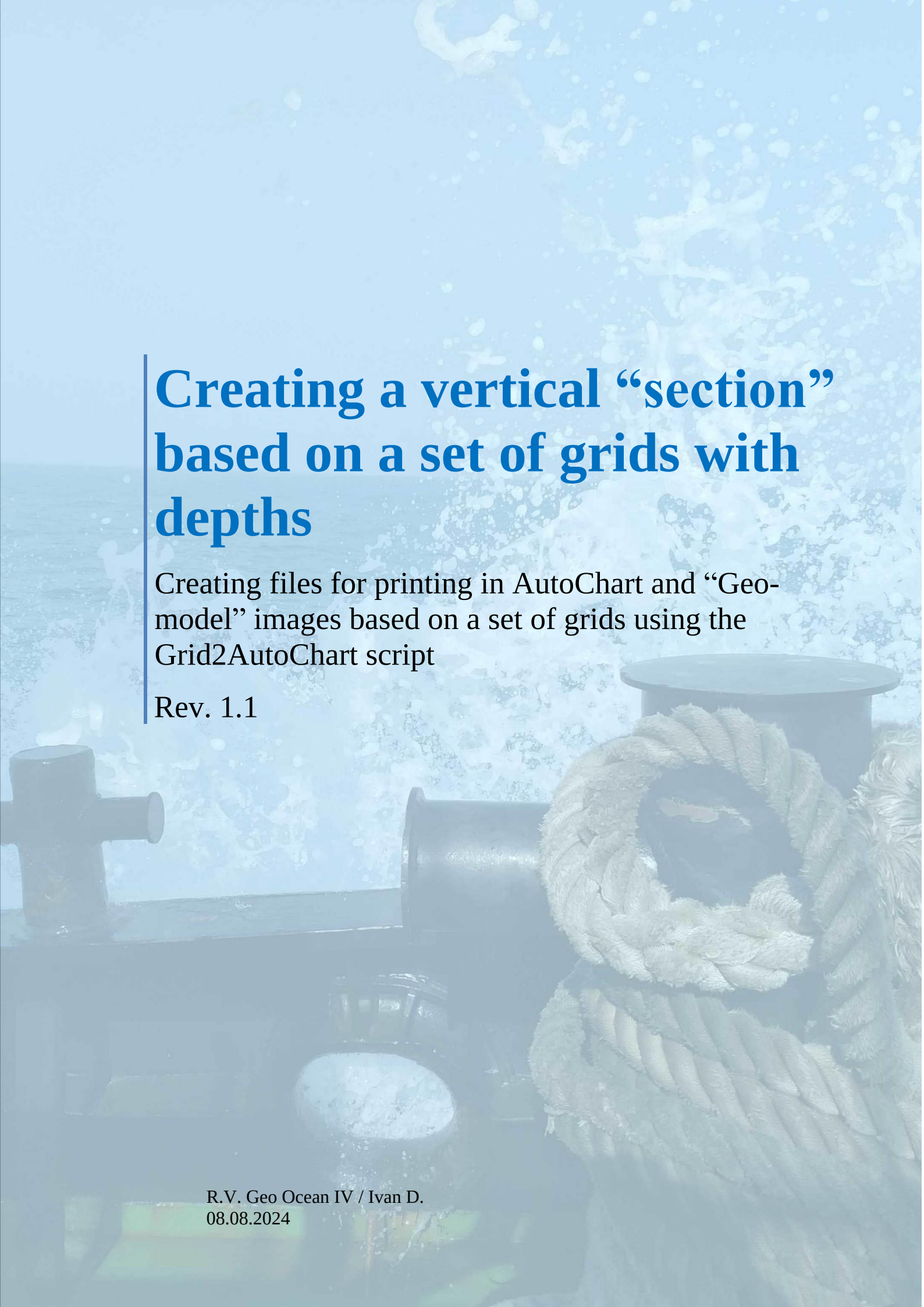


The background of the slide is a photograph of a ship's deck. In the foreground, there is a large, thick coil of light-colored rope. To the left, a dark metal structure, possibly part of a winch or a railing, is visible. The background shows the sea and a bright sky with some white clouds. The entire image is overlaid with a semi-transparent blue filter.

Creating a vertical “section” based on a set of grids with depths

Creating files for printing in AutoChart and “Geo-model” images based on a set of grids using the Grid2AutoChart script

Rev. 1.1



Contents

Introduction 3

1. Source data format.....4

2. Calculation results format.....7

3. Script commands 8

4. Geo-model image editing 14

Appendix 1. Grid2AutoChart script code.....15

Revision History

Version	Date	Description of changes
1.0	2022-11-10	Document created // Ivan D.
1.1	2024-08-08	Document revised // Ivan D.

Figures list

Figure 0.1 AutoChart file fragment example 3

Figure 0.2 Geo-model example 3

Figure 1.1 Contents of the project root folder 4

Figure 1.2 Contents example of the subfolder with grids 6

Figure 1.3 Example of the file with grid (fragment) 6

Figure 3.1 Screenshot of the MatLab window with the “rp” structure variable contents 8

Figure 3.2 Screenshot of the MatLab window with the “Hrz” structure variable contents.....9

Figure 3.3 MatLab command window with information about the loaded grids (printed when executing the “Hrzs_Read” command).....9

Figure 3.4 MatLab command window with information about the drawn grid (printed when executing the “HrzSurface_Draw” command) 9

Figure 3.5 Route trackplot (red line) and surface map (blue rectangle).....10

Figure 3.6 The surface map zoomed-in image (see **Figure 3.5**) 10

Figure 3.7 Geo-model image in a given KP interval using the “Hrzs_Calc&Draw” command 11

Figure 3.8 Geo-model image (result of executing the “Hrzs_Draw” command) 12

Figure 4.1 “Save As” menu 14

Figure 4.2 Vector graphics in the Corel Draw program window 14

Introduction

The Grid2AutoChart script was created to solve the following problem:

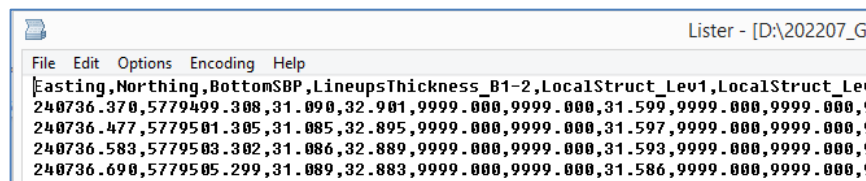
- prepare files with depths to seismic surfaces along a given route in a special format ([Figure 0.1](#)) for AutoChart software; moreover, the X axis in the prepared files should display the KP (length) values along the route.
- prepare an image called “Geo-model” displaying horizons (seismic boundaries) plotted based on seismic surfaces ([Figure 0.2](#)).

