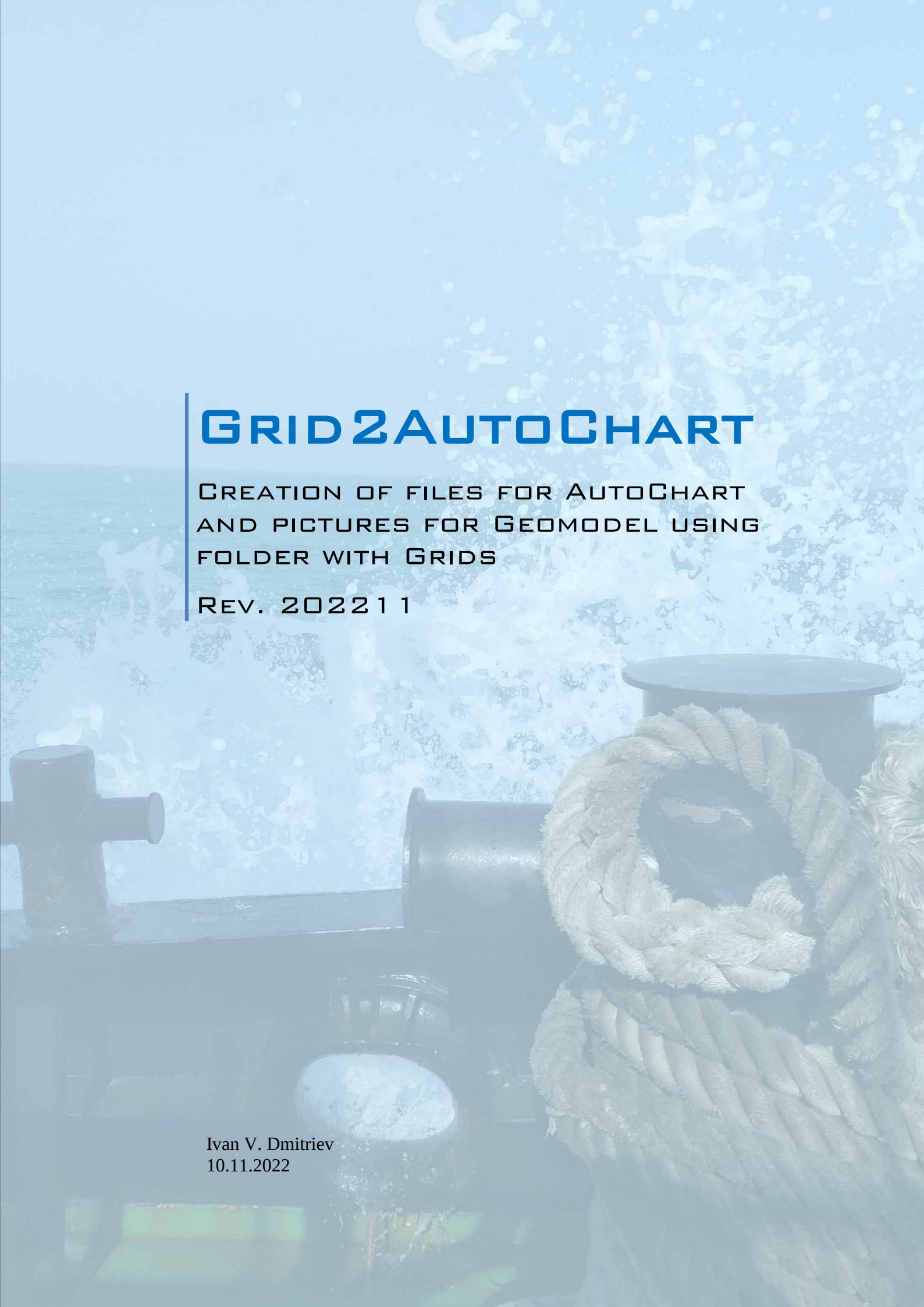


Grid2AutoChart

Creation of files for AutoChart and pictures for Geomodel using folder with Grids

Rev. 202211

Ivan V. Dmitriev
10.11.2022

Contents

Figures list

1. Overview

The script solves the following task: prepares files for AutoChart (the depth along route using KP-axis) in special format (***Figure 1.1***) and prepares picture named "Geomodel" (***Figure 1.2***). The input files are:

