# Isoliner - grids and isolines

## Contents


**Introduction** **8**
- Installation and location . . . 8
- Updating . . . 8
- Help and common dialog elements . . . 9
  - Interface and help language . . . 9
- All the tools . . . 9

**Quick start** **11**
- Boreholes on hand, a raster and contours wanted . . . 11
- Open country on hand, terrain and a base map wanted . . . 12
- A topographic plan with contours on hand, terrain wanted . . . 12
- A dense survey on hand, crests and toes wanted . . . 13
- A quarry on a topographic plan with no contours inside . . . 13
- A geological section wanted . . . 14
- Volumes of work wanted . . . 14

**General workflow** **14**
- Running from a model or a script . . . 15
- Field names . . . 15
- Precision of numbers in tables . . . 16
- The basemap . . . 16

**1. Grid and isolines** **16**
- The elevation from the geometry . . . 16
- 1.01 2D Kriging (points → raster) . . . 17
  - Local anisotropy . . . 18
  - Outliers as a separate layer . . . 18
  - Automatic values . . . 19
  - Clip to well hull . . . 19
  - Variogram and nugget . . . 19
  - Outlier removal . . . 24
  - Kriging standard error . . . 25
  - Trend removal (regression kriging) . . . 26
  - Block kriging . . . 26
  - Faults . . . 27
  - Why the fault is never turned into cells . . . 27
  - The price of the approximation . . . 28
  - How to try it on teaching data . . . 28
- 1.02 Minimum curvature (points → raster) . . . 28
  - Faults . . . 29
  - The membrane band along the line . . . 29
- 1.03 MBA: multilevel B-splines (points → raster) . . . 30
  - What it is good at . . . 30
  - Controlling the lattices . . . 30

Removing the trend . . . 31
What it does not give . . . 31
Bounds of the result . . . 31
Clipping . . . 32
1.04 Isolines from raster . . . 32
Isoline smoothing . . . 33
Contour polygons (bands) . . . 34
Straight edges on the belt drawing . . . 34
Belt solids . . . 34
Layer styling . . . 35
The value in the geometry Z . . . 35
Contour thinning . . . 35
Topographic labels . . . 35
Warning about flat levels . . . 36
Contour confidence . . . 36
Faults . . . 37
The corridor at a fault . . . 37
Smallest polygon thickness . . . 38
1.05 Declustering (weights) . . . 38
1.06 Variogram (experimental) . . . 38
Why the preview is needed . . . 39
A short theory . . . 39
Parameters . . . 39
The grouping field and a mixed-density survey . . . 41
Three typical geological situations . . . 41
Maximum distance and reaching the plateau . . . 42
Where the nugget came from . . . 43
When the fit is good for nothing . . . 43
The workflow with cross-validation . . . 43
The table of variogram models . . . 44
Structure of the table . . . 44
Order of work . . . 44
Parsing of rows . . . 45
1.07 Variogram map (anisotropy) . . . 45
What anisotropy is and why to see it . . . 45
How to read the map . . . 45
Parameters . . . 46
Estimating the azimuth, anisotropy and range . . . 47
When the range hits the window . . . 47
The surface raster . . . 48
1.08 Variogram cross-validation . . . 48
1.09 Method cross-validation (LOO) . . . 51
1.10 Example wells (demo) . . . 53
1.11 Example geophysical profiles (demo) . . . 55
Kriging or minimum curvature . . . 56
1.12 Graft a detailed surface into a regional one . . . 56
Two kinds of step, of different nature . . . 57
The extent and the resolution of the result . . . 58
The graft area has to be narrower than the survey . . . 58
What goes into the log and what to look at . . . 58
Topography: terrain from open data . . . 59

**2. Topography** **59**
2.01 Download DEM by extent . . . 59
2.02 Download base topography by extent . . . 62
2.03 Topo2Raster (terrain from vectors) . . . 63
Three-dimensional thalwegs . . . 64

The surface between structural lines . . . 65
The boundary of the build area . . . 65
2.04 Terrain preparation . . . 66
2.05 Slope and aspect . . . 67
2.06 Peaks and pits . . . 67
2.07 Demo relief . . . 68
A pair of surfaces for volumes . . . 68
The gully and ravine network . . . 69
Where the demo lands . . . 69

**2. Topography: diagnostics and repair** **69**
2.08 Split contours for validation . . . 69
Why split at all . . . 69
Why the split is by elevation, not by feature . . . 69
Outputs . . . 69
Working order . . . 70
Parameters . . . 70
2.09 Contour residuals against the DEM . . . 70
The numbers it reports . . . 70
Two figures instead of one . . . 70
The report . . . 70
The point layer . . . 71
Parameters . . . 71
2.10 Terracing check of a DEM . . . 71
Two independent signs . . . 71
The contour interval . . . 72
Flat areas are kept out of the count . . . 72
The report . . . 72
How to remove it . . . 72
Parameters . . . 72
2.11 Remove steps (clamped smoothing) . . . 73
How that is possible . . . 73
The treatment checks itself . . . 73
What the tool does not do . . . 73
Parameters . . . 73

**2. Topography (continued)** **74**
2.12 Cut and fill (earthwork volumes) . . . 74
The sign and the reference surface . . . 74
Bringing both to one grid . . . 74
Why figures differ from other programs . . . 74
The dead band . . . 74
Work areas and the balance . . . 74
Outputs . . . 75
Parameters . . . 75
A quick check on the demo . . . 75
Scope . . . 76
2.13 Crest and toe candidates . . . 76
2.14 Crests and toes into work . . . 77
2.15 Example open pit (demo) . . . 78
2.16 Profiling of slopes . . . 78
Spot heights as a second source . . . 79
Where the vertices go . . . 79
2.17 Remove bridges from a DEM . . . 79

**3. Additional analysis tools** **80**
3.01 Categorical indicator kriging . . . 80

How it is computed . . . 81
When a borehole misses its own zone . . . 83
What you get . . . 83
Vector boundaries from the probabilities . . . 84
3.02 External Drift Kriging . . . 84
How it differs from trend removal . . . 85
How it is computed . . . 85
Parameters . . . 86
The variogram on residuals . . . 86
3.03 Exceedance probability map . . . 86
How it is computed . . . 87
How to get the inputs . . . 87
Parameters . . . 87
Use . . . 88
3.04 Hydraulic gradient and flow direction . . . 89
Without permeability . . . 90
Parameters and smoothing . . . 90
Arrows from points . . . 90
The learning cycle . . . 91
3.05 Specific discharge (Darcy law) . . . 91
What is computed . . . 91
How to get the K and T rasters . . . 91
Parameters . . . 91
What the arrow layer holds . . . 92
Use . . . 92
3.06 Gaussian simulation (SGS) . . . 92
Parameters . . . 93
Kriging kinds: which one to pick . . . 93
Cheat sheet . . . 95
Equal distances at the edge of the selection . . . 96
When the system does not solve . . . 96
3.07 Density from measurements (variable support) . . . 96
Output and invariant . . . 96
How to read . . . 97
The “Density map” window (live preview) . . . 97
3.08 Example for density (demo) . . . 97
3.09 Example for 3.01 and 3.02 (demo) . . . 98

**4. Cross-sections** **98**
4.01 Cross-section along a line . . . 98
How beds are defined . . . 99
A section over a single surface . . . 99
The layer tree sets the order of the surfaces . . . 99
Outputs . . . 100
Vertical scale . . . 100
Several sections in one run . . . 100
Band colours: the bed reference . . . 101
Attributes of the output layers . . . 102
Parameters . . . 102
Trying it on a demo . . . 103
Relation to QGIS . . . 104
4.02 Boreholes on sections (drilling model) . . . 104
The drilling model . . . 104
Inclined holes . . . 104
Holes without intervals . . . 105
When collars and intervals do not match . . . 105
The tolerant reader . . . 105

Batch operation and alignment . . . 105
Colours and the legend . . . 105
Clipping by the drawing and the frame . . . 106
Column colours: the bed reference . . . 106
Parameters . . . 106
Colouring by grade . . . 107
4.03 Bed composition on the section . . . 107
Two modes . . . 107
Parameters . . . 107
The section definition and shared parameters . . . 108
The model manifest instead of a list of layers . . . 109
4.04 Intersect surfaces with the section . . . 110
Parameters . . . 110
4.05 Vector intersection with the section . . . 110
Parameters . . . 111
The name and the style of the source layer . . . 112
Clipping by the terrain and by the bottom line . . . 112
Feature attributes on the section . . . 113
4.06 Intersect a TIN with the section . . . 113
Parameters . . . 114
Chains and closed sections . . . 114
4.07 Project objects onto the section . . . 115
Every section of the definition . . . 115
The name and the style of the source layer . . . 116
4.08 Unproject from the section . . . 116
The round trip: correcting a contact on the section . . . 116
A pit: every wall at once . . . 116
4.09 Shaft wall unwrap (beta) . . . 117
4.10 Example for a section (demo) . . . 117
Boreholes with sampling intervals . . . 118
Demo layers and their attributes . . . 118
Parameters . . . 119
4.11 Bed reference template . . . 120
What is inside . . . 120
The body is filled in by the geologist . . . 120
Composite bodies are grey . . . 120
The same file outside QGIS . . . 121
4.12 Attitude from an outcrop trace . . . 121
4.13 Bed grades at the collars . . . 121
Why the join goes by depths . . . 122
A missing grade is not a zero . . . 122
Parameters . . . 122
What the output layer holds . . . 122
Geological model . . . 123

**5. Geological model (beta) 123**
5.01 Consistency of a bed stack . . . 123
What is counted . . . 123
Where the order of beds comes from . . . 123
What comes out . . . 123
What the tool does not settle . . . 124
5.02 Example section (demo) . . . 124
What is in the example . . . 124
The grids in the fold zone are wrong on purpose . . . 124
What comes out . . . 124
How to use it . . . 125
5.03 Assembly of a stack from the relief . . . 125

Why this is reliable . . . 125
What is paid for it . . . 125
What counts as thickness . . . 125
Contact lines . . . 125
5.04 Correction of a stack by the statistics of thicknesses . . . 125
5.05 Model manifest . . . 126
5.06 Folding of a surface . . . 126
Hydrology . . . 127

**6. Hydrology** **127**
6.01 Flow and accumulation (D8) . . . 127
6.02 River network . . . 128
6.03 Basins and watersheds . . . 128
6.04 Gauge point report . . . 129
How it works . . . 129
What is computed . . . 129
Three lengths that are easy to confuse . . . 130
Parameters . . . 130
A quick check on the demo . . . 130
Scope . . . 130
Computation to SP 33-101-2003 . . . 131
The length of the stream network . . . 131
6.05 Catchment of a line or an outline (ditches, open pits) . . . 131
How it works . . . 131
An outline instead of a line . . . 132
Burning the trace . . . 132
What is computed . . . 132
Parameters . . . 132
A quick check on the demo . . . 133
Scope . . . 133
6.06 Report on an existing catchment . . . 133
The fields of the output layer . . . 134
The outlet . . . 134
Computation to SP 33-101-2003 and the stream network length . . . 135
6.07 Flow lines from points, lines and outlines . . . 135
6.08 Cross-sections and rating curves . . . 136
How it is computed . . . 136
The division . . . 136
Roughness and slope . . . 137
Levels . . . 138
The section drawing as one layer . . . 138
The drawing footer . . . 138
What matters about the method . . . 138
What comes out . . . 138
The rating curve plot . . . 139
6.09 Flood extent polygon . . . 139
The level is not constant along the river . . . 139
6.10 Import section tables . . . 140
6.11 Example river (demo) . . . 140
How to check the group . . . 140
What the group does not have yet . . . 140

**7. Fractal analysis** **141**
7.01 Fractal dimension . . . 141
Reading the map . . . 141
Picking the window and the lags . . . 141
Workflow . . . 142

Parameters . . . 142
7.02 Box-counting of masks . . . 142
Where the mask comes from . . . 142
7.03 Dimension of lines and boundaries . . . 143
An isoline-smoothing diagnostic . . . 143
Other uses . . . 143
7.04 Minkowski dimension (vectors) . . . 143
7.05 Example for fractals (demo) . . . 144

**8. Data exchange** **144**
8.01 Read LandXML . . . 145
What is read . . . 145
Parameters . . . 145
The standard elements of the schema . . . 146
The fields of the output layers . . . 146
The coordinate order . . . 147
Units . . . 147
Curves . . . 147
Cross sections on the ground . . . 147
8.02 Write LandXML . . . 148
What is written . . . 148
Cross sections and the alignment . . . 148
Limits . . . 148
8.03 Example LandXML (demo) . . . 148
The variants . . . 148
Where the shapes come from . . . 149
How to work with it . . . 149

**9. Subsidence** **149**
9.01 Tilt and curvature from subsidence . . . 150
How it is computed . . . 150
Sign . . . 150
Check against profile lines . . . 150
9.02 Horizontal strain . . . 151
Estimate from curvature . . . 151
From measured intervals . . . 151
9.03 Example subsidence trough (demo) . . . 152
How to check . . . 152

**Typical situations and solutions** **153**

**Appendix. Isoliner in 15 minutes** **153**
Scenario 1. I have points . . . 153
Scenario 2. I need a relief . . . 154
Scenario 3. I need a section . . . 155
Where to go next . . . 155

**Appendix. Demo layer fields** **155**
Topography and Hydrology group outputs . . . 155
Sample wells (demo) - tool 1.10 . . . 156
Electrical-prospecting profiles - tool 1.11, electrical mode . . . 156
Subsidence profiles - tool 1.11, subsidence mode . . . 156
Section data - tool 4.10 . . . 156
Fractal data - tool 7.05 . . . 157
Density (demo) - tool 3.08 . . . 157

**Appendix. Variable-support density: step by step** **157**

Step 1. Generate the demo (3.08) . . . 158
Step 2. Density from points (3.07) . . . 158
Step 3. Density from lines . . . 158
Step 4. Density from polygons and dasymetry . . . 158
Step 5. Mix types by appending . . . 158
Step 6. Effective sigma . . . 159
Reading the result . . . 159

**For enterprises** **159**
Log and toolbar . . . 159

**License and support** **159**

# Introduction

Isoliner is a Processing provider for interpolating point data, building isolines and working with terrain. The kriging core is the KB2D algorithm from GSLIB. It works on roof elevations, thicknesses, geomechanical properties, chemistry and any numeric well attribute. The tools are split into seven groups.

- **Grid and isolines** - the main processing flow, from declustering to isolines.
- **Topography** - terrain from open data, its checking and repair, earthwork volumes and open pits.
- **Additional analysis tools** - computations from indicator kriging to variable-support density.
- **Cross-sections** - building geological sections.
- **Geological model** (beta) - consistency of a bed stack and a teaching example of a section.
- **Hydrology** - flow, river network, basins and catchments, cross-sections, rating curves and flooding.
- **Fractal analysis** - dimensions of surfaces, masks and lines.
- **Data exchange** - reading and writing LandXML.

A few terms used below. A variogram describes how much more strongly values differ as the distance between points grows. The sill is the level it reaches (close to the data variance). The nugget (from the “nugget effect”) is the jump of the variogram at zero - the scatter at arbitrarily small distances caused by measurement noise and microvariability.

## Installation and location

The main way is from the official QGIS repository. Open Plugins → Manage and Install Plugins → the **All** tab, type “Isoliner” in the search, select the plugin and click **Install**. When installed from the repository, QGIS itself reports new versions and updates the plugin at the press of a button.

The alternative way is from a ZIP file. Plugins → Manage and Install Plugins → Install from ZIP. This is handy for offline installation and pre-release builds.

After installation the tools appear in the **Processing** panel, under the **Isoliner** provider, in the same seven groups. They run in QGIS 3.16 and newer. Nothing has to be installed alongside it, everything is computed with the means QGIS already ships with.