The initial data for the Grid2AutoChart script are:

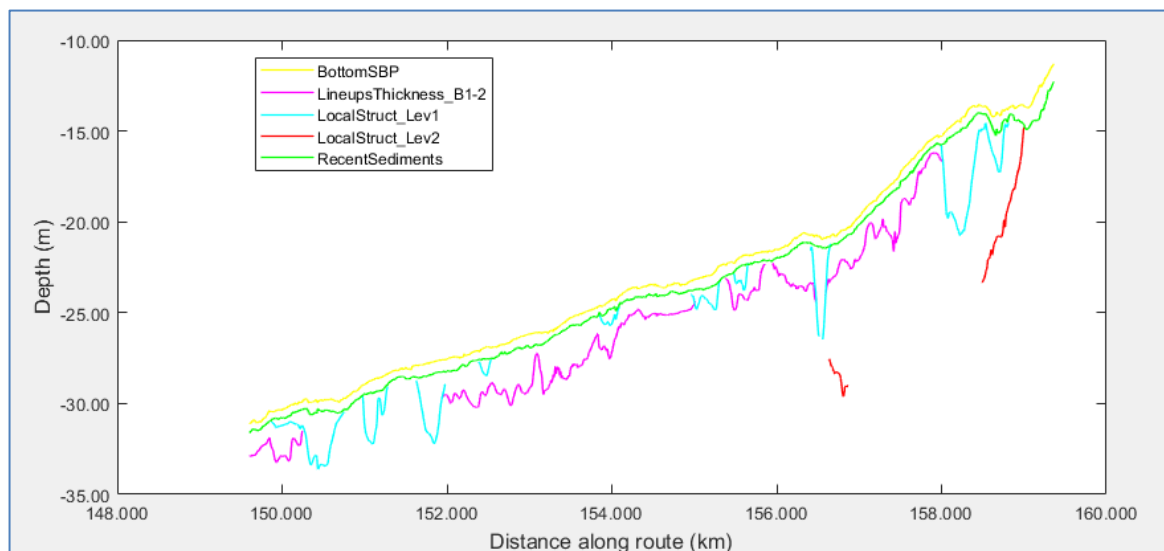
- a set of files containing grid surfaces (used to create each graph/horizon along the route, see [Figure 0.2](#)); grids should contain square cells with axes directed to Easting and Northing,
- a file with a “route” containing KP and coordinates (for example, along a pipeline laying line),
- a file with settings for displaying horizons (names, colors and line thickness).

The script can generate output data in a given KP range. Also, separate grids (indicating the KP range) can be loaded for the different route sections (different Survey Blocks) and then the calculation results are combined into a single file for AutoChart or a single Geo-model.

To run the Grid2AutoChart script, one must install MatLab 2018a (or later) software and the ge0mllib library.



[Figure 0.1](#) AutoChart file fragment example



[Figure 0.2](#) Geo-model example

1. Source data format

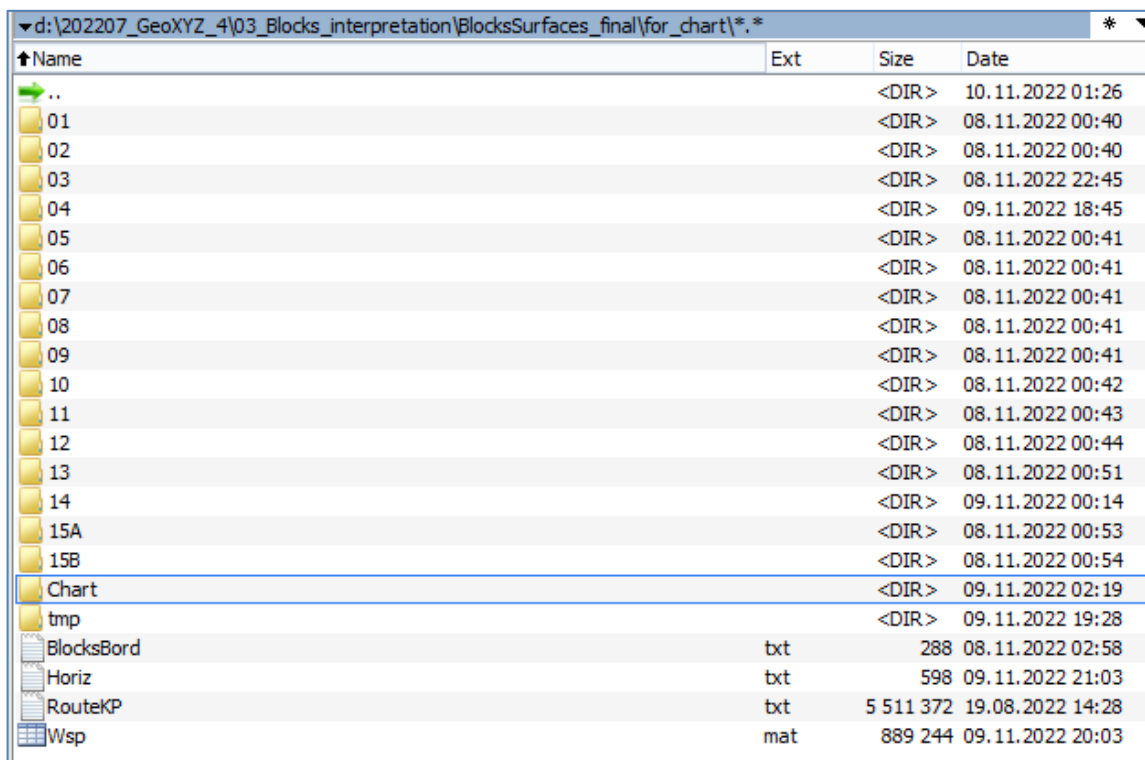
1) Structure of folder with source data

Files with source data are located in the project root folder; it includes:

- a set of subfolders with grid files for each Block;
- a file named “RouteKP.txt” located in the project root folder and containing route KP and coordinates;
- a file named “Horiz.txt” with settings for displaying seismic horizons in Geo-model, located in the project root folder.

The script creates a “tmp” folder for temporary files created when the script runs. If one saves the current processing state, the script creates the “Wsp.mat” file. It is recommended to prepare an additional “BlocksBord.txt” file including the boundaries of each survey block by KP (KP values by blocks will be used in the command parameters for the Grid2AutoChart script).

An example of the root folder contents is shown in [Figure 1.1](#). It contains “block folders” from “01” to “15B”, the “tmp” folder created by the script, “RouteKP.txt”, “Horiz.txt”, “BlocksBord.txt” and “Wsp.mat” created for saving the current processing state.