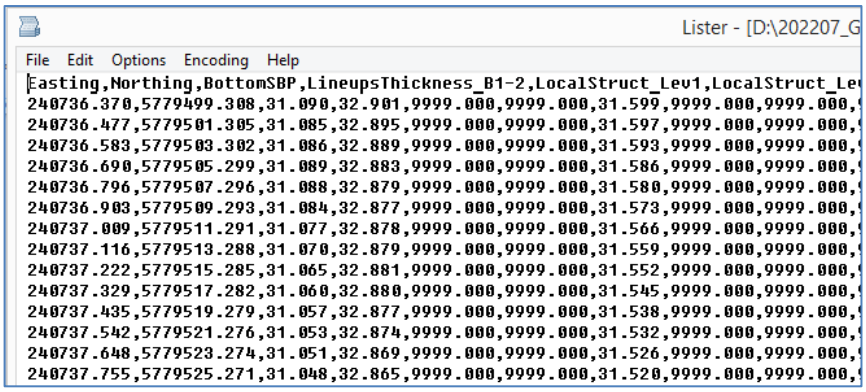
-- a number of files contained grids (used to create each plot/horizon along route, see ***Figure 1.2***); grids must include square cells with axis direction to Easting and Northing,

-- a file with pipeline route KP and coordinates,

-- a file with settings of horizons (there are horizons names, colors and lines width).

The script work with KP-ranges; the grids can be loaded for different parts of route (surveyed Blocks) separately (using KP-ranges) and combined into a single AutoChart-file or Geomodel.

To run script you need to install MatLab 2018a software (or later version) and ge0mlib library.


Lister - [D:\202207_G

File Edit Options Encoding Help

```
Easting,Northing,BottomSBP,LineupsThickness_B1-2,LocalStruct_Lev1,LocalStruct_Le
240736.370,5779499.308,31.090,32.901,9999.000,9999.000,31.599,9999.000,9999.000,
240736.477,5779501.305,31.085,32.895,9999.000,9999.000,31.597,9999.000,9999.000,
240736.583,5779503.302,31.086,32.889,9999.000,9999.000,31.593,9999.000,9999.000,
240736.690,5779505.299,31.089,32.883,9999.000,9999.000,31.586,9999.000,9999.000,
240736.796,5779507.296,31.088,32.879,9999.000,9999.000,31.580,9999.000,9999.000,
240736.903,5779509.293,31.084,32.877,9999.000,9999.000,31.573,9999.000,9999.000,
240737.009,5779511.291,31.077,32.878,9999.000,9999.000,31.566,9999.000,9999.000,
240737.116,5779513.288,31.070,32.879,9999.000,9999.000,31.559,9999.000,9999.000,
240737.222,5779515.285,31.065,32.881,9999.000,9999.000,31.552,9999.000,9999.000,
240737.329,5779517.282,31.060,32.880,9999.000,9999.000,31.545,9999.000,9999.000,
240737.435,5779519.279,31.057,32.877,9999.000,9999.000,31.538,9999.000,9999.000,
240737.542,5779521.276,31.053,32.874,9999.000,9999.000,31.532,9999.000,9999.000,
240737.648,5779523.274,31.051,32.869,9999.000,9999.000,31.526,9999.000,9999.000,
240737.755,5779525.271,31.048,32.865,9999.000,9999.000,31.520,9999.000,9999.000,
```

Figure 1.1 AutoChart file fragment (example)