## Updating

A plugin installed from the repository reports new versions itself. An icon appears in the status bar, and the version is listed on the **Upgradeable** tab of the plugin manager. Updating is a single click. A plugin installed from ZIP is updated the same way, over the old one.

The plugin reloads cleanly on the fly, no QGIS restart is required. After an update the tools are available in the panel straight away.

## Help and common dialog elements

Each tool's dialog has a **Help** button that opens this manual (the PDF bundled with the plugin). The right-hand panel of the dialog shows a short help text for the tool, and a hint for a field appears when the pointer rests on it. The manual and the version details are also available without opening a tool. The **Plugins** menu holds an **Isoliner** submenu with **Density map**, **Basemap**, **About** (version, links, changelog and the manual), **Language** and **Log**.

### Interface and help language

The interface language and the help language are chosen separately, in the **Plugins → Isoliner → Language** window. The same window opens from the button with the letters «A» and «Я» on the Isoliner toolbar. The interface covers the menus, the tool names and the field labels, while the help covers the right-hand panel of the dialog, the field hints and the PDF manual. This way the QGIS interface can stay in English while the help is read in Russian, or the other way round.

Each of the two languages has three values: **As in QGIS**, **Русский** and **English**. By default both follow the language of QGIS, as before. The language names in the list are written in the languages themselves, so a reader finds their own language even when the window opens in one they do not know.

The menus and the tool names in the Processing panel are rebuilt as soon as **OK** is pressed, with no need to restart QGIS. The help changes the next time a tool is opened.

Raster band choice in all the tools is a drop-down with band names. A multiband bed grid shows roof, bottom and the parameter layer names in the lists.

## All the tools

Below is the whole provider as it stands in the **Processing** toolbox.

### 1. Grid and isolines

- `1.01` 2D Kriging (points → raster)
- `1.02` Minimum curvature (points → raster)
- `1.03` MBA: multilevel B-splines (points → raster)
- `1.04` Isolines from raster
- `1.05` Declustering (weights)
- `1.06` Variogram (experimental)
- `1.07` Variogram map (anisotropy)
- `1.08` Variogram cross-validation
- `1.09` Method cross-validation (LOO)
- `1.10` Example wells (demo)
- `1.11` Example geophysical profiles (demo)
- `1.12` Graft a detailed surface into a regional one

### 2. Topography

- `2.01` Download DEM by extent
- `2.02` Download base topography by extent
- `2.03` Topo2Raster (terrain from vectors)
- `2.04` Terrain preparation
- `2.05` Slope and aspect
- `2.06` Peaks and pits
- `2.07` Demo relief
- `2.12` Cut and fill (earthwork volumes)
- `2.13` Crest and toe candidates
- `2.14` Crests and toes into work
- `2.15` Example open pit (demo)
- `2.16` Profiling of slopes

- `2.17` Remove bridges from a DEM

**2. Topography: diagnostics and repair**

- `2.08` Split contours for validation
- `2.09` Contour residuals against the DEM
- `2.10` Terracing check of a DEM
- `2.11` Remove steps (clamped smoothing)

**3. Additional analysis tools**

- `3.01` Categorical indicator kriging
- `3.02` External Drift Kriging
- `3.03` Exceedance probability map
- `3.04` Hydraulic gradient and flow direction
- `3.05` Specific discharge (Darcy law)
- `3.06` Gaussian simulation (SGS)
- `3.07` Density from measurements (variable support)
- `3.08` Example for density (demo)
- `3.09` Example for 3.01 and 3.02 (demo)

**4. Cross-sections**

- `4.01` Cross-section along a line
- `4.02` Boreholes on sections (drilling model)
- `4.03` Bed composition on the section
- `4.04` Intersect surfaces with the section
- `4.05` Vector intersection with the section
- `4.06` Intersect a TIN with the section
- `4.07` Project objects onto the section
- `4.08` Unproject from the section
- `4.09` Shaft wall unwrap (beta)
- `4.10` Example for a section (demo)
- `4.11` Bed reference template
- `4.12` Attitude from an outcrop trace
- `4.13` Bed grades at the collars

**5. Geological model (beta)**

- `5.01` Consistency of a bed stack
- `5.02` Example section (demo)
- `5.03` Assembly of a stack from the relief
- `5.04` Correction of a stack by the statistics of thicknesses
- `5.05` Model manifest
- `5.06` Folding of a surface

**6. Hydrology**

- `6.01` Flow and accumulation (D8)
- `6.02` River network
- `6.03` Basins and watersheds
- `6.04` Gauge point report
- `6.05` Catchment of a line or an outline (ditches, open pits)
- `6.06` Report on an existing catchment
- `6.07` Flow lines from points, lines and outlines
- `6.08` Cross-sections and rating curves
- `6.09` Flood extent polygon
- `6.10` Import section tables
- `6.11` Example river (demo)

**7. Fractal analysis**

- `7.01` Fractal dimension
- `7.02` Box-counting of masks
- `7.03` Dimension of lines and boundaries
- `7.04` Minkowski dimension (vectors)
- `7.05` Example for fractals (demo)

**8. Data exchange**

- `8.01` Read LandXML
- `8.02` Write LandXML
- `8.03` Example LandXML (demo)

**9. Subsidence**

- `9.01` Tilt and curvature from subsidence
- `9.02` Horizontal strain
- `9.03` Example subsidence trough (demo)

*Tools in total: 79*

# Quick start

There are many tools in the plugin, and picking one out of the general list is hard. This section is arranged the other way round, not by tools but by tasks. Find the one that looks like yours and follow the steps - the rest can be left unread.

One rule runs through all of it - **the tools do not guess, they ask and they report**. Nearly every one writes into the log what exactly it decided and why the count of objects came out as it did. Open the log every time, especially at first. The answer to “why did I get the wrong thing” usually lies there.

## Boreholes on hand, a raster and contours wanted

The commonest task and the oldest - turn points with measurements into a surface. Two tools are enough.

**1.01 2D Kriging** builds a raster from the points. Ordinary kriging estimates the value in a cell as a weighted average of the nearest measurements. It does not invent the weights but derives them from how quickly the values drift apart with distance. You supply the value field and three parameters of the model. The **range** is the distance beyond which the measurements tell you nothing. The **sill** is the overall spread. The **nugget** is the spread that remains at zero distance - measurement error plus variability finer than the spacing of the network. Take a cell about a quarter to a fifth of the mean spacing between the boreholes. Finer adds no accuracy and takes longer.

Along with the surface the tool produces the **kriging error map**. Always look at it. The surface itself looks equally smooth where the boreholes are dense and where the result rests on a single distant measurement, and only the error map shows the difference.

**1.04 Isolines from raster** turns the surface into contours with labels and, if wanted, into polygons of ranges. The interval is set as a number or as a step. For terrain there is a topographic labels checkbox there too: the labels are turned so that their top faces the high side.

It can be tried without your own data. **1.10 Example wells (demo)** produces a ready set of points with a realistic spatial structure.

**Further on, when justification is required.** The parameters of the model need not be guessed by eye. **1.06 Variogram** derives them from the data themselves and draws a plot that shows whether there is any spatial relation in the data at all. If the experimental points fall without any pattern, no method will create one. **1.07 Variogram map** shows the anisotropy - the similarity reaches further along one direction than across it, and that has to be taken into account. **1.09 Method cross-validation (LOO)** removes each borehole in turn, predicts its value from the rest and prints

the discrepancy. That is a direct answer to how far the map can be trusted. A successful set of parameters is stored as a profile, so it need not be typed again.

## Open country on hand, terrain and a base map wanted

There are no surveys of your own but a map is needed - the early and the most common case. Everything is taken from open sources right inside QGIS, with no archives downloaded by hand.

**2.01 Download DEM by extent** fetches the elevations over the given extent - Copernicus GLO-30 and other open sets. What arrives is a raster in metres, already brought to a metric coordinate system. Thirty metres of cell is a scale of about 1:25000: hills, valleys and watersheds read well, the crest of a quarry or a road embankment does not. That has to be understood at once, otherwise the impossible is expected of the result.

**2.02 Download base topography by extent** takes the vector setting from OpenStreetMap over the same frame: water bodies, rivers, roads, buildings, forest. Water arrives as polygons, composite ones included - a lake with an island stays a lake with a hole rather than turning into a solid patch. This is the layer that later goes into the interpolation as the water edge.

A downloaded DEM nearly always needs **2.04 Terrain preparation**. Open sets carry voids, noise and local pits that do not exist in nature. Filling the depressions here is not cosmetics. Without it the flow is not computed at all - the water runs into the first false pit. After that **6.01 Flow and accumulation** gives directions and accumulation. **6.02 River network** turns the accumulation into thalweg lines with orders. **6.03 Basins and watersheds** cuts the territory into catchments.

**2.06 Peaks and pits** finds the characteristic points of the terrain - the very spot heights that a paper map labels with a number.

**And here comes the main step, the one everything before it was done for.** A downloaded DEM is hydrologically wrong. It holds false pits, rivers run uphill in places, lakes have a slope. Curing that with filters is pointless - filling the depressions treats the symptom, not the cause. The right way is different. Assemble the skeleton of the terrain out of the downloaded raster and **recompute the surface from scratch**, taking into account what has been learned about it.

Everything gathered at the previous steps goes into **2.03 Topo2Raster**, each along its own input. Contours from 1.04 as hard nodes. Spot heights from 2.06 as hard nodes too, and with a weight above one if you want the peaks to outweigh contour vertices in a shared cell. Thalwegs from 6.02 as a downstream descent constraint, so that on the new terrain the rivers are guaranteed to run downhill. Water polygons from 2.02 as the water edge, so that the lakes lie flat rather than sloping. What comes out is a **hydrologically correct terrain**, which the downloaded raster by itself does not give.

The contours from **1.04** are needed twice in this chain. First as a way of looking at the country, then as an input to the recomputation. Take the interval by the scale and the character of the country - five metres on a plain, twenty five in the mountains. The tool prints the range of elevations to the log. Too fine an interval only does harm here. It carries the noise of the original raster into the new terrain, the very thing the work was meant to remove.

To check that it became better rather than merely different, use the same tools as in the next scenario, **2.09 Contour residuals against the DEM** and **2.10 Terracing check of a DEM**. And **6.01 Flow** over the new terrain should give rivers without breaks and without false lakes. That is the most telling check of all.

## A topographic plan with contours on hand, terrain wanted

The task is the reverse of the first one. Here you do not build a surface from sparse points but recover it from dense lines that somebody has already drawn.

**2.03 Topo2Raster** does this work - a multigrid interpolation in the spirit of ANUDEM. The main difference from kriging is that every type of input has a role of its own. Contours and spot heights give hard nodes: the elevation there is pinned. Thalwegs impose a downstream descent, so that

the rivers on the built terrain do not run uphill. Cliffs work as a barrier: the drop along them is not smeared by the smoothing. The water edge lays lakes flat or tilts them along the channel.

Set the cell explicitly, otherwise the default may turn out coarser than what you want to see. This is the first of the common mistakes. On a kilometre-wide area the automatic size gives a thirty-metre cell, and a seven-metre bench simply does not exist in such a grid. The tool warns about it in the log, but it is better to set the cell at once.

The result must not be checked by eye. **2.08 Split contours for validation** holds part of the lines back. **2.09 Contour residuals against the DEM** measures how far the built surface departed from the ones held back. This is the same device as cross-validation in kriging, and it answers the same question. Separately there is **2.10 Terracing check of a DEM**, which looks for the characteristic defect of interpolation over contours, where the surface steps along the original lines, and **2.11** removes it.

## A dense survey on hand, crests and toes wanted

A UAV or laser-scanning survey gives terrain in which the benches are visible but not digitized. Digitizing them by hand is slow, and Isoliner can find them itself.

The mechanics in **2.13 Crest and toe candidates** are these. The evidence of a break is not the slope but the **rate at which it changes**. On an even face, however sheer, the slope is constant and the evidence is small. It is large where the slope changes, that is on the crest itself and on the toe itself. The sign of the curvature splits the lines found into crests and toes automatically.

Two parameters here are not obvious. The **probe base** is the half-width of the window in which the drop across the line is measured. The drop in the `drop` attribute is computed within it, so on a ten-metre bench with a three-cell base it will read three metres rather than ten. Set the base by the width of the face in cells. The **drop cut-off** is a noise filter, not a criterion of significance. A formal definition of a crest does not exist, and deciding what counts as a bench stays with the human. The tool therefore deliberately returns more than needed. It puts the numbers for the selection into the attributes and prints the percentiles to the log, and you then move the layer filter while watching the map, recomputing nothing.

Next, **2.14 Crests and toes into work** takes the elevations off the terrain and assembles the forms. The pairs are determined by descending the slope rather than by proximity. Water from a crest runs exactly to its toe, while on a curved wall with narrow berms the nearest toe by distance often belongs to the neighbouring bench. A form comes out as one toe with several crests at it, because the tracing cuts a long crest into pieces, yet all the pieces descend to the same toe. Unpaired lines do not vanish, they go into a separate layer where the attribute records the reason - the descent did not reach a toe, or it scattered over different toes.

The whole chain can be checked on synthetic data. **2.15 Example open pit (demo)** builds a pit with benches, berms, a ramp, a dump and a ditch, and along with it the true structural lines. The completeness of the detector is measured as a number against them, not by eye.

## A quarry on a topographic plan with no contours inside

A case that looks hopeless until you look into it. By the standard, contours inside areal quarries, cuts, fills and dumps **are not described** - only the outer outline is agreed. So the terrain from such a plan has a hole exactly where the working is, and there is nothing to fill it with except crests and toes.

The catch is that the slope lines in the drawing carry no elevations. But the contours that adjoin them do. The standard requires the contours to be brought up to the object line with node points. **2.16 Profiling of slopes** gathers those points and recovers the profile of the whole line from them. The result is a varying elevation taken from the data itself rather than one number per object.

Then **2.14**, but this time **without a DEM**. When the lines carry elevations of their own the descent is not needed, and the toe is the nearest line lying below the crest. That matters, because otherwise

the work goes in a circle. To build the terrain you need pairs, and for pairs by descent you need terrain that does not exist yet.

**2.03** closes it. Contours as usual, and the forms go into the **Top of forms** and **Bottom of forms** inputs. The surface between the crest and the toe is built by the distance method. The body goes into the interpolation as hard nodes, and the border works as a barrier. There is no significance threshold in the tool on purpose: give two sides and a surface is built, give one and it works as a barrier. The decision is made by the person drawing the second line.

### A geological section wanted

A section in Isoliner is not an illustration but a coordinate system. **4.01 Cross-section along a line** turns a line on the map into a drawing plane with distance-elevation axes. Besides the drawing itself it produces a service layer, the **section definition**. All the other tools of the group take that layer as an input and place their objects in the same coordinates. Hence the rule. 4.01 goes first, everything else after, and the “Section definition” field takes the output of 4.01, not the original line.

The vertical exaggeration is set in three ways, and the choice is not obvious. A factor is simple but requires knowing the length of the section, since on a kilometre-long profile a factor of 3 gives a flat ribbon. **The ratio of the drawing’s dimensions** picks the factor itself, so that the width relates to the height as required, and it works on a section of any length - start with that. And remember that the exaggeration distorts **all** the angles. At a factor of 10 an inclination of 5 degrees lands on paper at 41, which is why angles are not measured on a drawing with a protractor.

Then the content is placed onto the section. **4.02** puts the boreholes with their sampling intervals. **4.04** draws surfaces from rasters. **4.05** places vector objects by their exact intersection with the line. The last one has a rule by object type. A flat line gives a vertical, because the where is known and the depth is not, a three-dimensional one gives a point at the real height, and a polygon gives a vertical band. When the dip and the dip direction are given, the vertical turns into an inclined trace and the band into a parallelogram.

