates to this document only). A converted wave traverses model layers twice: as incident wave and as reflected (head) wave and may experience conversion at some horizons on its way. To describe the multitude of converted waves, we need additional means.

### Wave codes

*Wave codes* are used to represent waves in the user interface. A monotype P-wave is encoded with a string "NT", where  $N$  is an ordinal number of a horizon at which it is formed,  $T$  is wave type: D – for diving, R – for reflection, H – for head. For diving wave,  $N = 0$ . Here are examples: 0D, 1R, 3H, 10R.

Now let us address a wave. For each model layer, conversion coefficient  $C$  must be defined. By default,  $C = 1$ , but if there is a wave which propagates in this layer as S-wave,  $C$  is set to a ratio of velocities of S- and P-wave in this layer.

A reflected (head) converted wave formed at  $k$ -th horizon is depicted with two binary (i.e. consisting of 0 and 1) strings  $s$  and  $s^*$ , each of length  $k$ .  $s$  pertains to incident,  $s^*$  – to reflected (head) wave. The string elements are called conversion flags;  $i$ -th flag is 0, if the wave propagates in the  $i$ -th layer as P-wave, and 1 if it is an S-wave,  $i \leq k$ . Consider wave  $W$  reflected from the third horizon and propagating upward as S-wave. Then, conversion flag strings for  $W$  are:  $s = "000"$ ,  $s^* = "111"$ . If we treat a binary string as binary representation of a decimal integer,  $W$  can be encoded as "3R-0-7". Repeating the same arguments for an arbitrary converted wave, we obtain its code in the form "NT-F-F\*", where  $F$  and  $F^*$  are integers whose bits in binary representation are conversion flags.

Such encoding is not transparent but it is compact, and this is important. Of course, encoding is performed automatically, and the user can view what is behind wave code at

any moment. XMF uses 31-bit integers for  $F$  and  $F^*$ , which implicitly restricts a number of converting horizons by 31.

With given conversion coefficients and flags, wave propagation velocity can be determined by (#) unambiguously.

## Interpretation Workflow

In this section, it is shortly explained, how user work in XMF is organized as an iterative process. Below the process steps are listed and commented on.

### 1. Creating a project

A project is created for processing of a seismic line or its part. The user enters project name, path to a folder containing line seismograms and defines a source of positional data: trace headers or text files. At project create time, XMF creates environment for further work, in particular, necessary data structures on disc. All subsequent work evolves within project and all created data are placed in *project folder*. A new project is created not only for new data, but also for producing a new variant of interpretation.

### 2. Data input

The following actions are performed at this step:

- seismogram verification;
- extraction of raw positional data from trace headers or SRR files into Raw positional Data Cache;
- building line coordinate system and transformation of coordinates;
- building Positional Database;
- creating auxiliary data sets, in particular, a detailed source list – a base data structure in supporting of the interpretation process.

All these actions are carried out by two alternative XMF programs.

### 3. Creating initial model

Initial model is layered model with constant layer velocities. The user builds a model wireframe basing on a priori (with respect to XMF) information. In the last resort, the wireframe can consist of a set of horizontals. A number of layers can be changed later, but at that, velocity function is lost and replaced with constants.

### 4. Binding waves to layers

Clearly, the interpreter determines initial model as a result of studying the wave field. At this step, XMF must be informed about that. In a special program, for each or some layers, the user defines a list of waves generated at the layer floor. For each wave, its type and

conversion parameters are fixed as well as drawing attributes of rays and traveltime curves used by graphic programs. Wave binding can be changed in the course of processing.

## 5. Iterations

The term "iteration" is used for a sequence of user actions, which is repeated many times to attain the main goal: minimal deviation of computed TX-curves from phase lineups on seismic records. The actions of one cycle include:

- a) wave binding or changing it;
- b) editing model (wireframe and/or velocity);
- c) changing forward problem task, i.e. samples of sources and waves participating in ray-tracing;
- d) solving forward problem; the result is the Ray Catalog database, containing all ray traced and their ray paths;
- e) examination of ray picture by ray samples defined by the user;
- f) studying wave TX-curves drawn over seismograms and deviation analysis.

Actions a) и c) are not obligatory. All data generated within one iteration are placed in the *iteration folder* inside the project folder.

After step 4, the user creates the first iteration folder and carries out actions a) – f). Then he or she repeats step 5 as many times as needed. At that, XMF creates an appropriate data structure, which is examined in the [next section](#).

## 6. Next iteration

In the standard interpretation process, the user studies TX-curves on seismograms in the N-th iteration and makes the relevant model corrections in the (N+1)-th iteration. XMF does not forbid editing the model in same iteration, but the user should understand, that immediately after the model changes are saved, the ray picture and TX-curves become invalid, and forward problem solution must be obtained anew. It is sometimes convenient to perform several trial iterations in the same folder, but every time the model file is overwritten, a possible branch point of the interpretation process is lost, and the process is not, normally, unidirectional.

## 7. Pre-study

Pre-study means a method of building a model approximation through fitting a set of one-dimensional sections. The last document chapter is completely devoted to this technology. Typically, pre-study is applied to turn the initial model into much better approximation than it usually is. However, pre-study may be used for a model of any project iteration. This is an autonomous unit of XMF, not connected formally to the process described above. It may be placed somewhere *between* two iterations: the one to be refined and the resulting one.

## Data and Processing Control

[Processing tree](#) – [Work folder menu](#) – [Project menu](#) – [Zero iteration menu](#) – [Pe-study menu](#)– [Iteration menu](#).

The head program XMF displays data structure as a tree (fig. 1a), whose nodes match folders of the file system. The root node images XMF *work folder* – a container of all XMF projects. The next level is filled with project nodes. When such node is expanded, iteration nodes are displayed.

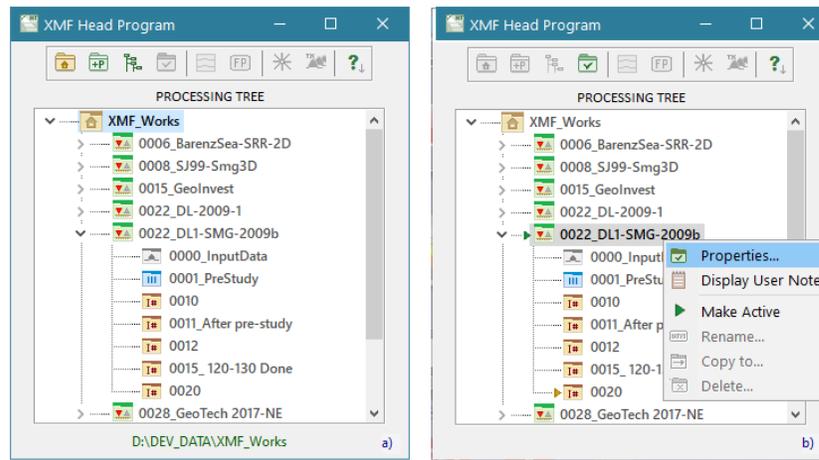


Fig. 1. a)representation of data structure; b)processing tree, active nodes, project node menu.

Project and iteration names begin with four-digit string – a *system number* of an object. They are assigned by XMF and serve as convenient references. The rest, optional, part of a name is an editable *user name*. The first 10 iteration numbers are reserved by the system. The first iteration folder of each project has the standard name "0000\_InputData" and is called the *zero iteration folder*. It contains the input data: Positional Database, initial model and the like. The node *0001\_PreStudy* node is a folder for preliminary study. The rest, with system numbers  $\geq 10$ , are regular iteration folders. Each iteration folder is created by the user. System numbering may have gaps because of deleted nodes.

### Processing Tree

The tree on fig. 1a is called *Processing Tree* because, additionally to mapping data structure, it performs functions of a dispatcher and a controller of the interpretation process. For that, it is provided with all required tools, accessed through 5 context menus. Depending on which item is *current* or *selected*, one or another menu is invoked on a right-click. On fig.1b, the project menu is invoked. Tool bar button duplicate menu commands for most important operations.

A project or iteration node is *active*, if data in its folder can be created or modified. An active project is set by the user with a special menu command. Once it is done, the last project's iteration becomes active automatically. Therefore, it is only the last iteration of

the current project, in which the actions making up [step 5](#) of interpretation workflow, can be performed. All other tree nodes of the current or any other project permit only viewing data. Active nodes are marked with small arrow to the left of the node icon as the menu command on fig. 1b.

A set of each menu depends on whether a project or iteration is active. In this way the program secures the discipline. Simple commands are performed by the head program, but, normally, it runs other XMF modules, watching that no conflict happens between the running modules. One reason that a command is disabled is just to avoid conflicts, for example editing the same data.

Below, commands of each menu are listed with short description and, sometimes a link ">>" to detailed discussion.

### Work folder menu

Table 1.

<i>Command</i>	Description
<i>Select Work Folder</i>	Invokes a dialog, in which the user selects an item from the list of earlier defined work folders or defines a new one with the help of local folder browser.
<i>Create Project</i>	Starts the module creating a new project. In particular, it creates a new project folder with zero iteration folder inside it <a href="#">&gt;&gt;</a> .
<i>Create Project As Copy</i>	Creates a new project as a copy of an existing one, either in full or only with zero iteration, i.e. input data.
<i>Rebuild Tree</i>	Rebuilds Processing Tree anew using the work folder tree on disk.

### Project menu

Table 2.

<i>Command</i>	Description
<i>Properties</i>	Displays project properties defined at create time. The user can change only import/export folder, the rest properties are project's constants <a href="#">&gt;&gt;</a> .
<i>Display User Notes</i>	Runs a simple text editor, like Windows Notepad, for work with User Notes file. The editor has a context menu with the standard

	commands. A project and each non-zero iteration have their own user notes files.
<i>Make Active</i>	Accessible, if active project is free from running modules. Makes current project and its last iteration active.
<i>Rename</i>	Allows editing project user name if the project is not active and is free of running modules. The system number stays intact.
<i>Copy to</i>	Copies an inactive project folder (with its content) to a network location.
<i>Delete</i>	Removes an inactive project from the tree and its folder – from disc, if only no module is working with the project data.

### Zero iteration menu

The menu commands provides input of positional data and their viewing.

Table 3.

<i>Command</i>	Description
<i>Extract Positional Data</i> <i>Import Positional Data</i>	Only one of these commands appears on the menu, depending on the source of positional data. Each command executes fully <a href="#">step 2</a> of interpretation workflow <a href="#">&gt;&gt;</a> .
<i>View Raw Positional Data</i>	Runs Raw Positional data Cache Viewer <a href="#">&gt;&gt;</a> .
<i>View Positional Database</i>	Runs Positional Database Viewer <a href="#">&gt;&gt;</a> .
<i>View Seismograms</i>	Launches Line Seismogram Viewer for viewing the ordered sequence of line seismograms.
<i>Create Initial Model</i>	Provides creating initial model with constant velocity layers with the help of Wireframe Editor and Velocity Grid Builder <a href="#">&gt;&gt;</a> .
<i>View Initial Mode</i>	Launches the graphic module Model Editor depicting model grid with cells filled with color matching velocity values <a href="#">&gt;&gt;</a> .
<i>Velocity Color Spectrum</i>	Runs Velocity Color Spectrum Manager for editing the default velocity color spectrum <a href="#">&gt;&gt;</a> .

<i>Create First Iteration</i>	Creates the first iteration enabling the user to begin the interpretation process.
-------------------------------	--

### Меню узла pre-study

Таблица 4.

<i>Command</i>	Description
<i>Run Pre-study Controller</i>	Launches pre-study head program
<i>Apply Pre-study Results</i>	Creates a new iteration with user name <i>After pre-study</i> and put in it the target of partial target model <a href="#">&gt;&gt;</a> .
<i>Help on Pre-study</i>	Opens the document chapter devoted to pre-study.

### Iteration menu

The first four commands starts servicing operations, the rest run operations making up an iteration.

Table 5.

<i>Command</i>	Description
<i>Create New Iteration</i>	Creates a new iteration by copying the current one. The new iteration gets its system number and becomes active. The current time and the source iteration number are added to the new iteration's user notes.
<i>Rename</i>	Allows creating or editing an iteration's user name, if it is free from running modules.
<i>Delete Iteration(s)</i>	Allows removing iterations with numbers in range [N1, N2], where N1 is the number of current iteration, while $N2 \geq N1$ and is defined by the user. By default, $N2 = N1$ . Iterations, busy with active modules are not deleted. An iteration is removed together with its folder.
<i>Display User Notes</i>	Displays the user notes relating to the current iteration.
<i>Edit Model</i>	Runs Model Editor for editing velocity and geometry (i.e. model wireframe) <a href="#">&gt;&gt;</a> .

<i>Bind Waves to Layers</i>	Runs Waver-Layer Binder to define waves formed at each (or some) layer's floor <a href="#">&gt;&gt;</a> .
<i>Forward Problem Task</i>	Runs Forward Problem Tasker for making up forward problem task, i.e. wave and source samples for which rays will be traced <a href="#">&gt;&gt;</a> .
<i>Solve Forward Problem</i>	Runs Forward Problem Solver implementing ray-tracing in accord with the task <a href="#">&gt;&gt;</a> .
<i>Forward Problem Log</i>	Allows viewing ray-tracing log to estimate a number of lost rays. <a href="#">&gt;&gt;</a> .
<i>View Ray Catalog</i>	Runs Ray Catalog Viewer for examination of the Ray Catalog database in the form of numeric tables <a href="#">&gt;&gt;</a> .
<i>View Ray Picture</i>	Runs Ray Picture Viewer for examination of the Ray Catalog database in graphic form, that is, as a ray picture <a href="#">&gt;&gt;</a> .
<i>View TX-Curves</i>	Runs Line Seismogram Viewer (LSV) for viewing TX-curves drawn over the seismograms. Only waves and seismograms defined by the forward problem task participate in the show <a href="#">&gt;&gt;</a> .

## Software

### Modules

This section helps better understand, how the XMF software functions. XMF includes the head program and more than 20 programs, performing essential or servicing tasks. They are called XMF *modules*. They are launched, mostly, by the head program, which, is an XMF controller and project manager. However, modules can launch other modules as well. For example, Model Editor runs Wireframe Editor.

A module works with the data of a certain node of the processing tree. Keeping that in mind, we use such phrases as "module works on a project node" or "module cannot be launched on this iteration" and the like. XMF allows running several modules on the same or different nodes, if no conflict occurs. Normally, only one module instance on a node is allowed, but there are exceptions.

## Task monitor

It would be a complicated job to track the state of active (i.e. running) modules, because they are launched in a random way by different launchers. But tracking is important for dealing with emergency situations that may lead to data loss. That is why module launch is carried out by a single program, a launch server, called XMF Task Monitor (TMR). The statement "program A runs module B" means that A sends a request to TMR, in which it identifies itself, work node and a module to launch". TMR either execute the request or rejects it. When the user double-clicks the XMF icon on desktop, TMR is started and its first automatic job is to run the head program. TMR stays active, while the user works with XMF, but its window is folded down to the icon on the Windows task bar. The session ends, when the user closes the head program's main window.

TMR supports XMF work, but the user had nothing to do with it, until he or she needs to get information on what is currently goes on, or kill a hanging module, or close an active module that behaves suspiciously. In that case, the user should click on the XMF icon on the task bar. TMR restores its window at the left bottom corner of the screen (fig. 1).

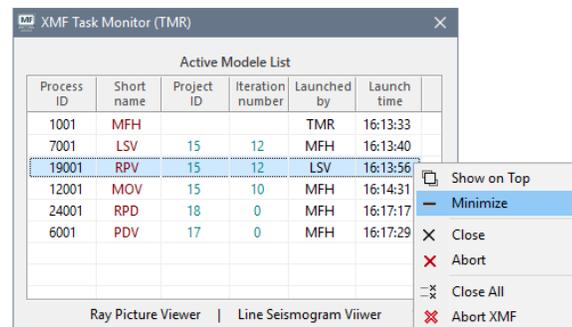


Fig. 1. TMR main window and context menu.

The window contains the list of active modules with the following columns:

- 1) internal active module ID;
- 2) module short name – a shortcut of the full name, which is displayed on the status bar at the moment the item gets selected;
- 3) – 4) work node address (project and iteration system numbers);
- 4) short name of the launcher;
- 5) launch time.

The first list item is always the same: head program (MFH) is launched by TMR at the session start. Permissible user actions are represented by the context menu commands. The first four relate to the selected item. When the *Abort* command is applied, a module is forcibly terminated with possible data loss. The last two commands relate to all active modules:

- *Close All* – close all active module in regular way (the head program stays active);
- *Abort XMF* – terminates XMF (all XMF programs are terminated forcibly).

When the user closes TMR window, the program gets minimized again staying active.

XMF is designed for use on a work station, not across the network. Work folders are selected using the local folder browser. Import/export operations can be carried out across the local network.

### XMF and XTomo-LM

The problem of building a layered model by kinematic analysis of a set of seismograms can be treated using both applications named in the topic caption. Clearly, better to use XTomo-LM with its exact inversion algorithms as ray tomography or reflected/head wave traveltimes inversion. However, exact methods require completeness of low-noise observed data. Additionally, one has to pay for exactness and efficiency by a slow procedure of picking times from the seismograms. On the other hand, XMF is applicable almost always, but "manual" model fitting is a laborious task even for a skilled user. At that, the problem itself is not simply ill-posed but has a very high degree of ambiguity. That's why joint inversion seems to be very helpful and even desirable.

XTomo-LM and XMF can be jointly applied for kinematic interpretation of a set of line seismograms in a certain class of layered models. Those model components that are well provided with observed data can be obtained using exact inversion algorithms of XTomo-LM, while the rest ones can be fitted approximately on the XMF platform. Joint use is illustrated by fig. 1.

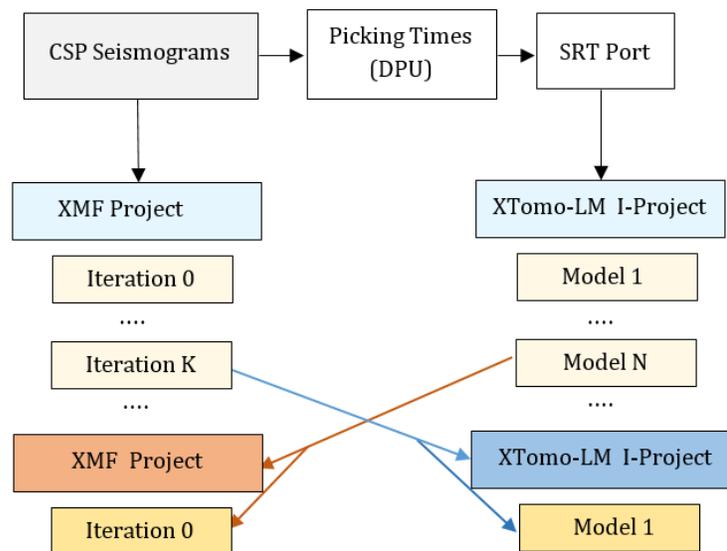


Fig. 1. Joint use of XTomo-LM and XMF.  
 DPU – XTomo-LM Data Preparation Unit, a separate application for picking times from seismograms.  
 SRT Port – the input XTomo-LM gateway; I-project – inversion project.

The exact meaning of the cross-arrows on the chart is the following:

- a model from each m-node of an XTomo-LM inversion project can be imported to the zero iteration of a newly created XMF project;
- when an XTomo-LM new inversion project is being created, its starting model can be imported from any iteration of an XMF project.

A common data source (a set of line seismograms) and the common target make both models close to each other even if it was not initially planned to use both applications. The XMF-XTomo-LM interface is not trivial, because, despite GUI similarity, the application architectures differ essentially. The details of XTomo-LM model import are given in the relevant [section](#) of the "User Interface" chapter.

## USER INTERFACE

### Creating Project. Data Input

[Work folder](#) – [Creating a project](#) – [Project properties](#) – [Creation a project as a copy of existing one](#) – [Positional data input](#) – [Viewing raw coordinate cache](#) – [Viewing Positional Database](#).

#### Work folder

At first run, with empty Processing Tree, XMF requires that the user define a work folder and displays a dialog, in which a folder is defined with the help of a local folder browser. A work folder is a container for XMF project folders. After it is fixed, the root node appears in tree, so the user can invoke the node menu to create the first project.

There may be several work folders. Their list is stored on disc and is loaded by the same dialog, so that the user can switch between different work folders at any time.

#### Creating a project

Before creating a project, the user should place project seismograms in a separate folder and summarize a priori information concerning initial model.

The *Create Project* command of the work folder node runs the module whose main window is shown on fig. 1a.

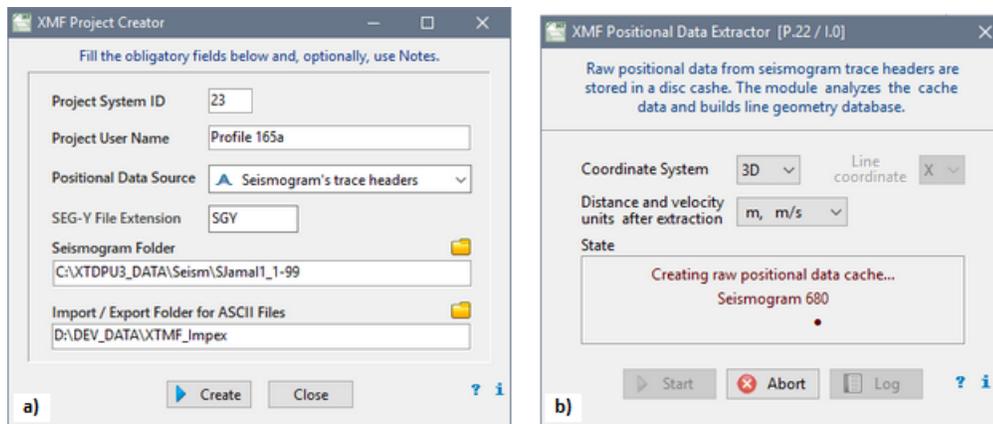


Fig. 1. Creating a project and positional data input.  
a) project properties; b) Positional Data extractor.

The top field is filled by XMF and contains project system number. The user enters project name in the next field keeping in mind that it is also a folder name, so that some characters are banned. Then he or she selects positional data source type and, if necessary, corrects the default seismic file extension. Further, two paths are to be

entered: to project seismograms and to import/export files. The latter must be reserved for text and graphic file that XMF exports and imports. Both folders must be outside any work folder. It might be useful to create a common import-export folder used by different projects. A click on the *Create* button completes the operation.

### Project properties

The data defined at create time are called project properties. They can viewed at any time by the *Properties* command of the project node menu. Only one property can be changed after creation, and this is project import/export folder.

### Creation a project by copying

The operation is executed by the *Create Project As Copy* command of the Work Folder node menu. It creates a new project *Q*, by copying a source-project *P* in the current or any other work folder, fully or partially. The command displays a dialog shown on fig. 2. There are two options:

- A. *Q* is an exact and full copy of project *P* with all its iterations;
- B. *Q* contains only zero iteration copied from *P*, in which initial model is replaced with the model from a specified iteration *I* of *P*.

Meaning of operation B is that one can continue work with iteration *I* in a new project (*Q*), in which its model serves as an initial model.

The user should:

- 1) select option A or B using radio-buttons;
- 2) select a work folder containing project *P*; if it is absent in the *Work Folder* drop-down list, the user can choose it in folder browser using the yellow folder button above;
- 3) select source-project *P* from the project list;
- 4) only in case of B: select iteration *I*, whose model will be the initial.
- 5) enter project *Q* name in the lower edit-field; by default, is the name of *Q* preceded with \$.

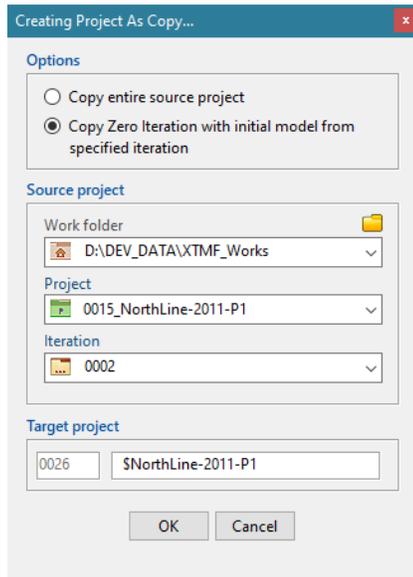
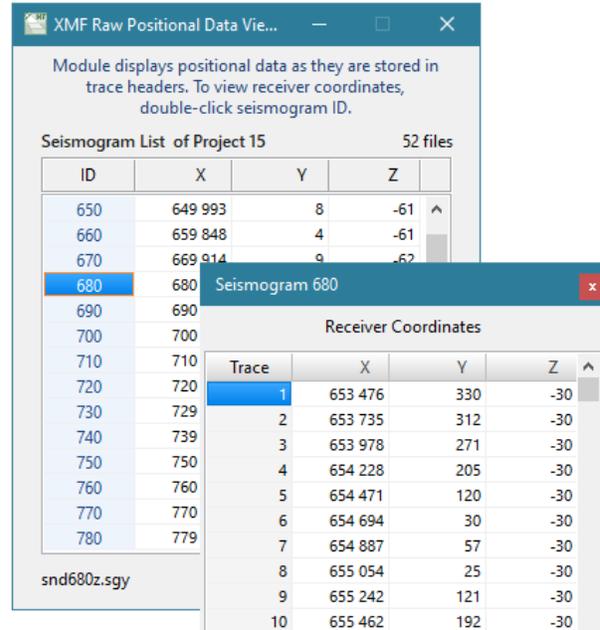


Fig. 2. Creation a project by copying.