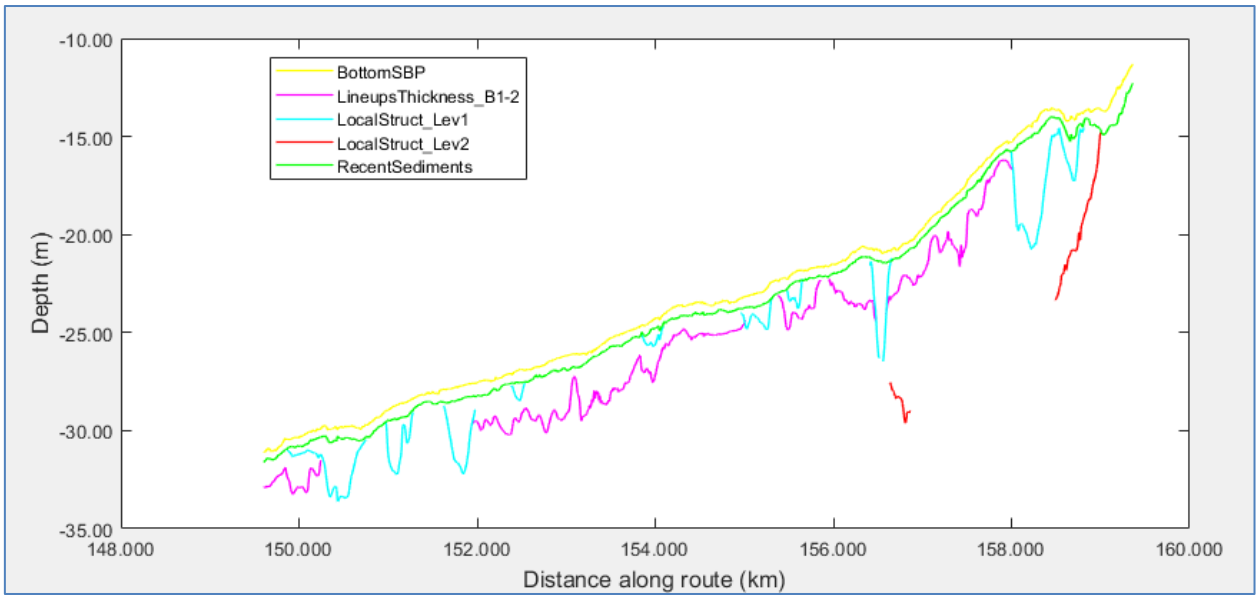


Figure 1.2 An example of "Geomodel"

2. Input and output data

1) Input data folder structure

The input files are located in the Root project folder. There are:

-- a number of subfolders with grid-files for each block;

-- a file with route coordinates in Root named “RouteKP.txt”;

-- a file with settings of horizons in Root, named “Horiz.txt”.

The script creates a folder “tmp” for the own temporary files. If you save the current processing state, then the script will create a file “Wsp.mat”. It is recommended to prepare an additional file “BlocksBord.txt” which includes each surveyed Block borders in KP (the KP values will be used to write commands for script).

The example of Root folder is shown in ***Figure 2.1***. There are “Blocks folders” from “01” to “15B”, “tmp” folder was created by script, “RouteKP.txt”, “Horiz.txt”, “BlocksBord.txt” and “Wsp.mat” were created to save the current processing state.