The easiest start is the demo. **4.10 Example for a section (demo)** produces six surfaces, five beds, three section lines, boreholes, zones and the layers for checking the dips. The whole chain runs through it in five minutes, and there is nowhere to go wrong.

### Volumes of work wanted

**2.12 Cut and fill** compares two surfaces and computes how much was taken out and how much was laid in. Usually that is the terrain before the works and after, or a design surface against the actual one.

It is computed cell by cell, the difference of the elevations is multiplied by the area of the cell, and the positive and the negative parts are summed separately. Hence the obvious but important point - **the accuracy of the volume rests on the accuracy of both surfaces**. If one of them was built from sparse points, a handsome number in the report means nothing. The kriging error map and the residuals over the held-back contours are exactly about this.

Set the boundary of the computation with a polygon, otherwise the volume will be summed over the whole overlap of the rasters, including areas the works never touched. This is the second of the common mistakes. The difference of two DEMs is almost never zero outside the working, the survey noise remains there, and over a large area it accumulates into noticeable cubic metres.

## General workflow

A typical scenario has two steps. First **2D Kriging** builds a raster from a point layer and a numeric Z field, that is a regular grid of values. Then **Isolines from raster** builds isolines from that raster and, if needed, filled contour polygons.

The steps are independent, and **Isolines from raster** works with any raster, not only with a kriging result.

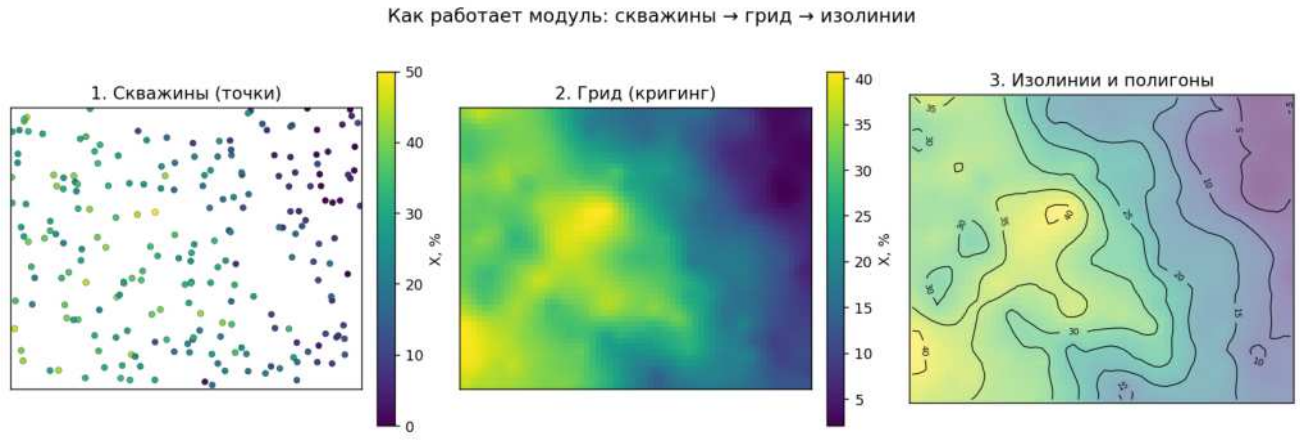



Figure 1: The whole process on a generated example: wells with measurements (left) are turned into a continuous grid by kriging (centre), from which isolines and contour polygons are built (right).

## Running from a model or a script

The parameters window remembers the values you entered and offers them the next time the same tool is run. That is convenient for repeated runs over the same data and works as before.

A call from the model designer or from a script is different. There the parameters the call did not set are taken from the declaration of the tool itself, not from what was typed into the window the day before. Otherwise the same script would compute differently on different machines, in different QGIS profiles and simply on different days, and the log would not show it. When a missing parameter differs from what the window remembers, the tool says so in the log and names the parameter.

Hence the rule for models. Set in the call every parameter whose value matters for the result. The declared default of a tool is known and written down in this manual, while the memory of the window is known only to the machine where it was formed.

## Field names

The fields of the layers the tools produce have short Latin names: **eta_mm**, **hole_id**, **roof**. Expressions, labels, styles and the next tools in a chain refer to these names. That is why they do not depend on the QGIS language, and a project built in a Russian QGIS opens in an English one without breaking.

Such names are awkward to read in a table, so the fields carry aliases. In an English QGIS the attribute table, the forms and the identify window show “Subsidence, mm”, “Borehole”, “Roof elevation, m”, and in a Russian one the Russian names. The fields of every tool that produces vector layers are labelled this way.

Only a field the plugin created gets an alias of its own. A field with the same name as a field of an input layer is taken to be yours. If it has an alias in the input layer, the output gets the same one, and if not, the field is left as it is. That is why a label follows a field along a chain: benchmarks from 9.03 arrive in 9.01 already labelled, and the section definition from 4.01 is labelled in 4.02 as well. Your own field with your own label keeps it in the output too.

If an output is saved to a GeoPackage file, the labels are written into the file itself as well. Such a file, opened in another project or handed to a colleague, shows the same field names without the original project. Temporary layers live only in the project, and so do their labels.

Temporary layers carry the aliases too. If such a layer is fed into a tool that copies fields into a temporary output, for example a buffer or an extract by attribute, QGIS writes the warning “Aliases are not compatible with scratch layers” for every field. The computation goes on as usual, and to avoid the warnings save the temporary layer to a file.

### Precision of numbers in tables

Numbers in the fields of the output layers are rounded to a sensible precision by the units of the field. Elevations, thicknesses and distances in metres are written to the centimetre, subsidence in millimetres to a tenth, tilts in mm/m to a thousandth, angles to a tenth of a degree, areas and volumes to a tenth. Quantities without a natural scale, such as hydraulic conductivity, a weight or a slope, are written with four significant digits. Twelve decimals on a roof elevation add no accuracy, and in a table or in an export to Excel they get in the way.

The fields of your input layers are not rounded. If a field of an output has the same name as a field of an input, the plugin leaves its values as they are.

### The basemap

The **Isoliner** toolbar carries a **Basemap** button - a map or satellite imagery under the data in one move. Tick the sources you need, press **Add**, and the layers go into the project. The window stays open, because basemaps are chosen while looking at the map. Add one, look, add another.

Seven sources ship with the plugin - OpenStreetMap, OpenTopoMap and five ESRI layers: satellite imagery (**ESRI World Imagery**), a topographic base, a street map, hillshade and slopes. The imagery has a second style, **Semi-transparent**, switched in the layer properties on the styles tab. The semi-transparent imagery goes over the data rather than under it, and positions are checked against it.

The **Put below the other layers** checkbox drops what was added to the very bottom of the tree, so that the basemap does not cover the work.

Your own source is added without touching the code. Set the layer up in QGIS as usual, then right-click in the layer tree and choose **Export - Save as Layer Definition File**. Put the resulting `.qlr` into the `basemaps` folder of the plugin. The order in the list is set by the numeric prefix of the file name. A layer definition is chosen for a reason. It already holds the connection string, the coordinate system and the style with its transparency, while an address assembled by hand loses all of that.

Basemaps are served in the web projection while a project usually works in a local system. If the coordinate system of the project has a datum, the transformation is computed and the basemap lines up. If it has none, the imagery shifts silently, so the window warns about it in advance, before anything is added.

## 1. Grid and isolines

### The elevation from the geometry

The value field Z is optional in kriging, minimum curvature, the variogram, the variogram map, both cross-validations and declustering. When no field is given, the elevation is taken from the third coordinate of the point.

Data often arrive exactly like that. A survey or an export gives PointZ features where the value is already in the geometry, and creating a column for it makes no sense.

Points may carry both a Z in the geometry and a value field, and the two often mean different things. The Z is usually the survey elevation, the field is the quantity being interpolated. For thicknesses and grades the field is given explicitly. A layer with no Z and no field is rejected with a message rather than failing in the middle of the run.

## 1.01 2D Kriging (points → raster)

Ordinary (OK) or simple (SK) kriging over a point layer. Coincident points (the same XY) are averaged over Z. At grid nodes the values of the source points are reproduced exactly (with a zero nugget).

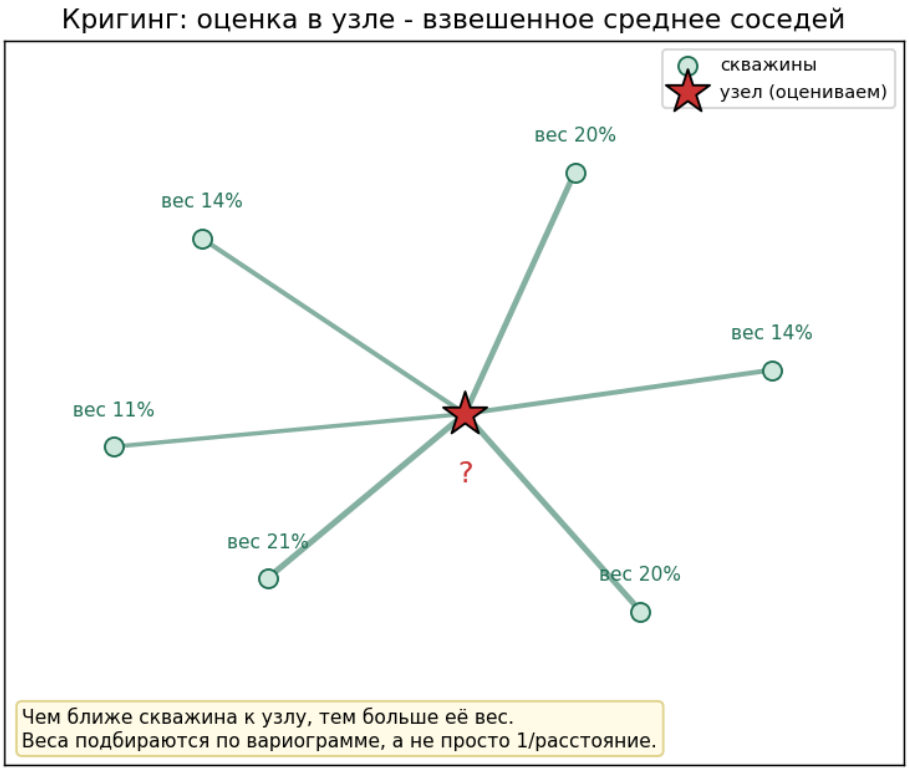



Figure 2: Kriging estimates the node value as a weighted mean of the nearest wells: the closer the well, the larger its weight. Weights come from the variogram.

Main parameters:

| Parameter | What it sets | Default / advice |
|---|---|---|
| Point layer | Source points (wells) for interpolation. | - |
| Selected features only | Compute only over the layer's selected points. | off |
| Value field Z (empty = from the geometry) | The numeric attribute that is interpolated: roof elevation, thickness, geomechanical property, chemistry, etc. | remembered between runs |
| Value transform | ln for log-normal quantities (K, T, grades with a long tail): ln(Z) is kriged and the estimate is returned via exp. | none |
| Kriging type | Ordinary (OK) - estimates the mean locally itself. Simple (SK) - uses the specified **Mean**. | OK |
| Search radius | Radius of the search window for neighbouring points around a node. 0 = whole sample. | 0 (whole sample) |

| Parameter | What it sets | Default / advice |
|---|---|---|
| Min. number of points | If the window has fewer points, the node stays empty (nodata). | 1 |
| Max. number of points | How many nearest points enter the kriging system. | 24 |
| Cell size | Grid step. 0 = auto = min(extent)/50. | smaller = smoother, but slower |
| Raster extent | The computation rectangle. By layer by default. | by layer |
| Clip to well hull | The raster is clipped to the convex hull of all points - removes extrapolation in empty corners. | recommended on |
| Hull buffer | Expand the hull outward by N map units. | 0 |
| Clip mask | Your own polygon instead of the hull (takes priority) - handy for concave areas. | - |
| Variogram models (table) | Table layer holding fitted models. When given, the variogram fields of the dialog are not read. | not set |
| Profile from the table | Name of the set in the table. May be left empty when the table holds a single profile. | empty |

### Local anisotropy

Normally the anisotropy is set by a single azimuth over the whole area. Where ore zones or geochemical haloes curve, that is not enough, the direction of elongation varies across the ground.

A strike field in the points switches on the local mode. Every sample carries its own value in degrees from north, and in every cell the kriging averages it over the same measurements it takes into the computation. The whole system is then solved with one model. Different models for different pairs of measurements would make the matrix not positive definite.

The strike is understood as an axis rather than a direction. Its period is a hundred and eighty degrees, so 170 and 10 differ by twenty degrees rather than by a hundred and sixty, and the averaging goes through the doubled angle. A plain mean would give the perpendicular to the truth here, and the error would be a quiet one. The map would stretch across the structure and look perfectly regular.

The anisotropy ratio stays common to the area, only the direction varies. A second varying parameter would make it possible to fit the map to expectation, and cross-validation would not show it.

Where the directions of the neighbours disagree, no local rotation is applied and the global model works. The tool does not invent a direction the data do not hold.

Check the result by cross-validation on a held-out sample. Local anisotropy makes the map look like the geological idea, and therein lie both its strength and its danger. With a wrong strike the kriging will stretch the anomalies along the error.

### Outliers as a separate layer

The points discarded by trimming can be obtained as an output - the coordinates, the value and the reason, below or above which bound. The output is off by default.

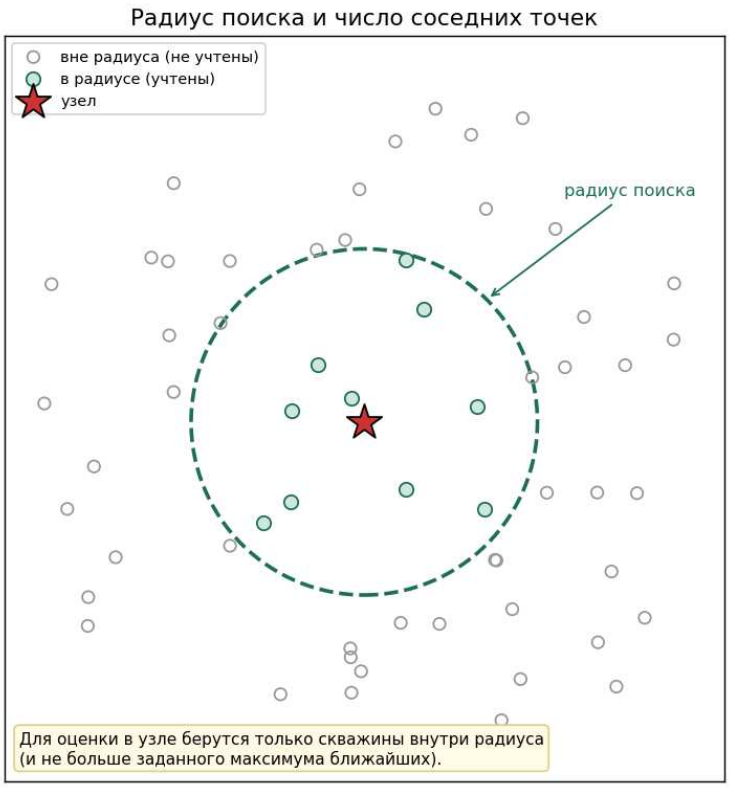



Figure 3: For each node only wells within the search radius are taken, and no more than the set number of nearest ones. Points beyond the radius do not take part.

They are worth looking at. Percentile trimming does not tell a measurement error from a genuine anomaly, and that decision is a geological one. A cluster of discarded points in one place usually means an unaccounted body rather than bad data.

**Automatic values**

Cell size = min(extent width, height) / 50.

Variogram correlation range = max(extent width, height) / 3.

Search radius (when 0) = the extent diagonal, i.e. the whole sample is taken.