Name	Ext	Size	Date
..		<DIR>	10.11.2022 01:26
01		<DIR>	08.11.2022 00:40
02		<DIR>	08.11.2022 00:40
03		<DIR>	08.11.2022 22:45
04		<DIR>	09.11.2022 18:45
05		<DIR>	08.11.2022 00:41
06		<DIR>	08.11.2022 00:41
07		<DIR>	08.11.2022 00:41
08		<DIR>	08.11.2022 00:41
09		<DIR>	08.11.2022 00:41
10		<DIR>	08.11.2022 00:42
11		<DIR>	08.11.2022 00:43
12		<DIR>	08.11.2022 00:44
13		<DIR>	08.11.2022 00:51
14		<DIR>	09.11.2022 00:14
15A		<DIR>	08.11.2022 00:53
15B		<DIR>	08.11.2022 00:54
Chart		<DIR>	09.11.2022 02:19
tmp		<DIR>	09.11.2022 19:28
BlocksBord	txt	288	08.11.2022 02:58
Horiz	txt	598	09.11.2022 21:03
RouteKP	txt	5 511 372	19.08.2022 14:28
Wsp	mat	889 244	09.11.2022 20:03

[Figure 1.1](#) Contents of the project root folder

2) Contents of the RouteKP.txt file

The ASCII file includes three columns: KP value in meters, Easting and Northing coordinates. Possible separators: comma, space or tab.

Example of the RouteKP.txt file contents fragment:

```
-465.000,357634.503,5725771.438  
-464.000,357633.562,5725771.099  
-463.000,357632.622,5725770.759  
-462.000,357631.681,5725770.420
```

-461.000,357630.740,5725770.081
-460.000,357629.800,5725769.741

3) Contents of the Horiz.txt file

The ASCII file includes three columns: horizon name, horizon color code, horizon line width. Separator: tab character. Horizons must be written in the same sequence in which they should be displayed on the legend in the Geo-model image.

Example of the Horiz.txt file contents:

```
BottomSBP      'y'      1
LineupsThickness_B1-2  'm'      1
LocalStruct_Lev1    'c'      1
LocalStruct_Lev2    'r'      1
RecentSediments 'g'      1
GravelBottom_B5-6-7-8 'b'      1
GravelTop_B5-6-7-8  'k'      1
RecentSediments_BelowLineup_B5-6-7  [0 255 0]/255      1
CoveredDune_B6-7    'y'      1
Till_Lev0_B8-9-10   'm'      1
Till_Lev1_B11      'c'      1
Till_Lev2_B11-12    'r'      1
Till_Lev3_B12-13    'g'      1
Till_Lev4_B11-12-13-14 'b'      1
AcousticQuietRock_B13-14-15A 'k'      1
AcousticQuietSand_B15A      [100 100 100]/255      1
InternalLayer_RecenSediments_B15A      [50 50 50]/255      1
RecentSediments_Lev1_B15B      [150 150 150]/255      1
RecentSediments_Lev2_B15B      [200 200 200]/255      1
```

The horizon color code can be specified in three ways:

-- As a symbol. For example: “r” (see color codes in the table below).

y	yellow
m	magenta
c	cyan
r	red
g	green
b	blue
w	white
k	black

-- As an RGB code with a value from 0 to 1. Example: [0.235 0.337 0.555].

-- As an expression with RGB code calculation, for example [150 150 150]/255 (each value in brackets will be divided by 255, thereby obtaining an RGB code with a value from 0 to 1).

4) Contents of the BlocksBord.txt file

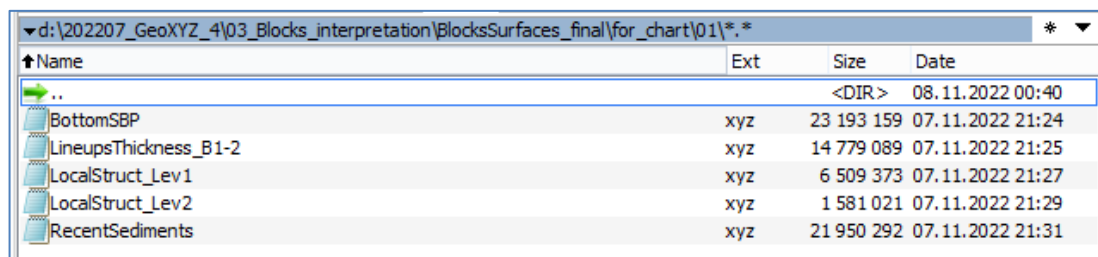
The ASCII file is used as a “memo” when setting parameters in commands to the Grid2AutoChart script. The file includes the following data: KP starting point in meters for the Block, KP ending point in meters for the Block, name of the Block.

Example of the BlocksBord.txt file contents:

```
[000750 004766]    15B
[004767 013075]    15A
[013076 019302]    14
[019303 030440]    13
[030441 040929]    12
[040930 050842]    11
[050843 060076]    10
[060077 071825]    09
[071826 083600]    08
[083601 092726]    07
[092727 104288]    06
[104289 115459]    05
[115460 123159]    04
[123160 139232]    03
[139233 149601]    02
[149602 159400]    01
```

5) Subfolders with grid files

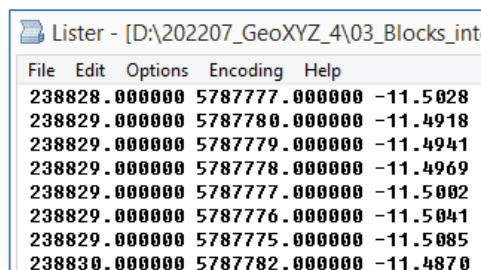
Each subfolder contains grids for one Block (route section); the KP starting and ending points of the block are defined in the “BlocksBord.txt” file. The name of the file with the grid must match the horizon name; horizon names are defined in the “Horiz.txt” file. The contents example of the subfolder with grids is shown in *Figure 1.2*.



Name	Ext	Size	Date
..		<DIR>	08.11.2022 00:40
BottomSBP	xyz	23 193 159	07.11.2022 21:24
LineupsThickness_B1-2	xyz	14 779 089	07.11.2022 21:25
LocalStruct_Lev1	xyz	6 509 373	07.11.2022 21:27
LocalStruct_Lev2	xyz	1 581 021	07.11.2022 21:29
RecentSediments	xyz	21 950 292	07.11.2022 21:31

Figure 1.2 Contents example of the subfolder with grids

Each grid includes the following columns: Easting coordinate, Northing coordinate and depth (*Figure 1.3*). The grid should be represented by square cells with the axes directed to Easting and Northing.



File	Edit	Options	Encoding	Help
238828.000000	5787777.000000	-11.5028		
238829.000000	5787780.000000	-11.4918		
238829.000000	5787779.000000	-11.4941		
238829.000000	5787778.000000	-11.4969		
238829.000000	5787777.000000	-11.5002		
238829.000000	5787776.000000	-11.5041		
238829.000000	5787775.000000	-11.5085		
238830.000000	5787782.000000	-11.4870		

Figure 1.3 Example of the file with grid (fragment)

2. Calculation results format

1) File for AutoChart program

The ASCII file includes the following columns: Easting, Northing, horizon 1, ..., horizon N, KP. Separator: comma; missing values are recorded as “9999.000”.