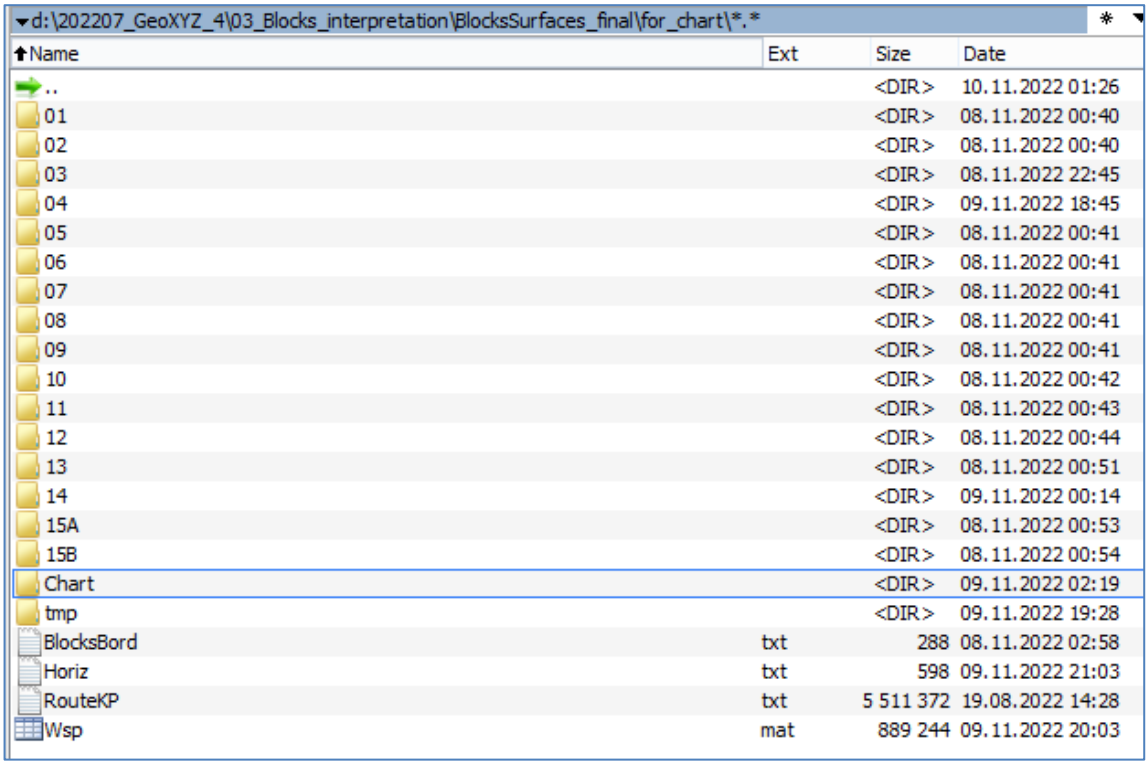


Figure 2.1 Root project folder example

2) RouteKP.txt

The ASCII-file includes the follow columns: Meter-point-number, Easting, and Northing. Possible delimiters: “,” or Space or Tab-symbol. A file part example:

-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) Horiz.txt

The ASCII-file includes the following columns: Horizon Name, Horizon Color, Horizon Line Width. Delimiter: Tab-symbol. It is better to define Horizons line by line, as you want to see it in Legend for picture. A file example:

```
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 can be defined using 3 ways:

1) Characters. 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

2) RGB code with value from 0 to 1. Example: [0.235 0.337 0.555].

3) Equation with RGB code calculation. Example: [150 150 150]/255 – each value will be divided in 255 and we will have RGB code with value from 0 to 1.

4) BlocksBord.txt

The ASCII-file includes the following columns: Start KP in meters, End KP in meters, Horizon Name. File example:

```
[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 Grids

Each subfolder includes Grids for one Block (part of the route); the start-KP and end-KP of Block are defined in "BlocksBord.txt". The name of Grid-file is equal to Horizon name. The names of Horizons are defined in "Horiz.txt". The example of sub-folders is shown in ***Figure 2.2***.

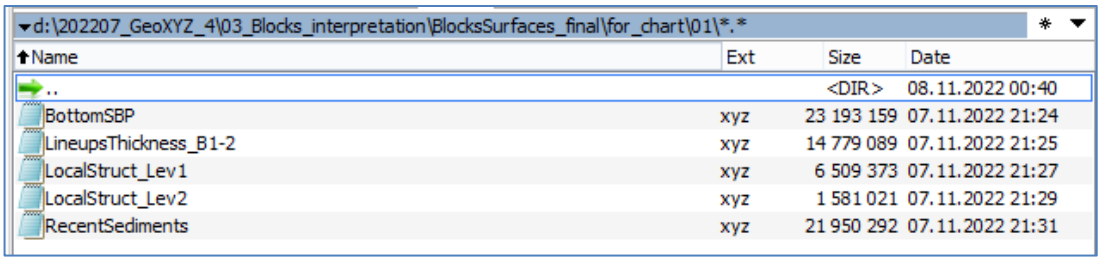


Figure 2.2 An example of sub-folder with Grids

Each grid includes the following columns: Easting, Northing and Depth (***Figure 2.3***).

Grid must include square cells with axis direction to Easting and Northing.

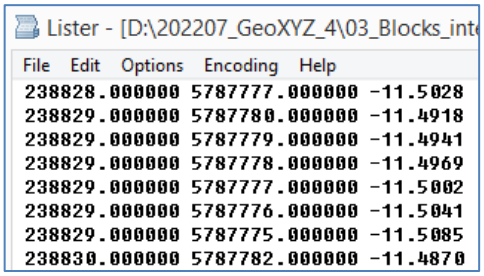


Figure 2.3 An example of a grid-file (fragment)

6) Output file for AutoChart

The ASCII-file includes the following columns: Easting, Northing, Horizon1, …, HorizonN, KP. Delimiter: ','; missed values write as "9999.000". The first line is Header which includes a names of horizons. The header is formed in accordance with "Horiz.txt" sequence. The example with a number of lines from an 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,RecentSediments_Lev2_B15B,KP
240736.370,5779499.308,31.090,32.901,9999.000,9999.000,31.599,9999.000,9999.000,9999.000,9999.000,9999.000,9999.000,9999.000,9999.000,9999.000,9999.000,9999.000,9999.000,9999.000,149.601
240736.477,5779501.305,31.085,32.895,9999.000,9999.000,31.597,9999.000,9999.000,9999.000,9999.000,9999.000,9999.000,9999.000,9999.000,9999.000,9999.000,9999.000,9999.000,9999.000,149.603
240736.583,5779503.302,31.086,32.889,9999.000,9999.000,31.593,9999.000,9999.000,9999.000,9999.000,9999.000,9999.000,9999.000,9999.000,9999.000,9999.000,9999.000,9999.000,9999.000,149.605

3. Script commands

There is a following format for script commands:

{'CommandName', Parameter1, …,ParameterN}; ScriptName;

The ScriptName will be Grid2AutoChart.