**Clip to well hull**

Kriging computes the whole rectangular extent, so outside the data area the values are extrapolation and produce artefacts (long “fan” isolines in empty corners). The **Clip to well hull** option builds the convex hull of all points (with an optional buffer) and clips the raster to it. The extrapolation disappears. If the actual boundary of the area is concave, set your own polygon in the **Clip mask** - it takes priority over the hull.

**Variogram and nugget**

Kriging relies on a variogram model - it describes how strongly the Z values in two points differ depending on the distance between them. By this model a weight is assigned to each neighbouring well. The model is set in the **Advanced Parameters** section by three numbers - the nugget C0, the sill (C0 + C) and the correlation range a.

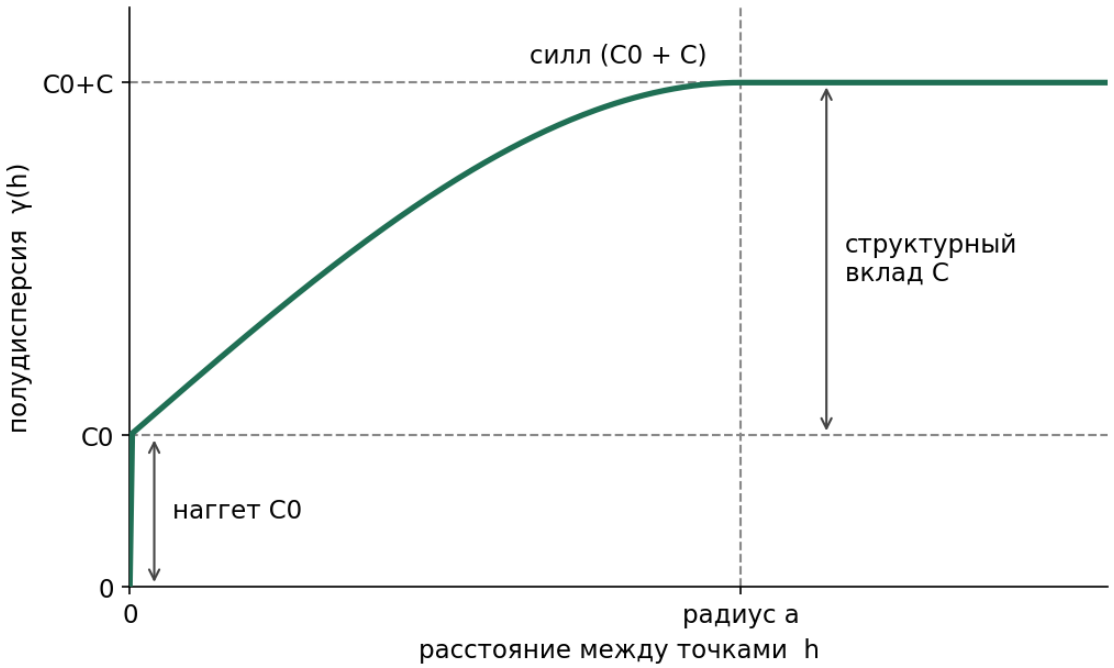



Figure 4: The variogram scheme: nugget C0, structural contribution C, sill (C0+C) and correlation range a.

**Nugget C0** The nugget is the value the variogram curve tends to as the distance tends to zero. In theory the discrepancy at zero distance should be zero, a point compared with itself. In practice a step remains, and it reflects the fact that data at very small distances still do not match. The causes are measurement and digitizing error, microvariability finer than the well spacing, and the discrepancy of duplicates at one point.

How the nugget affects the result:

C0 = 0 (default) - kriging is an exact interpolator, and the surface must pass exactly through every well. An isolated well with a Z outlier turns into a cone (a “bull’s eye”).

C0 > 0 - kriging stops reproducing the value at the measurement point exactly and becomes a smoother. Near a well the estimate is pulled toward the local mean. The larger the nugget fraction C0 / (C0 + C), the stronger the smoothing.

C0 = the whole sill (pure nugget) - the spatial link is lost, the surface degenerates into a plain mean. This is too much.

**Important - units.** Nugget C0 and sill are set in **absolute units of the data variance** (squared units of Z), not in 0-1 fractions. The default “1” for the sill is a placeholder that almost always needs changing. Set the total sill (C0 + the structure contributions C) **close to the data variance**. The level of smoothing is determined not by the absolute nugget value but by its **fraction of the sill** C0 / (C0 + C). The practical order is this. Take the total sill ≈ the variance, then the nugget = 0.2-0.4 of it - that is 0.2-0.4 × the variance, an absolute number, not 0.2-0.4 as such. The smaller the nugget, the more detail, but also more local peaks. The larger it is, the smoother the surface, but real structure may be smoothed away. The tool prints the data variance to the Log at start, and that is your reference for choosing the sill.

**Structures, range and anisotropy** The sill (plateau) is the level the variogram reaches. It is the sum of the nugget C0 and the structure contribution C. A structure is set by a model (spherical, exponential, Gaussian or power), a contribution C, a range a, an azimuth and an anisotropy.

**The meaning of the sill and its order of magnitude.** The sill is the upper limit of the differences

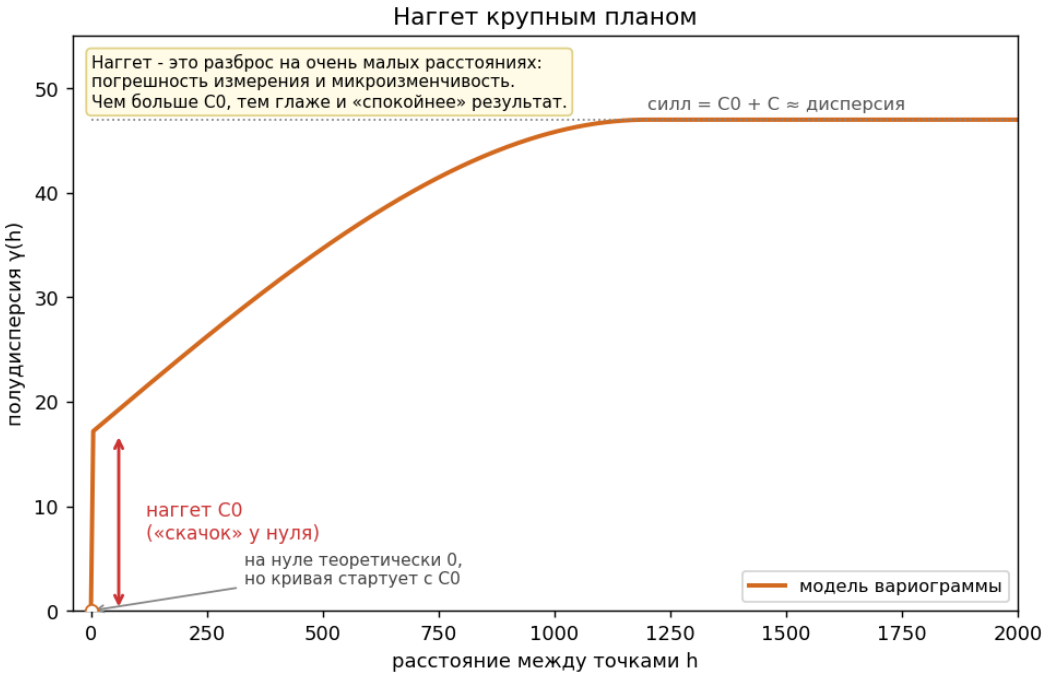



Figure 5: The nugget close up: the model starts at zero not from 0 but from a “jump” C0. This is the scatter at very short distances (measurement error, microvariability). The plateau C0+C ≈ the data variance.

between points, that is how much, on average, distant wells differ. It is practically equal to the ordinary data variance. Take an example for KCl. The mean ≈ 25 %, the variance ≈ 47.6 (%²), so σ ≈ 6.9 %, and the total sill is set ≈ 47.6. If the nugget C0 ≈ 17 (about 0.35 of the sill), then the structural contribution of the first structure C ≈ 47.6 - 17 ≈ 30. The absolute scale does not affect the grid itself, only the ratio of the nugget to the contribution matters for the estimates. But it is needed so that the standard-error map and the MSDR in cross-validation are at the real scale (total sill ≈ variance → MSDR ≈ 1). So do not leave the sill at the default 1, raise it to the data variance.

**Choosing a model.** The spherical and exponential models suit most tasks. The power model has no sill or range in the usual sense. It is used when variability grows with distance and does not reach a plateau (non-stationary increments), so the contribution and range fields for it are conditional. Use the Gaussian model with caution. At a zero or very small nugget it gives a numerically unstable system and artefacts, oscillations and negative weights. When the Gaussian model is chosen, the tool therefore enforces a small minimum nugget. Set one yourself where possible.

**Data type and mode.** Different data need different settings. Smooth structural surfaces (roof and floor elevations, thicknesses) are better modelled with a long range or a power model under a wide search radius (global, 0), and the surface then comes out smooth. A short radius with local search on such data gives “bull’s eyes” and discontinuities in the estimate when the set of neighbouring wells changes. For grades and chemistry (geomechanical properties, gas hazard) kriging works in its own right. Here a correct nugget matters, and for a strongly skewed distribution a data transform as well (see below on outliers).

The correlation range a is the distance at which the variogram reaches the plateau. Beyond it points practically do not influence each other. At 0 the automatic value max(extent)/3 is taken.

Anisotropy is set by the major-axis azimuth and the ratio of ranges (minor/major). A value of 1 is isotropic (the influence is the same in every direction). A value below 1 shortens the correlation across the major axis - useful for elongated geological structures.

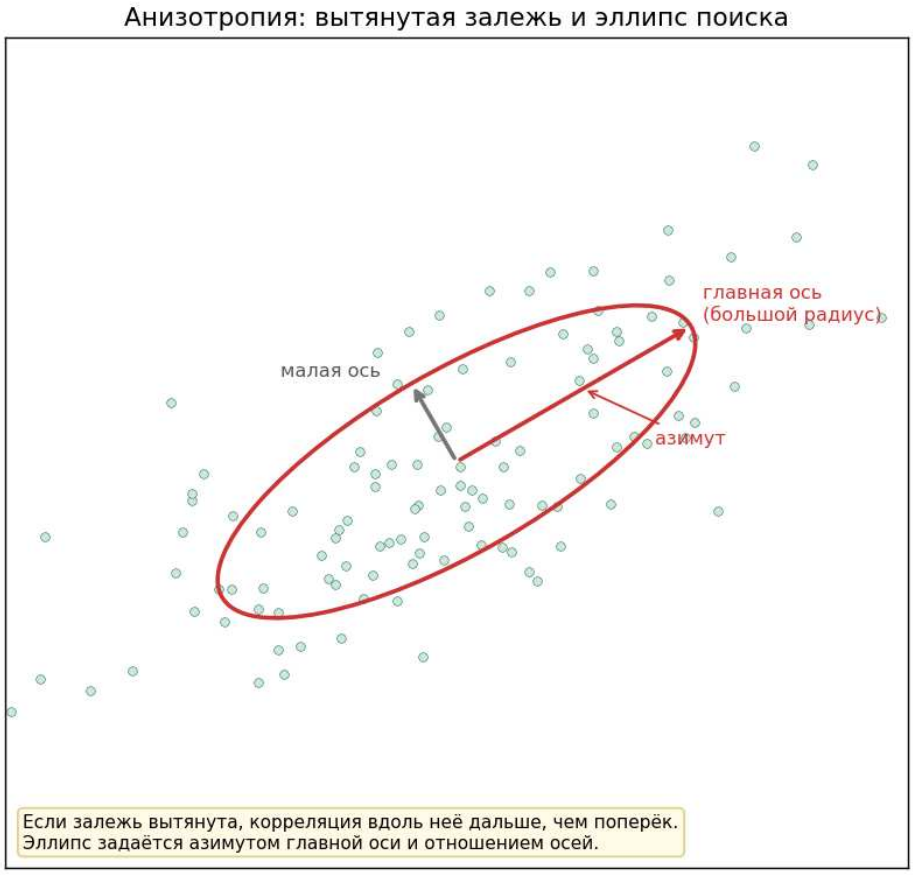



Figure 6: Anisotropy: if the body is elongated, correlation runs farther along it than across. The search ellipse is set by the major-axis azimuth and the axis ratio (minor/major).

| Parameter | What it sets | Default / advice |
|---|---|---|
| Mean for simple kriging | Used only with the SK type. | 0 |
| Nugget C0 | The “noise”/jump of the variogram at zero. Suppresses local peaks. In absolute variance units. | 0. For smoothing 0.2-0.4 of the sill |
| Structure i · model | Variogram shape: spherical, exponential, Gaussian, power. | spherical |
| Structure i · sill/contribution C | The structure’s contribution to the sill (abs. variance units). The sum C0+C ≈ the data variance. | str. 1 = 1 (replace with ≈ the variance) |
| Structure i · correlation range a | The distance to reach the plateau. 0 = auto = max(extent)/3. | 0 (auto) |
| Structure i · azimuth, ° | The direction of the anisotropy major axis. | 0 |
| Structure i · anisotropy (minor/major) | The ratio of ranges across/along the axis. 1 = isotropic. | 1 |

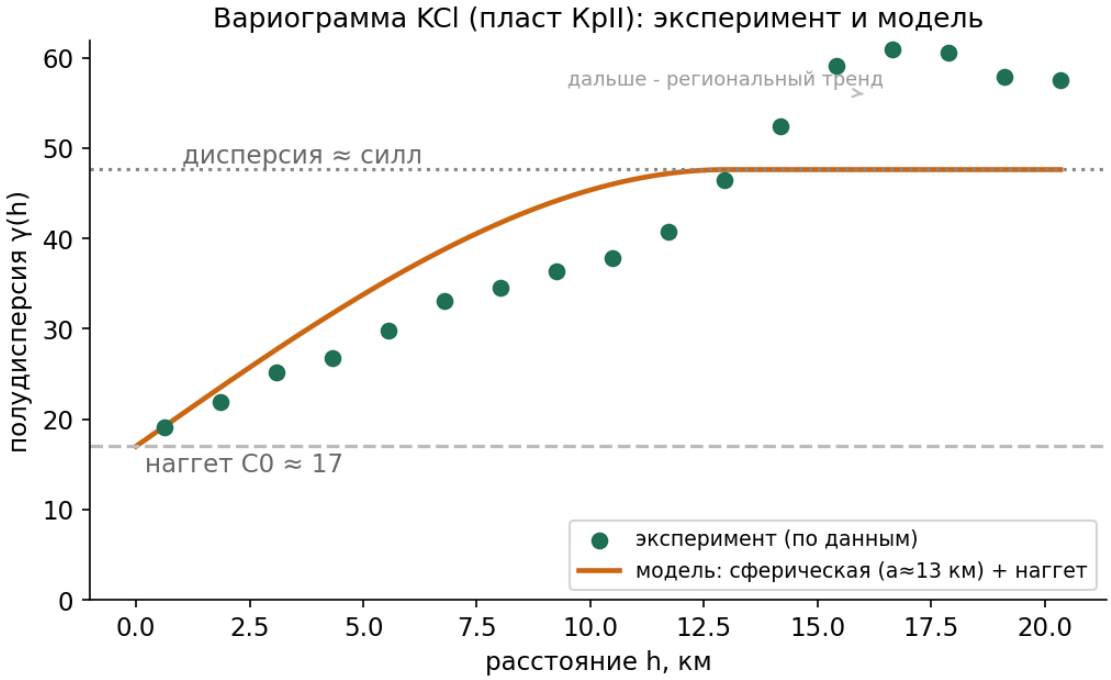



Figure 7: Experimental KCl variogram for the KrII seam and the fitted model: nugget C0≈17, the sill matched the data variance, range ≈13 km. Points beyond the sill are a regional trend.