Fog. 3. Viewing raw coordinate cache.

### Positional data input

Depending on the type of positional data source, the zero iteration menu' first command is either *Extract Positional Data* (from trace headers), or *Import Positional Data* (from SSR files). The commands run different modules with almost identical user interface. Fig. 1b shows the main window of the Positional Data Extractor module working with seismogram trace headers. What the user has to do is:

- 1) define field coordinate system dimension selecting "3D" or "2D" from the drop-down list;
- 2) if "2D" is selected, define which coordinate is a line coordinate: X or Y (i.e. from which trace header field it is extracted);
- 3) fix length unit, selecting from (conditional) meter or kilometer;
- 4) click on the *Start* button.

Positional Data Importer's task differs only in that step 2 is absent, because in the SRR files for two-dimensions the redundant coordinate is always Y. However, the "Y" column in file is obligatory; it can be filled with zeros.

If the user has selected "3D", while the actual dimension of the problem is 2, both modules work correctly, if only the redundant coordinate in Y column or trace headers is 0.

### Viewing raw coordinate cache

The *View Raw Positional Data* command of the zero iteration menu displays coordinates exactly as they are written in trace headers or SRR files. It launches a module whose main

window displays the seismogram list (fig. 2). The first column contains source IDs, the rest – source coordinates. A double-click on a source ID results in displaying the list of receivers which register a signal from the double-clicked source. Both lists support standard navigation methods: scroll bars and keys.

### Viewing Positional Database

Positional Database (PDB) is the main coordinate store and a source of coordinate information for all XMF modules. Database Viewer's main window is shown on fig. 3. The window displays two (independent) tables: *Source List* and *Receiver List*. Each line of both tables contains a device identifier, integer raw coordinates *X* and *Y* and real line coordinates *Line X* and *Line Z*. Raw *Z* is not shown: it differs from *Line Z* only by decimal point position. Below, special tools of the module are described.

Source List (52)					Receiver List (2094)					
Field ID	X	Y	Line X	Line Z	ID	X	Y	Line X	Line Z	Trace
780	779984	-2	129.9910	-0.075000	593	802783	11	152.7900	-0.030000	
790	789865	-12	139.8720	-0.080000	594	803035	5	153.0420	-0.030000	
800	800040	-6	150.0470	-0.085000	595	803286	0	153.2930	-0.030000	
810	809962	-16	159.9690	-0.086000	596	803542	3	153.5490	-0.030000	
820	819980	3	169.9870	-0.085000	597	803799	3	153.8060	-0.030000	
830	830006	9	180.0130	-0.086000	598	804064	3	154.0710	-0.030000	1
840	840098	2	190.1050	-0.086000	599	804409	11	154.4160	-0.030000	2
850	849894	0	199.9010	-0.091000	600	804672	19	154.6790	-0.030000	3
860	859912	19	209.9190	-0.109000	601	804948	30	154.9550	-0.030000	4
870	870115	0	220.1220	-0.125000	602	805202	36	155.2090	-0.030000	5
880	880012	7	230.0190	-0.142000	603	805447	33	155.4540	-0.030000	6
890	889920	19	239.9270	-0.154000	604	805690	26	155.6970	-0.030000	7

Fig. 3. Representation of Positional Database; a) normal mode; b) ray mode.

#### 1. Ray mode.

Let a line for source *S* is selected in the source list. After checking the *Ray mode* box, Receiver list acquires a new column called *Trace*, as on fig. 3b. The column contains ordinal numbers of all traces of seismogram *S* in the way that each trace number appears in the same line as the corresponding receiver. Thus, for each source *S*, the table on the right marks all receivers/traces that registers excitation of *S* (i.e. rays coming from *S*).

#### 2. Information on line coordinate system.

The *Geometry* button displays a dialog with information on transformation of raw coordinates to the line coordinate system:

- observation system area in raw coordinates;
- scaling factor (1 or 0.001);
- coefficients of the normal equation of the line;
- the origin of the line coordinate system (in raw coordinates);
- the unit vector of the line (in raw coordinates);
- observation system area in line coordinates;
- space resolution.

### 3. Export

The *Export* button exports the Positional Database content in the form of SRR files. The export files lie in a separate folder inside the project import/export folder with the default name "P.N\_PosData", where N is a project system number. Even if positional data had been imported from SRR files, the operation still has sense, because the exported files are verified and the RAY file is optimized.

## Viewing Seismograms

[Overview](#) – [Opening a seismogram. Viewing options](#) – [Frame and its tools](#) – [Selected frame and shared tools](#) – [Selected frame menu](#) – [The default settings](#).

### Overview

Line Seismogram Viewer (LSV) is a basic XMF module. It provides viewing line seismograms and also displaying computed traveltimes curves on them. In this section, only the first part of LSV functionality is discussed. Viewing seismograms is possible only after of a project's Positional database is built. To view seismograms (as well as any SEG-Y/PC file) outside XMF projects, the user runs Seismic Multi-Viewer, which is also included in XMF together with its own help file.

The main window of Line Seismogram Viewer (LSV) may look like on fig. 1.

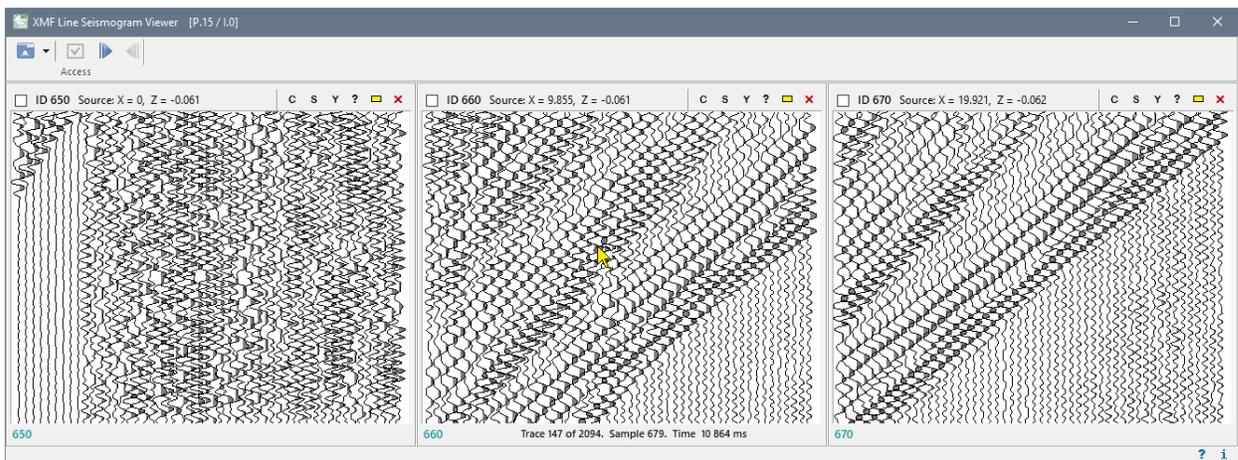


Fig. 1. Line Seismogram Viewer.

It shows that LSV can display several files at a time. Seismograms in LSV are always ordered by source x-coordinate. Each seismogram is displayed in its *frame* – a child window. Maximal number of frames depends on its width and screen width. A part of a seismic record within a frame is called a *record view*. A record view displayed at frame create time is called *record initial view*. Each view is identified with its *vertex* – the left-

bottom point in (trace #, sample #) coordinates. Frame width and initial view can be adjusted.

### Opening a seismogram. Viewing options

In a regular LSV mode, only the first section of the module's tool bar is visible. It is called the *access bar* and contains four buttons (fig. 1). The first button, when clicked on its icon, drops down the access menu with commands of opening seismograms. The first four menu commands are discussed in this section.

**1. Open seismogram.** The command displays a dialog with a list of seismogram IDs (fig. 1). After the user selects a seismogram, LSV adds (or, to secure the order, inserts) a new frame with the initial seismogram view in it. If there is no place for a new frame, the command is disabled. To open a new file, one of the existing frames must be closed. It is only this command that appends *the recent file list*. It is dropped down, if the user clicks on the small arrow of the first button.

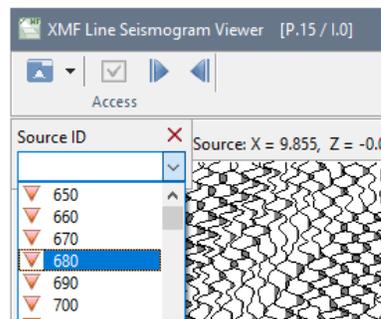


Fig. 2. Access bar and select dialog.

**2. Open Range [1,2,3,...].** The command switches on the *range mode*: LSV displays as many seismograms, starting from the first, as possible. Additionally, the first of the two scroll buttons on the access bar gets enabled. Suppose, seismograms ##1, 2, 3, 4 has been opened by the command. Then, a click on the first scroll button opens seismograms ## 2, 3, 4, 5. Thus, scroll buttons, indeed, allows scrolling seismograms along the seismic line. The range mode is switched off after the user does something that breaks required seismogram order, for example, closes a frame that is not the first or last.

**3. Open Range [N, N+1,...].** The command differs from the previous in that it displays seismograms starting from the *N*-th. The starting seismogram is selected as in **1**.

**4. Close All Frames.** The command closes all created frames.

The LSV main window can be resized or maximized. At that, sizing of frames is controlled by LSV, not the user.

## Frame tools

To view a seismic record, one needs tools for navigation and adjustment of trace display. The frame itself owns only some navigation tools. The rest instruments belong to the main window and are shared by all frames. In this section only frame's own tools are discussed. Some of them are on a small tool bar in the right part of the frame top panel. Its last button closes a frame. The other are explained below.

- 1. Current record point coordinates.** When the cursor moves over the record, the status bar shows trace number, sample number and time value of the record point. If the tool button, the one before the last, is downed, trace number and time value will be displayed also in the hint window near the moving cursor.
- 2. Navigation with the mouse wheel.** The mouse wheel scrolls the record vertically within the frame. If, at that, the Shift key is pressed, scrolling is horizontal. The scroll increment is small and fixed, therefore this way of scrolling is good for small and middle frame size. Otherwise, the screen update time becomes too great, and with it – the wheel response time.
- 3. Navigation by dragging a record point.** If the Ctrl key is pressed down, dragging a record point implies record scrolling in the direction of dragging. This is the most effective way of scrolling a record. It does not depend on frame size and allows moving in any direction, for example, along a phase lineups.
- 4. Emulation of Ctrl and Shift.** Instead of keeping a finger on those keys, LSV allows switch them on or off with the **C** and **S** buttons of the frame toolbar. If a button is downed the key is pressed, if it is released, so is the key.
- 5. File and trace properties.** The **Y** button of the frame toolbar displays a 4-page dialog, shown (partly) on fig. 3. Page names explain their content.

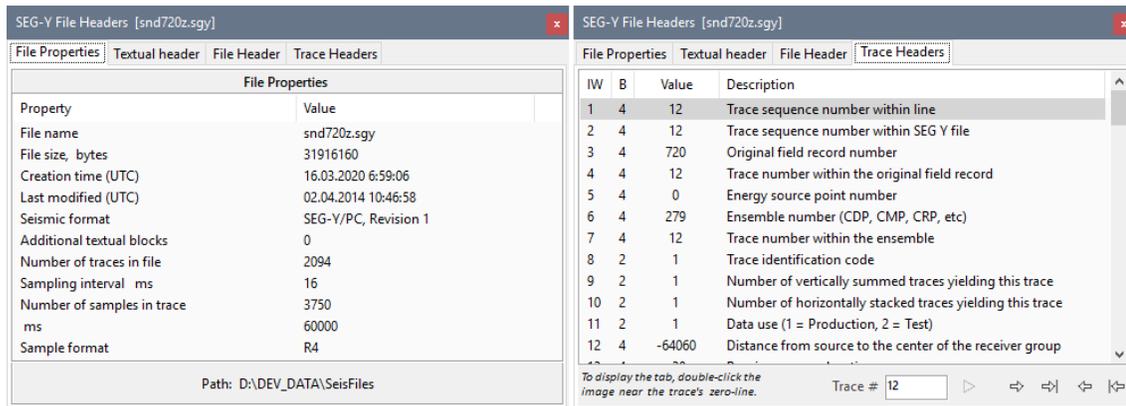


Fig. 3. SEG-Y file properties dialog: general properties page (left) and trace headers page (right).

Note that textual header is readable only if it uses the ASCII encoding. Binary headers are arranged by [information words](#) i.e. meaningful fields of different length. The 4-th page enables the user to navigate along the file and view any trace header, using the navigation panel at the page bottom. To jump to a required trace, the user enters its number into the edit field and clicks on nearest arrow button. It is enabled only if a number in the field differs from the current trace number. Other arrows allows moving to next, previous, last or first trace. In order to get quick access to trace header while viewing the record, double-click the required trace near its zero line.

**Selected frame and shared tools**

To make it clear for LSV, which frame is to be adjusted, a frame must selected. A check box at the frame top-left is just for that. Once it is checked, the entire LSV tool bar became visible (fig.4).



Fig. 4. LSV tool bar.

The tool bar is divided into sections. The buttons of the *Access* section, but one, were discussed above. The button with the check box icon will be explained later. The buttons of the *Navigation* section need not any comment, but, possibly, This button initiates moving the record to a view, specified in a dialog (fig. 5a). The user can choose to scroll the record to (1) initial view; or (2) a view starting from the trace with minimal offset; or (3) a view with a given vertex.

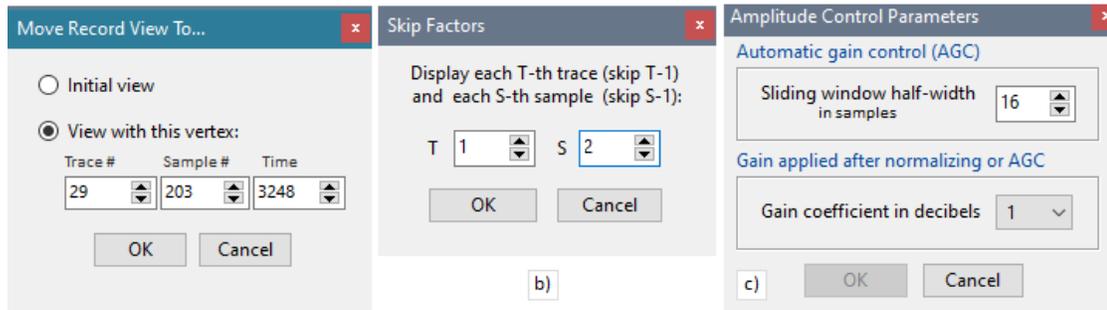


Fig. 5. Navigation and adjustment dialogs.

a) moving to a new record view b) setting skip values; c) setting AGC window half-width and gain.

The buttons of the *Rescaling* section allows changing view scale increasing or decreasing distance between traces or/and samples. The *Skip* button is used for displaying a record view with regular omission of traces or/and samples. The button drops down a menu with commands allowing skipping 0 to 2 traces or/and samples. The *Skip More* command displays a dialog, in which the user defines the omission value (fig. 5b). The *Reset* command cancels skipping.

The *Trace Display* section controls trace display. The the button drops down a menu with commands setting the following options:

- *Wiggle trace* – display with negative phases and filled positive phases (fig. 1);
- *Variable area trace* – displays only filled positive phases;
- *Upward* – time grows upward (fig. 1);
- *Downward* – time grows downward.

The first two and last two options are independent of each other.

The **A** button drops down amplitude control menu with the following commands:

- *Normalizing* – before a trace segment is drawn, all its samples are divided by maximal amplitude of the trace segment in the view. This is the default mode. Clear that, when shifted, a view may change in full, especially the noisy parts of traces;
- *AGC* – automatic gain control – a trace sample at (any) time  $t$  is normalized by average amplitude in  $[t-h, t+h]$ ;  $h$  is AGC window half-width;
- *Parameters* – opens a dialog shown on fig. 5c, in which the user can adjust AGC window half-width and gain coefficient; gain is applied after normalizing or AGC; 0 decibels is the default value;
- *Reset* – sets the default signal dynamic parameters, as on the initial view.

The last button **R** is a switch of seismogram *reducing*; the reduction velocity is taken from the LSV default settings.

### Selected frame menu

When there is a selected frame, the access section tool button  gets enabled. It drops down a menu with the following commands:

- *Open Next (Previous) Seismogram* – opens a new frame with the neighboring (in the sense of seismogram ordering) seismogram, if there is enough space on screen;
- *Close Frame* – closes the selected frame;
- *Synchronize Record Positions* – results in that all visible frames display views with the same vertex as the selected frame;
- *Make Record Position Initial* – makes the current view of the selected frame its initial view, or all frames' initial view .
- *Make Frame Settings Default* – makes the selected frame view options the default settings (see below) and updates the existing frames.

### The default settings

Default view options are those applied just after opening a seismogram. They can be changed in a dialog (fig.6) that is invoked by the *Default View Settings* command of the access menu.

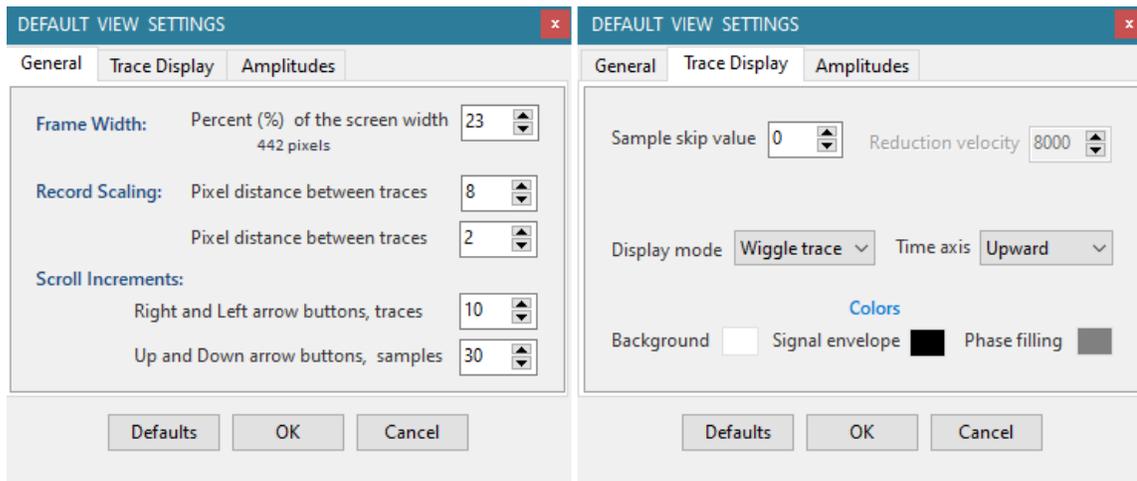


Fig. 6. Dialog *Default Settings*, tabs 1 and 2.

View settings are distributed between three tabs shown oh fog. 6. The *Frame Width* parameter defines frame width and, hence, the view of the entire main widow. It is measured in percents to the screen width. It is always used by the *Open Range* commands displaying a sequence of seismograms. The rest of settings are already discussed in the previous sections. The *Sample skip value* defines how many samples are skipped over when displaying a trace. Actually, it allows increasing signal sampling interval. A click on the *OK* button saves the settings and repaints the existing frames with new parameters. The *Defaults* button replaces all parameters with "factory settings".

Default view settings are automatically saved to the project folder; therefore, they are adjusted to project seismic records.

## Initial Model: Creating

### The user's actions

Creating the initial model is the third step of the interpretation [workflow](#). The positional database is already created. The further user actions are:

- 1) in the head program's zero iteration menu select *Create Initial Model*; the command starts Wireframe Editor (WED), which immediately displays the *Model Domain and X-Grid Definition* dialog;
- 2) In the dialog, the user defines the rectangular area, which surely contains the model, selects a number of [wireframe x-net](#) nodes and closes the dialog with *OK*;
- 3) back in WED, the user draws the initial wireframe curves and clicks on the  tool button, which results in displaying the *Layer Velocities* dialog;
- 4) in this dialog, the user defines a value of layer velocity for each layer and closes the dialog with *OK*;
- 5) WED launches Velocity Grid Builder and after it terminates (its work takes a moment), closes: the initial model is built; to view it, the user runs Model Viewer from the same head program's menu.

Work with WED is described in details in the next [topic](#). Here we dwell on the steps 2) and 4).

### Model domain and wireframe x-net

What will follow is based on the section [Model wireframe](#). The *Model Domain and X-Grid Definition* dialog is shown on fig. 1a.

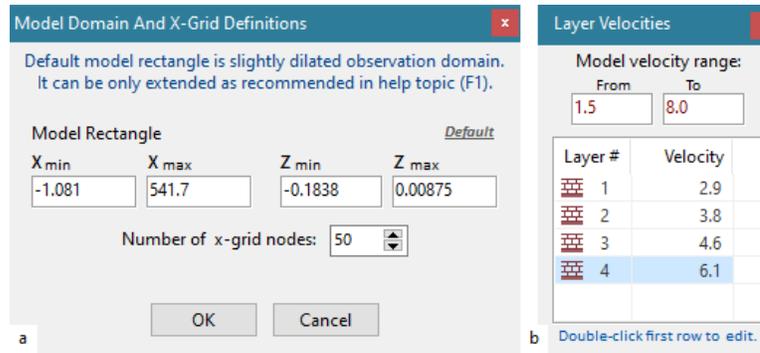


Fig. 1. Creating initial model.  
 a) model rectangle definition; b) layer velocities definition.

When it is displayed, the edit fields under *Model Rectangle* describe *observation system area* – minimal rectangle (in line coordinates) containing all sources and receivers (shortly, devices). The user's task is to define a rectangle, containing a model area. The range  $[X_{min}, X_{max}]$  must be slightly widen in order that devices were separated from the borders. The reserve must exceed the resolution, as minimum. The same relates to  $Z_{max}$  and, additionally, the upper bound must be easily distinguished visually from the daylight surface on the WED drawing pad. Finally,  $Z_{min}$  – the lower model border – must be selected so that the additional layer is well distinguished.

Now the *Number of x-grid nodes* field must set. The wireframe will be later edited on the x-net defined now, independent of the velocity grid density. The values defined in the dialog cannot be changed later within the current project.

**Layer velocities. Column editor**

The *Layer Velocities* dialog (fig. 1b) contains edit fields for model velocity range and a table of layer velocities (a number of layered is already known). In the course of interpretation, all velocity values must fall inside the range defined here, so enter velocity bounds with some reserve. The dialog is displayed with the same default value assigned to all layer velocities. To change them, use table *column editor* explained in the next section.

The column editor is started by a double-click on a column cell. An edit field appears over the cell, as on the snapshot. The user types in the value and then moves the field up or down the column with arrow-keys. Table 1 contains means of control.

Таблица 1.

Action	Effect
Double-click on an a cell or F2 on selected cell	Starts column editor. Below, the editor is regarded active.

Esc or double-click	Exit with canceling of changes.
Enter	Exit with saving changes.
Down and Up arrows	Move to the next or previous cell after saving changes. If the current cell is the last or first one exit the editor.

If the value typed in is not valid, the editor issues a sound and waits for the user's action. Layered velocities can be entered at one pass of the table. A click on *OK* closes the dialog.

### Initial Model: Import

Initial model can be imported from an m-node of an XTomo-LM project. About when such import makes sense and even is necessary see the "XMF and XTomo-LM" [section](#). The *Initial Model|Import from XTomo-LM* command of the zero iteration menu starts module XTI (XTomo-LM Model Importer). Before explaining the module's user, consider some details of the problem.

#### Discrepancy in model representations

Implanting an XTomo-LM model into XMF is, generally, impossible without some distortion of the former. First, in XTomo-LM, velocity grid is constructed and edited by the user, while in XMF it is stretched automatically over the wireframe. In particular, in XMF, grid verticals are equidistant, while in XTomo-LM they are, generally, not. And this is the first reason of distortion. The second reason relates to location of sources and receivers (devices, for short). In XTomo-LM there is an option of using a part of devices, which allows reducing model domain. Thus, XMF devices may prove to be beyond a model to import. Of course, if joint use of both applications is planned from the beginning such actions should be avoided. Finally, note the wave encoding in both applications is entirely different.

### XTI output

Let  $M$  be an XTomo-LM model to import,  $M^*$  is an XMF model generated by XTI. The following statements give an idea of how XTI works.

- If (1) verticals of  $M$  grid are equidistant and (2) devices of XMF project fall inside  $M$ , then  $M^* = M$ , i.e. have identical grids.
- If condition (1) is not fulfilled,  $M^*$  grid is obtained from  $M$  grid by repositioning the latter's verticals in the way to make them equidistant. Therefore, in this case velocity gets distorted horizontally.
- If condition (2) is not satisfied,  $M^*$  grid is obtained from  $M$  grid by adding some columns of the same width or/and moving the first h-line up in the way to secure that all XMF project devices are inside  $M^*$ .

Obviously, if (1) takes place,  $M$  domain is a subset of  $M^*$  domain and their grids are identical, possibly, except for the first h-line.

### XTI user interface

XTI main window is shown on fig. 1. The user's duty is to locate an XTomo-LM model to import with the following actions:

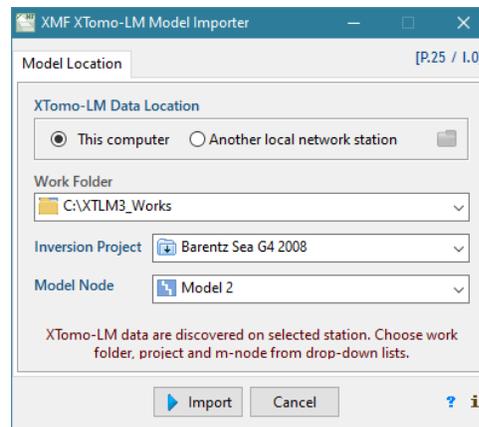


Рис. 1. Импорт модели из XTomo-LM .