The first line contains a header including the names of the horizons. The names of the horizons and their sequence in the header are formed in accordance with the names and sequence in the “Horiz.txt” file.

An example with multiple lines from the output file:

```
Easting,Northing,BottomSBP,LineupsThickness_B1-  
2,LocalStruct_Lev1,LocalStruct_Lev2,RecentSediments,GravelBottom_B5-6-7-8,GravelTop_B5-6-7-  
8,RecentSediments_BelowLineup_B5-6-7,CoveredDune_B6-7,Till_Lev0_B8-9 -10,Till_Lev1_B11,  
Till_Lev2_B11-12,Till_Lev3_B12-13,Till_Lev4_B11-12-13-14,AcousticQuietRock_B13-14-  
15A,AcousticQuietSand_B15A,InternalLayer_RecenSediments_B15A,RecentSediments_Lev1_B15B,RecentSedim  
ents_Lev2_B15B,KP  
240736.370,5779499.308,31.090,32.901,9999.000,9999.000,31.599,9999.000,9999.000,9999.000,9999.000,9999.00  
0,9999.000,9999.000,9999.000,9999.000,9999.000,9999.000,9999.000,9999.000,149.601  
240736.477,5779501.305,31.085,32.895,9999.000,9999.000,31.597,9999.000,9999.000,9999.000,9999.000,9999.00  
0,9999.000,9999.000,9999.000,9999.000,9999.000,9999.000,9999.000,9999.000,149.603  
240736.583,5779503.302,31.086,32.889,9999.000,9999.000,31.593,9999.000,9999.000,9999.000,9999.000,9999.00  
0,9999.000,9999.000,9999.000,9999.000,9999.000,9999.000,9999.000,9999.000,149.605
```

2) Geo-model image

The image can be obtained as a screenshot from the MatLab graphics window, which is drawn during the Grid2AutoChart script execution. Also, the graphics window contents can be saved in *.eps format and further edited in a graphic editor for vector images.

3. Script commands

The Grid2AutoChart script command format is as follows:

```
{'Command name', Parameter1,..., ParameterN}; Script name;
```

The script name is Grid2AutoChart.

1) Defining basic project settings

```
{'SU_Set','d:\202207_GeoXYZ_4\for_chart',2};Grid2AutoChart;
```

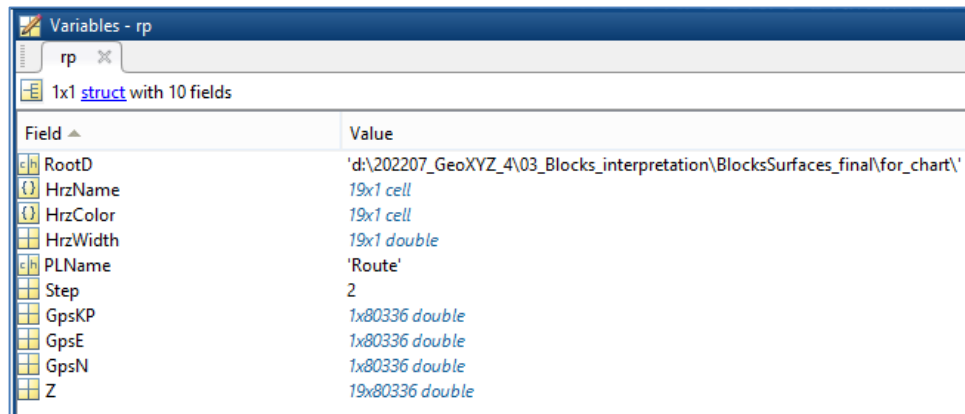
where

SU_Set – script command name;

d:\202207_GeoXYZ_4\for_chart\ – path to the project root folder (*Figure 1.1*);

2 – step for KP values in the output file for AutoChart.

The command reads the “RouteKP.txt” RPL file and the “Horiz.txt” horizons list, creates the “tmp” folder. A structure variable containing empty values for the horizons is created in the computer's RAM memory; The KP range is taken from the “RouteKP.txt” file. Further, empty horizon values will be filled with values obtained from grid files using script commands. Information about Geo-model horizons is written to the “rp” structure variable (*Figure 3.1*).



Field	Value
RootD	'd:\202207_GeoXYZ_4\03_Blocks_interpretation\BlocksSurfaces_final\for_chart'
HrзName	19x1 cell
HrзColor	19x1 cell
HrзWidth	19x1 double
PLName	'Route'
Step	2
GpsKP	1x80336 double
GpsE	1x80336 double
GpsN	1x80336 double
Z	19x80336 double

Figure 3.1 Screenshot of the MatLab window with the “rp” structure variable contents

2) Reading files with grids into the Hrз structure variable

```
{'Hrzs_Read','01\'};Grid2AutoChart;
```

where

Hrzs_Read – script command name;

01\ – the name of the “01” subfolder in the project root folder.

The command reads all grids from a subfolder (data for one Block) and saves them in the “tmp” folder in MatLab format to optimize work with the computer’s RAM. Grid headers are stored in RAM in the “Hrз” structure variable (*Figure 3.2*).

Fields	Name	Head
1	'BottomSBP'	1x1 struct
2	'LineupsThickness_B1-2'	1x1 struct
3	'LocalStruct_Lev1'	1x1 struct
4	'LocalStruct_Lev2'	1x1 struct
5	'RecentSediments'	1x1 struct

Figure 3.2 Screenshot of the MatLab window with the “Hrz” structure variable contents

The serial number, names and size of loaded grid cells are printed in the command window (**Figure 3.3**).

```
01 >> BottomSBP; step=1
02 >> LineupsThickness_B1-2; step=1
03 >> LocalStruct_Lev1; step=1
04 >> LocalStruct_Lev2; step=1
05 >> RecentSediments; step=1
>>
```

Figure 3.3 MatLab command window with information about the loaded grids (printed when executing the “Hrzs_Read” command)

The **HrzSurface_Draw**, **Hrzs_Calc&Draw** script commands described below will work with the grids loaded by the last **Hrzs_Read** command (i.e., with the grids for one survey block - from one subfolder in the root folder). The **Hrzs_Calc&Draw** command performs calculations on the “Hrz” structure variable data and places the calculation results to the “rp” structure variable. Thus, when plotting a Geo-model, it is necessary to sequentially load the grids of each block using the **Hrzs_Read** command and place the calculation results to the “rp” structure variable (**Hrzs_Calc&Draw** command). Based on the information in the “rp” structure variable, one can create an AutoChart file and a “Geo-model” image for the entire route using script commands.