This is how the scheme looks on real data. A variogram is built from the wells. For pairs of points the semivariance is computed and averaged over distances, and the result is a cloud (green points) under which a model (the curve) is fitted. From the model the kriging parameters are set. The height of the “jump” at zero is the nugget C0, the plateau is the sill, usually close to the data variance, and the distance to the plateau is the range a. If at large distances the points rise above the sill, as here, that is a regional trend (non-stationarity). It is either accounted for separately, or

the search radius is limited.

### Outlier removal

Outliers are anomalously high or erroneous values that distort the estimate. A few grade "bonanzas" can pull the whole grade map onto themselves, and clear errors such as a negative thickness spoil the surface. The **2D Kriging** tool lets you bound such samples right during the computation, without editing the source data. The parameters are in the **Advanced** section.

Removal and capping are a crude practical tool against clear errors. For grades and chemistry be careful. Extreme values are often not noise but signal (e.g. contamination spots), and blindly clipping the distribution tails is not worth it. Strongly skewed data are better transformed than clipped. A logarithm or a Box-Cox transform brings them close to normal. Another way is indicator kriging. Both lie beyond removal, but that is exactly how heavy tails are handled in the geostatistics of ores and contamination.

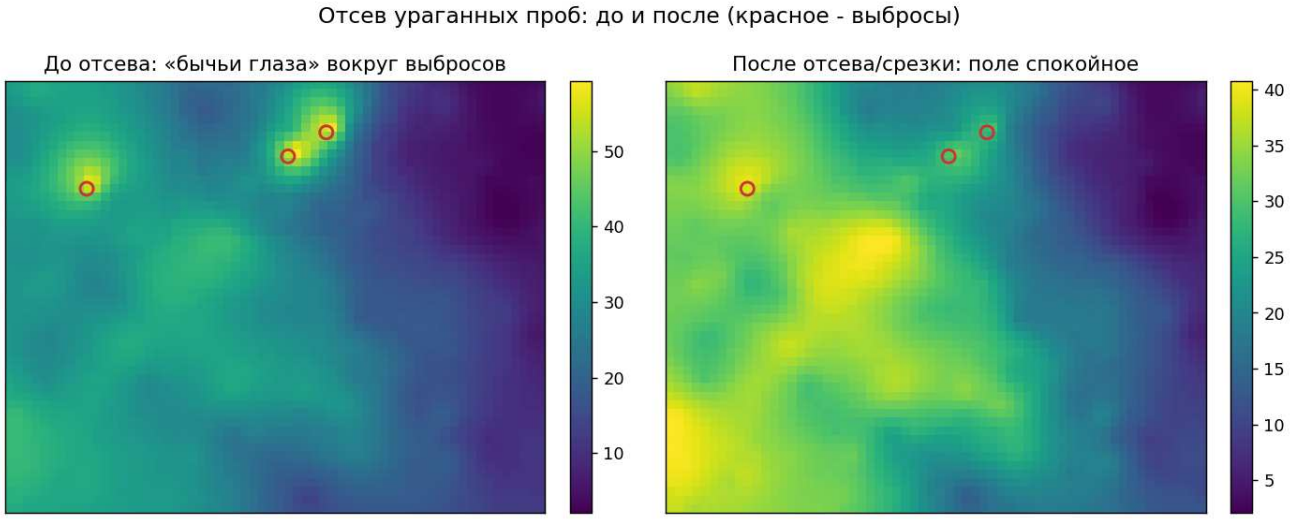



Figure 8: Outlier removal by example: on the left three outliers give "bull's eyes" (hot spots), on the right after capping to the upper bound the field is calm.

**Two modes.** **Remove** - samples outside the allowed range are discarded (for clearly broken records). **Cap (capping)** - values outside the range are clamped to the bound, while the point itself stays in the computation. Capping is the classic technique for grade outliers, the point's position is not lost, but its influence is limited. The mode is switched by the **Cap to bound (capping) instead of removing** checkbox.

**Absolute bounds.** The **Lower value bound** and **Upper value bound** set thresholds in Z units directly. An empty field means the bound is not set. They take priority over the percentile. For thickness, for instance, set the lower bound to 0, and negative values go away. Set the upper bound to, say, 30, and a clear outlier at 122 m goes away.

**Percentile bounds.** The p-th percentile is the value below which p% of all samples lie. For example, the 5th percentile is the threshold below which only the 5% smallest values lie. The 95th is the threshold above which the 5% largest lie. The **Outliers: clip percentile, %** parameter sets the number p. The bounds are taken symmetrically, from the p-th to the (100-p)-th percentile. So p = 2 treats as outliers the 2% lowest and the 2% highest samples. Everything below the 2nd and above the 98th percentile is either removed or capped. The larger p, the more aggressive the clipping. P = 0 disables the percentile mode. The convenience is that you need not know the absolute thresholds. They are computed from the data itself and suit any distribution and scale.

**Two-sidedness - important for chemistry.** The percentile mode cuts both tails, upper and lower. For grades this is dangerous. KCl = 0 in replacement zones is real geology, and clipping the lower

tail would wrongly raise the “empty” areas. So for grade clip only from above. Leave the **Lower value bound** empty and set the **Upper** as an absolute. The percentile works too, but keep in mind that it affects the bottom as well. For elevations and thicknesses two-sided clipping is usually appropriate.

**Order and Log.** The filter is applied before averaging coincident points. The tool’s Log reports how many samples were removed or capped and within which bounds - handy for checking.

### Kriging standard error

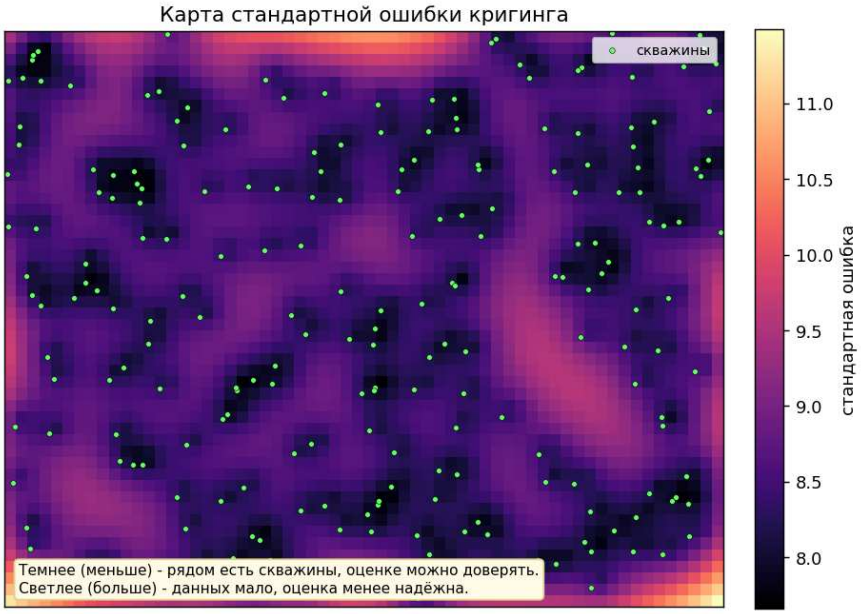



Figure 9: The standard-error map: dark near wells (green points) - the estimate is trustworthy, light in empty corners - few data.

Besides the estimate itself, kriging gives at every node the error variance - a measure of uncertainty. Its square root, the standard error, is output as an optional second raster (the **Kriging standard error** parameter of the **2D Kriging** tool). The units are the same as the interpolated quantity Z.

The key property is this. The standard error depends on the geometry of the well layout and the variogram model, but not on the Z values themselves. So it is a map of the observation network’s reliability, not of the data scatter. At a well point (with a zero nugget) the error equals zero - there the value is known exactly. As one moves away from wells it grows, and in areas without data it reaches a maximum (roughly the square root of the sill).

**How to read it.** Dark (small) values - the estimate is trustworthy, there are enough wells nearby. Light (large) - the estimate rests on distant points, effectively extrapolation. These are the first candidates for infill drilling. It is more convenient to compare relatively (where it is larger or smaller), because the absolute value depends on the variogram scale (the sill S1_SILL).

**Important.** This is a model estimate, and it is as valid as the variogram you set (nugget, range, anisotropy). At a nugget above zero the error at wells is not zero - the nugget sets a lower “floor” of uncertainty. The standard error is not a strict confidence interval, but as a relative uncertainty map it is very useful.

**Styling.** Give the layer graduated symbology by value (e.g. from dark to red) - and it is immediately clear where the map is reliable and where not.

### Trend removal (regression kriging)

Ordinary kriging estimates the mean locally, within the search window, so it follows a smoothly varying mean on its own. The difficulty appears when the field has a pronounced regional component, such as a general dip of the seam across the area. Then the experimental variogram of the raw value gets inflated. The range is overstated, there is no sill, the shape resembles a power model, and a stable model is hard to fit.

The **Remove polynomial trend** checkbox removes the regional component by least squares before kriging. The residuals are then kriged and the trend is added back to the estimate. The residual variogram returns to its normal shape, it reaches a sill with a nugget, and the range reflects the true scale of correlation rather than the span of the trend. The **Trend degree** field selects a plane or a quadratic surface.

The **Value transform** list adds a logarithm for quantities spanning orders of magnitude, such as hydraulic conductivity or transmissivity. With **ln** selected, the natural logarithm of the value is kriged and the estimate is returned through the exponential. This is the median, geometric estimate, correct for log-normal fields. The standard error is converted back to the original units. The logarithm removes the need to build an ln field by hand in the calculator and applies to positive values only. Set the variogram and nugget in ln units when the logarithm is on.

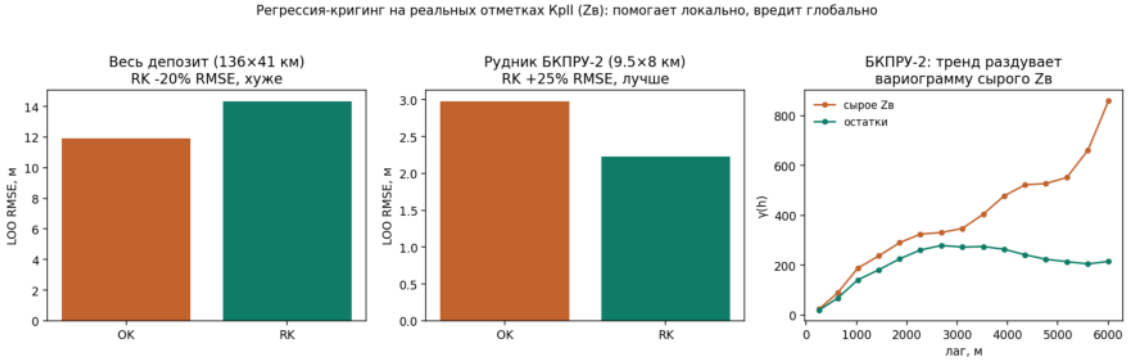



Figure 10: Trend removal on real data: a seam surface with a pronounced regional dip. The trend is removed by a polynomial, the residuals are kriged, and the trend is added back to the estimate.

Trend removal helps where the dip is uniform, within a single mine district or a local area. It is not meant for the whole deposit at once. Neighbouring blocks are offset in height, and a single polynomial describes them poorly. Local kriging already holds the varying mean, so removing a global trend there tends to hurt. The variogram shows whether it is warranted. If the raw value has a range comparable to the size of the area and no clear sill, a trend is present and worth removing. If the raw variogram already reaches a sill with a small nugget, there is nothing to remove.

Degree 1 is usually enough. Degree 2 captures curvature but can absorb part of the real structure into the trend, so after using it look at the residual variogram. If the sill and nugget become less defined, go back to degree 1. The same checkbox is in the **Variogram cross-validation** tool. There the trend is refit at each validation step, so the gain or loss from removing it is visible directly in the RMSE.

After trend removal, fit the variogram on the residuals. In this mode the standard-error raster is the kriging error of the residuals, the trend is treated as deterministic and adds no error of its own.

### Block kriging

Ordinary kriging estimates the value at a point, at the centre of a grid cell. In mining, however, what is needed is usually not a point value but the average over an area - a mining block, a panel or a reserve-estimation cell. The grade of a useful component in a block is the average over its area, and estimating it as the value at a single point is not quite correct. This is what the **Block kriging** checkbox is for.

When enabled, the mode estimates the average over the grid cell rather than the value at its centre. The cell is conceptually split into an N×N grid of points, the count set by the **Block discretization, N×N per cell** field, and the covariances in the kriging system are averaged over those points. The system is thus solved not for one point but for the whole block at once. This is the classical GSLIB block-kriging scheme.

The block estimate has two consequences, both useful for reserve estimation. The surface comes out smoother than the point one, because averaging over the block damps small fluctuations. And the kriging standard error comes out lower than the point one, because an area average is estimated more reliably than a value at a single point. There is one price. Block kriging does not reproduce the samples at nodes exactly. The average over a block, even one centred on a borehole, does not equal the value at the borehole itself, and that is as it should be.

A 4×4 discretization is almost always enough. A larger N takes longer to compute while accuracy grows only slightly. At N equal to one block kriging degenerates into point kriging, so the minimum value of the field is two, and the mode itself is off by default.

Block kriging combines with trend removal, the residuals are kriged over the block and the trend is added back to the estimate. It also combines with grid smoothing, but block averaging alone is usually enough and extra smoothing is not needed.

### Faults

A fault here is a line along which the surface breaks. It is supplied as a line layer in the **Faults** field, and the same layer is accepted by 1.02 and 1.04.

The fault acts as a barrier of influence. For every cell being estimated the tool draws a segment to each measurement. If the segment crosses the line, that measurement does not enter the sample of the cell. The cell sees only its own wing, the values on the two sides diverge, and the surface gets a break along the line.

The wing of a cell follows by itself from the side its centre lies on. There is no need to split the area into blocks and grid them separately, and the tool needs no list of wings.

**The line need not cut across the area.** For a dying fault the influence goes round its end. A measurement beyond the end of the line stays visible, and above the end the surface closes up. That is ordinary geology - the displacement along a fault plane falls to zero and the break dies out.

Right at the end the isolines crowd together into a narrow bundle. That is how a dying fault is drawn on a structural map. There is one caveat here. In the model the throw does not fall gradually but at once, because the barrier is binary. Up to the last vertex a measurement beyond the line is not visible at all, past it the measurement is fully visible. So the whole decay falls on a single cell. The bundle is shorter than it would be for a real fault, where the displacement dies out over a length comparable with the fault itself.

### Why the fault is never turned into cells

The barrier is tested by exact geometry - does the segment cross a link of the line. There is no raster mask of barrier cells, and that is deliberate.

The mask brought two troubles. A cell cannot carry a diagonal, so a slanted fault fell into the grid as a series of steps, and everything built over such a grid inherited that stepping. The other trouble was worse. The ray of visibility starts at the centre of the cell being estimated, and if that cell is itself a barrier cell, the very first test hits the mask. Such a cell saw not a single measurement in the whole area and went to nodata. A stepped slit was left in the grid along the fault.

With exact geometry there are no barrier cells at all. The grid comes out solid, there are no empty cells along the line, and the break shows as a jump of values between neighbouring cells on opposite sides.

A measurement standing exactly on the fault is visible from both wings. A borehole on the line belongs to both sides and there is no reason to discard it.

### The price of the approximation

There is one approximation in the method. Only the neighbours are selected by visibility. The covariances between the measurements themselves, which make up the kriging system, stay Euclidean, they are computed over the straight distance, as if there were no fault.

Zeroing them is not allowed, the matrix would lose positive definiteness, the system would become singular and the kriging would fall apart. This is exactly what the developers of well-known packages ran into, and some of them removed kriging with faults altogether, sending users to minimum curvature instead.

What that means in practice is this. Near the line the weights of the neighbours are computed without regard for the break lying between two of those neighbours. At distances of the order of the correlation range the difference is negligible. On a very dense network right at the line the contrast may be slightly lowered. The break itself does not go anywhere, it is held by the selection of neighbours rather than by the covariances.