- 1) point whether the model sits locally or on another node of the local network by clicking a radio-button;
- 2) only in case of network location: in Windows folder browser, select the *ProgramData* on the target network station;
- 3) select XTomo-LM work folder from the drop-down list *Work Folder*;
- 4) select inversion project containing the the model;
- 5) select m-node with the model to import;
- 6) click on the *Import* button.

Additionally, XTI imports XTomo-LM project wave list. Only monotype waves are imported because conversion flags are included in XMF wave code, while in XTomo-LM, conversion flags are defined only in the forward problem task. Wave drawing attributes in both application are different. Wave color is kept intact, while curve styles may be distorted.

## Wireframe Editor

[Adding a curve](#) – [Current curve](#) – [Resolution error](#) – [Pointwise editing](#) – [Segments](#) – [Numeric Editing](#) – [Curve operations](#) – [Parameters of operations](#) – [Curve drawing attributes](#) – [Keyboard shortcuts](#).

The graphic module Wireframe Editor (WED) is started either by the command of creation initial model or by *Edit Geometry* command of Model Editor. WED's main window (fig. 1) main part is drawing pad for wireframe curves. Its surface looks like rectangle grid with square cells. Cell size is defined by the grid parameter – *number of x-net nodes*. or *x-net step*. In the initial model mode the parameter is defined by the user as explained [here](#). When model geometry is being edited, number of x-net nodes is equal to a number of model grid verticals. X-net step is shown below the drawing pad. Vertical grid step, shown on the left margin, is variable; its value depends on x-net step and window size.

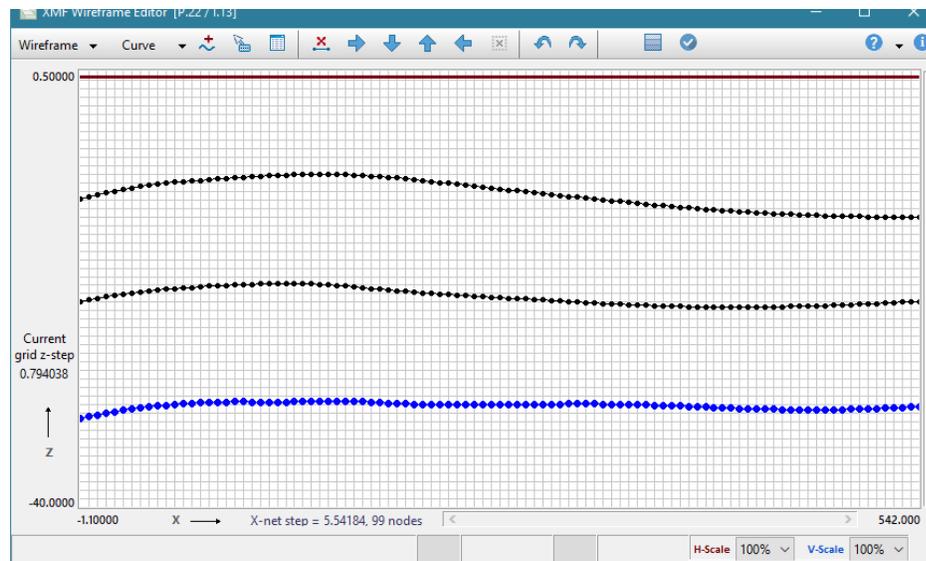


Fig. 1. WED Main window.

The status bar at window's bottom displays hints for buttons and menus and the cursor context information: nearest vertical number counting from the left border; x-coordinate; nearest horizontal number counting from the top one; z-coordinate.

The user can increase horizontal and vertical scales. Scale is selected in the drop-down lists *H-scale* and *V-scale* at right side of the status bar. The magnified image can be scrolled

within the drawing pad with scroll-bars, mouse wheel and dragging the pad (see [Keyboard shortcuts](#)). WED support undo-redo stack of small length.

All edit commands are associated with tool bar buttons, two of which – *Wireframe* и *Curve* – drop down menus. The top wireframe curve is daylight surface line (DSL). It is built automatically (see [here](#) how) and cannot be changed.

### Adding a curve

The button  displays a message demanding that the user double-click a point through which the curve will pass. After the message box is closed, WED waits a double-click for some time, with the button still downed. After the user double-clicks on a pad point, WED draws the default curve to be edited later.

### Current curve. Resolution error

The curve just added becomes the *current* or *selected*. It is drawn in special color (blue, by default). All edit operations have this curve as their object. To make a curve current (and hence, editable), the user should capture any segment of it with the rubber-band and click on the *Select Curve to Edit* command of the rubber-band menu.

The wireframe curves are not allowed to intersect each other and, moreover, they are banned to approach each other too close. More precisely, on each vertical, the distance between points of the adjacent curves must exceed  $\alpha \cdot R$ , where  $R$  is [resolution](#) value,  $\alpha \leq 10$ , and is defined by the program's settings. If, while editing a curve, the user comes too close to another curve, he or she gets the error message "The point is beyond curve permissible area".

### Pointwise editing

The points of a wireframe curve lie on verticals passing through the nodes of the x-net. The main edit tools are the the following:

- a double-click on the drawing pad creates a point of the current curve on the nearest vertical; if there was a point on the vertical, it is replaced with the new one;
- a double-click on the drawing pad with the Shift key pressed deletes a point of the current curve from the nearest vertical.

Using those tools, the user can quickly draw a new curve in this way:

1. add a new curve to the wireframe (fig. 1a);
2. remove all points, but the first and last ones, from the default curve by clicking on the  button (fig. 1b).
3. fix some curve breaking points by double-clicking those points on the drawing pad (fig. 1c).

4. define the curve on the other verticals using linear or spline interpolation (fig. 1d); for that, apply the Curve/Interpolation command. Done.

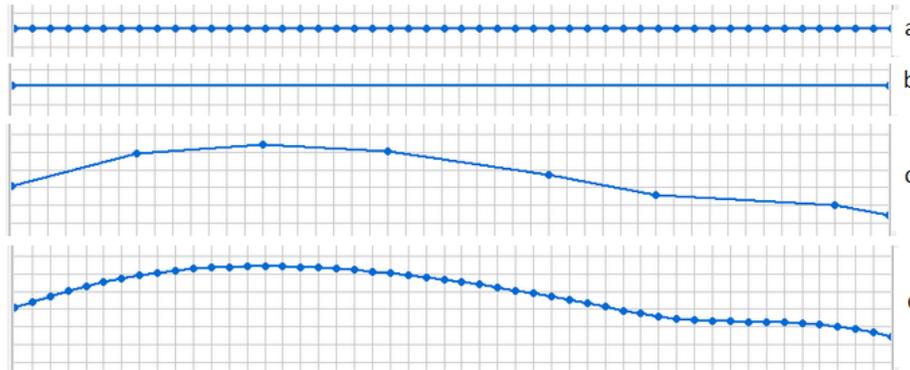


Fig. 2. Quick way of drawing a curve.

- a) default curve;
- b) all curve points but the first and the are removed;
- c) curve sketch
- d) final view after spline interpolation.

Curve Equation		
Selected: # 1    Top		
#	X	Z
1	-1.10000	-11.1812
2	1.67092	-11.1812
3	4.44184	-11.1812
4	7.21275	-11.1812
5	9.98367	-11.1812
6	12.7546	-11.2543
7	15.5255	-11.3273
8	18.2964	-11.4004
9	21.0673	-11.4735

Double-click or press F2 to edit Z.  
 Show point on the image

Fig. 3. Curve equation.

With pointwise editing, the user can change every point of an existing curve. The real coordinates of the cursor, moving over the drawing pad, are displayed on status bar, so that editing can be performed accurately.

### Numeric editing

The  button displays a floating window with current curve tabular equation (fig. 2). The table columns are: x-net node number, point x- and z-coordinates. Additionally to the current curve, the window can display the daylight surface line which cannot be made current. The curve switch is represented by a couple of radio0-buttons above the table. Points that have been deleted have empty cells in the Z column. If the *Show point on the image* box is checked, the curve point matching the item selected in the table is pointed to with a small red arrow on the drawing pad. Numeric editing means changing z-coordinate values with the [column editor](#), which is started by a double-click on a Z-cell or pressing the F2 key. The program blocks changes which imply breaking a curve's permissible area.

### Segments

Some edit operations can be applied both to a curve and its *segments*. For that, it must be selected. To do that, capture it by the rubber-band and issue the *Select Segment* command of the rubber-band menu. The segment will be redrawn in selection color (red, by default) and becomes an object of an operation. A segment can consist of one point only. In the segment mode, the commands, which are not applicable to segments, get disabled. Pointwise editing does not work too: a double-click results in deselecting a segment.

## Curve operations

Table 1.

Button or menu command	Operation
	Adds the default curve to the wireframe.
	Shows/hides the hint window with cursor coordinates.
	Displays a floating widow with the tabular curve equation which allows numeric changing of z-coordinate of any curve point. >>
	If there is a selected segment, deletes its points from the curve. Otherwise, deletes all curve points but the first and last one.
	Moves vertically a selected segment or the entire curve. The shift value at one click is the operation parameter.
	Moves the curve horizontally. The shift value at one click is the operation parameter. When moved to the right, the rightmost points are lost, while new points at the beginning are the copies of the first point before the operation. Left shift is similar.
	Kills segment selection; same as double-click on the drawing pad.
	Undo/Redo operations.
	Allows model previewing when in "Edit Geometry" mode.
	Ends editing and saves the result.
	<i>The Curve menu</i>
<i>Interpolation</i>	Reconstructs absent points using linear (Ctrl+I) or spline interpolation. The user cannot change curve or exit until the curve is defined at all nodes of the x-net.
<i>Approximation</i>	Replaces the current curve, possibly, defined on some nodes only, with a cubic spline approximating the curve in the sense of least squares. A number of a spline equidistant conjugations is the operation parameter. Applied to the entire curve only.
<i>Smoothing by</i>	Smooths a curve or a segment with sliding average or median. The sliding widow half-width is an operation parameters. Applied both to a curve and a segment.

<i>Parameters</i>	Displays a floating window with parameters of operations(Ctrl+P) >>>.
	The <i>Wireframe</i> menu
<i>Import model Top</i>	Allows importing a curve from an <a href="#">MG</a> file to replace the default DSL, but only if the curve is located above DSL and inside the model rectangle. Because the MG format admits several curves in one file, the user is supposed to enter the curve number in the file.
<i>Import interface Curve</i>	Allows importing a curve from an <a href="#">MG</a> file to become one of wireframe curves. Import is possible if a curve does not intersect the other curves and does not generate resolution error.
<i>Remove Curve</i>	Deletes the current curve from the wireframe. Do not mix with the  operation which deletes curve points, but not the curve itself.
<i>Clear</i>	Deletes all curves but DSL.
<i>Export</i>	Saves the entire wireframe, but DSL to a binary file.
<i>Import</i>	Imports an earlier saved wireframe, if only its domain is contained in the current one.
<i>Drawing Attributes</i>	Invokes a dialog for changing drawing parameters >>>.

**Parameters of operations**

Access to parameters of operations is provided with the the *Curve/Parameters* command. Fig. 2a shows the window.

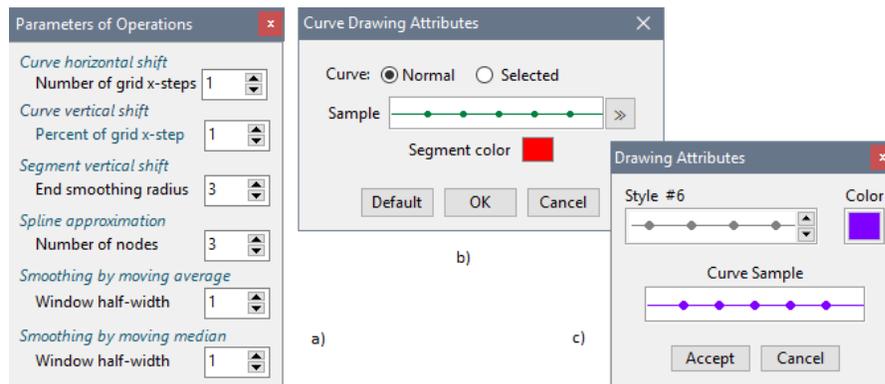


Fig. 4. Editor parameters and settings: a) operation parameters; b) curve drawing attributes; c) standard dialog for selecting styles and colors.

It is floating and can on top of drawing pad all the time; changing parameters in it does not require confirmation.

Table 2. Parameters of operations.

<i>Operation</i>	Parameter
<i>Curve horizontal shift</i>	Value of shift at one click in a number of x-net steps; 1 is the default.
<i>Curve Vertical shift</i>	Value of shift at one click in percents of x-net step; 1% is the default.
<i>Segment Vertical Shift</i>	Additional to the previous: number of points used for smoothing the curve near the segment ends.
<i>Polynomial approximation</i>	Polynomial degree; must be less than a number of curve points.
<i>Spline approximation</i>	A number of spline conjugation nodes; must be less than a number of curve points minus 3.
<i>Smoothing by moving average</i>	Half-width of sliding window in number of curve points.
<i>Smoothing by moving median</i>	Half-width of sliding window in number of curve points.

### Curve drawing attributes

The notion of *drawing attributes* is used in all graphic modules which draw curves. There are two such attributes: *style* and *color*. For changing drawing attributes, the standard dialog is invoked (fig. 4c). It contains the *Style #N* field and a square filled with the current color, where *N* is the style number in the fixed list of styles. A style is represented by an image of a segment in gray color. To change for another style, use the spin-button that scrolls the style list within the field. To change color, click on the square to display the Windows Color dialog, in which a new color is to be selected. At the dialog bottom a curve segment with selected drawing attributes is shown. A click on the *Accept* saves the attributes.

In WED, the current drawing attributes can be displayed by the *Wireframe/Drawing Attributes* command (fig. 4b). There are three kinds of curves: normal, current and a segment. For the latter, only color should be defined by clicking on the colored square (red, by default). A sample of curve is shown in the *Sample* field. The view depends on which radio-button at the top is checked. To change attributes of each curve, click on the double-arrow button on the right. It invokes the standard dialog discussed above.

**Горячие клавиши**

Combination	Command
Ctrl + ↑	Increment vertical scale.
Ctrl + ↓	Decrement vertical scale.
Ctrl + →	Increment horizontal scale.
Ctrl + ←	Decrement horizontal .
Ctrl +Z	Cancel zoom.
Ctrl + I	Reconstruct curve by linear interpolation.
Ctrl +P	Display operation parameter window.
Ctrl +S	Cancel segment.
Ctrl +T	Displays curve equation as a numeric table.
double-click	Add/Replace curve point on the nearest vertical.
Shift + double-click	Remove curve point from the nearest vertical.
Shift + mouse wheel	Scroll magnified image horizontally.
Ctrl + Drag	Scroll magnified image in the direction of dragging.

**Binding Waves to Layers**

Binding waves to model horizons (or layers) is a part of [iteration](#). Fig.1 shows snapshots of all windows and menus of Wave-Layer Binder (WLB). The module is launched from the iteration menu of Processing Tree.

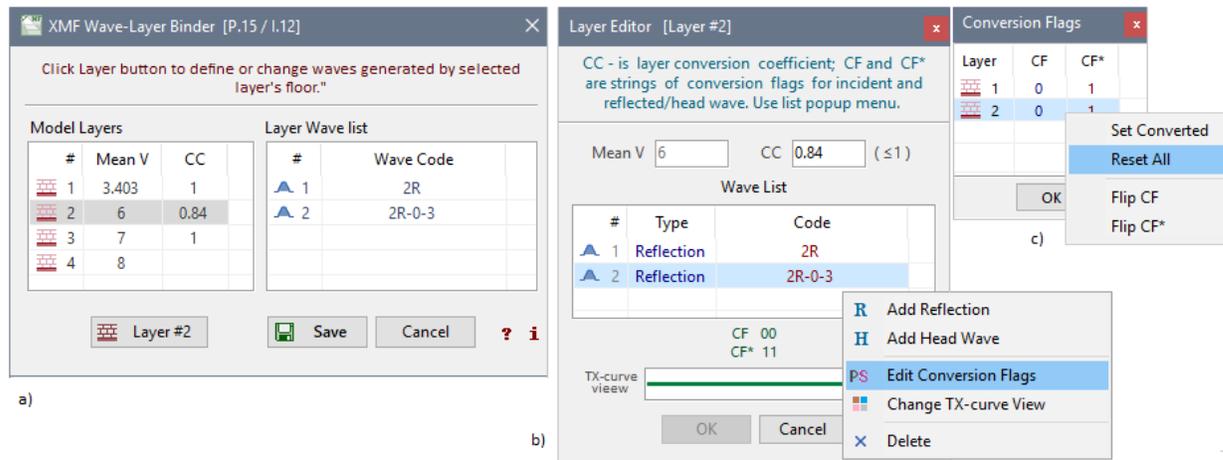


Fig. 1. WLB windows and menus; a) main window; b) Layer Editor dialog with its operational menu; c) Conversion Flags dialog with its menu.

The main window (fig. 1a) contains the layer list with average velocities and [conversion coefficients](#) (on the left) and wave list with [wave codes](#) (on the right). The latter is dependent on the former in the following way: it contains waves bound to a selected layer. The *Layer* button displays the *Layer Editor* dialog, in which binding is, actually, carried out. After binding or editing, the user clicks on *Save* to save changes.

The *Layer Editor* dialog (fig. 2b) contains a list of waves bounded to the current layer and the *CC* edit field for changing the layer conversion coefficient. By default,  $CC = 1$ , there are converted waves propagating through the layer, it must be changed to represent a ratio of S- and P-wave and, hence, be less than 1. The *TX-curve view* field shows a selected wave TX-curve sample. TX-curve color is also the color of wave rays. Under the list, the binary strings of [conversion flags](#) of the selected wave are shown: *CF* – for incident and *CF\** – for secondary wave. For P-wave the strings consist of zeros.

The popup list menu offers the following commands:

- *Add Reflection* – adds a reflection;
- *Add Head Wave* – adds a head wave;
- *Edit Conversion Flags* – changes the selected wave's conversion flags; invokes the dialog discussed below;
- *Change TX-Curve View* – change the selected wave's [drawing attributes](#); it is not necessary to assign drawing attribute at this point: it may be done later;
- *Delete* – remove the selected wave from the list.

The conversion flags dialog (fig. 1c) allows editing the conversion flags *CF* and *CF\** for each layer from top down to the current. Flags are shown in the table "layer number – *CF* – *CF\**". The user changes *CF* and *CF\** using the table context menu with the commands:

- *Set Converted* – defines one particular converted wave – the reflected or head S-wave formed at the floor of the current layer;
- *Reset All* – nullifies all flags;
- *Flip CF, Flip CF\** – each command changes its flag for the opposite in the selected table line.

## Viewing Model

[Overview](#) – [Main menu](#) – [Context menus](#) – [More about zoom and selecting](#) – [Selected column and vertical profile](#) – [Model numeric view](#) – [Sources and receivers](#). [Hint window](#) – [View options](#) – [Text export](#) – [Graphic export](#).

### Overview

In XMF graphic modules, the model is depicted in the form that is convenient for editing: as a grid whose cells are filled with color matching a certain velocity value interval. A set of those intervals makes up the *color velocity spectrum*. XMF uses the default spectrum, but the user can change it or create its own from scratch in special program.

The three key XMF modules: Model Viewer, Model Editor and Ray Picture Viewer are based on the above visual model representation. This topic is devoted to the first and the simplest of the three – Model Viewer (MOV). It is launched on any project iteration, including the zero one, but excluding the active iteration, on which Model Editor runs instead.

The main module window (fig. 1) contains three images: (1) model domain with grid and horizons; (2) velocity vertical profile in the column under the cursor; and (3) velocity color spectrum. On the snapshot, the latter is in the *continuous* mode: velocity jumps are smoothed. All images are interactive. The spectrum modes are switched in its context menu.

In the status bar at the bottom, additionally to tips for menu commands and buttons, the cursor context information is displayed: real coordinates, grid column and row numbers, cell velocity value. Note that on the snapshot, the grid is not uniform: its density is very high at top and drops with depth. Besides the main menu and toolbar, the program controls include the drawing pad context menu, the rubber band menu, the spectrum and vertical profile menus.

The rubber-band is used for zoom and selection of grid subsets. Selected grid subsets are painted in special color (light-red, by default). Though selecting is intended for editing velocity, its mechanism is built in MOV.

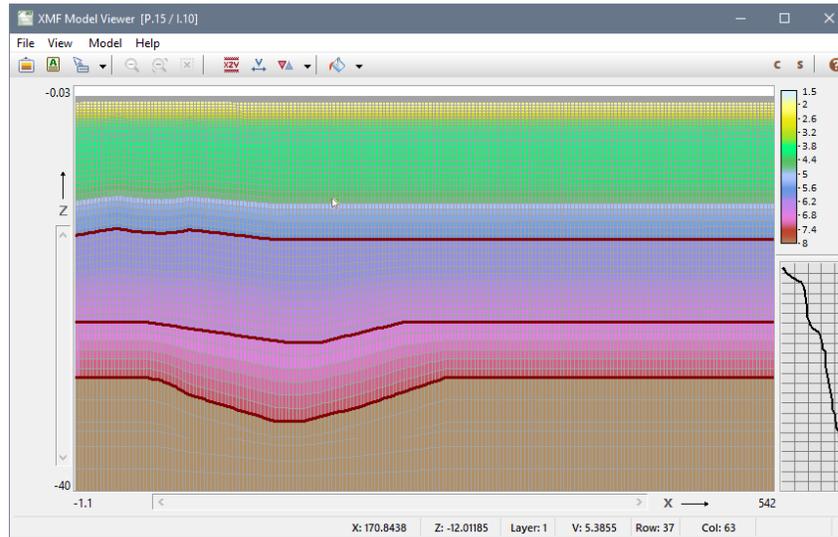


Fig. 1. Model Viewer's main window .

In the below tables, all MOV menu commands are listed with short descriptions. The >> link points to detailed comments. Most tool bar buttons repeat menu commands, but the two ones: they will be regarded in a proper place.

**Main menu**

Table 1. Main menu commands.

Command	Description
	File menu
<i>Reload Model</i>	Reloads model file.
<i>Export Model Image</i>	Exports scaled model image, optionally, provided with axis and captions, in a graphic file >>.
<i>Export Model to ASCII File</i>	Exports model's components to text files >>.
	View menu
<i>Velocity Color Spectrum</i>	Shows/hides velocity color spectrum. If the spectrum is hidden, vertical velocity profile takes up the entire right margin.
<i>Right Margin</i>	Shows/hides the right margin of the model image.
<i>Update View</i>	Redraws the three images of the MOV main window.

	Model menu
<i>Grid Metric Information</i>	Displays numeric information on the grid.
<i>Model Numeric View</i>	Displays dialog with numeric model representation <a href="#">&gt;&gt;</a> .
<i>Velocity Row Slice</i>	Displays dialog which allows viewing the curves $V_{row}(x)$ for any grid row.
<i>Model View Settings</i>	Displays a dialog with model, sources and receivers view options. <a href="#">&gt;&gt;</a> .

### Context menus

Table 2. Context menu commands

<i>Commands</i>	Description
	Rubber-band menu
<i>Magnify</i>	Stretches out the model subset within the rubber-band up to the entire drawing pad. At that, scrolling tools become active: scroll bars, mouse wheel (with Shift pressed for horizontal scrolling) and dragging the image with Ctrl downed. Magnifying can be iterated. <a href="#">&gt;&gt;</a> .
<i>Magnify Horizontally</i>	The same with vertical zoom blocked.
<i>Magnify Vertically</i>	The same with horizontal zoom blocked.
<i>Select Subgrid</i>	Selects the maximal subgrid contained strictly inside the rubber-band. Any selection is canceled by a double-click on the model image <a href="#">&gt;&gt;</a> .
<i>Select Horizontal Strip</i>	Selects the horizontal strip defined by the left rubber-band edge.
<i>Select Vertical Strip</i>	Selects vertical strip lying strictly between the side edges of the rubber-band.
	Image menu in the absence of selection
<i>Zoom out</i>	Cancel magnifying. The same does the  tool bar button; the  button does a step back in the sequence of successive

	magnifications.
<i>Select Cell</i>	Selects a cell right-clicked for the menu. A double-click on a cell does the same.
<i>Select Column</i>	Selects a column right-clicked for the menu.
<i>Select Row</i>	Selects a row right-clicked for the menu.
<i>Select Layer</i>	Selects a model layer right-clicked for the menu.
<i>Select All</i>	Selects the entire grid.
<i>Select in Dialog</i>	Selects a grid subset in a dialog <a href="#">&gt;&gt;</a> .
	Image menu in the presence of selection
<i>Unselect</i>	Kills whatever is currently selected.
<i>Velocity in Selection</i>	Displays mean, minimal and maximal velocity values in the selected subset.
<i>Select in Dialog</i>	Selects a grid subset in a dialog <a href="#">&gt;&gt;</a> .
<i>Selected Horizon As Table</i>	Accessible if a horizon is selected (see the next section). Displays a dialog in which the horizon is represented as a numeric table "N-X-Z", where N is a column number (= vertical's number). If a box under the table is checked, the horizon point, matching the selected table item, is pointed to by an arrow.
	Spectrum panel menu
<i>Line Spectrum Continuous</i>	Spectrum mode switch. There are two modes: line (discrete) and continuous.
<i>Reload</i>	Reloads the spectrum from the file. Used when the user wish to update spectrum which was changed during the MOV session.
<i>Hide</i>	Hides the spectrum.