1) Define basic parameters for processing

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

SU_Set – a command name of the script;

d:\202207_GeoXYZ_4\for_chart\ – path to project Root Folder (***Figure 2.1***);

2 – step for meters for KP column in output file.

The command read RPL-file (RouteKP.txt) and Horizons list (Horiz.txt), create a "tmp" folder. The structure contained empty horizons is being created in the Computer operative memory which will be filled with the use of script commands. The "Route-structure" is put in the variable "rp" (***Figure 3.1***).

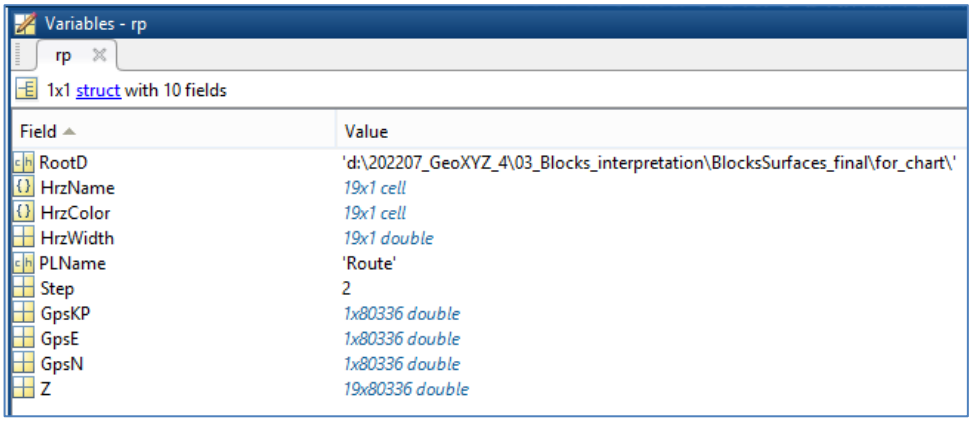


Figure 3.1 "Route-structure" in the variable "rp"

2) Grid-files read in the structure

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

Hrzs_Read – a command name of the script;

01\ – subfolder "01" in the Root folder.

The command reads all Grids from sub-folder (for one Block) and saves it to "tmp" folder in MatLab format to optimize work with Computer operative memory. The Grids headers and structure keep in operative memory in the "Grids-structure" in variable "Hrz" (***Figure 3.2***).

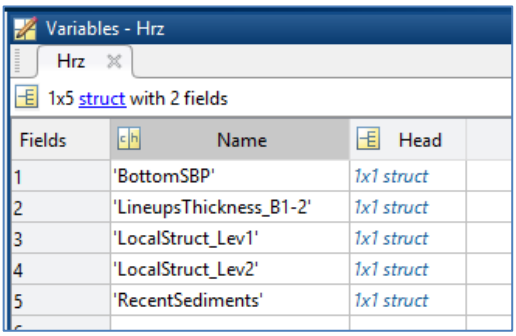


Figure 3.2 "Grids-structure" in the variable "Hrz"

The names, numbers and cell size (for Horizons/Grids where loaded) are printed in 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: Hrzs_Read command echo

The described following script commands HrzSurface_Draw, Hrzs_Calc&Draw dealt with current loaded grids – for one Block surveyed (from one sub-folder in a Root folder). The command Hrzs_Calc&Draw gets data from current "Grids-structure" and puts it in a "Route-structure". We need to load each Block one by one using Hrzs_Read and put data in the "Route-structure" using Hrzs_Calc&Draw to create single AutoChart file for all route.

3) Surface drawing

{'HrzSurface_Draw',1};Grid2AutoChart;

HrzSurface_Draw – a command name of the script;

1 – the surface number (see *Figure 3.3*).

The drawn Horizon/Grid name is printed in Command Window (*Figure 3.4*).

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

Figure 3.4 MatLab Command Window: HrzSurface_Draw command echo

The new window with route-line and horizon image will be opened when command is being processed (*Figure 3.5*). The zoom and pan tools can be used to create better view for screenshot or figure saving (*Figure 3.6*). This view can be used as a "map" for Geomodel.