3) Surface/grid drawing

`{'HrzSurface_Draw',1};Grid2AutoChart;`

where

HrzSurface_Draw – script command name;

1 – number of the loaded surface/grid (see **Figure 3.3**).

The name of the drawn surface/grid is printed in the MatLab command window (**Figure 3.4**).

```
>> {'HrzSurface_Draw',1};Grid2AutoChart;
Name of horizon: BottomSBP
fx >>
```

Figure 3.4 MatLab command window with information about the drawn grid (printed when executing the “HrzSurface_Draw” command)

After executing the command, a new graphics window with the route track and the specified surface image opens (*Figure 3.5*). One can use zoom and move tools in the graphics window; one can take screenshots or save the window contents in *.eps format for further editing (*Figure 3.6*). Such images can be used as a “surface map” together with a Geo-model image.

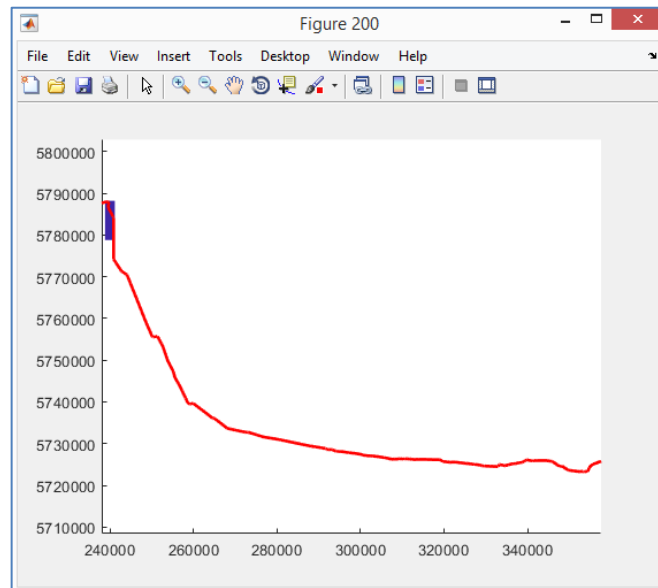


Figure 3.5 Route trackplot (red line) and surface map (blue rectangle)

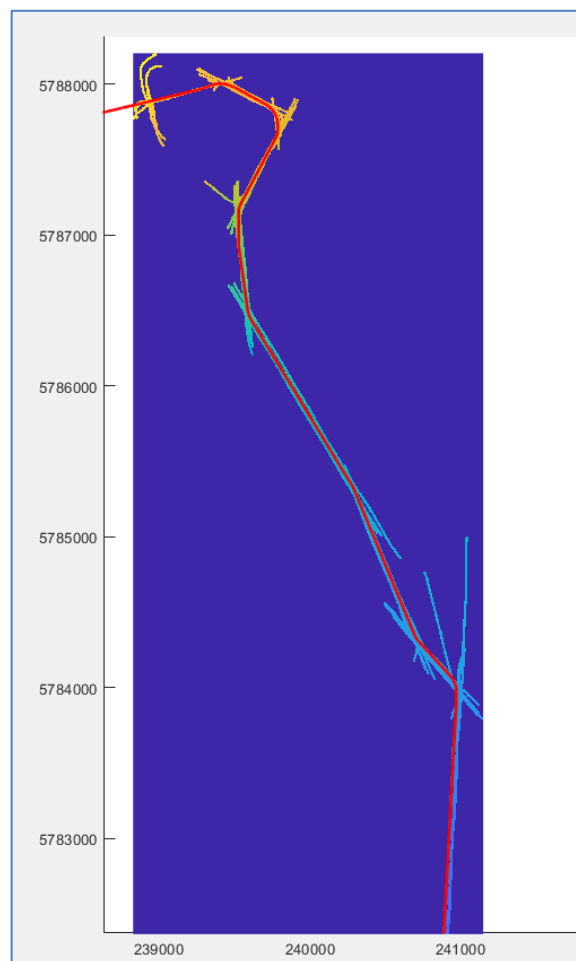


Figure 3.6 The surface map zoomed-in image (see *Figure 3.5*)

4) Seismic horizons calculation for a given KP interval

```
{'Hrzs_Calc&Draw',[149602 159400],[15 15]};Grid2AutoChart;
```

where

Hrzs_Calc&Draw – script command name;

[149602 159400] – KP interval in meters used to calculate horizons based on grids and place the calculation results in the “rp” structure variable;

[15 15] – maximum length in points for “missing values” and “sections”; Based on these values, a “horizon optimization” is performed in two steps. The first step is that any “missing” horizon values less than 15 points long will be filled in using linear interpolation; the second step is that any horizon “sections” less than 15 points long will be deleted.

When executing “Hrzs_Calc&Draw” command, the script performs the following actions:

- depth/time values for horizons are calculated based on grids (in a given KP interval), the calculation results are placed in the “rp” structure variable;
- the length of “missing values” and “sections” is analyzed; based on these values, “horizon optimization” is performed in two steps (see the description above);
- the Geo-model is drawn in the specified KP interval (*Figure 3.7*).

When using the command, the following features occur:

- zoom and move tools can be used to create the most convenient view for screenshots;
- the legend can be moved with the mouse to any part of the picture;
- the legend includes only drawn horizons (not all horizons from “Horiz.txt” file);
- the color and width values for horizon graphs are read from “Horiz.txt” file using the “SU_Set” command.