**More about zoom and selecting**

A new selection can be done only after the previous is killed. A horizon is regarded as selected if the row, for which the horizon is a roof, is selected. Column and row ranges of any selection are shown under the model image. Some subsets can be selected with hot keys (Table 3).

Table 3 . Alternative ways of zoom out and selecting.

Operation	Shortcut
Kill magnifying	Ctrl + Z
Scrolling	Ctrl + dragging an image point in the required direction.
Select a cell	Double-click on the cell.
Select a column	Ctrl + double click on the column.
Select a row	Shift + double-click on the row.
Unselect all	A double-click on the image.
Select subgrid	1. A double-click on the top left cell (it will be selected). 2. Ctrl + a double-click on the bottom-right cell.

Instead of pressing down Ctrl or Shift, the user can down the **C** and **S** buttons in the end of the tool bar.

Sometimes, numbers of columns and rows bounding a subgrid to select are known. In that case, it is more convenient to use the *Select in Dialog* command of the image context menu. The dialog, invoked by it, is shown on fig. 2a.

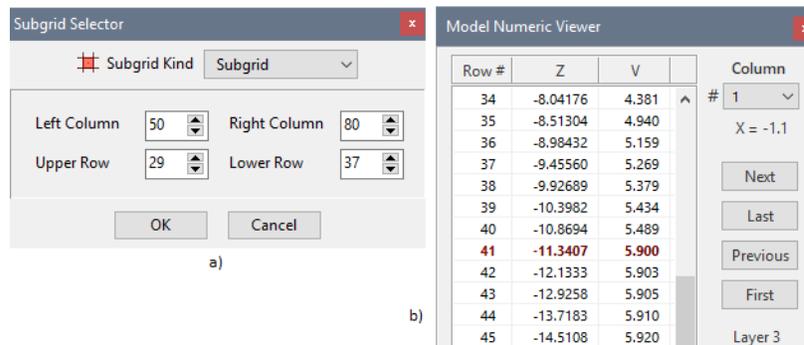


Fig. 2.MOV dialogs; a) select dialog; b) model numeric view.

The user can select any standard grid subset picking its kind from the drop-down list at the top. What is left is to set the number of bounding columns and rows in the spin-edit

fields. If before the call, there was a selection (a small red icon at the top is visible), the dialog field initial values are defined by the existing selection.

**Selected column and vertical profile**

Velocity vertical profile on the right model image margin, changes in step with the moving cursor. However, if a column is selected, the profile gets frozen displaying the selected column only. The cursor can move over the profile drawing pad, showing the coordinates *V* and *Z* in the hint window. Moreover, a popup menu is activated in this mode, allowing to select a way of drawing the profile: as step curve or as piecewise linear curve.

**Model numeric view**

The dialog *Model Numeric Viewer* (fig. 2b) represents the model as a sequence of velocity columns. The dialog window is a floating one, letting the user switch to the main window and back.. Each column is displayed as a table "Row number – *Z* – *V*". Viewing begins with the first column. Some rows are typed in bold: they underlie the model horizons (wireframe curves). The tools under the *Column* heading implement navigation over the model. To move to a new column, the user selects its number from the drop-down list. To move to the next, previous, first and last columns, the user clicks on the matching button.

**Sources and receivers. Hint window**

The  tool button and its menu allow displaying sources and receivers over the model image as colored triangles: the one standing on its vertex for sources, and the one standing on the base – for receivers. The exact shotpoint is the triangle vertex, and the exact receiver position is the middle of the triangle's base. The button displays all devices and takes them away at the repeated click. The button menu commands allow displaying only sources or only receivers. When the observation system is dense, the triangles get overlapped, and viewing makes sense only on the zoomed model image, just as shown on fig. 3a.

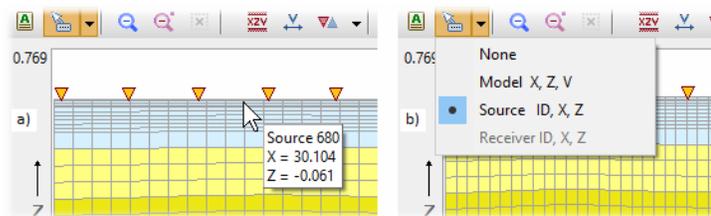


Fig. 3. Sources and receivers on magnified model image. a) sources on the daylight surface line; b) hint window option menu.

In the "only sources" and "only receivers" modes, the user can get information on each device: source ID or receiver number and their coordinates. Suppose that, as in fig.3, the "only source" mode is set with the button menu. The  tool button drops down the menu controlling the hint window content (fig 3b). The user clicks on the *Source ID, X, Z* command and then brings the cursor nearer to a source *from the left side of it*. The hint

window appears with the required information (fig. 3a). The mechanism works, if only devices are remote enough of each other, which can be achieved by proper magnification.

If the user selects the *Model* command in the hint control menu, the hint window with model point coordinates and velocity value will follow the cursor moving all over the model image.

### View options

The user adjusts model view option in the *Model View Settings* dialog with two tabs shown on fig. 4.

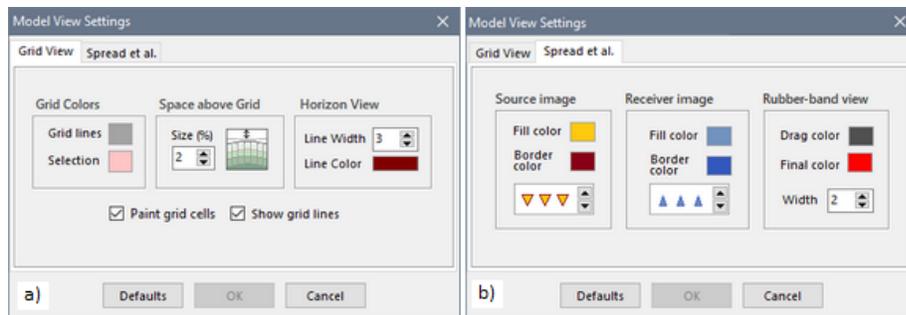


Fig. 4. MOV view options;  
a) grid view; b) sources, receives and rubber band views.

The first tab contains the following grid parameters:

- grid lines color;
- select color;
- size of a space above grid for displaying sources and receives; the size is defined in percents of the image height;
- horizon line width and color ;
- flags of filling grid cells and drawing grid lines; the same flags make up the menu dropped down by the  button.

The second tab allows changing colors and sizes of source and receiver triangles, and rubber-band colors (drag color and drop color). The *Defaults* button returns the "factory settings".

### Text export

The model components – velocity and horizons – can be exported to a text file for exchange with other applications. The *File / Export Model to ASCII File* command displays a dialog, in which the user selects an export option from the drop-down list and the defines the export file. The options are:

- velocity export to a file of the LMF format;
- velocity export to a file of the DAT format;

- horizons export to a file of the MG format;
- horizons export to a file of the BLN format.

LMF is XMF model format; MG is XTomo-LM format for a set of curves. DAT and BLN formats are used, in particular, by the Surfer™ (Golden Software®) presentation graphic application. All formats are described in Appendix 2. The *Export* button creates a file of the selected format. The exported file can be immediately viewed, if the *Open File after Export* box is checked.

### Graphic export

The *File|Export Model Image* command invokes a two-tab dialog shown on fig. 5.

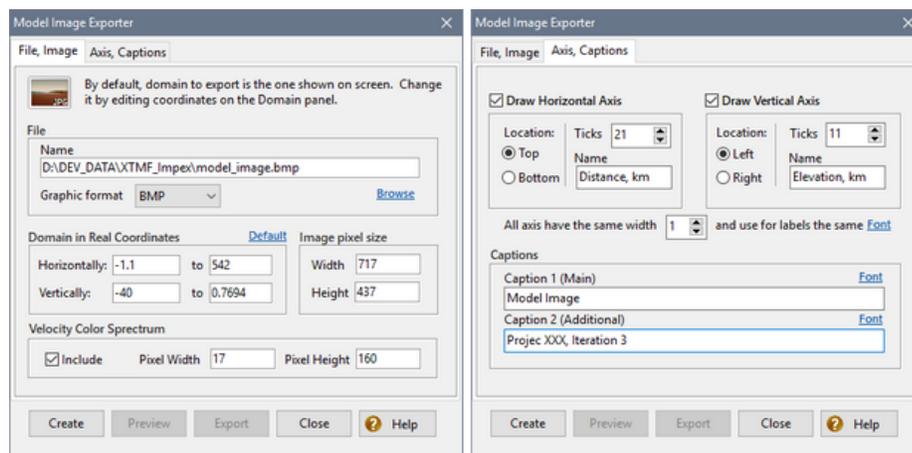


Fig. 5. Graphic export dialog.

If just after the dialog is displayed, the user clicks on *Create*, and then, after it becomes enabled, on the *Preview* button, the Windows Paint will appear with the model image which is the exact copy of the MOV model image, in the same state (view options, zoom) and with the spectrum. The new model image will be provided with axis with the default number of ticks in the default font.

The dialog allows changing the default adjustments; the user can set:

- which model subset is to be imaged (the *Domain in Real Coordinates* panel, tab 1);
- the pixel size of the target image (the *Image Pixel Size* panel);
- is the spectrum exported (the *Include* check box); the pixel size of color strip of the spectrum (the *Velocity Color Spectrum* panel);
- which axis to draw (the *Draw ... Axis* check boxes, tab 2);
- for each axis: location, number of ticks, name; line width;
- captions: the main and additional (can be empty);
- fonts of axis labels and captions; fonts are selected by a click on the *Font* links; the current font appears in a link's hint window.

After the image properties are set, the user clicks on the *Create* button to create the image in memory, then on *Preview* to check the view. If it is approved, the image can be exported in any of the graphic formats from the *Graphic Format* list, tab 1. There are four popular formats: BMP, PNG, JPG, GIF. Export file name is defined in the Windows file dialog displayed by the *Browse* link.

## Color Velocity Spectrum

### Overview

If the default velocity color spectrum does not fit, the user changes it or create a new spectrum with the help of the servicing module Velocity Spectrum Manager (VSM). Because the spectrum is stored together with the initial model, VSM is launched on the project's zero iteration. If the project is not active, VSM runs in view mode. The module's main window and its key dialogs are presented on fig. 1.

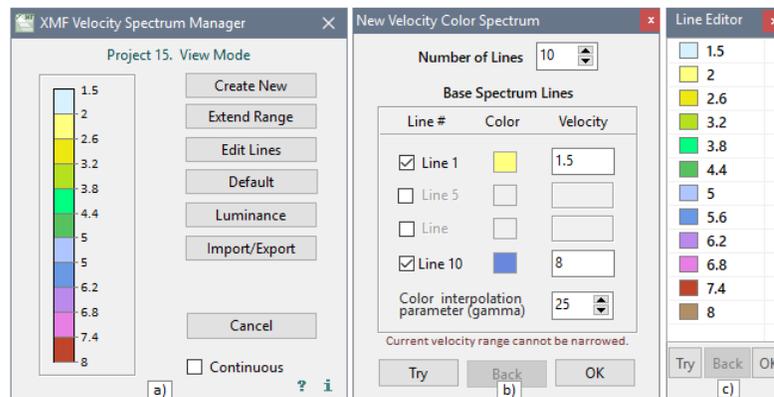


Fig. 1. Velocity Spectrum Manager.  
a) main window; b) creating a new spectrum; c) editing spectrum lines.

The spectrum is a set of pairs

$$(+)\quad \mathbf{S} = \{ (v_1, c_1), (v_2, c_2), \dots, (v_{n+1}, c_{n+1}) \},$$

where  $c_k$  is color code; the color  $c_k$  relates to the velocity interval  $[v_k, v_{k+1}]$ . Such spectrum has  $n$  spectrum lines. On a snapshot of the VSM main window the default spectrum is shown with 11 lines. The 12-th color is also defined; it is used for color interpolation in the continuous mode.

The window buttons carry out the operations described in Table 1 and, in some details – in the three sections below.

Table 1.

<i>Button</i>	Operation
<i>Create New</i>	Creates new spectrum (details are below).
<i>Extend Range</i>	Allows extending velocity range.
<i>Edit Lines</i>	Displays a dialog for editing spectrum lines - velocity values and colors (details are below).
<i>Default</i>	Replace the current spectrum with the default one.
<i>Luminance</i>	Allows changing luminance if the spectrum image; in a dialog, clicking on the "+" or "-" button increase or decrease luminance of the image in the main window (fig. 1a). To accept, click on <i>OK</i> , to cancel, close the dialog with the system button.
<i>Import/Export</i>	Allows exporting the current spectrum to a user file and load it later to replace the current spectrum. In the import/export dialog, first, select a file name, then use the appropriate button.

### Creating new spectrum

To create a new spectrum, the user has to define 2 to 4 pivot colors. The rest are obtained by color interpolation. In the *New Velocity Color Spectrum* dialog (fig. 1b), first, define a number of spectrum lines in the edit field. Then check the boxes matching the pivot colors. The first and last colors are obligatory and checked always. To select pivot color, the user clicks on the color square and then makes selection in the Windows color dialog. The velocity range is defined by the first and last values and cannot be narrower than the current range. Other pivot values are selected automatically to provide the uniform velocity distribution over its range. The last item to define is the color interpolation parameter (*gamma*). It shifts the intermediate interpolated colors to one of the pivot colors. Gamma can be adjusted experimentally. The *Try* button allows the user to test the new spectrum: it replaces the current one in the main window temporarily. If it is approved, the user clicks on *OK* to accept the result, else on *Back*, to cancel the replacement and continue working on the new spectrum.

### Editing spectrum lines

The *Line Editor* dialog displays the spectrum as a list of velocity values with icons filled with line colors (fig. 1c), exactly like in (+). Changing the first and last velocity values, the user widens velocity range. When editing other velocity values, the user avoids setting them too close to each other. VSM tracks velocity resolution. Editing is implemented by commands of the list context menu presented in Table 2.

Table 2. Commands for editing spectrum lines.

<i>Command</i>	Description
<i>Edit Value</i>	Allows changing velocity value of the selected line. Edit mode can be also switch on by a left-click on the selected item.
<i>Edit Color</i>	Allows changing the selected line color in the Windows color dialog. The same effect can be obtained by a double-click on the selected item.
<i>Insert before</i> <i>Insert after</i>	Inserts a new line before or after the selected line with the default velocity value and color. The user changes them as needed.
<i>Delete</i>	Deletes the selected line.

### **Saving. Velocity range. Black grid cells**

The *Save* button of the main window saves the changes to disc, replacing the current spectrum. It may happen, that graphic modules with the spectrum image are currently running. Then the user should update the spectrum in every such module with the *Reload* command of the spectrum context menu.

The spectrum defines the permissible velocity range. If the user, while editing velocity in Model Editor, assigns velocity a value outside the spectrum range, the grid cell will be filled with black color to signal the error. The user should either correct the cell velocity, or extend the spectrum range.

## **Model Editor**

Model editing includes changing geometry, grid density and velocity. Changing geometry means editing the model wireframe with Wireframe Editor. Model Editor (MED) is built upon [Model Viewer](#) (MOV) and inherits all its features, appending MOV's menus with editing commands. MED can be started only from the active iteration menu with the *Edit Model* command.

### **Edit commands**

These command are added to the *Model* section of the main menu главного меню, model image context menu at the presence of a selection and the spectrum menu. The *Model* menu contains commands related to the entire model, context menu commands relates to a selected grid subset.

Table 1. Edit commands

<i>Commands</i>	Description
	<i>Model</i> menu
<i>Edit Geometry</i>	The command is accessible, if the model image is in normal state: no zoom, no selection. It starts Wireframe Editor which displays the current wireframe. After editing the curve(s), the  button starts building velocity grid and then the model image in MED is updated, possibly, with some delay. If a number of model layers is changed, velocity function gets lost, and the user enters new constant velocity values, as in <a href="#">initial model</a> . The further actions are the same.
<i>Change Grid Density</i> <i>Double Number of Col</i> <i>Halve Number of Col</i>	Shows the submenu with commands that double or halve a number of columns. The lower bound for a number of columns is a number of x-net nodes.
	Model image context menu at the presence of grid selection
<i>Edit Velocity Profile</i>	Accessible, if a column is selected. Displays a dialog for editing vertical velocity profile <a href="#">&gt;&gt;</a> .
<i>Edit Velocity</i>	Accessible at any selection. Displays <i>Velocity Editor's</i> window <a href="#">&gt;&gt;</a> .
<i>Change Row Density</i> <i>Double Number of Rows</i> <i>Halve Number of Rows</i>	Accessible if a horizontal strip is selected, in particular, a layer. doubles a number of rows in the selected strip; halves a number of rows in the selected strip.
	Context Spectrum menu
<i>Assign to Selected</i>	Accessible at any selection. Assigns velocity value, defined by the cursor position on the spectrum image, to all cells of the selection. Before assignment, displays a dialog for refinement of the value. The same effect can be obtained by a double-click on the proper spectrum image point.
<i>Spectrum Manager</i>	Starts the <a href="#">VSM</a> module for editing the spectrum.

### Velocity editor

The editor window is displayed by the *Edit Velocity* command of the model image context menu at any grid selection. The window is floating, so the user can switch to the main window and come back to Velocity Editor .

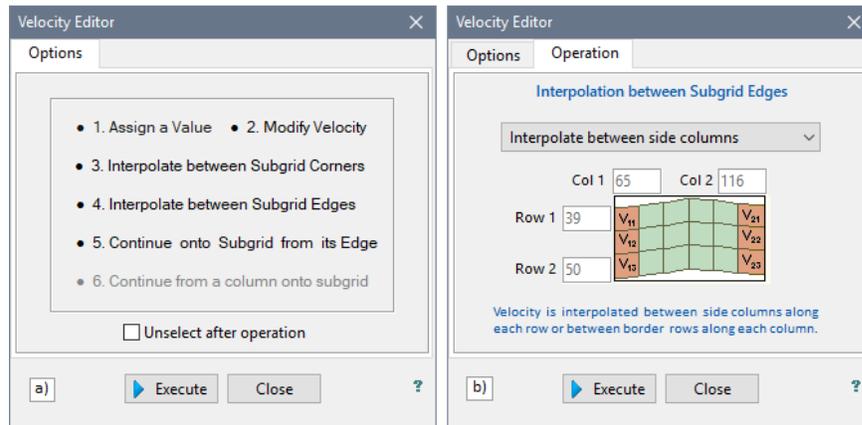


Fig. 1. Velocity editor; a) main page; b) a page of an operation.

Just after the start, the dialog shows its option page (fig. 1a) with a list of operations. Some of options may be disabled. That depends on the form of selection. When the user clicks on the option, the operation page opens, on which the user defines operation parameters. After that, the user clicks on *Execute* to perform the operation. Then the user can change selection in the main window and continue work back in Velocity Editor. Table 2 elaborates editing options.

Table 2. Options of editing velocity

In the description of operations we use "column" and "row" meaning their parts within the selected subset.

Option	Description
<i>Assign a Value</i>	Assigns (the same) velocity value to all cells of the selection. The user enters the value in the edit field. It must be inside velocity range, defined by the spectrum. Otherwise, it is silently replaced with the nearest velocity range bound. That relates to all operations.
<i>Modify Velocity</i>	On the operation page the user, first, selects the type of modification: absolute or relative (the default). Then the user enters the value either in velocity unit or in percents.
<i>Interpolate between Subgrid Corners</i>	The operation replaces velocity values in each selection cell but the corner ones with the interpolated values of the four corner

	cells. The user can, first, change the current values of the corner cells.
<i>Interpolate between Subgrid Edges</i>	Velocity values are interpolated either between the first and last column, or between top and bottom rows. The user select variant of interpolation from the drop-down list on the operation page (fig. 1b).
<i>Continue onto Subgrid from its Edge</i>	Either each column becomes a copy of its bounding column (in the sense of velocity values), or each row becomes a copy of a bounding row. The user selects the source column or row for continuation from the drop-down list on the operation page.
<i>Continue from a column onto subgrid</i>	Continuation results in that any velocity column of the grid can become a copy of the selected one; or only those velocity columns, that are to the left or to the right of the selection. The user selects one of the three variants in the drop-down list on the operation page. <i>Important example:</i> velocity is defined in Velocity profile editor and then is continued onto the entire grid or its part.

**Velocity profile editor**

The *Velocity Profile Editor* (fig. 2) allows editing velocity profile numerically. The dialog is displayed by a command of the model image context menu, if a grid column is selected.

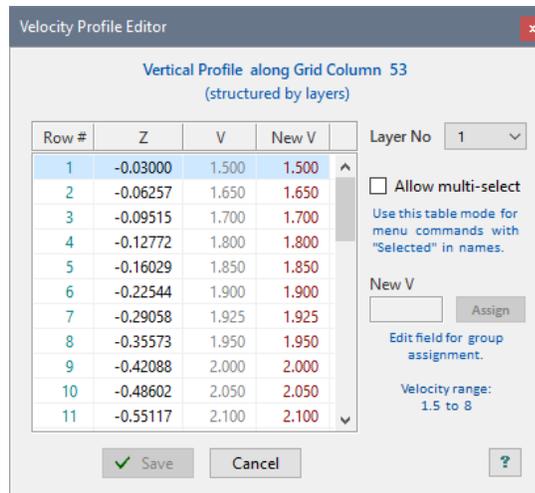


Fig. 2. Velocity Profile Editor.

In the dialog, a velocity column is shown as a table "Row # – Z – V – New V". In the beginning, the *New V* column is a copy of the *V* column; it is that column that is about to be edited. After saving it replaces *V* in the grid.

The column is displayed by model layers. Layer numbers make up the drop-down list at the right top corner. After the user selects a number in the list, the matching part of the column appears in the table. All editing tools are targeted at the current part of the column. When the user selects another layer, the changes are saved in the column buffer.

Under the layer list, there is the *Allow multi-select* check box. Before executing a group operation, multi-select mode should be switch on. In the context menu, such operation commands contain the word "selected".

A double click on a cell of the *New V* column starts the [column editor](#), which provides the column "manual" editing. The context menu commands (Table 3) help make editing much more effective.

Table 3. Context menu commands. Group commands are marked with the asterisk. "Column" means part of a column within the selected layer.

<i>Команда</i>	Описание
<i>Edit Value</i>	Same as double-click: starts the column editor. If the <i>Multi-Select</i> mode is on, the command does not work.
<i>Assign to Selected*</i>	Makes the field <i>New V</i> to the right of the table active. It accepts a velocity value to assign to all selected cells with the <i>Assign</i> button.
<i>Copy Selected*</i>	Copies the selected values from the <i>V</i> column to the <i>New V</i> .
<i>Copy All</i>	Copies the entire <i>V</i> column to the <i>New V</i> .
<i>Clear Selected*</i>	Clears the selected cells of the <i>New V</i> column. Clearing the first and last cells is blocked (see <i>Fill Gap</i> ).
<i>Clear All But Ends</i>	Clears the entire <i>New V</i> column but the first and last cells.
<i>Fill Gap</i>	Fills a block of empty cells by means of interpolation between the cells bounding the block. The command allow to define the column very fast by clearing it, setting some characteristic values and filling gaps.
<i>Fill All Gaps</i>	Fills all gaps in the <i>New V</i> column.

## Forward Problem

[Forward problem task](#) – [Forward Problem Solver](#) – [Ray-tracing log](#) – [Ray Catalog](#)

The forward kinematic problem (FP) suggests tracing rays of the specified waves from the specified sources to all relevant receivers.

### Forward problem task

To make up FP task, the user should define a set of waves and a set of sources. Task may include all waves and sources, but that happens at the concluding stages of interpretation. Most iterations are of trial or testing types and work with small groups of waves and sources.

Task is made up with the help of module Forward Problem Tasker (FPT). It allows viewing task on any iteration and creating/changing it on the active one. The iteration menu command is *Forward Problem Task*. The module main window is shown on fig. 1.

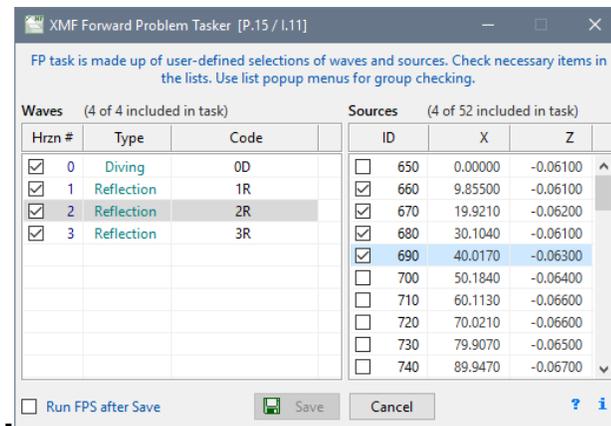


Fig. 1. FPT main window.

The window contains a list of all defined waves at the left and a list of all project sources at the right. Each item of each list is provided with a check box for marking. A checked box means the item is included in the task. To help perform multi-step operations, each list is equipped with a context menu. The wave list menu contains only two commands: *Check All*, *Uncheck AL*.

The source list can be very long, therefore, it is set in multi-select mode, and its menu has, additionally, the following commands:

- *Check Selected* – includes in the task all selected items (plus to already included);
- *Check Only Selected* – includes all selected items and excludes the rest ones;
- *Check Subsequence* – includes the subsequence of items which consists of each  $J$ -th one starting from the  $K$ -th;  $J$  and  $K$  are defined in a dialog.