### How to try it on teaching data

The tool **1.10 Example wells** can create a fault by itself. The **Fault throw** field sets the size of the shift. The generator draws a line across the area without bringing it to the edges and adds the throw to the value at every well on one side. Above the end of the line both sides are equal, so the teaching data contain both the break and its dying end.

The line is produced as a separate output **Fault (demo)**. Supply it to **Faults** here, then the same line to 1.04, and the whole chain comes together.

To test the barrier itself take a throw noticeably larger than the correlation range, the jump will be obvious. To test the corridor in 1.04 a throw of the order of the contour interval is enough.

The `throw` attribute in the output fault layer is for reference. The tools do not read it, their barrier is purely geometric and knows nothing of the size of the displacement.

## 1.02 Minimum curvature (points → raster)

The tool builds a grid by minimum curvature. The surface behaves like a thin elastic plate passing through the data with the least bending, that is a solution of the biharmonic equation. The method is not exact and the data are honored approximately, but the surface comes out as smooth as possible. That is why it is traditionally used for maps of geophysical fields and any smooth quantity. It is a deterministic alternative to kriging without variogram fitting. Kriging, unlike it, gives an estimate with a standard-error map.

**Tension** mixes in a membrane term. At 0 it is pure minimum curvature, at 1 a taut membrane with fewer overshoots between samples. Boundary tension is set separately and helps remove edge overshoots. The solution is iterative, by successive over-relaxation (SOR) with a nine-colour sweep of the grid. Nodes of one colour do not fall into each other’s stencil, so they are updated at once and stably. The grid is recomputed until the largest node change drops below the **residual threshold** or the iterations run out.

Free nodes start from the nearest data value, so convergence is fast on dense data. On very sparse data more iterations are needed, so raise their limit or the residual threshold.

| Parameter | Purpose | Default |
|---|---|---|
| Point layer | Samples with a value. | - |
| Value field Z (empty = from the geometry) | Numeric field to interpolate. | - |

| Parameter | Purpose | Default |
|---|---|---|
| Extent | Result rectangle. | from layer |
| Cell size | 0 = auto, min(extent)/50. | 0 |
| Tension | 0 = minimum curvature, 1 = membrane. | 0 |
| Residual threshold | 0 = auto, 0.01 percent of the data range. | 0 |
| Maximum iterations | Cap on the number of SOR passes. | 100000 |
| Boundary tension (Adv.) | Tension at the grid edge. | 0 |
| Relaxation factor (Adv.) | SOR acceleration, sensibly 1.5 - 1.9. | 1.85 |
| Anisotropy (Adv.) | Y/X axis ratio in the membrane term. | 1 |
| Faults | Lines along which the surface breaks. | optional |
| Grid (minimum curvature) | Output raster. | - |

The output is an ordinary grid ready for **1.04 Isolines from raster**. The log prints the grid size, the number of data nodes, the number of iterations and the final residual. If the iteration cap is reached while the residual is still above the threshold, the tool warns about it.

Next to the cell size field the size of the grid is shown - how many cells the given resolution produces. It is recomputed as you type. The method is iterative, and the number of cells has to be seen before the run rather than in the log after it. A cell twice as large means four times less work. Above four million cells the tool warns separately.

Clipping by a mask works as in kriging - a polygon from the project or the convex hull of the points with a buffer. Beyond the area of the data the method continues the surface anyway, and that field is better removed at once.

### Faults

Faults are supplied by the same line layer as in 1.01, but they work differently. In kriging the barrier decides which measurements a cell can see. Here there are no measurements, there is a grid of nodes tied together by a stencil, so the barrier lives on the links.

An edge between two neighbouring nodes crossed by the fault line drops out of the stencil. The link between the wings is cut, each wing is solved on its own, and the surface breaks along the line. The fault stays a line and is never turned into cells. A slanted line therefore does not fall into steps, and the wing of a node follows from the side it lies on.

A node that falls exactly on the line is assigned to one side. In appearance it is a detail, but it is a significant one. A fault drawn along an axis of a mine grid passes exactly through the node centres, and under a strict test it would block no edge at all. Assigning such a node to both sides is not an option either. It would be left without neighbours and would freeze at its starting value, leaving a seam of uncomputed nodes along the fault.

### The membrane band along the line

Minimum curvature has a feature that kriging does not. Its stencil reaches two cells out in each direction rather than one, so right at the line it would step over the break anyway, even with the neighbouring edge blocked.

Single points cannot be dropped from the stencil, it would stop approximating the biharmonic equation and the solution would lose its meaning. So within a band two nodes wide along the line the solution switches to a membrane over the unblocked neighbours. The membrane works over the

four nearest nodes and holds the break exactly. Away from the line the ordinary minimum curvature does the work.

There is one practical consequence. Right at the line the surface is slightly less smooth than away from it, though on a map that is not noticeable. The same device is used in **2.03 Topo2Raster** at cliffs, and for the same reason.

The line, as in kriging, need not cut across the area. Above the end of the fault the edges are not blocked and the surface closes up there.

## 1.03 MBA: multilevel B-splines (points → raster)

The third way to get a surface from points, after kriging (1.01) and minimum curvature (1.02). Read it together with them, and the number simply comes last because the tool appeared later.

The method of Lee, Wolberg and Shin, 1997. A coarse lattice of control points is taken and a cubic B-spline approximating the data is built over it. It approximates coarsely, so the residual is computed - the difference between the measurement and the current surface. Then the lattice is doubled and the residual is approximated anew. It goes level after level, and each next one picks up what the previous could not.

| Parameter | What it sets | Default / advice |
|---|---|---|
| Points with measurements / Value field | The input. | - |
| Extent | By the points by default. | by the points |
| Cell size | The detail of writing into the raster. 0 - five hundred cells along the longer side. | 0 |
| Initial lattice in X, in Y | The radius of influence. Different counts per axis give anisotropy. | 4 and 4 |
| Number of levels | Smoothness. One or two give a trend, eight and more sit the surface on the measurements. | 8 |
| Clip by the convex hull of the points | Beyond the cloud the surface must not be kept. | on |
| Lower and upper bounds of the result | The physically possible range of the quantity. Empty - no bound. | empty |
| Margin around the hull (adv.) | Widens the clip. | 0 |
| Stop by the residual (adv.) | Ends the refinement early. 0 - do not check. | 0 |
| Limit on the size of the raster, millions of cells (adv.) | Beyond it the tool refuses to work. | 50 |

### What it is good at

No system of equations is solved at all. A coefficient of the lattice is computed explicitly, as a weighted sum over the points that fall into the support of its spline. So the work is linear in the number of points while the memory depends only on the size of the lattice. Kriging solves a system over the neighbours in every cell, and on ten million measurements these are incomparable things, MBA takes a million points in seconds.

### Controlling the lattices

The initial lattice sets the radius of influence, and the coarser it is, the further a measurement spreads. Different numbers of cells per axis give anisotropy, which on a survey grid stretched along

the strike is exactly what is needed, and it is set directly, without a variogram model.

The number of levels controls the smoothness. Every level is twice as detailed and twice as close to the data, so one or two give a smooth trend while eight and more sit the surface on the measurements.

### Removing the trend

A coefficient of the lattice is computed by a formula linear in the value. So the error grows with the **magnitude** of the elevation rather than with its spread. Sometimes the support is one-sided - the edge of the area, or an empty place inside the cloud of points. There the denominator is small, the coefficient flies apart in proportion to the elevation itself, and the surface dives towards zero in the void.

The number of levels does not fix it, every next level inherits the same defect. Take a test terrain of about two hundred metres with an empty circle of radius 250 m. The error inside the circle reached 30 m, and at its centre 169 came out instead of 199.

So the tool removes a trend by a plane before the fit and returns it into the coefficients of the zero level. The error in the same void becomes less than a metre. There is no setting for this. Where the values are near zero, removing the trend changes nothing, and where they are far, it removes an error proportional to the magnitude itself.

It follows that **clipping by the convex hull and the bounds of the result do not repair a void inside the cloud**. The hull cuts off what lies outside, and the bounds turn a pit into a flat patch at the level of the bound. There is nothing to restore the shape from, an empty place needs data.

### What it does not give

Neither an error of the estimate, nor a model of covariance, nor weights that can be justified. This is an approximator rather than an estimator, and it does not know how good its answer is. Cross-validation of the method (1.09) and a map of the error do not apply to it. Hence the main use in exploration - **build a trend with it and refine the residuals by kriging**, for which there is kriging with an external trend.

The surface is smooth by construction, with a continuous derivative. For terrain and trends that is a plus, for grades a minus. Peaks get smoothed, and in reserve estimation that is exactly what geostatistics gets blamed for.

An exact hit into the measurements must not be expected even where the data are constant. On a constant value the deviation reaches a third of the range at one level, drops tenfold at three and disappears at eight. That is the nature of the method rather than an error of the computation. The edge coefficients underget their share, and every next level picks up the residual.

### Bounds of the result

An approximator does not know that a grade is never negative and a fraction is never above one. Between measurements a smooth surface overshoots the range - especially where markedly different values stand close together, and especially near the edge of the cloud. On lognormal data such as grades it easily goes fifteen hundred below zero.

The bounds cut the result to the physically possible. Cutting is more useful than the «as it came out» look, but it is also a sign of trouble. The log reports how many cells were cut from below and from above, what share of the area that makes and how far the surface went. If a tenth of the area or more is pressed against a bound, the surface there is not estimated but cut, there are no measurements nearby and the spline behaves arbitrarily. What helps is not a bound but a coarser lattice, fewer levels or a narrower extent.

Some quantities are never negative and vary by orders of magnitude - grades, permeability. For those it is better not to cut but to build the surface over the logarithm of the quantity and return

it through the exponent. Then zero is unreachable by construction and no plateau appears at the bound.

#### Clipping

Beyond the cloud of points the edge coefficients have no data and the surface behaves arbitrarily, and on a survey grid that shows as lifted corners of the raster. So the result is clipped by the convex hull of the points by default, with a margin around it set separately. The clip can be turned off, but then beyond the outline of the grid you get a result that must not be regarded as an estimate.

## 1.04 Isolines from raster

Builds isolines (lines) and, by default, contour polygons. Levels are set by a uniform step or by an explicit list. Parameters:

| Parameter | What it sets | Default / advice |
|---|---|---|
| Raster | The input raster (e.g. a kriging result). | - |
| Isoline step | A uniform step over Z. 0 = set **Explicit levels**. | - |
| Base level (offset) | Anchors the level grid (levels are multiples of the step from the offset). | 0 |
| Explicit levels | A space-separated list of levels. Takes priority over the step. The decimal separator is a comma or a dot. | - |
| Index isoline every N-th | Every N-th line is flagged is_index = 1 (for thickening). 0 = off. | 5 |
| Min. line length | Drop lines shorter than the threshold (map units). 0 = no filter. | - |
| Bicubic isoline smoothing | Densifies the grid (×2…×4) by bicubic interpolation before contouring - the main isoline-smoothing method, removes “octagons” from a coarse grid. Works for both lines and polygons. off = no densification. | off (×4 on a coarse grid) |
| Line rounding (Chaikin), iterations | An extra light line rounding (Chaikin). Weaker than bicubic smoothing; usually not needed if it is on. 0 = off. | 2 |
| Write the value into the geometry Z | The level is written into the Z of the line vertices. Needed for DXF export: otherwise AutoCAD and Credo place the contours at zero elevation. | off |
| Contour thinning, fraction of a cell (adv.) | Drops vertices that deviate from the straightened line by less than the given fraction of a cell. The shape is kept, the file gets much smaller. 0 = off. | 0.25 |

| Parameter | What it sets | Default / advice |
|---|---|---|
| Side of hachures and labels (adv.) | The downslope side for the depression-style hachures and the direction of the topographic labels, both at once. | automatic |
| Value field name | The name of the level attribute in the output lines. | ELEV |
| Band (adv.) | The band number of the input raster. | 1 |
| Isolines / Contour polygons | The output layers. Polygons are built by default into a temporary layer. | - |
| Belt solids (closed shells) | A third output: the same belt as a closed solid. Skipped by default, the layer is created only if a destination is set. | skipped |

Output fields:

| Layer | Field | Type | Holds |
|---|---|---|---|
| Isolines | ELEV | number | The level value of the line (name set by the **Value field name**). |
| Isolines | is_index | integer | 1 on index isolines (every Nth), otherwise 0 - for thickening. |
| Contour polygons | ELEV_MIN | number | Lower level of the band. |
| Contour polygons | ELEV_MAX | number | Upper level of the band. |
| Belt solids | ELEV_MIN, ELEV_MAX | number | Elevations of the bottom and top caps of the solid. |
| Belt solids | shell | integer | 1 on a solid - tells it from a flat belt without inspecting the geometry. |

**Isoline smoothing**

The main way to smooth isolines in this tool is **bicubic smoothing**. Before contouring, the grid is densified by bicubic interpolation (×2…×4), and the contours are built on the finer grid. On a coarse grid isolines otherwise look like “octagons”, because vertices are placed at cell edges. Densification removes this angularity topologically cleanly. Nodata boundaries and internal data “windows” are preserved. Densification affects both lines and contour polygons, and the band boundaries still coincide with the isolines. The cost is more cells (×4 = 16 times more), so on a very large grid start with ×2.

In addition there is a light line rounding by the **Chaikin** algorithm (number of iterations). It is weaker than the bicubic one and usually not needed if densification is on. It makes sense as a fast alternative on a coarse grid when you do not want to densify.

Smoothing of the field itself (Gaussian, over the raster) is a separate operation, done not here but in the **2D Kriging** tool. There it goes over the grid before contouring and removes not angularity

but field bumpiness, the “bull’s eyes” around wells. Bicubic smoothing and Gaussian field smoothing complement each other - the first removes grid angularity, the second data bumpiness. The contoured kriging raster is not changed in the process, only a temporary copy is smoothed.

### Contour polygons (bands)

Contour polygons are filled bands between neighbouring isolines. They are built not by classifying raster “steps” but by polygonizing the smoothed isolines themselves together with the outline of the raster’s valid area. Line ends are snapped to the outline, the network is noded and polygonized.

A band takes its level range from the elevations of the isolines that bound it. A band lies between two neighbouring isolines, their elevations are its bottom and top, and they are read exactly, with no regard to the cell size. Raster sampling remains the fallback for bands that have the area outline rather than an isoline on one side. How many bands were decided each way goes to the log.

Why not by sampling. An isoline is drawn by interpolation between cell centres, while sampling returns the value of the cell itself. These are two different surfaces. While a band is wider than a cell the difference goes unnoticed. Once a band lies entirely inside one cell, that cell’s value belongs to the neighbouring range and the band gets the wrong one. On the map that is a shard of an alien colour among its neighbours. At a fine interval such a band is the rule rather than the exception: a one-metre interval over a hundred-metre range on a fifty-cell grid makes almost every band thinner than a cell.

A face that has no isoline on any side and is thinner than a cell is not a band. It is discarded and counted in the log. Such a face is closed by the overshoot tails at the area outline rather than by isolines.

Thanks to this the polygon boundaries coincide with the isolines, and the coverage is continuous (no holes). The polygons carry the ELEV_MIN and ELEV_MAX fields and go into a temporary layer by default. To skip building them, clear the **Contour polygons** field.

### Straight edges on the belt drawing

Not every line of the output is a contour. The belt polygons are assembled from three sources at once - the contours themselves, the outline of the area with data, and the fault lines. The first are smooth, the other two are straight.

The outline of the area follows the edges of the raster cells and therefore gives a stepped or strictly straight border. It appears at the edge of the area and around patches without data, where a node did not gather the required number of neighbours.