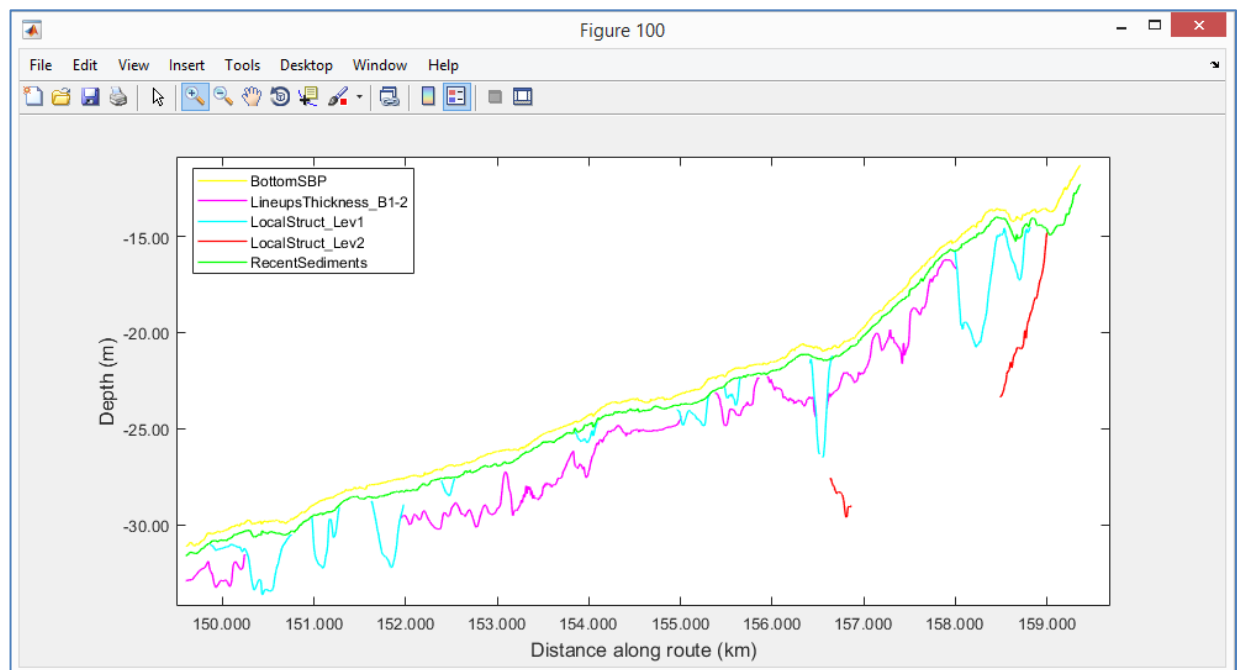


Figure 3.7 Geo-model image in a given KP interval using the “Hrzs_Calc&Draw” command

5) Drawing horizons in a given KP interval (Geo-model drawing)

```
{'Hrzs_Draw',[149602 159400]};Grid2AutoChart;
```

where

Hrzs_Draw – script command name;

[149602 159400] – KP interval in meters in which horizons need to be drawn (it can be the entire route).

The legend displayed by the command in the graphics window includes all horizons specified in the “Horiz.txt” file (**Figure 3.8**).

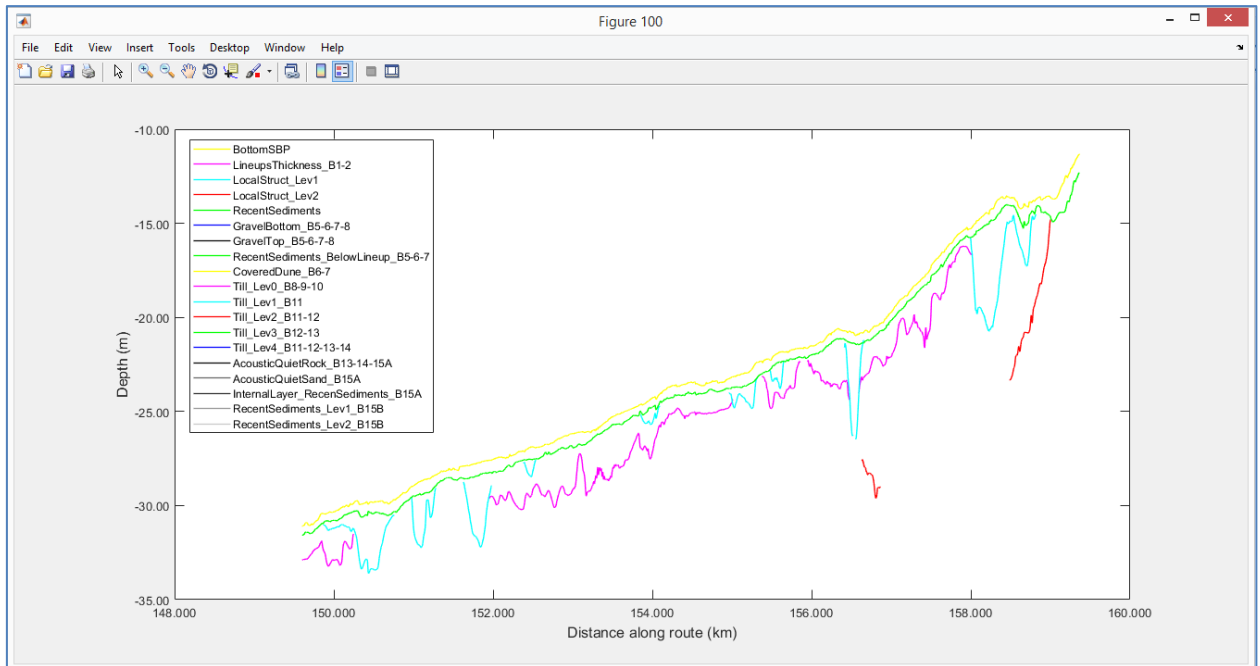


Figure 3.8 Geo-model image (result of executing the “Hrzs_Draw” command)

6) Saving horizons to an AutoChart file

```
{'ChartFile_Write','Chart\B01.txt',[149602 159400]};Grid2AutoChart;
```

```
{'ChartFile_Write','Chart\Route.txt',[000750 159400]};Grid2AutoChart;
```

where

ChartFile_Write – script command name;

Chart\B01.txt – AutoChart file name; the file will be recorded to the “Chart” subfolder located in the project root folder;

[149602 159400] – KP interval in meters used to record data to the AutoChart file.

In the example given, the 'Chart\B01.txt' file with the KP interval [149602 159400] will contain data only for 01 block. The 'Chart\Route.txt' file with the KP interval [000750 159400] will contain data for the entire route (for all blocks' KP specified in “BlocksBord.txt”).

7) Saving the MatLab workspace with structure variables

```
{'WspSave'};Grid2AutoChart;
```

where

WspSave – script command name.

The command saves the current state of the “rp” and “Hrz” structure variables in the file “Wsp.mat” in the project root folder. To restore/load the MatLab state (actually, the “data processing” state), one needs to use the command

```
load('d:\202207_GeoXYZ_4\for_chart\Wsp.mat');
```

d:\202207_GeoXYZ_4\for_chart\Wsp.mat – path to the file with the “rp” and “Hrz” variables.

4. Geo-model image editing

When preparing graphics, the Geo-model image detailed editing may be required. The most effective method seems to be the following one:

- go to the “Save As” menu in the drawing window ([Figure 4.1](#));
- save the graphics window in a file with the extension *.eps (Enchanted Post Script);
- open the saved file in a program working with vector graphics (for example, Corel Draw; [Figure 4.2](#)) and perform the Geo-model necessary editing.

Using this method, one can edit all Geo-model elements as vector graphics. At the same time, at the first stage it is recommended to remove several invisible rectangles that form the frame for the Geo-model image.