The *Save* button saves the task to disc. If the *Run FPS after Save* box is checked, the XMF head program launches Forward Problem Solver immediately after FPT terminates.

**Solving Forward Problem**

Kinematic forward problem suggests tracing rays defined in the task through the current model. Computation time is a critical parameter of FP algorithm. It depends on the velocity grid density and task size.

Ray-tracing itself is only a part of the problem. The second part is creating such output data set that is optimal for accessing individual rays. Such optimized data set is called Ray Catalog, which lists rays and stores ray paths. Ray Catalog size may take up hundreds of megabytes on disc. That's why access time is so important. If a number of rays is large, creating Ray Catalog may take time comparable to ray-tracing.

To reduce computation time, the Forward Problem Solver (FPS) module make use of parallel computations. The task is divided into elementary jobs of tracing rays of one wave from one source to all receivers. An elementary job is executed by module Primary Ray Tracer (PRT). FPS launches concurrently *N* instances of PRT, where *N* is a number of logical processors (LP) in the system. When an PRT instance completes its job, FPS appoints the next one to it, and so FPS does for each instance until the task is expired. Clear, efficacy of parallelization depends on a number of LP in the system. If a number of different pairs (wave, source) in the FPS task exceeds *N*, FPS takes almost all CPU time. Therefore it does not make sense to launch other heavy applications in parallel with FPS.

**FPS user interface**

FPS is launched on an active iteration only. Usually, it does not runs concurrently with other modules to decrease load on the system. FPS main window snapshot is shown on fig. 2. At the window's top, there is FPS task in the form of two drop-down lists. Ray tracing begins after a click on the *Start* button.

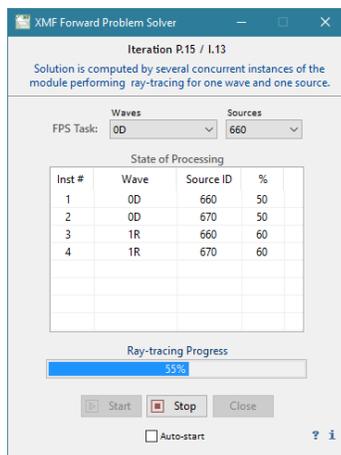


Fig. 2. FPS main window.

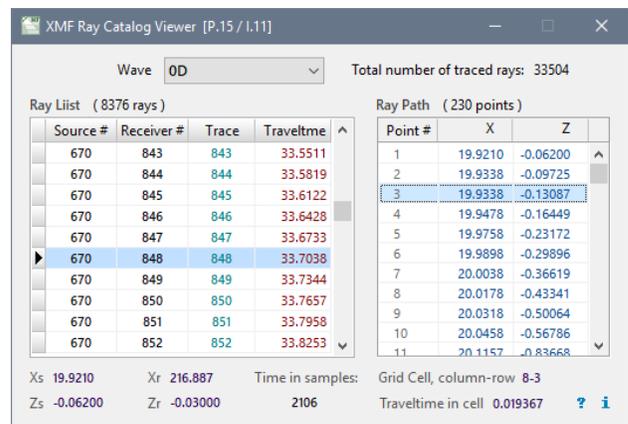


Fig. 3. Ray Catalog Viewer main window

On fig. 2, task consists of 4 elementary jobs: rays of 2 waves from 2 sources are to be traced. The table *State of Processing* has 4 rows, each for one concurrently working PRT instances (the station has CPU Intel Core i7 with 8 LP). Thus, the entire task is being executed at one go. I-th row shows which job is executed by i-th PRT instance, that is shows wave code and source ID. The last column displays percentage of completion. After i-th instance terminates, the column shows "OK" or "Error". In the latter case, the error message can be read in in the hint window that appears when the cursor is hovering above the row.

The progressor under the table shows total percentage of task completion and on the stage of creation Ray Catalog – a fraction of rays added to it. The *Stop* button makes all instances to shut down, possibly, with a delay. If the *Auto-start* box at the bottom is checked, then in the next session FPS begins ray-tracing immediately after start, without waiting for a click on the *Start* button. After Ray Catalog is created, FPS displays a message box in which the user can inform XMF about further actions: display the log or/and view TX-curves or simply close the module. The *View log* and *View TX-curves* check boxes are just for that. Those flags as well as the *Auto-start* flag are saved and, together with the *Run FPS after Save* flag in Forward Problem Tasker, allow move processing forward without the head program.

### Ray-tracing log

The log shows, how many rays has been traced for each source and wave and how many has been lost. The latter is important. Too large number of lost rays suggests that the model is wrong. The log is displayed by the Forward Problem Logger module. The launching command is in the active iteration menu. If a number of traced rays in the final FPS message does not look suspicious, there is no need in viewing the log at each iteration.

### Ray Catalog

Studying forward problem solution is the most important part of an iteration. XMF offers three tools for that: (1) direct viewing Ray catalog in numeric form; (2) exploring a ray picture and (3) studying TX-curves on seismograms. The tools are based on Ray Catalog database. In this section we consider (1). The next two topics are devoted to (2) and (3).

The Ray Catalog Viewer (RCV) module represent Catalog in the form of numeric tables. It is launched from the iteration menu of Processing Tree. The module's main window is shown on fig. 3. The left part of the window is taken up by the Ray List table. The table structure is "source ID – receiver number – seismogram trace number – travel time". A "source–receiver" pair unambiguously defines a ray of a wave, so this is indeed, a ray list

of one wave. A wave is selected in the drop-down list at the window's top. A set of waves and sources is defined by the forward problem task.

The coordinate information is displayed under the table and relates to the selected item of the table:  $(X_s, Z_s)$  is a source point,  $(X_r, Z_r)$  is a receiver point. Additionally, there is a travel time expressed in signal samples. To make navigation along a large table easier, it is provided with a context menu, containing three commands:

- *Next Source* – moves to the first item with the next source ID;
- *Previous Source* – moves to the first item with the previous source ID;
- *Move To* – move to a ray with a "source–receiver" pair specified by the user in a dialog.

The table on the right displays the ray path for the ray selected in the ray list. The table structure is "ray point # – X – Z". Clear, the first point is the source, the last point is the receiver. Ray point is a point of exit from a grid cell. The cell matching a selected point is displayed under the table in the form "column-row". Below travel-time in the cell is displayed.

## Ray Picture

[Overview](#) – [Ray output control](#) – [Ray samples](#) – [Drawing attributes](#) – [Other features](#).

### Overview

The XMF Ray picture is a model image, like in Model Viewer (MOV), with a subset of rays from Ray Catalog drawn over it. Ray Picture Viewer (RPV) is a module responsible for presenting different views of ray picture. Drawing all rays from the Ray Catalog make sense only when a number of rays is comparatively small. Otherwise, rays covers the entire model areas completely and drawing time become unacceptable for normal work, requiring multiple screen updates. Zoom is only a partial help, because the user loses the whole picture.

RPV completely blocks output of a ray subset numbering 5000 rays or more. Instead, it offers a mechanism for creating meaningful ray samples.

Usually, XMF forbids running more than one module instance on one node. RPV is an exception: up two three instances can run on an active iterations to examine different views of the same ray pictures.

### Ray output control

RPV is an elaboration of MOV and inherits all its features. New functionality is behind the one new tool bar button: . A click on the icon allows creating a ray sample. A click on the arrow drops down menu, described in table 1.

Table 1. Commands controlling ray output.

<i>Command</i>	<i>Description</i>
	<i>Ray picture state switch</i>
<i>Show All Rays</i>	Display all rays from Ray Catalog.
<i>Show Last Ray Sample</i>	Display rays of the lately created sample.
<i>Hides rays</i>	Hide rays
	<i>End of switch</i>
<i>Only Rays Crossing Selection</i>	The command is accessible if only there is a selection on the grid. The command implements a filter that hides those rays from the current sample, which do not pass through a selected subset. To switch off filtering, the user applies the same command once more or double-clicks the image to kill the selection.
<i>Drawing Attributes</i>	Displays a dialog for changing rays and TX-curves drawing attributes <a href="#">&gt;&gt;</a> .

### Ray samples

The criterion for sampling rays is a logical statement "W and S and R", where W denotes wave clause, S and R denotes source and receiver clauses. The  button displays three-page dialog shown on fig. 1. Each page is devoted to one clause. The first page lists waves from the forward problem task (fig. 1a), and the user has to check those that will be included in the sample. The same is valid for the second page and sources. Both lists have context menus with the *Check All*, *Uncheck All* commands. Because there can be many sources in the task, there is a tool for checking a subsequence at the bottom of page 2. It allows checking each  $J$ -th source in the list beginning from the  $K$ -th one.

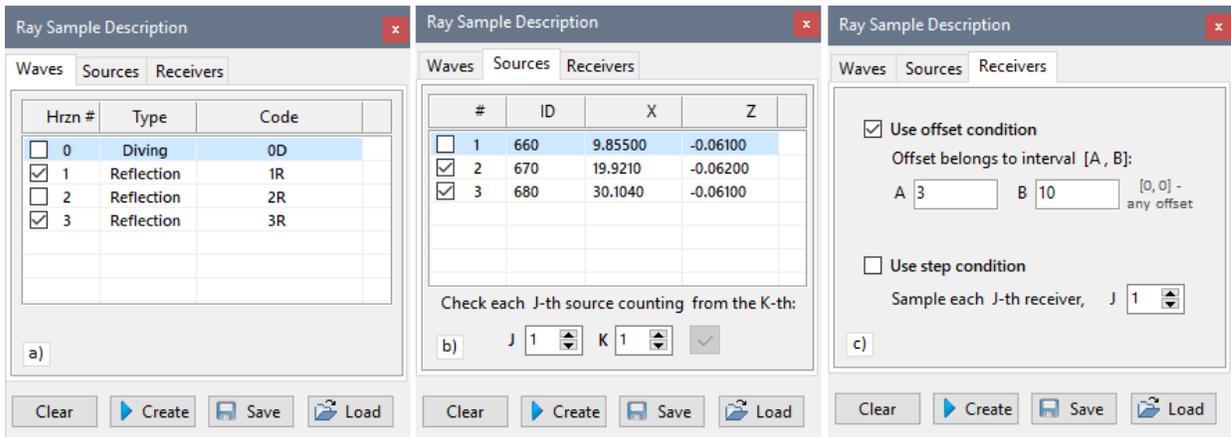


Fig. 1. Dialog for ray sample criterion.  
 a) condition on waves; b) condition on sources; c) condition on receivers.

Two conditions can be imposed on receivers on page 3 (fig. 1c). The first one limits receiver offset: it must fall into the  $[A, B]$  interval, where  $A$  and  $B$  are specified by user in the edit fields. The other condition requires that only every  $J$ -th receiver in the list ordered by numbers participate in the sample. In order that receiver condition become actual, the user has to check a box on the left of it.

The sample description is saved, so the next time the dialog is displayed, it shows the previous sample. The user can save sample description to a user file and later load it. A special folder is reserved in the iteration folder for sample descriptions. The *Save* and *Load* buttons display the content of the folder as the list of user-defined sample names. The name is assigned to a sample description when it is saved.

RPV supports another type of ray samples dealing with travel times, but the criterion for them is composed in another module.

**Drawing attributes**

Rays and TX-curves drawing attributes are applied for the first time in RPV, so the module allows to define or edit them here, with the *Drawing attributes* command. It invokes a dialog (fig. 2) with the list of waves, introduced earlier (not limited to forward problem task). A sample of TX-curve is shown in the *TX-curve line* field. To change it, the user double-clicks it and use the [standard dialog](#) for selecting styles and colors. As for rays, they use wave color, while line style is common for all waves: line without nodes with width of 1 pixel, by default. The user can change the line width in the spin-edit field *Ray line width*.

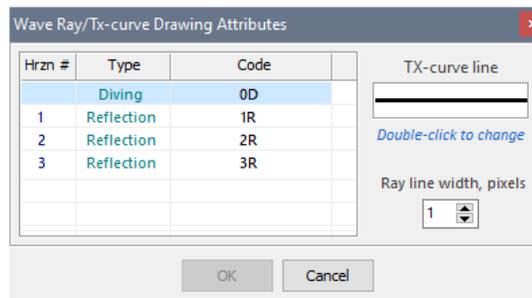


Fig. 1. Dialog for setting/editing drawing attributes.

## Other features

1. Subgrid description. In the model image context menu, in the presence of a selection, a new command is added: *Save Subgrid Description*. It saves specification of a selected subgrid as a text record to the iteration user notes. A record looks like this:

```
<2020-01-12 14:46>
Subgrid 1 {rows: 32 to 43, cols: 73 to 97}.
XZ-Domain {X: 198.406 to 267.679, Z: -14.723 to -7.58721}.
```

It is in RPV, that the user has to memorize information for future corrections of the model in the current or next iteration. The automatic records together with the user's notes can help planning further actions.

2. Ray picture export. The [graphic export](#) command, inherited from MOV, exports the picture of the current ray sample to a graphic file.
3. Ray text export. The [ASCII export](#) dialog, inherited from MOV, has an additional option of exporting rays of the current sample to a BLN file.

## Studying TX-curves

[Overview – Frame rubber-band – TX-curve visibility manager – Frame context menus – Displaying rays matching specified TX-curve points](#)

### Overview

If Line Seismogram Viewer ([LSV](#)) discovers Ray Catalog in the iteration folder, it works in the TX-curve output mode. That means that, when creating a frame, it, first, output a seismic record view, then extracts from Ray Catalog wave travel times for the source, forms TX-curves and output them on top of a record view (fig. 1).

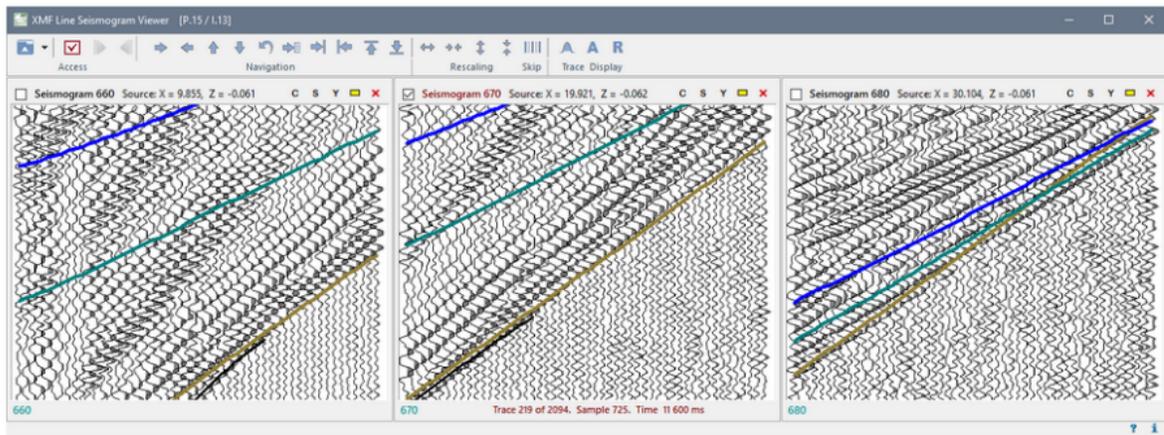


Fig. 1. Computed traveltime curves on CSP seismograms.

The LSV и RPV (Ray Picture Viewer) modules are intended for studying forward problem solution. The following tools can be helpful in this job:

- frame rubber-band and frame context menu;
- TX-curve visibility manager;
- special ray samples (together with RPV).

Note also the enabled *TX-curve Drawing Attribute*, command of the access menu which allows changing TX-curve view "in-place". (Drawing attribute can be changed in three modules, each one dealing with the same data.)

### Frame rubber-band

Frame rubber-band is used as a means of selecting a TX-curve or its segment on the seismic record. However, rubber-band has restricted resolution. For example, it is hardly possible to separate curves of different waves in the zone of first arrivals (the third frame on fig. 1). That is the reason, that executing a command of the rubber-band menu is interrupted by a dialog in which the user makes his intentions exact (fig. 2a) – "the refinement dialog".

The dialog contains the domain selected by the rubber-band (on the left) and the list of waves captured by the rubber-band (if no TX-curve is captured, menu does not appear). The user can change the selected domain and exclude some waves from the pending operation. After a click on *OK*, the operation is executed.

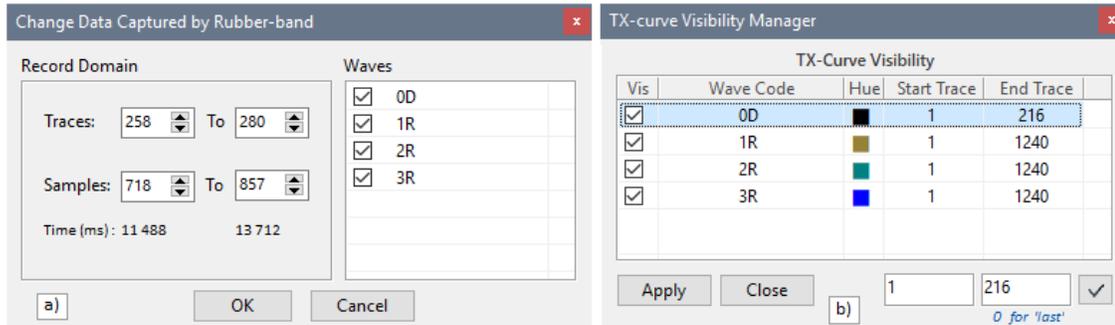


Fig. 2. LSV tools: a) refinement of the rubber-band data; b) TX-curve visibility manager.

**TX-curve visibility manager**

The computed TX-curve is, normally, defined for all traces, while phase lineups on a seismic record may be traced at comparatively short ranges. Non-informative parts of TX-curves can be hidden to make the view more intelligible. On fig. 1, all TX-curves are shown in full but the diving-wave's curve which is hidden after it leaves first arrivals. The TX-curve visibility manager dialog (fig. 2b) is invoked by a command of the [selected frame menu](#) or a command of context frame menu. The dialog contains a list of waves from the forward problem task, provided with visibility check boxes (the *Vis* column) and visibility ranges [Start Trace, End Trace]. By default, all boxes are checked, while ranges include all file traces. The user can change visibility by checking the boxes or/and editing the bounding traces in the edit fields under the list. At that, the last trace number is encoded with 0. Additionally, list has a context menu that allows:

- set left range bounding trace to 1;
- set right range bounding trace to the last trace number;
- set the default range.

Some of the manager functions can be carried out without the dialog, though the frame menus.

**Frame context menus**

Table 1. Frame menu commands.

Command	Description
	Frame context menu
<i>Set Record Position as Initial</i>	Makes the current record view the initial view. This is a replica of a command of the selected frame menu.
<i>Manage TX-curve Visibility</i>	Displays TX-curve visibility manager dialog. This is a replica of a command of the selected frame menu.

<i>Show All TX-curves</i>	Shows all TX-curves, each within its own visibility range.
	Frame rubber-band menu
<i>Hide Curve</i>	Hide TX-curves captured by the rubber-band (i.e. wave visibility flags are reset).
<i>Set Visibility Interval</i>	Visibility ranges of the waves, whose TX-curves are captured by the rubber-band are set to [m, n], where m and n are the leftmost and rightmost traces inside the rubber-band.
<i>Show Entire TX-Curve</i>	Shows TX-curves, captured by the rubber-band, in full.
<i>Show Matching Rays</i>	Displays rays matching captured TX-curve points (details are below).
<i>Capture Refinement</i>	Allows/bans displaying the <a href="#">refinement dialog</a> .

**Displaying rays matching specified TX-curve points**

The *Show Matching Rays* command of the rubber-band menu makes RPV to display a sample of rays matching TX-curve points captured by the rubber-band (fig. 3). In the left part of the snapshot an LSV frame is shown with the rubber-band stretched in the way to capture three TX-curves and the rubber-band menu dropped down. On the right, RPV displays a ray sample consisting of rays matching exactly TX-curve points.

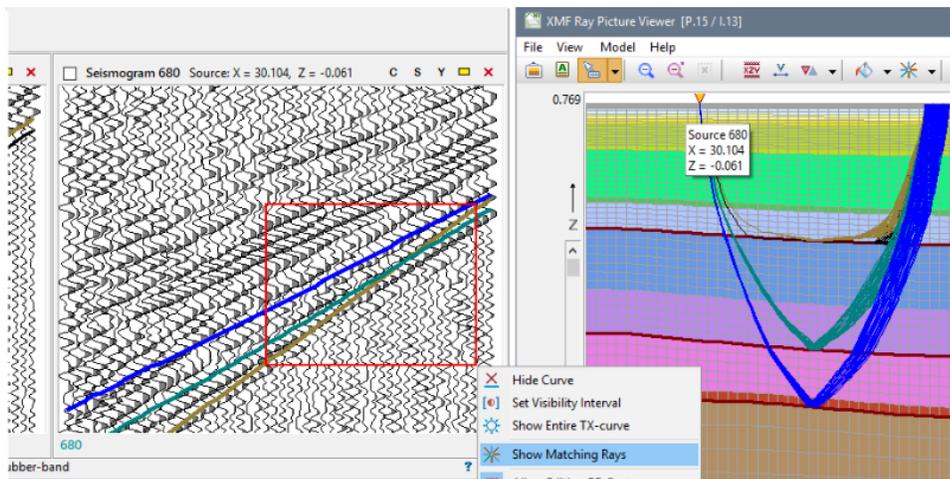


Fig. 3. Ray sample matching specified TX-curve points.

The operation described is helpful for studying TX-curve points diverging significantly from the target phase lineups, because it allows binding "bad" curve points to a model subdomain, let in indirect way.

However, for reflections which illuminate a boundary locally, binding a TX-curve segment to a boundary segment is direct, if velocity is fixed. Additionally, a diving wave illuminates different depths with different rays, and so binds a TX-curve segment to a segment of velocity vertical profile.

The user can examine, at a time, a few ray samples matching TX-curve points on the one or a few seismograms. If RPV with the first ray sample is left active and, after stretching the rubber-band anew, the *Show Matching Rays* command is issued repeatedly, a message box appears with a question of launching a second RPV instance. If the user reply is "yes", a new RPV instance is launched with a new ray sample. If, the user's reply is "no", a new ray sample replaces one of the existing (but not the one in the minimized RPV instance).

## PRE-STUDY

### Problem Statement

In the first three topics of the chapter, the concept of pre-study is explained. The rest topics are devoted to the user interface of the programs that together implement that concept.

#### Problem definition

Let  $M_0$  be the initial model of an XMF project. It is described by velocity function  $V(x, z)$  defined on a two-dimensional curvilinear grid and by a set of indexes of grid h-lines representing seismic boundaries. For a given shotpoint  $s = s(X_s, Z_s)$ , *s-vertical* is the line with equation  $x = X_s$ , *s-section* is section of  $M_0$  by *s-vertical*. S-section is described by one-dimensional velocity distribution  $V_s(z) = V(X_s, z)$  and a set  $\{Z_{b1}, Z_{b2}, \dots, Z_{bn}\}$  of z-components of intersection points of boundaries with the *s-vertical*; each  $Z_{bk}$  is a grid node. Now let us state the pre-study problem.

1. For each  $s$  from a given shotpoint subset, build a new *s-section* by fitting a simplified model for each layer of the layered model; fitting criterion is visual proximity of a computed TX-curve to wave lineups on the seismogram generated by  $s$ .
2. Using the built *s-sections*, replace velocity profiles on each  $M_0$  vertical and, thereby, build a new two-dimensional model  $M_1$ , which will be, supposedly, much better approximation to the solution, than  $M_0$ .

This way of posing a problem is natural because XMF input data have the form of a set of common shotpoint seismograms. Expectations of better approximation from #2 are justified because initial model  $M_0$  is based on superficial estimation of the wave field and a priori information, while  $M_1$  is built after detailed study of the seismic record. As for "simplified models" from #1, they are considered in the next section.

Within our problem, model  $M_0$  is called *reference model*, while  $M_1$  is called *target model*. Formally, reference model is not necessarily the initial model, it may be a model from any project iteration.  $M_0$  and  $M_1$  can have different number of layers.

In the user interface and further in the document, the terms *1D section* and *local section* are used as equivalents of *s-section*, when reference to a shotpoint is not essential.

### Standard models

The notion of *standard* or *elemental* model (SM) comprises three models, each of which is associated with a wave that can be traced on the seismogram record.

D-model is associated with the diving wave. The only model parameter is strictly decreasing velocity function  $V(z)$ . Such model is normally used for study of low velocity zone (LVZ).  $V$  strictly increases with depth.

R-model is associated with the reflection from a flat boundary. Its parameters include:

- depth under day surface along s-vertical;
- slope;
- velocity in overburden  $V = V(z)$ .

H-model is associated with the head wave. It has the same set of parameters as R-model and, additionally, boundary velocity value.

R- and H-models are used to study a layer of the layered model, whose floor is a reflector or a refractor, respectively. Those models are not, generally speaking, one-dimensional, because slope may be nonzero. All models used in local section fitting together with TX-curves computed for the models are stored in Standard Model Database (SMDB). Such database is created for each shotpoint participating in pre-study.

### Local section

Fitting standard models for the layered model goes in accord with layer-by-layer interpretation principle. First D-model for LVZ is fitted, if diving wave is traced on the seismogram. Then, successively, top to bottom, R- or H-models for the layers are built. In each layer, a boundary to find is the layer floor, and velocity to find is the layer velocity, because velocity above the roof is already known. In the end of the process, we will have one optimal standard model for each layer. They make up the local section, which is saved to Local Section Database (LSDB). This database is used for building the target model  $M_1$ .

The textual local section description exported by one of pre-study programs can be viewed [here](#). It is very helpful to study it right away, even if some details are still not explained. Building local section is described in details in "Fitting local section".