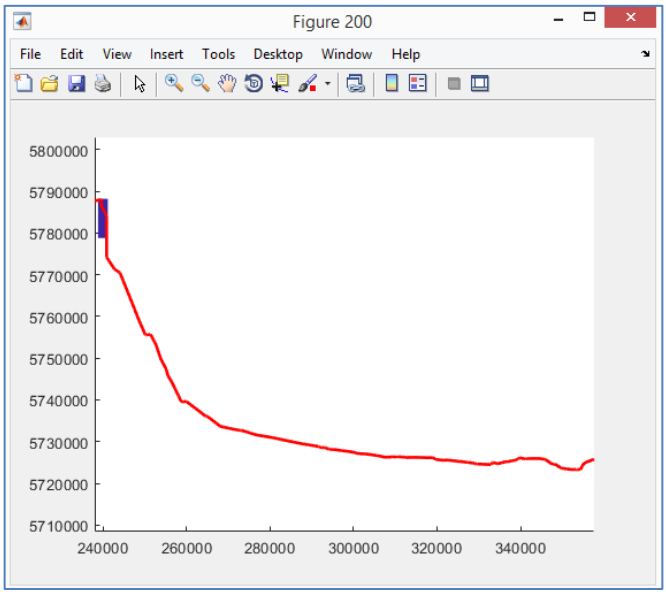


Figure 3.5 Surface figure for the part of route (red line)

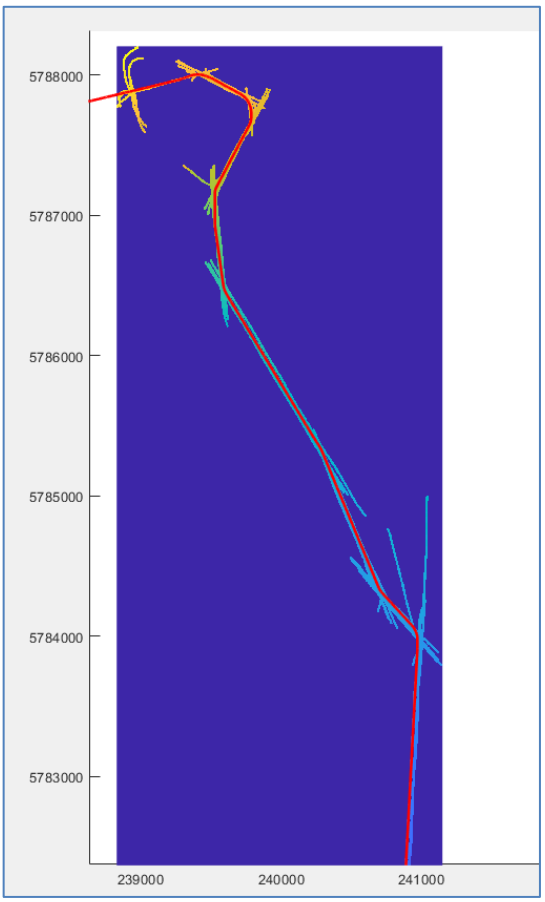


Figure 3.6 Screenshot of zoomed surface figure (see Figure 3.5)

4) Calculation of horizons for KP-interval

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

Hrzs_Calc&Draw – a command name of the script;

[149602 159400] – the interval of KP in meters is used to get data from Grids and to put it in "Route-structure"; the Grids data out of KP-interval will be ignored;

[15 15] – the gap-and-parts maximum size in points. The first step – any gaps in horizons smaller than 15 points will be covered by using linear interpolation; the second step – any parts of horizons less than 15 points (which means "parts" bordered with "gaps") will be deleted.

The command makes the following action:

-- get data from Grids (in defined KP-interval) and put it in "Route-structure";

-- analyze "gap-and-parts"; interpolate gaps smaller than defined and delete "parts" (which means "parts" bordered with "gaps") smaller than defined;

-- draw figure with Geomodel in defined KP-interval (*Figure 3.7*).

There are follow features:

-- the zoom and pan tools can be used to create a better view for screenshot or figure saving;

-- the legend can be dragged (by using a mouse) in any part of the figure;

-- the legend includes on-screen horizons only;

-- Color of horizons and Line Width used was preliminary read from "Horiz.txt".

Figure 3.7 Figure with Geomodel in defined by KP-interval

5) Horizons drawing for KP-interval

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

Hrzs_Draw – a command name of the script;

[149602 159400] – the interval of KP in meters used to draw the figure (for example, it can be all route).

The legend for this command includes all horizons defined in "Horiz.txt" (*Figure 3.8*).