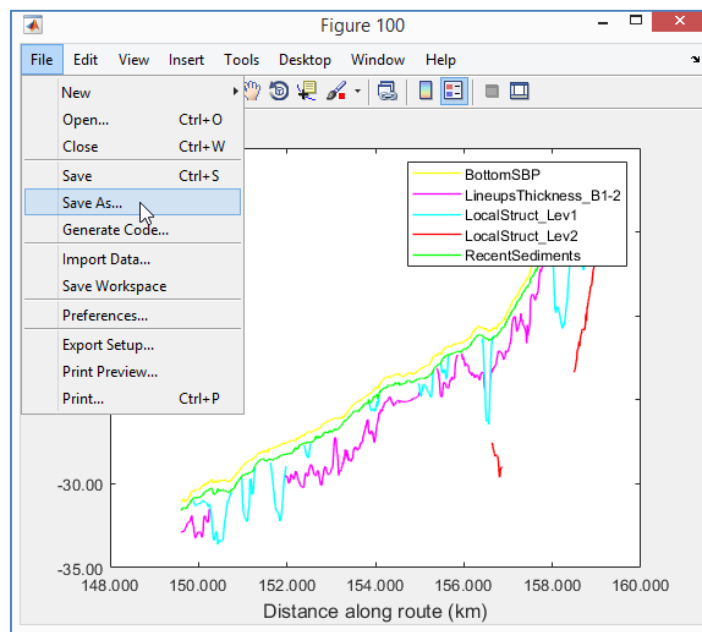


Figure 4.1 “Save As” menu

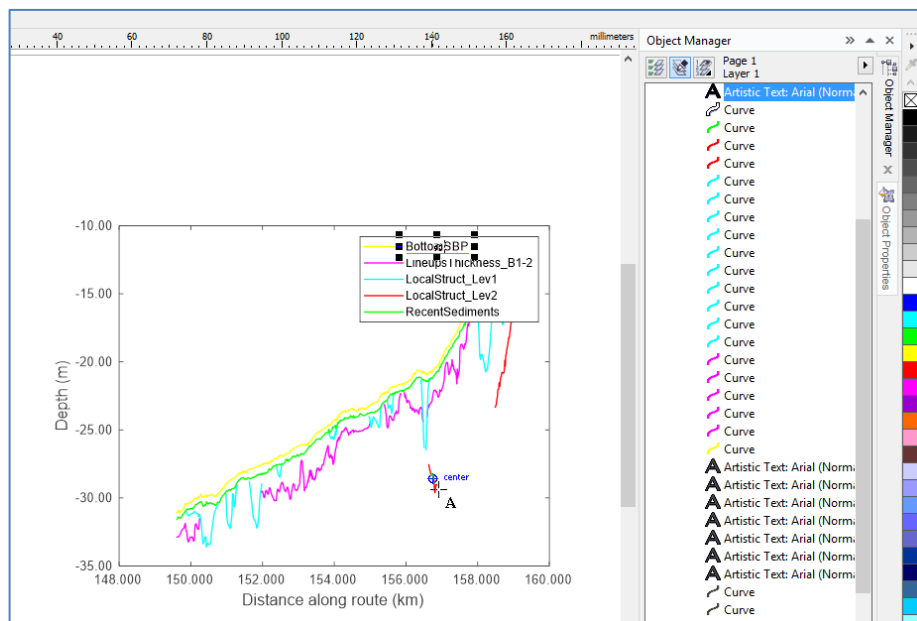


Figure 4.2 Vector graphics in the Corel Draw program window

Appendix 1. Grid2AutoChart script code

```
%script Grid2AutoChart;
%The script is solving follow task: prepare files for AutoChart (depth along route using KP-axis) in special format and prepare picture named “Geo-model”.
%==== There are follow input data:
%-a number of files contained grids (used to create each plot/horizon along route); grids must include square cells with axis direction to Easting and Northing,
%-file with route KP and coordinates,
%-file with horizons settings (there are horizons names, colors and lines width).
%==== There are follow script's commands:
%SU_Set- read RPL and horizons list;
%Hrzs_Read- read surfaces to structure from sub-folder;
%HrzSurface_Draw- draw XYZ surface and Route-line;
%Hrzs_Calc&Draw- calculate and draw horizons for KP-interval;
%Hrzs_Draw- draw horizons for KP-interval;
%ChartFile_Write- write horizons to file;
%WspSave- save workspace // for loading >> load('...Wsp.mat');
%See formats description in 10-03_Grid2AutoChart.pdf

gKey=ans;
ifstrcmp(gKey{1},'SU_Set')%Read RPL and horizons list
%{'SU_Set',d:\202207_GeoXYZ_4\03_Blocks_interpretation\BlocksSurfaces_final\for_chart\,2};Grid2AutoChart;
tryRootD=gKey{2};catch,RootD=input('Root folder name=');end;trystp=gKey{3};catch,stp=input('Step for RPL points=');end;
fid=fopen([RootD'\Horiz.txt'],'r');Hrz=textscan(fid,'%s %s %f','Delimiter','\t','MultipleDelimsAsOne',0,'EndOfLine','\r\n');fclose(fid);
A=dlmread([RootD'\RouteKP.txt']);A=A(1:stp:end,:);
rp=struct('RootD',RootD,'HrzName',Hrz(1),'HrzColor',Hrz(2),'HrzWidth',Hrz(3),'PLName','Route','Step',stp,'GpsKP',A(:,1),'GpsE',A(:,2),'GpsN',A(:,3),'Z',nan(numel(Hrz{1})),numel(A(:,3))));
for n=1:numel(Hrz{1}),rp.HrzColor{n}=eval(Hrz{2}{n});end;
dos(['mkdir',RootD'\tmp']);
clearvarsRootD stp fid Hrz A L1 L2 n;
%output >> rp
end;
ifstrcmp(gKey{1},'Hrzs_Read')%Read surfaces to structure from sub-folder
%{'Hrzs_Read','01'};Grid2AutoChart;
tryfName=gKey{2};catch,fName=input('Horizons (XYZ) files sub-folder name=');end;
dz=dir([rp.RootD'\fName]);dz([dz.isdir])=[];fName=[repmat(fName,length(dz),1) char(dz.name)];
Hrz=struct('Name',repmat('',size(fName,1),1),'Head',[]);
for nn=1:size(fName,1),
fNameN=deblank(fName(nn,:));L1=find(fNameN=='\');L2=find(fNameN=='.');tmp=zeros(numel(rp.HrzName));
for nnn=1:numel(rp.HrzName),ifstrcmp(rp.HrzName{nnn},fNameN(L1(end)+1:L2(end)-1)),tmp(nnn)=1;end;end;
ifsum(tmp)==0,error(['Name not found: 'fNameN(L1(end)+1:L2(end)-1)]);end;ifsum(tmp)<1,error(['Name duplicated: 'fNameN(L1(end)+1:L2(end)-1)]);end;
Hrz(nn).Name=fNameN(L1(end)+1:L2(end)-1);
bo=dlmread([rp.RootD'\fNameN']);
s1=abs(bo(1,2)-bo(1,1));ifs1==0,s1=abs(bo(2,2)-bo(2,1));end;disp([num2str(nn,'%02d')' >> 'Hrz(nn).Name'; step='num2str(s1)']);
[Hrz(nn).Head,Data]=gWfrXyz2Mat(bo,[0 0 s1 0],[0 0 0 s1]);
save([rp.RootD'\tmp'Hrz(nn).Name'.mat'],'Data','-v7.3');Data=[];
```

```

end;
clearvars fName dz nn fNameN L1 L2 nnn tmp bo s1;
%output >> HRz
end;
if strcmp(gKey{1}, 'HrzSurface_Draw') %Draw XYZ surface and Route-line
    %{'HrzSurface_Draw',1}; Grid2AutoChart;
    try nn=gKey{2}; catch, nn=input('Horizon number='); end
    disp(['Name of horizon: 'Hrz(nn).Name]); load([rp.RootD\tmp\Hrz(nn).Name'.mat']);
    x=Hrz(nn).Head.Wf(5)+(0:size(Data,2)-1).*Hrz(nn).Head.Wf(1); y=Hrz(nn).Head.Wf(6)+(0:size(Data,1)-1).*Hrz(nn).Head.Wf(4); [X,Y]=meshgrid(x,y);