## Pre-study technology

[-Opening](#) – [-Leading section](#) – [-Next sections](#) – [-Partial target model and supervision](#) – [-Congruence](#)

Here general pre-study algorithm (without s-section fitting technique) considered. Terms introduced in previous section are used without reference.

### 1. Pre-study opening

XMF allots one folder for all pre-studies in a project. It is shown as the *0001\_Pre-study* node in Processing tree (fig. 2a [here](#)). The head pre-study program is Pre-study Controller which is launched by a menu command on that node. The study begins with definition of the reference model "address", i.e. system number of an iteration it is located in. If the pre-study folder is not empty, that is, contains data relating to another reference model, it is cleared out. If necessary, Controller can save the old content to an archive file.

### 2. Fitting leading section

Let  $S$  be a subset of shotpoints participating in the pre-study. One selects a shotpoint  $s_0$  from  $S$ , whose seismogram demonstrates the most unambiguous wave identification matching, roughly, the reference model structure. This shotpoint and its seismogram are called *leading*. The  $s_0$ -section is fitted from scratch and this procedure becomes a pattern for the rest work to do. At that, some parameters of the procedure are fixed for later use, for example, TX-curve drawing attributes, or z-net density for velocity function. When ready,  $s_0$ -section is saved to Local Section Database ( $s_0$ -LSDB).

### 3. Fitting next sections

In the next step, one selects the shotpoint  $s_1$  from  $S$  nearest to  $s_0$  and *imports* all standard models making up  $s_0$ -section from  $s_0$ -LSDB to the newly created database for  $s_1$ . The imported models form a good initial approximation to  $s_1$ -section, so that amount of work and time to spend drops drastically. The same is performed for the next shotpoint  $s_2$  from  $S$ , nearest to  $s_1$ , and so on. Applying the import feature subsequently, one moves along the line to the right and left of the leading shotpoint while  $S$  is not expired.

### 4. Partial target model and supervision of fitting

One of the Pre-study Controller important features is to construct 2D model matching the content of Local Section Database (LSDB), if it contains at least 2 sections. Therefore, after fitting a new  $s_n$ -section, one can build a *partial* target model from  $s_0, s_1, \dots, s_n$ -sections. Fig. 1 shows Pre-study Controller main window and a fragment of partial target model, displayed by Model Viewer.

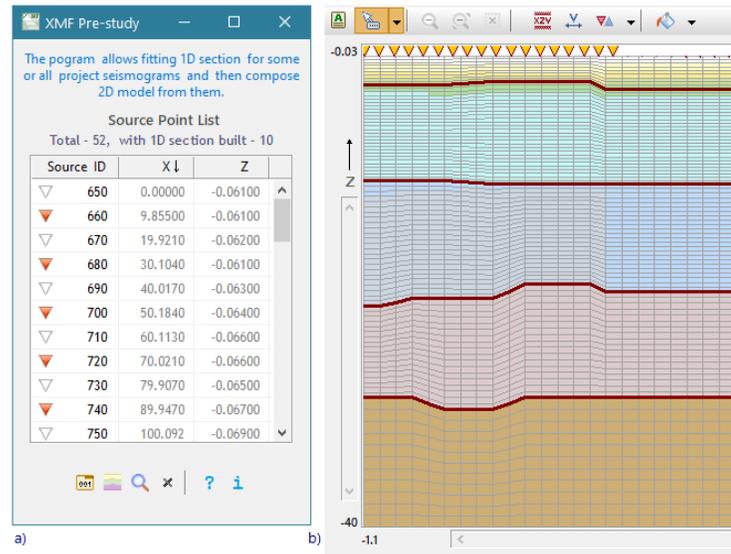


Fig. 1. Partial target model built after a subset of local sections.

a) Pre-study Controller main window; b) partial target model built after 10 local sections.

The Pre-study main window displays the full list of the project shotpoints, each with an icon explaining its status. The red ones mark *active* shotpoints, that is, those with built local sections. In the example, each second SP from the first 20 is active. The partial target model on the right is built after all active SP. The model is continued as constant to the right of the rightmost and to the left of the leftmost active shotpoint. A section of the target model by a grid vertical nearest to an active shotpoint  $s$  is close to the  $s$ -section with high enough accuracy.

Thus, target model creation can go in parallel with  $s$ -section fitting. This is important because it gives two-dimensional view of every step of 1D fitting. In this way, XMF compensates for deficiency of 1D approximation.

Pre-study Controller has another important feature: implanting the target model in XMF Processing Tree. On the user command, Controller creates a new project iteration with the user name "After pre-study" and puts the target model in it. The user name warns the users of the fact that initial list of waves and other settings can change during pre-study. After implanting, the user can solve two-dimensional forward problem for the target model and compare its solution to solutions of 1D forward problems, which may prove to be very informative.

## 6. Structural congruence of $s$ -sections

This XMF version requires that different  $s$ -sections must satisfy the following conditions:

- all  $s$ -sections have the same number of layers;
- number of nodes of  $z$ -grid at which  $V(z)$  is defined is the same for all  $s$ -sections.

The second requirement takes into account that the density of z-grid can be increased in the course of s-section fitting. These two conditions are referred to as (structural) *congruence* of s-sections.

## Fitting a local section

[Building a layer](#) – [LVZ](#) – [Work with velocity function](#) – [Local section](#) – [Model resolvability](#) – [Ray-tracing](#)

In this section shotpoint  $s$  and the matching seismogram are fixed. Process of fitting s-section is arranged in accord with layer-by-layer interpretation [principle](#).

### 1. Building a layer of the layered model

First, suppose that the layer has ordinal number  $n > 1$ , counting from the model top, and that the floor of the layer is a reflector. Therefore, one has to fit an R-model. One sets its parameters: vertical depth, slope and also the layer velocity function  $V_n(z)$ . This is the initial model iteration. Velocity above the the layer floor is  $V(z) = \{ V_1(z), \dots, V_{n-1}(z), V_n(z) \}$ , so it is, indeed, an R-model. One issues the command of solving forward problem, which implies, that reflection rays are traced, TX-curve is computed and drawn over the seismogram. One estimates its position against reflection phase lineups. Unlikely, that the initial iteration is successful. One implements further iterations, each of which includes:

- 1) correcting the R-model of the previous iteration;
- 2) getting forward problem solution;
- 3) studying the computed curve drawn over the seismogram;
- 4) making decision about the next step.

The decision is binary: either move to the next iteration, or move to next layer. If the studied layer was the last one, one moves to building the local section.

On each iteration only layer velocity  $V_n(z)$  is corrected. Behavior of reflected and head wave rays is not too much affected by local variations of velocity in overburden, therefore, it is reasonable to use the simplest forms of  $V_n(z)$ , such as constant, linear or piecewise linear function. Because standard models are simple, one usually can guess the direction in which its parameters are to be corrected. Staying at formal level, one can get to an optimal model at 10 – 20 iterations, the time being mostly spent on one's argumentation. We remind that if  $s$  is not a leading shotpoint, the initial iterations for each layer are known and they not, actually, bad, being borrowed from a neighboring SP.

The condition  $n > 1$  was made only to make the text simpler. In the first layer,  $V(z) = V_1(z)$ . Changing reflector for refractor makes no difference too.

**2. LVZ**

If diving wave is traced at small offsets, one can try to describe top model zone with gradient velocity in which continuous refraction returns rays back to day surface. We will use the regular term *low velocity zone* for this part of the section, or shortly, LVZ. Finding velocity in LVZ means fitting a D-model. Clear, this is a laborious job. If velocity function  $V^*(z)$  for LVZ has been fit, one has to take it into account when one fits all model layers. The  $z$ -range  $I^*$  of  $V^*$  can be easily found from the ray picture. If a seismic boundary happens to fall inside  $I^*$ , velocity of its overburden must be set to  $V^*$ .

**3. Work with velocity function**

R- and H-model have velocity in the overburden as parameter; in the course of layer-by-layer interpretation only layer velocity is determined. On the implementation level, those statements take the following form: one works with the complete velocity profile; when fitting model for a layer, one edits only the part of the profile below its roof. When fitting D-model, one edits the upper part of the profile.

In the example on fig. 1, the profile is shown at the moment of fitting layer #3; black lines match boundaries, that have been already built.

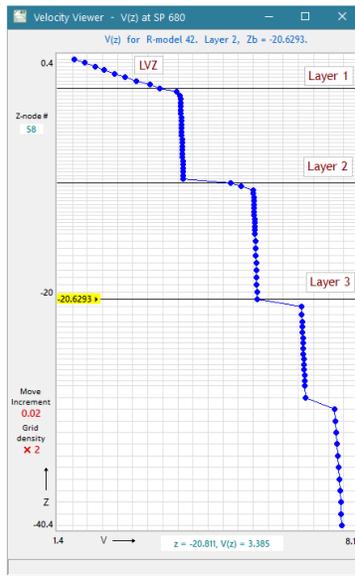


Fig. 1. S-section velocity profile. Currently, layer 3 is being fitted; it is not 1st iteration, because layer floor is already shown.

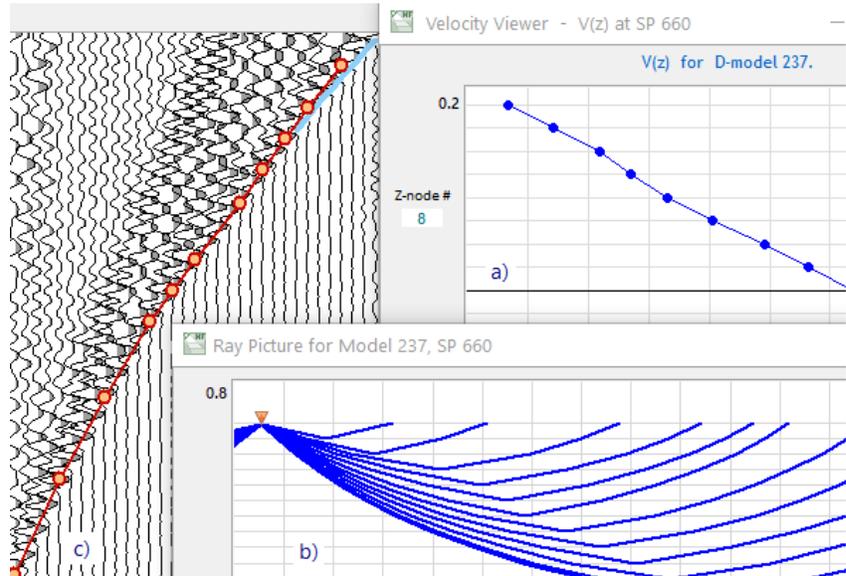


Fig. 3. Solution to forward problem for D-model. a) velocity in LVZ - upper part of full profile shown on fig. 1; z-grid step is 300 m.; b) a fragment of ray picture (total ray number - 11); c) diving wave TX-curve.

Where does velocity profile comes from? At the time one begins fitting the first standard model for the leading SP, one is offered two options to choose the initial velocity profile  $V_0(z)$ :

- a)  $V_0(z) = V_s(z) = V(X_s, z)$  – section of reference model velocity by s-vertical ( $\gg$ );
- b)  $V_0(z) = V_{pow}(z) = C \cdot z^\gamma$ ,  $0 < \gamma \leq 1$  – power function, in particular, linear, if  $\gamma = 1$ .

Power function parameters are defined by the user and it is defined on the same z-grid as  $V_s(z)$ . Subsequently fitting LVZ and model layers, one changes the initial profile  $V_0(z)$  layer by layer and, in this way, forms the final s-section profile.

The idea that one edits the same velocity profile in each layer is conceptually true. But, actually, the current state of velocity profile is a part of current standard model and is saved with other parameters to SMDB. Difference with standard model definition is that, instead of velocity in the layer floor overburden, the full profile is saved. Thus, we have as many profiles, as there are standard models in the database. That provides full freedom for the user to go up the model stack and start a new branch of fitting with all needed information at hand, in one database block.

#### 4. Local section

It is built, on the user command, from *final* iterations of every layer. The important part of this operation is testing velocity profiles of the participating models. The problem is that, when one edits a part of the velocity profile relating to a current layer, one can make a mistake and change points above the layer roof. There is no program restriction for that at edit time. Such error breaks the layer-by-layer interpretation rule, so it must be excluded. The right moment to do that is just before building local section. For every  $k > 1$ , it is verified that velocity profile for layer  $k$  is equal for that of layer  $k-1$  above the roof of layer  $k$ . Only after verification, the local section can be unambiguously provided with the only velocity profile – the one belonging to the final standard model of the last layer. That profile is a part of data added to local section database.

#### 5. Standard model resolvability

When fitting a D-model, one usually faces the negative effect of model discreteness. On fig. 1 and 2, velocity is shown as a continuous function for clarity. However, in accord with XMF model representation, it is a stepwise function, constant in each microlayer and equal to the value on its roof (shown as bold points). A continuous gradient model is approximated with a stack of thin layers, while continuous refraction is replaced with critical reflection from the interfaces. Suppose that z-grid is equidistant, with step  $dz$  between nodes, and that maximal depth of diving wave ray penetration is  $H$ . Then a number of diving wave rays is equal to  $H/dz$ . The problem is that this value may be times less than a number of traces on which diving wave is, actually, traced. On fig. 2, distance between ray exit points is much more than difference between trace offsets, and TX-curve points appears on several traces only. That means that *model resolvability* is much less than the planned resolvability of the observation scheme. *Model resolvability can be associated with the velocity z-grid density.*

Model discreteness affects R- and H-models too. In travel time analysis, reflection or refraction can happen only at points where velocity  $V(z)$  changes, that is, at z-grid nodes. When the user sets model parameters, he or she sets vertical depth it at his or her discretion, but just before solving forward problem, the value is replaced the depth of the nearest z-grid node. That explains, why boundaries go exactly through z-nodes on fig. 1. In case of equidistant z-grid,  $dz/2$  is accuracy of boundary positioning.

To increase model resolvability, one may want to increase z-grid density. XMF provides such feature, but with some restrictions: one is allowed to double density of the entire z-grid up to five times. The resulting density will be inherited by the target model. If it is excessive for lower layers, it may be decreased by Model Editor, which is capable of performing such operation locally.

The initial z-grid comes with  $V_0(z)$  from the reference model. On Velocity Editor's left margin (fig. 1) coefficient of density growth with respect to the initial is shown. The simplified density increase procedure (doubling), in particular, allows to control [congruence](#) of different local sections.

## 6. Ray-tracing

When comparing a computed TX-curve against a phase lineup on the seismic record, it is important to keep in mind TX-curve accuracy, that is, accuracy of ray-tracing. The problem of ray-tracing includes hitting the receiver points (RP). Only if a ray comes unambiguously at RP one can associate travel time with the event on a record trace. Situation with hitting RP depends on the type of a standard model.

*D-model.* Ray exit points are defined only by  $V^*(z)$ , that is by z-grid and velocity values. The only way "to hit RP" is to replace a ray exit point with the nearest RP and properly correct travel time. If number of rays is small (i.e. z-grid is too rare), such replacement can lead to undesirable visible effects.

*R-model* uses *regula falsi* method and secures hitting RP with accuracy of 10% of minimal distance between RP.

*H-модель.* Ray tracing method used secures exact RP hitting. That means that errors, if any, are of computational origin.

## Pre-study modules

### Overview

Functionality, described in the three first sections of the chapter, is implemented by the following XMF modules:

- *Pre-study Controller* (PSC) – pre-study supervisor; launched from Processing Tree menu called on the pre-study node (fig. 2a);

- *Pre-study Fitting Visualizer* (PFV) – visualizes seismogram record and TX-curves;
- *Standard Model Manager* (SMM) – main program for fitting a local section; implements user interface for definition of standard models, and editing them; solves forward problem, allows viewing standard model database; controls visualization;
- *Pre-study Ray Viewer* (PRV) – graphic module for ray picture display.

Module interrelations are shown on fig. 1. Relation  $A \rightarrow B$  means that  $A$  launches an instance of  $B$ ; relation  $A \bullet B$  is client-server relation. For example, SMM invokes PFV and PRV for visualization.

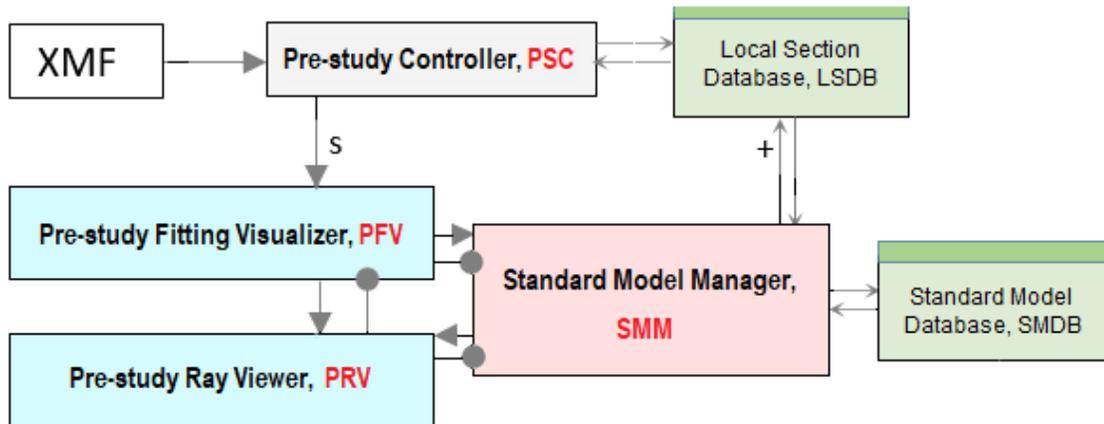


Fig. 1. Pre-study modules. Pre-study controller runs PRV for each shotpoint  $s$ ;  
SMM adds the built  $s$ -section to LSDB.

In this section, two programs – PSC and PFV – are discussed. SMM description takes up the next three sections. The last section is devoted to PRV.

**Pre-study Controller**

The program is partly discussed in [one](#) of previous sections. PSC launch is shown on fig. 1a.

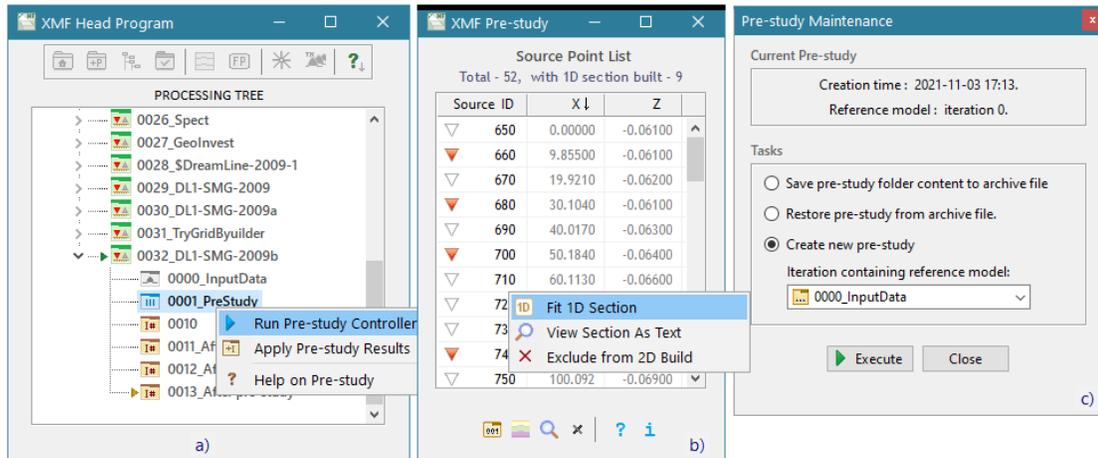


Fig. 2. Pre-study Controller; a) pre-study entry point in processing tree; b) главное окно контроллера pre-sudy; c) диалог Поддержка pre-study.

Main PSC window (fig. 2b) displays the list of project sources with their coordinates. Every shotpoint is provided with its *status in pre-study* imaged by an icon:

- ▽ – not used in pre-study;
- ▼ – s-section is built (*active* shotpoint);
- ▽ – s-section is built, but excluded from build of the target model.

Status information is extracted from Local Section Database. One manages the program with the list context menu and tool bar buttons under the list. The commands are described in table 1.

**Таблица 1. Команды управления Контроллера.**

Command	Description
	<i>Context menu</i>
<i>Fit 1D section</i>	Launches Pre-study Fitting Visualizer for the selected SP independently of its status; double-click on the item has the same effect. It is allowed to start several PFV instances for different shotpoints, which is helpful in comparing the results.
<i>View Section as Text</i>	Displays s-section textual representation of the selected SP.
<i>Exclude from Build</i>	Excludes active SP from the procedure of building target model.
	<i>Tool bar buttons</i>

	Invokes maintenance dialog (see below).
	Starts the procedure of building (partial) target model.
	Launches Model Viewer for (partial) target model preview. The preview is kept between program sessions.
	Closes the program.

**Maintenance dialog.** Dialog snapshot is shown on fig. 2c. If it s not the first pre-study of the project, the upper panel displays pre-study creation date and reference model address, that is, iteration number. Maintenance includes three operations:

- *creation of an archive file* of the 0001\_PreStudy folder; by default the file is placed in the project import-export folder; its name contains project ID and creation time; file format is not common, so it may be processed only in this dialog;
- *extraction of pre-study* content from an archive into the 0001\_PreStudy folder; its current content is destroyed;
- *creation of a new pre-study*; to do that, the user sets the reference model selecting an iteration it belongs to from the drop-down list; current folder content is destroyed.

To perform an operation, the user selects it with a radio-button and clicks on the *Execute* button.

**Pre-study Fitting Visualizer**

PFV is, basically, one-frame version of Line Seismogram Viewer ([LSV](#)). The frame displays a seismogram of the shotpoint selected in Pre-study Controller list. All LSV functionality of viewing seismic record and adjusting the view is fully kept. Below, main differences are listed.

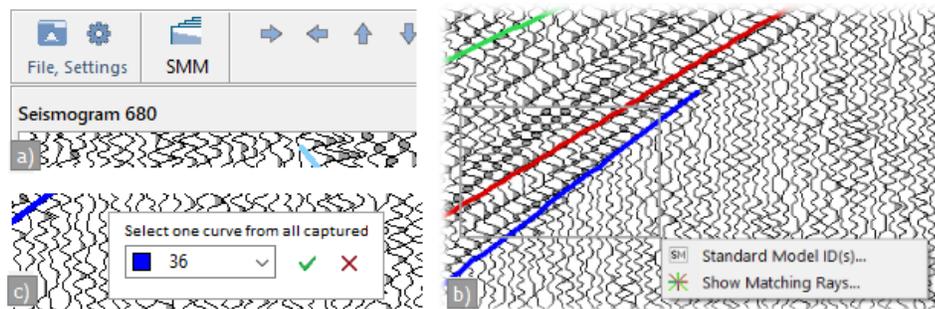


Fig. 3. Special PFV features; a) tool bar leftmost part; b) rubber-band menu; c) model ID dialog.

1. Because PFV displays only one seismogram, the commands of frame selection are absent, together with matching buttons and menu items. Left part of the tool bar looks as shown on fig 3a. The third tool bar button starts Standard Model Manager.

2. TX-curve drawing attributes, their visibility intervals, show/hide flags are controlled by SMM and so all these tools are removed.

3. To identify a curve in the image, one should capture its segment with the rubber-band and select one of two commands of the popup menu:

*Show Model ID(s)* – displays the list of IDs of curves whose points get inside the rubber-band (рис. 3с);

*Show Matching Rays* – invokes Ray Picture Viewer and makes it to display wave rays matching TX-curve points captured by the rubber-band; if points belong to different curves (that is, models), the user selects the model needed in the same dialog (fig. 3с).

4. The cursor context information includes, additionally, trace offset to make examination of seismic record and ray picture easier; the information appears in status bar and, on demand, in the hint window.

## SMM: Overview. Tools

Standard Model Manager (SMM) is a tool kit for fitting a local section including solving forward problems for standard models, managing standard model database (SMDB) and supporting proper user interface. For imaging forward problem solutions it invokes other pre-study modules: PFV and PRV. Studying SMM will take this and the following two sections. Here, module's main window is discussed and the full list of managing commands is presented.

### Main window

SMM is launched from Pre-study Fitting Visualizer (PFV), therefore, its main window is displayed over the seismogram image (fig. 1). This is convenient, because the two applications work together, as TV-set and remote control. To fix such relative position, check the *Stay on top* box at the bottom of the SMM window.

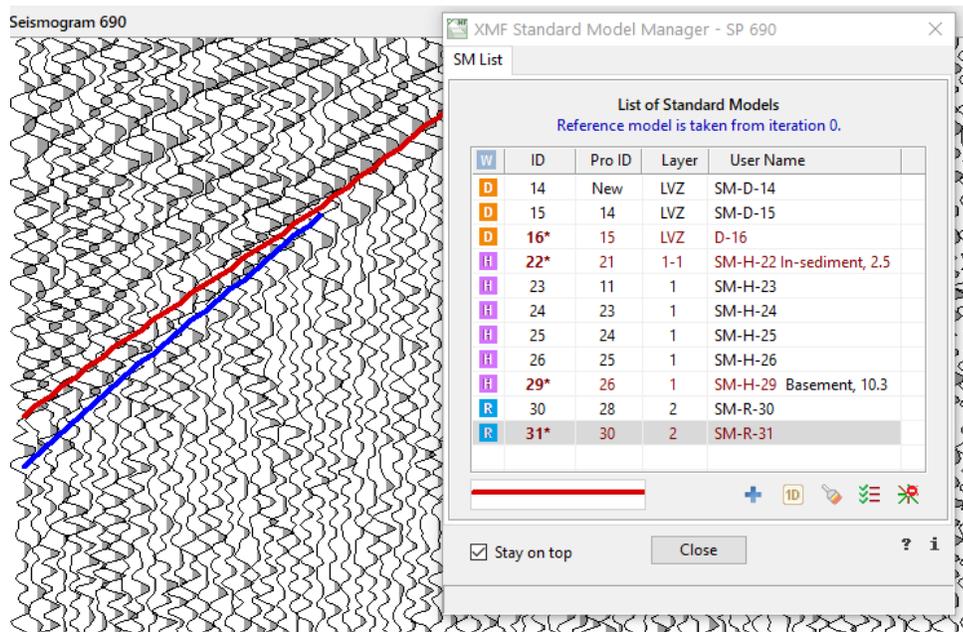


Fig. 1. SMM main window; standard model list display.