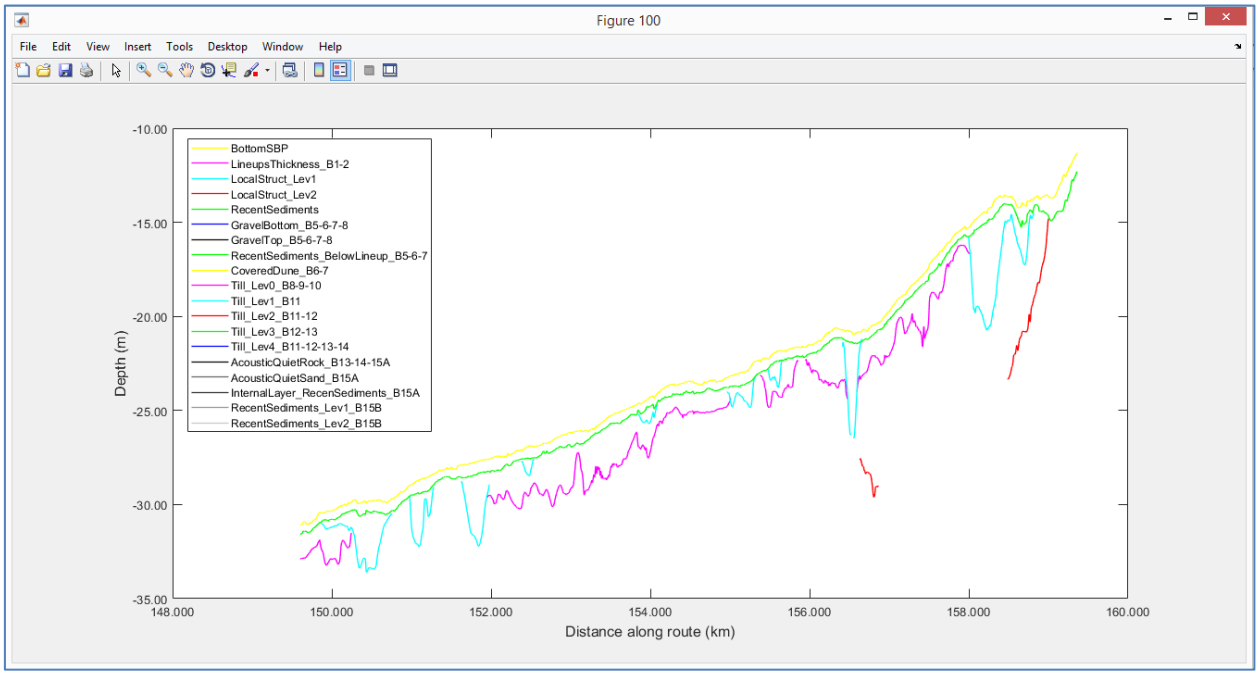


Figure 3.8 Figure with Geomodel (Hrzs_Draw command result)

6) Horizons saving to AutoChart-file

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

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

ChartFile_Write – a command name of the script;

Chart\B01.txt – AutoChart-file name; the file will write in sub-folder "Chart" located in the Root project folder;

[149602 159400] – the interval of KP in meters used to write data in AutoChart-file (the B01.txt is AutoChart file for Block 01 only).

The parameter [000750 159400] writes data to AutoChart-file for all KP used and defined in BlocksBord.txt.

7) Workspace saving

{'WspSave'};Grid2AutoChart;

WspSave – a command name of the script.

The command saved processing (current variables state) in the file "Wsp.mat" in Root project folder. To recover/load processing state need to use command

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

d:\202207_GeoXYZ_4\for_chart\Wsp.mat – path to the file in the Root project folder.

4. Geomodel picture edit

It can be necessary to make a careful Geomodel picture edit. The following way looks the best:

-- go to "Save As" menu in a Figure window (*Figure 4.1*);

-- save file with "*.eps" extension (Enchanted Post Script);

-- open file in software worked with vector graphics (for example, Corel Draw; *Figure 4.2*).

You can edit any elements of Geomodel as a vector graphic (at the first step it will be fine to delete several invisible rectangles which formed a frame for the figure).