A fault line cuts the belts along itself, and at a dying end the break does not vanish at once. On one side of the last segment the barrier still acts, on the other it no longer does, and the sets of neighbours of adjacent cells diverge. Hence both a crowding of contours and a small belt with a straight edge along the line are normal near the end of a fault.

This is a property of the model rather than a failure of the computation. If such a broken end gets in the way, extend the fault line beyond the area. More often than not a fault is drawn shorter on the map than it really is, and extending it is better than smoothing the break.

### Belt solids

A belt on its own is a surface, it has an outline but neither a bottom nor a volume. The third output gives the same belt as a closed solid, with a cap below at ELEV_MIN, a cap above at ELEV_MAX and side walls along every ring, inner ones included. Such a solid computes a volume, cuts by a plane with a closed section and travels to programs that only understand closed shells.

Closure is checked by a single rule - every edge of the shell belongs to exactly two faces. An inner ring gets its wall on a par with the outer one, otherwise the shell does not close around the hole. Solids that fail the check are counted and reported in the log.

The output is skipped by default. The field holds the skip-output entry and no layer is created until a destination or a temporary layer is set. Solids are noticeably heavier than flat belts, so turn them on for a task rather than always.

### Layer styling

For lines set rule-based symbology on is_index and give the index isolines (is_index = 1) a larger width. Label by the level field (ELEV).

Polygons are created with a single symbol. For range fills set graduated symbology by ELEV_MIN (or ELEV_MAX).

The isoline layer is automatically placed above the polygon layer so the lines show over the fill.

### The value in the geometry Z

The **Write the value into the geometry Z** tick lifts the vertices of every line to its own elevation. A contour is a line of equal level, so the Z of all its vertices is the same and equals the field value.

The Z elevation checkbox works on the belt polygons as well. A belt polygon is laid flat, at the upper bound of its band - the same elevation as the top cap of its solid. The stepped surface comes from the stack of belts, and belts and solids sit on each other without a gap.

The ring of a belt has to be planar. A non-planar ring is triangulated by the 3D scene in plan. Long thin triangles then stretch across the whole belt between vertices of different heights, which on screen looks like ribbons running away in height. The height is carried by the walls of the solid, not by a tilted patch.

This is needed to hand the contours over to AutoCAD and Credo. The standard QGIS export (**Project - Import/Export - Export project to DXF**) keeps the Z of the vertices, you only need to clear the **Force 2D** tick. Without Z in the geometry both programs place every contour at zero elevation, and the height has to be set by hand line by line.

The multipart structure of the layer is preserved. A contour of one level stays a single feature with all its branches, the feature count does not change and the labels do not multiply.

### Contour thinning

The contourer puts a vertex at every crossing of a level with a cell edge. On a detailed grid a line therefore carries more vertices than its shape needs. The **Contour thinning** parameter drops vertices that deviate from the straightened line by less than the given fraction of a cell. With a fraction of 0.25 and a five-metre cell everything deviating by less than 1.25 m is dropped.

The shape is kept while the layer gets several times smaller - which matters for the drawing, for the 3D scene and for the handover to AutoCAD. Thinning runs before polygonisation, so belt borders still coincide with the contours after it. Zero turns it off.

Where the contours run closer than the tolerance, thinning and rounding are checked against the grid. A line of a level stays between the midpoints to the neighbouring levels, and a straightening or a corner cut that would take it past that bound is not made. So on the steep side of a subsidence trough, or on a slope with a small contour interval, neighbouring contours do not cross each other, while on gentle ground the tolerance works in full. The log reports how many straightenings were cancelled and how many corners were kept.

### Topographic labels

On a topographic map the top of the figure on a contour always faces up the slope. Reading the map, a single label tells you which way is higher without checking the neighbouring contours.

QGIS measures the top of a label from the direction of the line, so text rotation cannot achieve this. What is needed is to give the lines a single direction relative to the slope, and that is what the

**Topographic labels** tick does. The layer keeps an **up_side** field, where 1 means the line was left as it was and 0 that it was reversed.

The automatic choice of side is sometimes wrong, and then it is switched by hand with the **Side of hachures and labels** parameter. The switch acts on both halves of the picture at once, the side of the hachures and the direction of the labels. It is applied exactly once. Reversing a line already swaps left and right, and with them the sign of the downslope side, so a second flip would return the hachures to where they were.

**It does not work unless upside-down labels are allowed** The labelling settings hold an option for showing upside-down labels, and by default it forbids them. Under that ban QGIS turns the text around by itself so that it reads left to right, and the direction of the line stops mattering.

A topographic label is upside down by definition on a slope facing south. So the showing of upside-down labels has to be allowed rather than fought.

In the **Structure / hypsometry** and **Depression (hachures down)** styles this is already set. If you label with a style of your own, enable the showing of upside-down labels in the rendering section, otherwise the tick gives nothing.

**The label sits on the line** In both styles the label is placed on the contour itself rather than above it, and breaks it. This is the familiar topographic device - the figure reads together with the line rather than beside it.

### Warning about flat levels

The tool checks by itself whether any level of the interval has landed on an area with a near-zero slope. If it has, the log gets the level, the number of cells touched and the share of the data. Nothing is blocked, the isolines are built as usual.

Why this is needed. On an area where the surface changes by millimetres, the position of an isoline is set not by the relief but by the noise of the matrix. The line loses its stability, breaks into a multitude of small rings and looks like a single thickened one. The classic source of such an area is a water surface. On a matrix derived from a stereo pair it stands at one elevation to within centimetres and can cover half the area.

Telling this defect from the normal work of the algorithm by eye is hard, because the algorithm did its job right and the data are at fault. That is why the check is reported in the log as numbers.

The practical remedy is usually simple - mask the water surface before building, or shift the levels. A shoreline is a map feature of its own, not a contour, and should not be drawn as one.

The check runs in a single pass over the array and is skipped on rasters larger than sixty million cells so as not to waste time. Any error inside it is suppressed, the build is never brought down by diagnostics.

### Contour confidence

A continuation of the same thought, but this time with something you can do about it. The **Contour confidence** parameter has three positions.

**Do not compute** is the default, nothing changes.

**drop_min and drop_mean fields only** leaves the lines whole but gives each one two fields, the smallest and the mean elevation drop per cell along it. The decision stays with you. Suspect stretches show up with an expression such as `drop_min < 0.005`, and you decide whether to hide or to show them.

**Fields plus a break on suspect stretches** additionally breaks the line where the drop falls below the threshold and marks the parts with a **lowconf** field. Nothing is deleted, what is marked can be hidden with a layer filter.

**Why the threshold is a fraction of the interval** What is measured is not the slope but the elevation drop per cell. This quantity has the same dimension as the contour interval, so it can be compared with the interval directly. A single threshold then works the same way on different data.

An example. At an interval of 0.5 m and a threshold of one hundredth the boundary runs at 5 mm per cell. A gentle slope gives centimetres per cell and passes, a water surface gives millimetres and does not. Setting the threshold as an absolute slope would mean tuning it anew for every area.

The threshold is set in the advanced parameters, **Drop-per-cell threshold, fraction of the interval**, 0.01 by default.

**Why only runs are broken** A single suspect vertex does not break the line, and neither do two. A break happens only where three or more weak vertices follow one another. Otherwise one random noisy cell would break up a contour on a perfectly normal slope, and the map would thin out for no reason.

The number three is fixed and is not offered as a parameter, there is nothing to tune it against.

The parts share the boundary vertex, so no gap appears in the geometry, neighbouring pieces meet point to point.

**Order in the pipeline and the summary** The marking runs after the isolines are built and before the short-line filter. Otherwise the fragments left by the breaking would enter the length statistics and some lines would be dropped twice for different reasons.

A summary goes to the log. It gives how many lines came in, how many parts came out, how many of them are below the noise, and what share of the total length turned out weak.

**What the tool does not do** It does not smooth flat areas for the sake of an even line. That would be a forgery of the data. The result would be a smooth and wrong contour instead of a broken one that matches what the matrix actually holds. The decision whether to show a weak stretch stays with a human.

### Faults

The same line layer that was supplied to 1.01 or 1.02 goes here into the **Faults** field. Here it does two things.

First, the isolines are cut along the line, exactly along it and not along cell edges. Second, the line itself enters the network from which the contour polygons are built, so the boundary of a belt runs exactly along the fault.

The break is entirely vectorial. The grid stays solid, there are no gaps in it, and everything computed off the grid works as usual.

### The corridor at a fault

One parameter needs an explanation, because without it the result looks odd.

The grid is solid, while the values on the two sides of a fault differ by the whole throw, and the jump falls on a pair of neighbouring cells. The contourer faithfully draws every intermediate level at once in that gap. With a throw of twenty metres and an interval of two, that is a dozen isolines within the width of a single cell. They carry no geological meaning, this is interpolation across a break the bed does not know about.

**Corridor width at a fault** is given in cells, one by default. A strip of that width is cut out of the isolines, and the ends of the freed lines are snapped to the fault itself. What is left on the map is a clean break along the line.

Zero switches the cutting off. There is no point taking less than a cell, the jump occupies exactly one cell. Noticeably more and isolines that run along the fault for a reason will start to disappear.

At the dying end a short stretch of crowded isolines remains, about the width of the corridor. That is right in meaning. The break has come to nothing there, the surface closes up and there is nothing to tear.

#### Smallest polygon thickness

The threshold is given in cells and filters out narrow strips of belts. The thickness is taken as twice the area over the perimeter. For a long narrow strip that equals its width at any length, while by area such a strip cannot be told from a normal belt.

The threshold is meant against fragments at a break where fault lines have not been supplied - at open pits, cliffs and the edge of the area. **It must be used with care.** On a steep surface with a fine interval a normal belt between neighbouring levels is itself narrower than a cell, and a threshold of one cell will then clear almost everything off the map. The tool warns in the log if it has filtered out more than half of the belts. By default the threshold is zero, that is, there is no filtering.

## 1.05 Declustering (weights)

The tool prepares data before interpolation. When samples are clustered unevenly, some blocks drilled denser than others, the naive global statistics shift toward the over-sampled areas. If rich zones were drilled denser, the mean and histogram are overstated, and that directly affects reserve calculation. Cell declustering (a port of GSLIB **declus**) gives each sample a weight inversely proportional to the local density, less in a cluster, more in sparsely sampled areas. A representative declustered mean is computed from the weighted data.

A grid of cells is laid over the area, a sample weight is proportional to one divided by the number of samples in its cell, and then the weights are normalized. The cell size is chosen automatically by a sweep over sizes, which picks the minimum declustered mean when clusters fall in rich zones and otherwise the maximum. The size can also be set manually. On a regular grid declustering changes nothing, all weights are equal.

| Parameter | Purpose | Default |
|---|---|---|
| Points with values | Samples. | - |
| Value field Z (empty = from the geometry) | Numeric field. | - |
| Cell size | Auto (sweep) or manual. | Auto |
| Cell size for manual mode | Cell side in manual mode. | 0 |
| Sweep objective (Adv.) | Minimum or maximum mean. | Minimum |
| Number of sizes in the sweep (Adv.) | How many cells to try. | 24 |
| Cell Y/X ratio (Adv.) | Cell anisotropy. | 1 |
| Grid-origin offsets (Adv.) | Averaging over grid shifts. | 4 |
| Points with weights | Point layer with a **wt** field. | - |
| HTML report | Summary, histogram, mean curve. | default |

The outputs are a point layer with a **wt** field and an HTML report. The report gives a summary (naive against declustered mean), a raw-against-weighted histogram and a mean-against-cell-size curve. The declustered mean from the log and the report goes into **1.01**, into the **Mean of simple kriging** field. The **wt** field feeds **3.06 Gaussian simulation** for a weighted normal-score transform. Outlier samples are cut separately, by percentile capping in the kriging and cross-validation tools themselves.

## 1.06 Variogram (experimental)

The tool builds an experimental semivariogram from points, fits a model to it if needed, and produces an HTML report with a chart. It does not compute a grid and is not part of the kriging computation

chain directly. Its job is diagnostic. It shows the structure of the data's spatial variability and helps set the variogram parameters deliberately, by the look of the cloud rather than by eye.

### Why the preview is needed

Kriging relies on a variogram model, on the nugget, the sill and the range. The interpolation weights and the standard-error map depend on them. The fitting of these numbers is easily handed to automation and forgotten about. On a clustered drilling grid this is dangerous. Clusters of close wells give a huge number of pairs at short distances and press down the near part of the variogram. An auto-fit on such a cloud easily yields a confidently wrong nugget. The preview removes this problem. The geologist sees the pair cloud itself, understands where the data are dense and where sparse, and fits the model knowing what lies beneath it.

That is why model fitting in the tool is given as a recommendation, not a finished result. The numbers it suggests should be checked against the look of the chart and only then carried into kriging.

### A short theory

The semivariogram describes how statistically related the parameter values are in two points depending on the distance between them. For a pair of points separated by a distance h, half the squared difference of their values is taken (the semivariance of the increment). These quantities are averaged over distance intervals (lags), giving the curve γ(h). It is a measure not of the "average difference" of values but of the statistical reliability of predicting a value from a neighbour. The smaller γ, the closer the link.

A typical curve has three characteristics. The nugget C0 is the value γ tends to as the distance tends to zero. It reflects variability at a scale finer than the network step, plus measurement error. The sill is the level the curve reaches at large distances. The full sill equals the sum of the nugget and the structural contributions and is ideally close to the data variance. The range (a) is the distance at which the curve reaches the sill - i.e. at which the spatial correlation drops practically to zero. Beyond it points are statistically unrelated. For the exponential and Gaussian models the sill is reached asymptotically, so for them the range is effective.

The nugget and contributions in the tool are set in absolute units of the parameter variance, not as fractions of one. The reference for the full sill is the data variance, which is shown in the report summary.

### Parameters

| Parameter | What it sets | Default / advice |
|---|---|---|
| Points with values | A point layer of wells or samples. | - |
| Z value field (empty: from the geometry) | The numeric attribute to analyse: roof elevation, thickness, grade. Empty - from the point Z. | remembered between runs |
| Grouping field (adv.) | Builds a separate curve for each field value (e.g. survey type) and overlays them on one chart. | off |
| Declustering weight field (adv.) | Weights from **1.05 Declustering** for a weighted estimate. | not set |
| Number of lags | Into how many distance intervals the pair cloud is split. | 15 |

| Parameter | What it sets | Default / advice |
|---|---|---|
| Maximum distance | The far edge of the variogram, in layer units (for metric coordinates - metres). 0 = half the extent diagonal. | 0 |
| Fit model (recommended) | Auto-fit of the nugget, sill, range and model type; the result is remembered for substitution into **2D Kriging**. | on |
| Model to fit (adv.) | Fix the model type or leave the auto-pick of the best by $R^2$. | Auto |
| Minimum points per group, % (adv.) | Groups smaller than the threshold are not built and are listed in the Log. The floor is 30 points. | 2 |
| Robust estimator (Cressie-Hawkins) (adv.) | An estimate less sensitive to rare anomalous pairs. | off |
| Show pair cloud (adv.) | Adds the source pairs (before averaging) to the chart. | off |
| Overlay a given variogram model (adv.) | Draws a model with a manually set nugget, sill and range over the cloud: handy to compare your model with the data. | off |
| Outliers (adv.) | Clip percentile, lower and upper value bounds, capping to the bound instead of removal. At the very end of the list. | off |
| Profile name for the table | Name of the set under which the fitted model and the current outlier settings are entered into the table of models. | empty |
| Append to the table of models | An existing table for the result. The rows of this profile are replaced. | not set |
| Variogram model (new table) | A new table. Created when there is nothing to append to yet. | created |
| Variogram table | Output: lag, γ value, number of pairs. | temporary layer |
| Report (HTML) | Output: the pair cloud, the fitted curve and the data-variance line. | temporary file |
| Outliers (output) | The removed points with coordinates, value and the reason for removal. | off |

Why the removed points are worth a look is explained in 1.01.