The main window has 4 tabs, but, if not in the process of creating a new standard model, only the first tab, tagged with *SM list*, is visible. It displays the list representing current content of SMDB. Under the list, one can see a panel with TX-curve view at the left and a tool bar at the right. The view panel relates to a list item currently selected in the list; a double-click on the view invokes the [dialog](#) for editing the curve drawing attributes.

The list is a table with the following columns:

- W* – standard model icon: D, R или H; icon color has no meaning;
- ID* – automatic model identifier, counts from 10;
- Pro ID* – prototype model ID, or *New*, or *Imp* [orted];
- Layer* – relative layer number (explained below);
- User Name* – model user name.

Each list item points to the SMDB data block relating to a standard model which stores model parameters and computed TX-curve. A model gets into the database and list only after TX-curve for the model is computed.

### Displaying fitting process

On fig. 1, the list displays the state of the database for shotpoint 690, according to the window's caption. Currently, fitting R-model for layer 2 is under way, and the last model iteration has ID 31. The ID is printed in bold and provided with the asterisk, which means that it is *marked*. Marked models (and only they) have their TX-curves drawn over the seismogram. Besides, local section is built only from the marked models. The default program settings work in the way that the last model iteration is marked automatically, while the previous is unmarked. Following this logic, a marked model is either a final

iteration for LVZ or a layer, as models 16, 22, 29, or the last one, as 31. The user can mark or unmark a model by a double-click. Respectively, model TX-curve is shown or hidden.

*Relative layer number.* It is not supposed, that a number of layers of the target model is a priori known. In the course of pre-study new (with respect to the reference model) horizons may be taken into account. Therefore, model list does not use ordinal layer numbers. Instead, it "counts" layers with respect to layers of the reference model. In the example on fig. 1, the *Layer* column contains LVZ, 1-1, 1, 2. In this succession, 1-1 and 1 mean that a new horizon is tracked within the layer 1 of the reference model, which becomes a floor of *sublayer* 1-1. Thus, a double number relates to a new sublayer, while a single one – to a layer of the reference model. If two new horizons appeared within the first layer of the reference model, the succession would be 1-1, 1-2, 1. Relative numbers are controlled by the user; they do not play an active part, they are needed to fix the order and are checked only before building the local section. After building, they are replaced with true ordinal numbers (see [example](#)).

Normally, process of fitting develops in the way, that the next iteration is created from the previous. The *Pro ID* shows that модель 14 is created anew, model 15 is created by correcting model 14, model 16 is created from model 15, model 23 – from model 11, which was removed. If removal were blocked, the entire fitting process for all layers would be displayed in the list.

According to logic of marking, unmarked models are rejected variants. There is no much sense to keep them, but, possibly, the last one, for the case of a failure. The removal command affects a continuous block of items. After 8-10 rejected iterations have been accumulated in the list, it is recommended to delete them simply to save disc space.

Each model is provided with automatic user name. It includes wave type and model ID. The default name can be edited straight in the table. On fig. 1, meaningful names are assigned only to the final layer model iterations.

### Local section

Local section is created from the marked R- or H-models. Fig. 2a shows the moment before that.

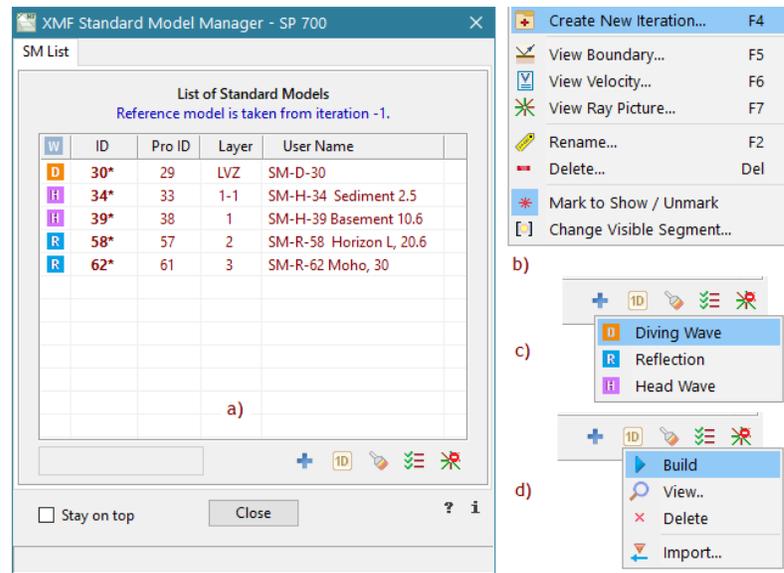


Fig. 2. Standard model list and SMM tools; a) list view before local section creation; b) list context menu; c) New SM button menu; d) ID section button menu".

All intermediate models are removed, though it not obligatory. The user issues the *Build* menu command (fig. 2d); SMM implements verification of layer numbering and consistency of velocity profiles; then, it builds the section and puts it to Local Section Database. As for velocity profiles, one should keep in mind, that if one rebuilds the  $k$ -th layer, one should rebuild layers with numbers  $j > k$ . Therefore, normally, model ID increases with layer depth.

Local section's velocity profile is that of the standard model for the last layer. D-model is not included in the local section, but its velocity  $V^*(z)$  is part of its velocity function.

**Import**

If SMM is launched for a shotpoint  $s$  for the first time, SMDB is empty and so is the list of standard models. According to pre-study technology, if the shotpoint is not leading, fitting begins with the import operation (last command of the button menu on fig. 2d). The user is allowed to select the source SP, whose SMDB will be used as a source of import. As recommended here, it is the nearest SP participating in pre-study. After the import command is issued, the list becomes (with some delay) populated with the imported standard models and looks like on fig. 1b with the following distinctions: D-model is absent; model IDs are 10 to 13. All models have the same velocity profile. The user begins with fitting D-model for LVZ (if any). Velocity prototype can be borrowed from any imported model. Then, successively, all imported model must be refined.

**Management**

In the below tables, short descriptions of each menu command and tool bar buttons is given.

**Table 1. Tool bar**

Button	Command description
	Initiates the process of creating a new standard model of the type selected in the drop-down menu (fig. 1c).
	Drops down menu for work with the local section (see fig. 2c and table 3).
	Redraws TX-curves of the marked models. Normally, there is no need to apply the command, because SMM updates the image automatically when necessary.
	Displays the settings dialog, which contains flags of marking and other switches.
	Removes all ray path files from the pre-study folder (to save disc space).

**Table 2. List context menu**

Command	Description
<i>Create New Iteration</i>	Initiates the process of creating a new iteration by means of correcting the selected model.
<i>View Boundary</i>	Displays boundary parameters of the selected model in a floating window; up to three windows can be displayed (for different models). Relative number of the layer is among the parameters and the only one that can be edited.
<i>View Velocity</i>	Displays model's velocity profile in a floating window of <a href="#">velocity editor</a> in view mode; up to three windows can be displayed (for different models).
<i>View Ray Picture</i>	Runs module Pre-study Ray Viewer ( <a href="#">PRV</a> ) to view model's ray picture. Up to three PRV instances can be displayed (for different models).
<i>Rename</i>	Turns the cell with model's user name into an edit field to change it. To end editing, apply keys: <i>Enter</i> to accept, <i>Esc</i> to cancel.
<i>Delete</i>	Removes model block with ID from [id1, id2] from SMDB. Models included in the local section are not removed (first delete the section). Values of id1 and id2 are set in a dialog, in which, by default, id1 = id2 = ID of the selected item.
<i>Mark / Unmark</i>	Marks the selected item or unmarks it. Respectively, shows or hides TX-curve on the seismogram.
<i>Change Segment</i>	<i>Visible</i> Changes TX-curve visibility interval. Invokes a dialog in which current visibility interval is shown together with the curve domain. That information is displayed separately for the forward and

	reverse TX-curve branches. Each interval has the form $[n1, n2]$ , where $n$ is trace number.
--	---

**Table 3. Work with local section** (see fig. 2d)

<i>Command</i>	Description
<i>Build</i>	Creates local section from the marked standard models and puts it to LSDB.
<i>View</i>	Displays local section textual description.
<i>Delete</i>	Removes local section from LSDB.
<i>Import</i>	Imports a stack of standard models from another shotpoint SMDB. In a dialog, the user can select a shotpoint to import from. It is recommended to select the nearest shotpoint. Being complicated, the operation takes some time.

## SMM: Standard model creation

### New model and new iteration

SMM offers two commands for creating a new standard model. The first button of the tool bar initiates the process of building a model from scratch. The user should select model type from the menu the button drops down ([fig. 2](#)).

The first command of the list context menu (*Create New Iteration*) begins correcting of the model currently selected in the list. In this case, model type and other parameters are known. The command start building a new model iteration.

After initiation, both cases differ insignificantly. SMM moves the user through the main window tabs as "wizard": boundary tab, velocity tab, forward problem tab. At that, tab tags are not active; to break the process, one applies the *Cancel* button which can be found on any tab.

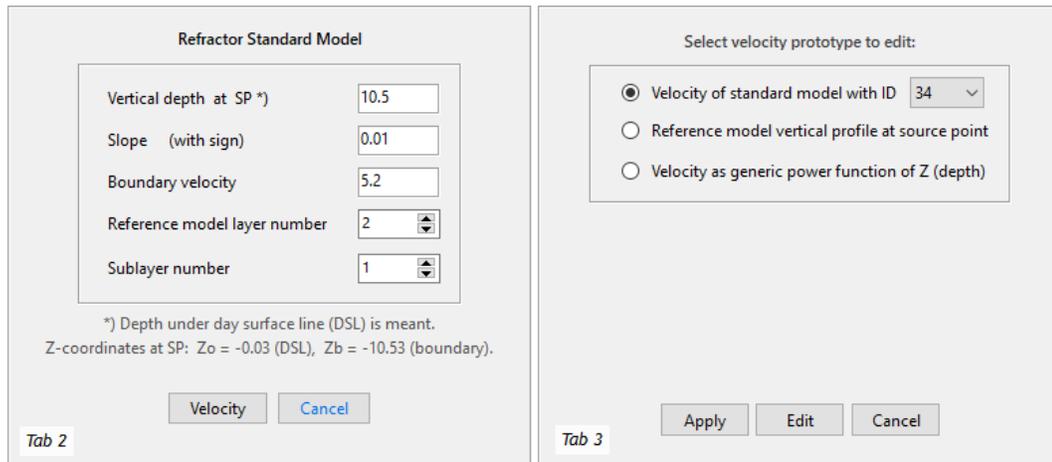


Fig. 1. SMM main window tabs; Tab 2 – boundary parameters; Tab 3 – selecting velocity prototype to apply or edit.

**Boundary**

If a new model is of type R or H, the first tab is #2 (fig. 1, left) – to set boundary parameters. In case of D-model this step is skipped. For a new model, all fields are nullified, layer and sublayer fields set to 1. Fig. 1 shows the case of new iteration, so all edit fields are filled with parameters of the selected model. If it is R-model, the boundary velocity field is blocked. When editing, one keeps in mind the following features.

The first field contains the value of boundary depth  $H_b$  under the day surface counting along s-vertical. In computations, however, not  $H_b$  but boundary z-coordinate  $Z_b$  at s-vertical is used. Z-coordinate of the day surface line  $Z_0$  is known (the first point of velocity profile) and so  $Z_b = Z_0 - H_b$ . Values of  $Z_0$  and  $Z_b$  are displayed in a footnote below the edit panel.

A boundary is positioned with accuracy of half-distance between z-grid nodes ( $\geq$ ). Before solving forward problem, SMM "rounds"  $Z_b$  to the value of the nearest node. One has to keep it in mind: *small depth increments do not change the problem and, hence, its solution; depth is discrete!* When boundary properties are displayed by the command *View Velocity*, true values of  $H_b$  and  $Z_b$  are shown, not those, one entered at the tab #2.

*Slope* is tangent of the acute angle the boundary makes with the positive direction of the OX axis. Angle is positive if a boundary rises in that direction. In practice, small angles are typical, so that slope is approximately equal to the value of the angle measured in radians. After editing, one clicks on the *Velocity* button to move to velocity definition.

**Velocity**

Velocity tab looks like on fig. 1-Tab3. One should select velocity prototype and either use it "as is" (the *Apply* button) or start editing it (the *Edit* button). Options for prototype selection:

- 1) velocity of a standard model with specified ID (is picked from a the drop-down list);
- 2) initial velocity profile  $V_0(z) \gg$ ;
- 3) power function  $V_{pow}(z)$ ;

**Default selection.** If a new iteration is being created, the option 1) is selected by default and ID is set to ID of the selected model. This is the right choice. One presses *Apply* или *Edit*. If a new model is being created, then if the model list is not empty, default selection is 1), but prototype ID is not fixed. This is the user responsibility. For example, if the first model for layer  $k$  is being created, one *must* select ID of the final model for layer  $k-1$ .

Options 2) and 3) are used for the [leading](#) s-section, when there is no prototypes. At the moment, when option 3) is selected, a new panel appears on the tab (fig. 2-Tab3).

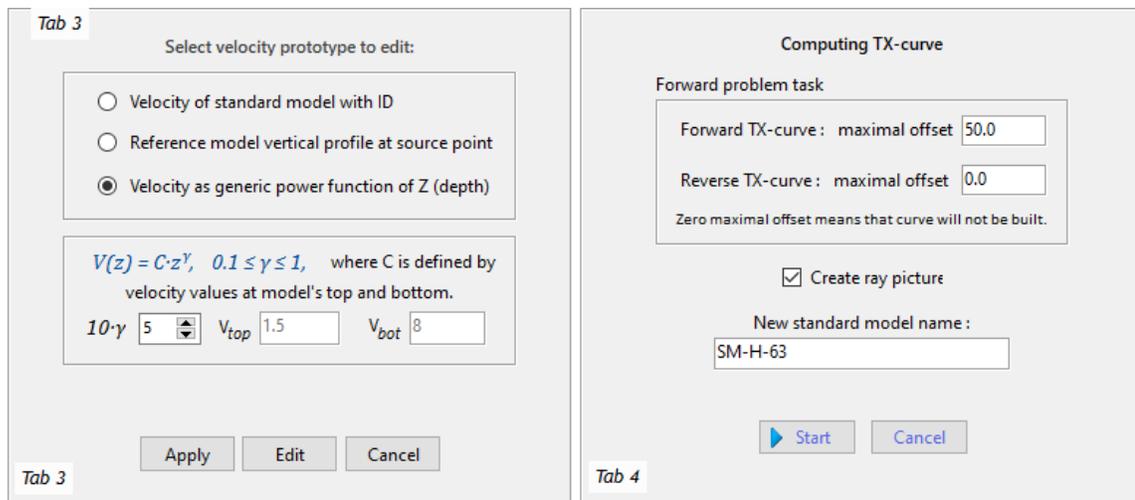


Fig. 2. SMM main window tabs; Tab 3 – power function parameters;  
Tab 4 – forward problem task.

Its purpose is to specify power function parameters. Function  $V(z) = C \cdot z^\gamma$  has two parameters: power exponent  $\gamma$ ,  $0 < \gamma \leq 1$  and constant  $C$ . Instead of  $\gamma$ , one selects integer  $10\gamma$  in the spin edit field. Instead of  $C$ , one defines velocity values at velocity profile top and bottom. By default, they are borrowed from  $V_0(z)$ . If one selects 10 in the spin-edit, velocity prototype is linear function.

If velocity prototype definition ends with click on the *Apply* button, the last tab opens. The *Edit* button invokes velocity editor (see next section).

### Solving forward problem

By the moment one gets to this tab, a new standard mode is defined. Now forward problem (FP) for the model is to be solved. FP task include:

- TX-curve domain definition,
- ray picture creation flag;
- model user name.

The curve domain is defined by maximal offsets for the forward and reverse branches. For the case of new iteration, the domain of the model-prototype is displayed in edit fields. For a new model, maximal; offsets must be typed in. Trace offsets are shown in the status bar of PFV module. If one of offsets is set to 0, the curve branch is not computed. If boundary slope is zero, it is, formally, enough to compute one branch only, because velocity does not depend on offsets. However, one must take into account that seismic record quality may differ for different record areas and this is more important than saving on symmetry.

There is no need to set ray picture flag every time. It is included in program's settings and can be set once. By default, it is set. Note, that rays are traces always, the flag defines whether they are saved to ray file or not.

The default model name can be accepted and changed later in the model list, if necessary. The *Start* button starts ray-tracing. Results are displayed in a dialog (fig. 3).

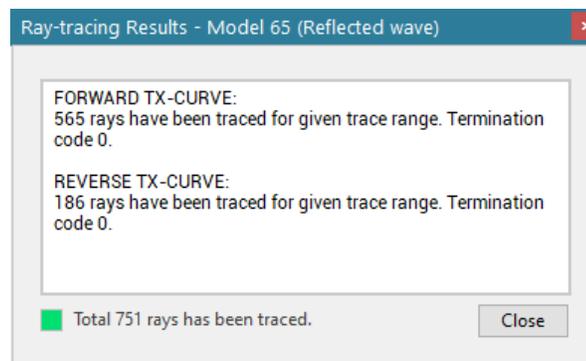


Fig. 3. Results of ray-tracing.

In the window, a number of traced rays and termination codes are displayed separately for forward and reverse branches. If total of traced rays is nonzero, the indicator at the bottom is green and the operation is regarded a success, even if one or both termination code are nonzero. A click on the *Close* button starts adding the new model and TX-curve to SMDB. It may take some time, as well as drawing TX-curve for the first time. After that, all tabs but the main close, and one sees a new item in the list.

If total of traced rays is zero, the indicator is red which means failure. After closing the dialog, model data are lost.

### Ray-tracing error

If  $V(z)$  is not constant, ray-trace errors may occur and nonzero termination codes. In R-models, it may happen that only part of rays can be traced, because at some starting angle, a ray may become critical before it reaches the boundary (termination codes 136–139). At that, all traced rays can be viewed in ray viewer and TX-curve is displayed on the seismogram. At large offsets, the necessary accuracy of hitting a receiver point may not be achieved. (152–155).

It happens comparatively often that the red indicator lights up for H-models. The reason is that critical ray exists only under special condition on  $V(z)$  above the boundary, velocity boundary and slope values. If a critical ray is not found, termination code is 160. Usually that means that the required refractor does not exist.

At last, user can make a mistake in model definition: a boundary proves to come out of  $V(z)$  domain or  $V(z)$  above the boundary is not less than boundary velocity at some points.

## SMM: Velocity editor

[Main window](#) – [Мобильная метка](#) – [Поточечное редактирование](#) – [Другие операции редактирования](#) – [Контекстное меню](#) – [Удвоение плотности z-сетки](#) – [Скорость как числовая таблица](#)

### Editor main window

Velocity editor (VE) is invoked as dialog in the process of creating of a new model. Also, it has view mode, and in this mode it is invoked by the *View velocity* command of the model list menu. In view mode, VE has no the tool bar with editing command (fig. 1a).

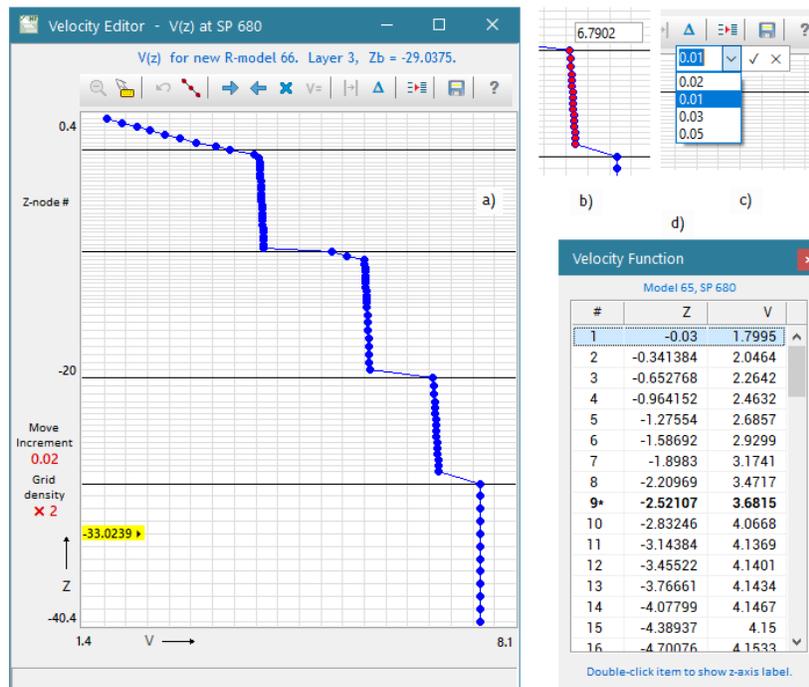


Fig. 1. Velocity editor; a) VE main window; b) assigning a value; c) changing move increment; d) velocity as table.

VE is a graphic module, in a way, similar Wireframe Editor, but VE works with the only curve  $V = V(z)$  – velocity profile. On fig. 1, the curve is drawn over a grid. Grid horizontal lines go through the nodes of the velocity z-grid. Some horizons are drawn in black. Black horizons have equations  $z = Z_b$  ( $\gg$ ) and go through the intersection points of boundaries the s-вертикаль. We speak of of boundaries of marked models.

VE is provided with the rubber-band which is used to 1) zoom in the picture; 2) select a curve segment for a group edit operation.

Clearly, the curve points lie on the horizontal grid lines. Though the curve is, by default, drawn as continuous function, it is constant in each microlayer ( $\gg$ ). One will see how to display the true  $V(z)$  view.

The cursor context information is shown on the bottom margin (real coordinates) and on the left margin at its top (z-grid node number counting from top). The left margin displays two LE parameters: 1) velocity increment, used by arrow-buttons and 2) ratio of current grid density to that of initial profile  $V_0$  ( $\gg$ ). The parameters are called, respectively, *move increment* and *grid density*. On the upper margin, above the tool bar, one can see boundary parameters of the model being currently created.

**Note.** The editor does not block changing velocity at any point. But one must remember, that according to layer-by-layer interpretation principle, one must not change velocity above the roof of the layer being currently fitted. ( $\gg$ ).

### Mobile label

Though there are only three labels on the OZ axis, LE is provided with *mobile label* (yellow field on fig. 1a). It is invoked by *Ctrl+Alt+Click* and has the following features:

- when invoked, it appears at the left edge of the horizontal nearest to the clicked point, displaying its z-coordinate; its small arrows point lies exactly on the line;
- the label can be moved using the same combination *Ctrl+Alt+Click* or drag-and-drop; in any case, the label points exactly at a grid line, the label moves discretely along z-nodes;
- once invoked, the label stays visible while VE is active.

### Pointwise editing

Double-click on the drawing pad creates a curve point on the nearest horizontal. Double-click with Shift key downed removes the point from the nearest horizontal. At that, the curve proves to be defined partly, with some points absent. That is allowed and, moreover, is very convenient. Absent point are restored by a command which uses linear interpolation. In this way, LE offers a fast tool of creating and editing of a piecewise linear function. The first and last curve points cannot be removed.

### Other edit operation

They are collected in the VE tool bar and are shortly described in the following table.

Таблица 1. Панель инструментов

Button	Description
	Zooms out the image.
	Shows/hides hint window with context information.
	Cancel several last changes.
	Restores all absent points using linear interpolation between nearest existing points.
	Moves selected segment or entire curve right or left, that is increases or decreases velocity. One click moves the curve by the value of <i>move increment</i> which is shown on the left LE margin.
	Delete points of a selected segment.
	Assigns user-defined velocity value to all points of a selected segment. The edit field appears near the first segment point (рис.1b). Keys <i>Enter</i> (OK) and <i>Esc</i> (cancel) end editing.
	Takes away segment selection. Another way is to double-click on the pad.
	Allows changing move increment used by arrow-buttons. Displays a dialog under the button with the drop-down list of existing values (fig. 1c). A new

	value is either selected from the list or typed in the list's edit field. After clicking on the ✓ button the value, if absent, is added to the list.
	Doubles z-grid density (see below in this section).
	Saves velocity profile, closes VE and if a new model is being created, opens the last SMM tab. If there are absent curve points, they are automatically restored.

### Drawing pad context menu

These command can be used both in view and edit mode.

Таблица 2. Команды контекстного меню

Command	Description
• <i>Continuous View</i> <i>Stepwise View</i>	This is the switch between continuous and stepwise view of $V(z)$ . The first is convenient, the second is true.
<i>Zoom out</i>	Zooms out the image.
<i>Hint Window</i>	Shows/hides hint window with context information.
<i>Velocity as Table</i>	Displays velocity in the form of numeric table (see below).
<i>Paste And Select</i>	Selects curve segment whose bounds are taken from the Window clipboard. This is a way of exchange with ray view, see next <a href="#">section</a> .
<i>Drawing Attributes</i>	Allows changing curve drawing attributes in the <a href="#">standard dialog</a> .