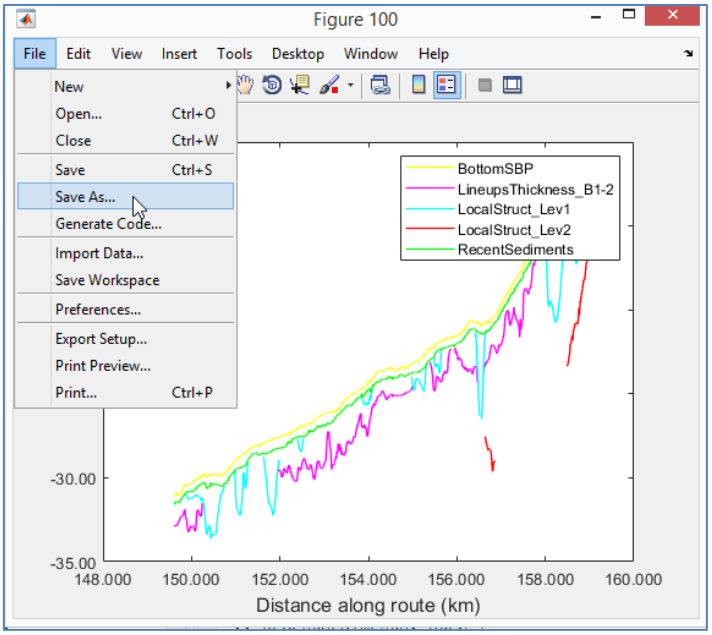


Figure 4.1 "Save As" menu

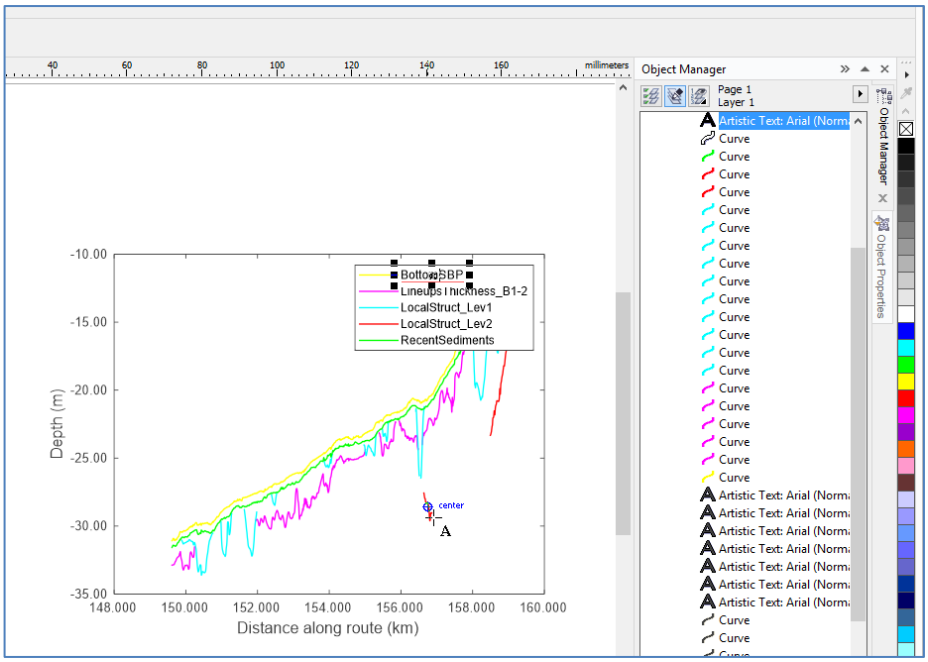


Figure 4.2 Vectors graphic in Corel Draw window

Appendix 1. 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 “Geomodel”.
%==== 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;
if strcmp(gKey{1},'SU_Set') %Read RPL and horizons list
   %{'SU_Set','d:\202207_GeoXYZ_4\03_Blocks_interpretation\BlocksSurfaces_final\for_chart\',2};Grid2AutoChart;
   try RootD=gKey{2};catch,RootD=input('Root folder name=');end; try stp=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']);
   clearvars RootD stp fId Hrz A L1 L2 n;
   %output >> rp
end;
if strcmp(gKey{1},'Hrzs_Read') %Read surfaces to structure from sub-folder
   %{'Hrzs_Read','01\'};Grid2AutoChart;
   try fName=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),if strcmp(rp.HrzName{nnn},fNameN(L1(end)+1:L2(end)-1)),tmp(nnn)=1;end;end;
      if sum(tmp)==0,error(['Name not found: ' fNameN(L1(end)+1:L2(end)-1)]);end; if sum(tmp)<1,error(['Name dupliceted: ' 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));if s1==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;
if strcmp(gKey{1},'ChartFile_Write') %Write horizons to file
    %{'ChartFile_Write','B01z.txt',[149602 159400]};Grid2AutoChart;
    try fName=gKey{2};catch,fName=input('Horizons file name=');end; try LLL=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;
if strcmp(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 10/11/2022
```