    figure(200); hold on; axis equal; imagesc(x,y,Data); plot(rp.GpsE,rp.GpsN,'-r','LineWidth',2); hold off; aaa=gca; aaa.YDir='normal';
    gMapTickLabel(200,'%0f',10);
    clearvars nn Data x y X Y aaa
end;
if strcmp(gKey{1}, 'Hrzs_Calc&Draw') %Calculate and draw horizons for KP-interval
    %{'Hrzs_Calc&Draw',[149602 159400],[15 15]}; Grid2AutoChart;
    try LLL=gKey{2}; catch, LLL=input('KP interval='); end; try ddd=gKey{3}; catch, ddd=input('Holes and parts='); end;
    L1=find(rp.GpsKP<=LLL(1)); L2=find(rp.GpsKP>=LLL(2)); LL=L1(end):L2(1);
    figure(100);
    for nn=1:size(Hrz,2),
        nnn=1; while ~strcmp(rp.HrzName{nnn},Hrz(nn).Name), nnn=nnn+1; end;
        load([rp.RootD\tmp\Hrz(nn).Name'.mat']);
        x=Hrz(nn).Head.Wf(5)+(0:size(Data,2)-1).*Hrz(nn).Head.Wf(1); y=Hrz(nn).Head.Wf(6)+(0:size(Data,1)-1).*Hrz(nn).Head.Wf(4); [X,Y]=meshgrid(x,y); %mesh(X,Y,Hrz(nn).Data);
        rp.Z(nnn,LL)=interp2(X,Y,Data,rp.GpsE(LL),rp.GpsN(LL),'linear',NaN);
        z=rp.Z(nnn,LL); zL=find(~isnan(z)); zzL=find((diff(zL)>1)&(diff(zL)<ddd(1)));
        for n=1:numel(zzL); z(zL(zzL(n)):zL(zzL(n)+1))=interp1([zL(zzL(n)) zL(zzL(n)+1)], [z(zL(zzL(n))) z(zL(zzL(n)+1))], zL(zzL(n)):zL(zzL(n)+1), 'linear'); end;
        rp.Z(nnn,LL)=z;
        z=rp.Z(nnn,LL); zL=find(isnan(z)); zzL=find((diff(zL)>1)&(diff(zL)<ddd(2)));
        for n=1:numel(zzL); z(zL(zzL(n)):zL(zzL(n)+1))=nan; end;
        rp.Z(nnn,LL)=z;
        if (nn>1), hold on; end; plot(rp.GpsKP(LL)./1000,rp.Z(nnn,LL),'-', 'DisplayName',Hrz(nn).Name, 'LineWidth',rp.HrzWidth(nnn), 'Color',rp.HrzColor{nnn}); Data=[];
    end;
    hold off; gMapTickLabel(100, {'%.2f', '%.3f'}, 10); lgd=legend; lgd.Interpreter='none';
    ylabel('Depth(m)', 'FontSize', 12, 'Interpreter', 'none'); xlabel('Distance along route (km)', 'FontSize', 12, 'Interpreter', 'none');
    clearvars LLL ddd L1 L2 nn nnn Data x y X Y LL z zL zzL n lgd;
end;
if strcmp(gKey{1}, 'Hrzs_Draw') %Draw horizons for KP-interval
    %{'Hrzs_Draw',[149602 159400]}; Grid2AutoChart;
    try LLL=gKey{2}; catch, LLL=input('KP interval='); end;
    L1=find(rp.GpsKP<=LLL(1)); L2=find(rp.GpsKP>=LLL(2)); LL=L1(end):L2(1);
    figure(100);
    for nn=1:numel(rp.HrzName), if (nn>1), hold on; end; plot(rp.GpsKP(LL)./1000,rp.Z(nn,LL),'-', 'DisplayName',rp.HrzName{nn}, 'LineWidth',rp.HrzWidth(nn), 'Color',rp.HrzColor{nn}); end;
    hold off; gMapTickLabel(100, {'%.2f', '%.3f'}, 10); lgd=legend; lgd.Interpreter='none';
    ylabel('Depth(m)', 'FontSize', 12, 'Interpreter', 'none'); xlabel('Distance along route (km)', 'FontSize', 12, 'Interpreter', 'none');
    clearvars LLL L1 L2 LL nn lgd;
end;

```

```

end;
ifstrcmp(gKey{1},'ChartFile_Write')%Write horizons to file
%{'ChartFile_Write','B01z.txt',[149602 159400]};Grid2AutoChart;
tryfName=gKey{2};catch,fName=input('Horizons file name=');end;tryLLL=gKey{3};catch,LLL=input('KP interval=');end;
L1=find(rp.GpsKP<=LLL(1));L2=find(rp.GpsKP>=LLL(2));LL=L1(end):L2(1);
tmp=-rp.Z(:,LL);tmp(isnan(tmp))=9999;tmp=[rp.GpsE(LL);rp.GpsN(LL);tmp;rp.GpsKP(LL)/1000] ;
Title='Easting,Northing,';for nn=1:numel(rp.HrzName),Title=[Title rp.HrzName{nn}',''];end;Title=[Title'KP'];
gDataTxtWrite({Title},tmp',[rp.RootD'\fName'],'%1.3f',',','\r\n');
clearvars fName LLL L1 L2 LL tmp Title nn;
end;
ifstrcmp(gKey{1},'WspSave')%Save workspace // load('...\Wsp.mat'); <<< for loading
%{'WspSave'};Grid2AutoChart;
save([rp.RootD'\Wsp.mat'],'rp','Hrz','-v7.3');
end;

```

```

% mail@ge0mlib.com 11/10/2022

```