### Z-grid density duplication

As discussed [here](#), the operation increases model resolvability. The operation cannot be undone with button . To double density, it is necessary to define velocity value at each new node  $Z$  between the previous node  $Z^-$  and the next one  $Z^+$ . SMM offers two options: 1) linear interpolation,  $V = (V^- + V^+)/2$ ; 2) leave the previous value,  $V = V^-$ . In the first case, velocity in every old microlayer changes, which implies changing ray picture too. In the second case, velocity stays intact. One selects one of two options in a dialog after issuing the command. The operation can be repeated up to 5 times.

### Velocity table

Mostly, editing is performed graphically. But sometimes one has to be sure that  $V(z)$  takes the needed values exactly. For that aim, the assignment command **v=** is introduced. On the same reason, one can view  $V(z)$  as a numeric table with the *Velocity as Table* menu command (рис. 1d). The table has columns #, Z, V, where # is z-grid node number. Lines displayed in bold correspond to black lines on the image. In same way, velocity profile of 2D model is displayed in Model Viewer. Double-click on a table line moves the mobile label to the matching horizontal.

## Ray picture

The purpose of Pre-study Ray Viewer (PRV) is to draw rays computed by forward problem solver and saved to a ray file. PRV is a graphic module with the usual features (fig. 1). The domain of the picture is defined, horizontally, by TX-curve domain and, vertically, – by  $V(z)$  domain. Ray picture usually takes up the upper part of the domain. For D-model it is bound from below by the maximal depth of diving wave ray penetration. For R- and H-models – by reflector or refractor. Note, that velocity is included in the cursor context information and can be seen in the status bar.

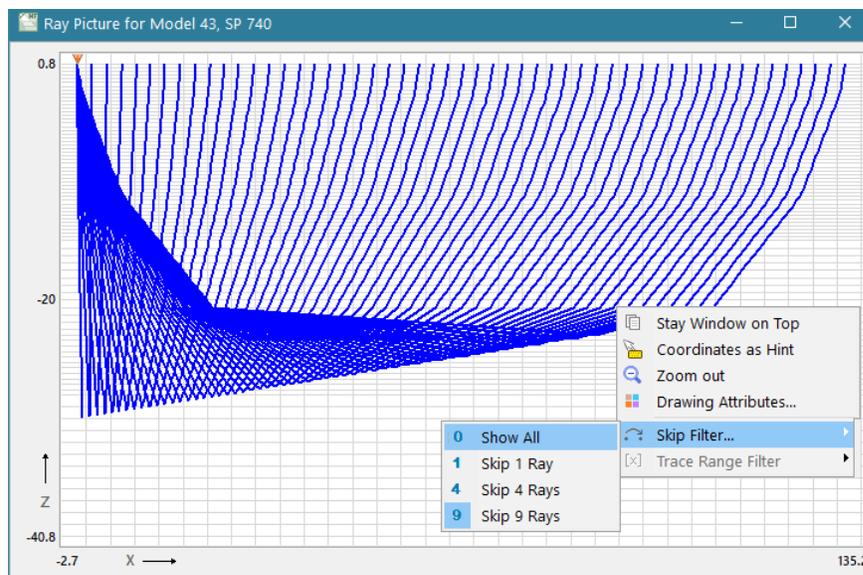


Fig. 1. RPV Main window with dropped down menus.

The rubber-band menu includes a very special command *Copy Z-span*. The command copies z-coordinates of upper ( $z_u$ ) and lower ( $z_l$ ) rubber-band sides to the Windows clipboard. These values can be retrieved by velocity editor's command *Paste And Select* which selects the  $[z_l, z_u]$  segment of the velocity profile. This feature helps in fitting velocity, especially in the case of D-model. The question "how velocity is to be changed to bring about the required change of ray picture" is in the background.

The picture view can be controlled by the context menu.

Таблица 1. Команды контекстного меню планшета

Command	Description
<i>Stay on Top</i>	Makes PRV window a topmost one.

<i>Coordinates Hint</i>	as Shows or hides the hint window with coordinates and velocity values.
<i>Zoom out</i>	Zooms out the picture.
<i>Drawing Attributes</i>	Allows changing ray drawing attributes in the <a href="#">standard dialog</a> .
<i>Skip Filter</i> <i>Show All</i> <i>Skip 1 Ray</i> <i>Skip 4 Rays</i> <i>Skip 9 Rays</i>	Raring filter. Open submenu with the commands: Shows all rays Skips 1 ray, i.e. shows every 2nd ray, while drawing; Skips 4 ray, i.e. shows every 5th ray Skips 9 ray, i.e. shows every 10th ray On fig. 1, each 10th ray is shown.
<i>Trace Range Filter</i> <i>Hide Filtered off</i> <i>Select Passed</i>	This filter is set not by the user, but by the <a href="#">PFV</a> module, which makes PRV to show only rays matching TX-curve points within the rubber band on the seismic record. These two commands switch between two option: – hides rays which are filtered off; – show all rays but select those passed by the filter.

## APPENDICES

### 1. Seismic File Format

The term SEG-Y/PC is used for a seismic file format, which follow the SEG-Y Standard with the only exception of ordering bytes in a word storing a number. The Standard requires that the first byte represent the higher bits of a number, which true for UNIX systems. The x86/Windows platform uses the opposite representation of bytes in memory. Therefore, software designed for Windows cannot directly read a file of the true SEG-Y. Real time converting would be too time-consuming. Thus, it is supposed that seismogram are converted to SEG-Y/PC beforehand.

Below it is explained how file and trace headers are interpreted by XMF. In the user interface, a notion of header *information word* (IW) is used. IW is a portion of header bytes with meaningful information. It can be of 2 to 6 byte length. In the following tables file, and trace header layouts are presented with partial omission of IWs not used by XMF.

#### File binary header layout

Table 1.

XMF interprets only words in light-blue cells.

IW	Bytes	Description
1	3201-3204	Job identification number.
2	3205-3208	Line number. For 3-D poststack data, this will typically contain the in-line number.
3	3209-3212	Reel number.
4	3213-3214	Number of data traces per ensemble. Mandatory for prestack data.
5	3215-3216	Number of auxiliary traces per ensemble. Mandatory for prestack data.
6	3217-3218	Sample interval in microseconds ( $\mu$ s). Mandatory for all data types.
7	3219-3220	Sample interval in microseconds ( $\mu$ s) of original field recording.
8	3221-3222	Number of samples per data trace. Mandatory for all types of data. Note: The sample interval and number of samples in the Binary File Header should be for the primary set of seismic data traces in the file.

IW	Bytes	Description
9	3223-3224	Number of samples per data trace for original field recording.
10	3225-3226	Data sample format code. Mandatory for all data. 1 = 4-byte IBM floating-point 2 = 4-byte, two's complement integer 3 = 2-byte, two's complement integer 4 = 4-byte fixed-point with gain (obsolete) 5 = 4-byte IEEE floating-point 6 = Not currently used 7 = Not currently used 8 = 1-byte, two's complement integer
11	3227-3228	Ensemble fold — The expected number of data traces per trace ensemble (e.g. the CMP fold). Highly recommended for all types of data.
	.....	
28	3501-3502	SEG Y Format Revision Number. This is a 16-bit unsigned value with a Q-point between the first and second bytes. Thus for SEG Y Revision 1.0, as defined in this document, this will be recorded as 010016. This field is mandatory for all versions of SEG Y, although a value of zero indicates “traditional” SEG Y conforming to the 1975 standard.
29	3503-3504	Fixed length trace flag. A value of one indicates that all traces in this SEG Y file are guaranteed to have the same sample interval and number of samples, as specified in Textual File Header bytes 3217-3218 and 3221-3222. A value of zero indicates that the length of the traces in the file may vary and the number of samples in bytes 115-116 of the Trace Header must be examined to determine the actual length of each trace. This field is mandatory for all versions of SEG Y, although a value of zero indicates “traditional” SEG Y conforming to the 1975 standard.
30	3505-3506	Number of 3200-byte, Extended Textual File Header records following the Binary Header. A value of zero indicates there are no Extended Textual File Header records (i.e. this file has no Extended Textual File Header(s)). A value of -1 indicates that there are a variable number of Extended Textual File Header records and the end of the Extended Textual File Header is denoted by an ((SEG: EndText)) stanza in the final record. A

IW	Bytes	Description
		positive value indicates that there are exactly that many Extended Textual File Header records. This field is mandatory for all versions of SEG Y, although a value of zero indicates “traditional” SEG Y conforming to the 1975 standard.
	.....	

### Trace header layout

Table 2.

IW in light-blue cells are used always. Light-red cells contain coordinate information and are NOT used, if positional data are imported from SRR files.

Слово	Байты	Описание	
1	1-4	Trace sequence number within line — Numbers continue to increase if the same line continues across multiple SEG Y files. Highly recommended for all types of data.	
2	5-8	Trace sequence number within SEG Y file — Each file starts with trace sequence one.	
3	9-12	Original field record number. Highly recommended for all types of data.	
4	13-16	Trace number within the original field record. Highly recommended for all types of data.	
	.....		
12	37-40	Distance from center of the source point to the center of the receiver group (negative if opposite to direction in which line is shot).	
13	41-44	Receiver group elevation (all elevations above the Vertical datum are positive and below are negative).	The scalar in Trace Header bytes 69-70 applies to these values. The units are feet or meters as specified in Binary File Header bytes 3255-3256). The Vertical Datum should be defined through a Location Data stanza
14	45-48	Surface elevation at source.	
15	49-52	Source depth below surface (a positive number).	
16	53-56	Datum elevation at receiver group.	
17	57-60	Datum elevation at source.	
18	61-64	Water depth at source.	

Слово	Байты	Описание	
19	65-68	Water depth at group.	
20	69-70	Scalar to be applied to all elevations and depths specified in Trace Header bytes 4168 to give the real value. Scalar = 1, +10, +100, +1000, or +10,000. If positive, scalar is used as a multiplier; if negative, scalar is used as a divisor.	
21	71-72	Scalar to be applied to all coordinates specified in Trace Header bytes 7388 and to bytes Trace Header 181-188 to give the real value. Scalar = 1, +10, +100, +1000, or +10,000. If positive, scalar is used as a multiplier; if negative, scalar is used as divisor.	
22	73-76	Source coordinate - X	The coordinate reference system should be identified through an extended header Location Data stanza (see section D-1). If the coordinate units are in seconds of arc, decimal degrees or DMS, the X values represent longitude and the Y values latitude. A positive value designates east of Greenwich Meridian or north of the equator and a negative value designates south or west.
23	77-80	Source coordinate - Y	
24	81-84	Group coordinate - X	
25	85-88	Group coordinate - Y	
	.....		
39	115-116	Number of samples in this trace. Highly recommended for all types of data.	
40	117-118	Sample interval in microseconds ( $\mu$ s) for this trace. The number of bytes in a trace record must be consistent with the number of samples written in the trace header.	
	.....		

### Requirements

1. The following sample format are permitted: 2-byte integer (I2); 4-byte integer (I4); 4-byte floating-point (R4).
2. If sample format is R4, representation of a floating-point number must follow the IEEE 754 Standard. At that, format code in word 10 of the file header must be set to 5; however, the value of 1 is allowed (for very old files).
3. All traces of a file are assumed to be actual (bytes 29-30 of a trace header are not interpreted).

4. All file traces have the same number of samples.

### Using coordinate information

The next table shows, how XMF utilizes coordinate information from trace header.

Table 3.

<n> means value of the word #n.

Positional data	IW	Bytes
Source field ID	3	9 - 12
Source X	22	73 - 76
Source Y	23	77 - 80
Source Z	= <14> - <15>	45 - 48
Receiver X	24	81 - 84
Receiver Y	25	85 - 88
Receiver Z	13	41 - 44

Scale coefficients in words 20 and 21 are not used. Coordinates are extracted as integers; distance unit is assumed to be meter (conditionally).

### Textual header(s)

Main textual header and additional text headers, if any, are interpreted as ASCII text.

## 2. ASCII File Formats

### SRR files

The files contain [positional data](#) of the observation system. File with the ".src" extension describes shotpoints, file with the ".rcv" extension – receiver points, file with the ".ray" extension lists pairs "source – receiver/trace". Examples of the three files are shown on fig. 1. All files contain tables. The first line consists of column names and is not interpreted, when file is being read. The SRC file lists sources as table with columns:

Src #	Field ID	X	Y	Z	File name	Rcv #	X	Y	Z	Src #	Trace1	Rcv1	Trace2	Rcv2
1	650	649993	8	-61	smg650z.sgy	1	653476	330	-30	1	1	1	1240	1240
2	660	659848	4	-61	smg660z.sgy	2	653735	312	-30	2	1	1	1240	1241
3	670	669914	9	-62	smg670z.sgy	3	653978	271	-30	3	1	1	1240	1240
4	680	680097	4	-61	smg680z.sgy	4	654228	205	-30	4	1	11	1240	1251
a) 5	690	690010	1	-63	smg690z.sgy	b) 5	654471	120	-30	c) 5	1	11	1240	1251
6	700	700177	2	-64	smg700z.sgy	6	654694	30	-30	6	1	11	1240	1251
7	710	710106	-10	-66	smg710z.sgy	7	654887	57	-30	7	1	15	1240	1255
8	720	720014	-6	-66	smg720z.sgy	8	655054	25	-30	8	1	18	1240	1258
9	730	729900	-5	-65	smg730z.sgy	9	655242	121	-30	9	1	20	1240	1260
10	740	739940	-6	-67	smg740z.sgy	10	655462	192	-30	10	1	20	1240	1260
11	750	750085	6	-69	smg750z.sgy	11	655708	217	-30	11	1	22	1240	1262

Fig. 1. SRR files: a) SRC file; b) RCV file; c) RAY file.

- *Src #* – source ordinal number in file;
- *Field ID* – source field ID (also seismogram ID);
- *X, Y, Z* – coordinates (integers);
- *File name* – seismogram file name (without path).

RCV file table is similar, but without the "Field ID" column. Ordinal numbers are temporary identifiers and are used for references in RAY file at input time only. If coordinate system is two-dimensional, Y is redundant coordinate; the "Y" column must be included in file but is ignored and may be filled with zeros.

RAY file lists for each source intervals of trace numbers and matching intervals of receiver numbers. Interval of trace numbers [Trace1, Trace2] must correspond exactly to interval of receiver ordinal numbers [Rcv1, Rcv2]. Fig. 1c demonstrates the simplest case, when only one interval is needed. However, if a seismogram *Q* with *N* traces has a block of too noisy traces [T1, T2], then the RAY file will contain at least two lines for source *Q*, for example:

Q	1	T1-1	1	R1
Q	T2+1	N	R2	R3

An irregular observation system may deliver various complications of RAY files, but it can be composed always.

**Model and velocity file formats**

XMF uses two such formats – LMF and DAT – both for export. LMF (Layered Model File) is the own own XMF format, containing all information on the model (fig. 2). File LMF describes model similar to the dialog [Model Numeric View](#); the difference is that the columns follow one after another vertically. Lines beginning with "//" are ignored. The first line contains number of grid columns, number of h-lines and row numbers for rows underlying the horizons.

```
//Structure
196,58,3,41,47,53
//Col Row      X      Z      V
  1  1      -1.1  -0.03    1.5
  1  2      -1.1  -0.062573  1.65
  1  3      -1.1  -0.0951459  1.7
  1  4      -1.1  -0.127719  1.8
.....
  1  58      -1.1  -38.0068    8
  2  1  1.67092  -0.03    1.5
  2  2  1.67092  -0.062573  1.65
  2  3  1.67092  -0.0951459  1.7
  2  4  1.67092  -0.127719  1.8
.....
```

Fig. 2. LMF file format.

The DAT is a known format representing a function of two variables; it is an input format of the presentation graphic system Surfer®. In XMF, it is used for exporting model velocity V(x, z) in the form of the table "X-Z-V". If one deletes from an LMF file on fig. 2 first three lines and first two columns, what is left is a DAT file.

**File for a set of curves**

The MG format (Model Geometry) is used in XTomo-LM to store one or several curves. XMF uses it for import of wireframe curves.

	Curve #	X	Z	197		
	1	-1.1	-13.39789			-1.1 -13.39789
	1	1.670918	-13.34247			1.670918 -13.34247
	1	4.441836	-13.28705			4.441836 -13.28705
	1	7.212755	-13.23164			7.212755 -13.23164
	1	9.983673	-13.17622			9.983673 -13.17622
	.....					
a)	1	542	-13.75722	b)	539.2291	-13.75749
	2	-1.1	-21.73779	197		
	2	1.670918	-21.73779			-1.1 -21.73779
	2	4.441836	-21.73779			1.670918 -21.73779
	2	7.212755	-21.73779			4.441836 -21.73779
	2	9.983673	-21.73779			7.212755 -21.73779
	.....					

Fig. 3. Format of files for a set of curves: a) MG file; b) BLN file.

An MG file is a table "Curve number – X – Z", as on fig. 3a. Curve points are sorted by X. The BLN format is an input format of the Surfer® system. BLN file consists of a set of tables "X-Z", separated by a line with a number of points of the curve below.

### 3. Error Codes

A few modules inform of errors referring to error codes, without details. Those details usually are not important for the user or are beyond the scope his or her capability to correct it. Most important errors of this kind are gathered in this topic.

#### License Manager

This XMF component watches over legality of software use. If an error occurs, License Manager blocks access to the software and displays error message "Access denied" with reference to error code. Below the selected codes are explained:

- 107 – dongle (USB key) does not respond;
- 111 – dongle driver not found;
- 133 – dongle driver service is not launched;
- 150 – license file not found;
- 150 – 159 – errors occurred while reading license file.
- 178 – 185 – license file does not match product release;
- 186 – product was not properly installed;
- 190 – allowed number of trial version installations is exceeded;
- 192 – 194 – time of use or allowed number of runs of trial version are exhausted;
- 200 – 209 – installation errors.

#### SEG-Y file open errors

- 1 – file not found;
- 6 – file is too small to be a SEG-Y file;
- 9 – too large number of traces (possibly, format error);
- 12,13 – impossible to interpret file header word;
- 14 – file of revision 1 with variable length of textual headers;
- 15 – number of samples in trace header differs from the value declared in file header.

#### XMF Task Monitor

TMR displays error codes too, but it is mostly system or internal error. Those codes are rather for the support service.

### 4. Example of s-section

1D SECTION AT SOURCE POINT 680

Source point (SP) location:  $X_s = 30.104$ ,  $Z_s = -0.061$ .  
Reference model location: iteration 0.

---

Layer #	SM ID	SM type	Horizon Depth	Horizon Z at SP	Horizon slope	Boundary velocity
1	32	H	2.49107	-2.52107	0	3.68
2	36	H	10.5871	-10.6171	0	5.25
3	62	R	20.5993	-20.6293	0	
4	64	R	29.0075	-29.0375	0	

---

'SM' stands for 'Standard model'.

Horizon in the i-th line is a floor of the i-th layer.

Depth counts from day surface, i.e. 2D model top h-line at SP.

#### VELOCITY FUNCTION V(z)

---

#	Z	V
1	-0.03	1.7995
2	-0.341384	2.0464
3	-0.652768	2.2642
4	-0.964152	2.4632
5	-1.27554	2.6857
6	-1.58692	2.9299
7	-1.8983	3.1741
8	-2.20969	3.4717
9*	-2.52107	3.6815
10	-2.83246	4.0668
11	-3.14384	4.1369
12	-3.45522	4.1401
13	-3.76661	4.1434
14	-4.07799	4.1467
15	-4.38937	4.15
16	-4.70076	4.1533
17	-5.01214	4.1566
18	-5.32353	4.1598
19	-5.63491	4.1631
20	-5.94629	4.1664
21	-6.25768	4.1697
22	-6.56906	4.173
23	-6.88045	4.1763
24	-7.19183	4.1796
25	-7.50321	4.1828
26	-7.8146	4.1861
27	-8.12598	4.1885
28	-8.43737	4.1908
29	-8.74875	4.1931
30	-9.06013	4.1955
31	-9.37152	4.1978
32	-9.6829	4.2002
33	-9.99428	4.2025
34	-10.3057	4.2049
35*	-10.6171	5.2492

---

36	-10.9284	5.4687
37	-11.2398	5.7482
38	-11.5512	5.7511
39	-11.8626	5.7539
40	-12.174	5.7567
41	-12.4854	5.7596
42	-12.7967	5.7624
43	-13.1081	5.7653
44	-13.4195	5.7681
45	-13.7309	5.7709
46	-14.0423	5.7738
47	-14.3537	5.7766
48	-14.665	5.7794
49	-14.9764	5.7823
50	-15.6045	5.788
51	-16.2326	5.7937
52	-16.8607	5.7995
53	-17.4888	5.8052
54	-18.1169	5.8109
55	-18.745	5.8166
56	-19.3731	5.8224
57	-20.0012	5.8281
58*	-20.6293	6.7902
59	-21.2574	6.8068
60	-21.8855	6.8136
61	-22.5136	6.8204
62	-22.9743	6.8254
63	-23.4349	6.8303
64	-23.8955	6.8353
65	-24.3561	6.8403
66	-24.8167	6.8452
67	-25.2773	6.8502
68	-25.7379	6.8552
69	-26.1985	6.8602
70	-26.6591	6.8651
71	-27.1197	6.8701
72	-27.5803	6.8751
73	-28.0409	6.8801
74*	-29.0375	7.5076
75	-30.0341	7.5086
76	-31.0307	7.5095
77	-32.0273	7.5105
78	-33.0239	7.5114
79	-34.0205	7.5124
80	-35.0171	7.5134
81	-36.0136	7.5143
82	-37.0102	7.5153
83	-38.0068	7.5163
84	-39.0034	7.5172
85	-40	7.5182

-----  
Velocity column vertical range is defined by reference model.  
Line numbers with asteriscs match horizons.



# Index

## - 1 -

1D section 67

## - A -

AGC 26

## - B -

binding waves to layers 39

black cells. 49

## - C -

column editor 30

congruence 68

conversion coefficient 8, 40

conversion flags 8, 40

## - D -

default settings (LSV) 28

drawing attributes 38, 61

## - E -

export 62

    graphic 47

    SRR 22

    text 47

## - F -

file

    BLN 47, 62, 99

    BMP 48

    DAT 47, 98

    GIF 48

    JPG 48

    LMF 47, 98

    MG 47, 99

    PNG 48

    SEG-Y 93

    SEG-Y/PC 93

    seismic 93

    SRR 97

forward problem

    log 58

    solving 57

    task 56

frame

    rubber-band 63

    selected 26

    vertex 23

## - G -

gain 26

grid

    density 7

    standard subsets 5

## - H -

head program 15

hint window (MOV) 46

## - I -

information word 25

initial model 29

initial view 23

iteration

    creation 14

    folder 10

    menu 14

    user notes 14

## - L -

layer velocity 6

layer-by-layer principle 71

leading section 69

line coordinates 2

local section 67

    congruence 68

Local Section Database 68

LSDB 68

LVZ 68, 72

## - M -

model

    domain 4, 29

- 
- model
    - editing 51
    - image 41
    - initial 29
    - layer 4
    - numeric view 46
    - reference 67
    - target 67
    - wireframe 4
  - model wireframe 33
  - module 15
    - Forward Problem Solver 57
    - FPS 57
    - Line Seismogram Viewer 23
    - LSV 23
    - Model Viewer 41
    - Ray Picture Viewer 59
    - RPV 59
    - Wave-Layer Binder 39
  - N -**
  - normalizing 26
  - P -**
  - positional data 2
  - Positional Database 22
  - pre-study 68
    - concepts 67
    - folder 69
    - PFV 74
    - PSC 74
    - RPV 74
    - SMM 74
    - technology 69
  - Pre-study Controller 69
  - processing tree
    - iteration node menu 14
    - project node menu 12
    - work folder node menu 12
    - zero iteration menu 13
  - project
    - creation as a copy 20
    - properties 20
  - R -**
  - range mode 24
  - Raw Positional Data Cache 21
  - Ray Catalog 58
    - ray sample 60
  - ray sample 60, 65
  - reducing 26
  - reference model 67
  - resolution 3
  - rubber-band 41
  - S -**
  - SEG-Y headers 93
    - information word 25
  - SEG-Y/PC 93
  - Seismic Multi-Viewer 23
  - selected frame menu 28
  - SMDB 68
  - spectrum 49
  - SP-shotpoint 67
  - SRR files 2, 21, 22, 97
  - s-section 67
    - leading 69
  - standaer model
    - creation 83
  - standard model
    - database 68
    - D-model 68
    - H-model 68
    - marked 79
    - ray-tracing 74
    - resolvability 73
    - R-модель 68
  - s-vertical 67
  - s-разрез
    - импорт 69
  - T -**
  - target model 67
    - partial 69
  - Task Monitor 16
  - TMR 16
  - trace display 26
  - TX-curve 62
  - V -**
  - velocity
    - color spectrum 49
-

velocity  
  column 6  
  layer 6, 30  
  profile 6, 46  
  profile, editing 54  
  range 49  
  spectrum 41

**- W -**

wave  
  binding 39  
  code 8  
  conversion flags 8  
  converted 8  
  P- 8  
  S- 8  
  TX-curve 62  
  type 7  
wireframe  
  editor 33  
  x-net 29  
work folder 19

**- X -**

x-net 4  
XTomo-LM  
  import of initial model 31  
  joint interpretation 17

**- A -**

активный узел 11

**- Д -**

дерево обработки  
  активный узел 11  
  узел итерации, меню 14

**- И -**

итерация  
  заметки пользователя 14  
  меню 14  
  создание 14