The parameters marked “adv.” are in the collapsed **Advanced Parameters** section.

The output is a **Report (HTML)** with the chart, the fitted curve and the data-variance line. Optionally there is also a **Variogram table** - the experimental variogram points as a geometry-less layer, one row per lag of each series. From it you can build your own chart in QGIS or export the values. Its columns:

| Field | Type | What it holds |
|---|---|---|
| **series** | string | The series: “all points” or the grouping-field value, if set. |
| **lag** | double | The mean distance between points in the interval (lag), in layer units. |
| **gamma** | double | The semivariance γ(h): the mean of half the squared value differences over the pairs of this lag (or the robust Cressie-Hawkins estimate, if enabled). |
| **npairs** | integer | The number of point pairs that fell into the lag. A small number of pairs means the variogram point is unreliable. |

### The grouping field and a mixed-density survey

The optional **Grouping field** builds a separate variogram for each field value and overlays them on one chart. This is needed when the sample is collected by networks of different nature and density, for example surface and underground exploration. By feeding the survey type into the grouping, you can see whether these populations share a structure or each has its own.

Mixing mixed-density networks does not create artefacts by itself, but it distorts the overall variogram. A dense network gives many pairs at short lags and shapes the near part of the curve, a sparse network works on the far lags. A single model stretched over such a cloud turns out to be a mixture of two structures and describes neither correctly. The grouping shows this mixture, and the decision whether it is legitimate to combine the populations stays with the geologist. Declustering is not applied to the pair cloud here. Its weights are meant to correct the histogram and the mean, not the variogram pairs, where each pair is equally valid regardless of grid density.

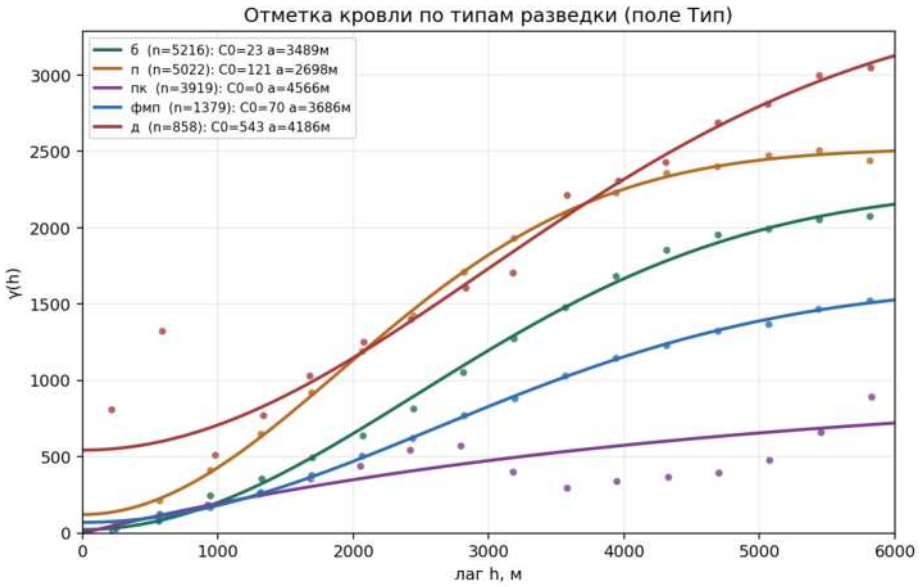



Figure 11: Roof elevation for the KrII seam, grouped by survey type: the underground network lies noticeably lower (a more homogeneous area), the detailed survey gives a high nugget. The different populations are visible at once.

### Three typical geological situations

Seam elevations, thicknesses and component grades have different geostatistical characteristics, and it is useful to see them side by side. The illustration shows variograms of three parameters of one industrial seam, computed in a single distance window.

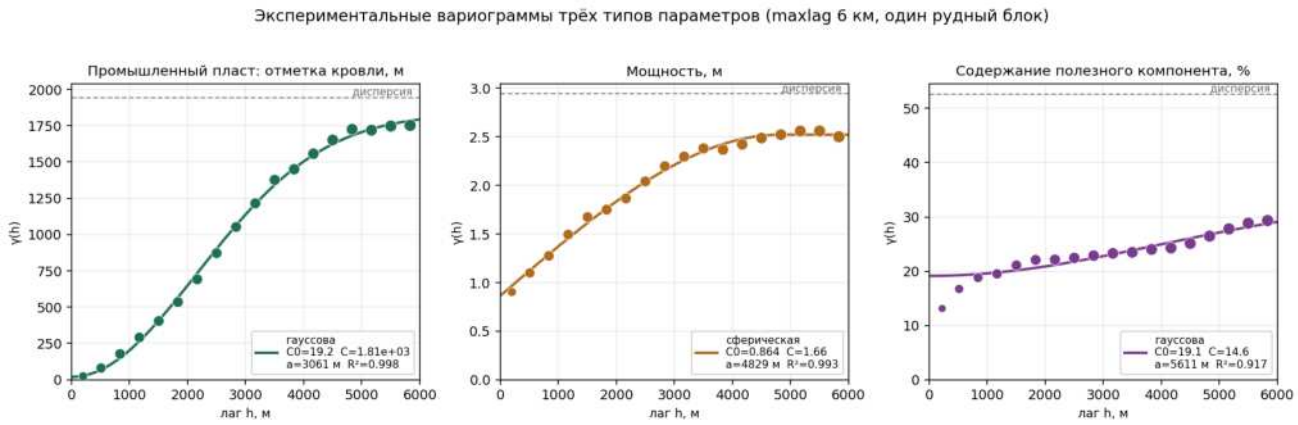



Figure 12: Three parameter types of one seam in a single window: roof elevation (almost zero nugget, smooth surface), thickness (nugget about a third, spherical) and grade (nugget comparable to the contribution, a noisy parameter).

Roof elevation is a smooth surface. The nugget is almost zero, the range large, the model close to Gaussian, the fit quality very high. Neighbouring wells give almost the same elevation, the variability is large-scale. Kriging works confidently. There is a subtlety here. A Gaussian model with an almost zero nugget is numerically unstable and gives the characteristic “bull’s eyes” on the map, so a small nugget should be set by hand.

Thickness is an intermediate case. The nugget makes up a noticeable fraction of the sill, the range is medium, the model more often spherical. About half the variability is structural, half small-scale. This is a typical working variogram.

The component grade is the noisiest parameter. The nugget is comparable to the structural contribution or exceeds it, the curve rises slowly, the fit quality is lower, and the model is poorly distinguishable from neighbouring types. The main variability sits at a scale finer than the sampling grid. Kriging smooths such a parameter heavily, and cross-validation shows a large error. Grade is predictably worse than elevations and thicknesses, and that is normal.

**Maximum distance and reaching the plateau**

The most common mistake is too large a maximum distance. If you leave the automatic value at half the diagonal, on an elongated deposit the window stretches over tens of kilometres. The lags begin to link points across barren gaps and inter-block breaks, so the variogram describes the regional trend instead of the local structure. The fit then yields a range larger than the window itself and a sill several times the variance. The sign of trouble is simple. The fitted model’s range is comparable to the window or exceeds it, which means the curve has not reached a plateau and the sill is obtained by extrapolation.

The cure is to reduce the maximum distance to the local scale and to check that the variogram has reached the plateau. Take a grade example for one seam. With a 6-kilometre window the fit gave a range of about 9 kilometres and a sill below the variance, so the curve had not yet reached the plateau. With a 12-kilometre window it did, giving a range of about 18 kilometres and a full sill close to the data variance. The real correlation range turned out larger than it looked in the narrow window, and the right answer came precisely from checking that the curve reaches the plateau.

At the same time the window must not step over large barren zones. On a drilling grid they are visible by the drop in point density. Build the variogram within a single ore block, otherwise the local geology mixes with regional tectonics.

**Where the nugget came from**

On a sparse grid the nugget is almost always set by a handful of pairs of points rather than by a cloud. Two wells a few tens of metres apart with incomparable values lift the first lag above the sill. All that is left for the fit is to describe this with an almost pure nugget, and kriging with such a model returns the mean instead of a map.

The tool therefore prints a breakdown of the first lag to the log. It gives the distance, the number of pairs in it and the value of gamma next to the data variance. If the gamma of the first lag already exceeds the overall variance, a separate warning says so, because only a nugget can describe that.

The heaviest pairs inside the first lag are then named - the distance between the points, both values, the gamma contribution and the coordinates of both points. The coordinates find the pair on the map at once, and from there it is a question about the data. A mixed-up horizon, a sign, the units - or genuine micro-variability, in which case the nugget is real.

**When the fit is good for nothing**

Two failures of the fit are silent. The parameters are printed, kriging runs without a single error and produces a flat field around the mean. The tool pulls both out into warnings.

The first is a fit quality R2 below 0.1. The model explains next to nothing, and its parameters cannot be carried into kriging.

The second is a nugget above half of the total sill. Correlation at short distances is not resolved, and there are two reasons for it. Either the grid is sparser than the structure someone is trying to see, or the data contain those very pairs. Kriging with such a model smooths the estimate towards the mean and produces bull’s eyes on the map, tight concentric rings around individual samples.

**The workflow with cross-validation**

The variogram gives a starting model, and **Variogram cross-validation** checks it. The order is as follows. First an experimental variogram is built with a maximum distance at which the curve reaches a plateau, and the fitted nugget, contribution, range and model are taken from it. Then these numbers are carried into cross-validation and the leave-one-out metrics are assessed. The fitted and validated model is entered into the table of variogram models. From there it is substituted into **2D Kriging** through the **Variogram models** and **Profile from the table** fields - see the section «The table of variogram models».

The mean error ME should be near zero, meaning there is no systematic error. The root-mean-square error RMSE shows the absolute accuracy. The MSDR deserves separate attention - the ratio of the squared error to the kriging variance. If it is noticeably above one, kriging underestimates the uncertainty and the standard-error map is understated.

Correcting the MSDR is done exactly, not by eye. In ordinary kriging, multiplying the whole variogram by a constant factor does not change the estimate. The weights depend only on the shape of the curve, not on its scale, and only the kriging variance changes. So it is enough to multiply the nugget and contributions by the current MSDR value, leaving the range and model unchanged, and repeat cross-validation. The ME, MAE, RMSE and R metrics do not shift, the MSDR comes to one, and the error map becomes trustworthy.

After scaling, the full sill may turn out above the data variance, and on a clustered grid that is not an error. The naive variance is understated because dense well clusters pull it down, while the true scatter over the area is larger. The excess of the sill over the variance here is a consequence of the uneven grid.

The finished and validated model then only needs to be carried into **2D Kriging** to compute the grid, and after that, if needed, into **Isolines from raster**.

If the data are clustered unevenly, set the optional **wt** weight field from tool **1.05 Declustering**. Each pair of points is then taken with a weight equal to the product of its endpoints’ weights, and

clusters do not inflate the near lags. The pair count in the report shows the raw number of pairs, while γ itself is computed with weights.

## The table of variogram models

A fitted variogram is the outcome of data analysis rather than a setting of the program. It is therefore kept as a table layer. It is saved with the project, goes into version control and moves between workstations, and it is edited in the QGIS attribute table on a par with the rest of the data.

### Structure of the table

One row describes one structure of a variogram. This representation does not limit the number of nested structures - a second structure is added as a row with the same profile name and `struct = 2`.

| Field | Type | Contents |
|---|---|---|
| `profile` | string | Name of the set. The model is selected by it at computation time |
| `field` | string | Value field the model was fitted to |
| `struct` | integer | Structure number, from one |
| `model` | integer | Model: 1 spherical, 2 exponential, 3 gaussian, 4 power |
| `sill` | real | Contribution C of the structure |
| `range` | real | Correlation range a |
| `azimuth` | real | Azimuth of the major axis, degrees |
| `anis` | real | Anisotropy, the ratio of the minor axis to the major one |
| `nugget` | real | Nugget C0. Common to the profile, repeated in each of its rows |
| `val_pct` | real | Clipping percentile for outliers. Common to the profile |
| `val_cap` | boolean | Capping to the boundary instead of removal. Common to the profile |
| `fitted` | string | Date of fitting |
| `author` | string | Who fitted it |
| `note` | string | Note: how the values were obtained, what was checked |

The fields `nugget`, `val_pct` and `val_cap` belong to the profile as a whole. A discrepancy between the rows of one profile is treated as a data entry error, a warning goes to the log and the value of the first row is used.

The `model` column accepts both a numeric code and the name of the model, in Russian or in English. The tool writes the code.

### Order of work

**1.06 Variogram (experimental)** fits a model and enters it into the table. When no table exists yet, the tool creates one through the **Variogram model** output. Once a table exists, it is given in the **Append to the table of models** field. The rows of this profile are replaced by the new ones,

while the rows of other profiles remain untouched. Fitting one profile again replaces its rows, so duplicates do not accumulate.

**1.07 Variogram map** writes the azimuth and the anisotropy ratio into the same table, and where possible the range of the major axis. The model, the nugget and the contribution are not changed: they are determined by the omnidirectional variogram, while the map describes only the geometry. The estimate off the map is indicative, which is recorded in the `note` field.

**1.08 Variogram cross-validation** reads the model from the table for checking.

**1.01 2D Kriging** and **3.01 Categorical indicator kriging** read the model at computation time. When a table is given, the variogram fields of the dialog are not used: two sources of the same values inevitably diverge, so the table takes precedence, and that is reported in the log.

#### Parsing of rows

A row with an unrecognised model, a non-positive range or a repeated structure number is skipped, and the reason together with the row number goes to the log. The computation continues over the usable rows. A profile left without a single usable structure does not reach the result.

The parser tolerates manual editing. The case of column names is ignored, a decimal comma and spaces inside numbers do not interfere. Synonyms of column names are not provided, since the table is produced by the plugin itself.

## 1.07 Variogram map (anisotropy)

The tool builds a variogram map - the semivariance surface γ as a function of the two-dimensional separation vector (h_x, h_y). An ordinary variogram averages all directions into one curve and loses directionality, while the map shows how the continuity of the parameter depends on direction. From it you can see whether there is anisotropy in the data and where the axis of maximum continuity points. The tool is diagnostic. It does not compute a grid but helps to set the azimuth and anisotropy in the 2D Kriging variogram structure deliberately.

#### What anisotropy is and why to see it

An isotropic variogram assumes the link between values depends only on the distance between points, not on direction. For folded and elongated geological bodies this is not so. Along strike the seam is sustained, across it it changes faster, and the same difference in roof elevations is gained over kilometres along the fold but over hundreds of metres across it. If this is not accounted for, kriging smooths the field equally in all directions and blurs the real elongation of the structure.

A variogram map reveals the directionality directly. For each pair of points not only the distance is taken but also the direction of the vector between them, and the semivariance of the increment is spread over a two-dimensional grid of lags. Where γ grows slowly and the map stays dark far from the centre, continuity is high. Where γ grows fast, continuity is low. The low-γ area as a whole stretches into an ellipse. Its long axis is the direction of maximum continuity - for folding this is the strike direction.

#### How to read the map

At the centre of the map lies the zero lag. A value at a point always equals itself, so γ here is zero and the centre is the darkest. As one moves away from the centre the points are separated farther and γ grows. The h_x axis points east, the h_y axis north, the scale on both axes is the same. The map is point-symmetric - a pair and its mirror image give the same semivariance, so the picture is the same in opposite directions.

Anisotropy is read from the shape of the dark area. If it is round - the structure is isotropic, direction plays no role. If it is elongated - along its long axis γ grows more slowly, i.e. in this direction values

are linked over a larger distance. Hints are drawn over the map, a white ellipse by the estimated ranges and a red dashed line along the major axis.

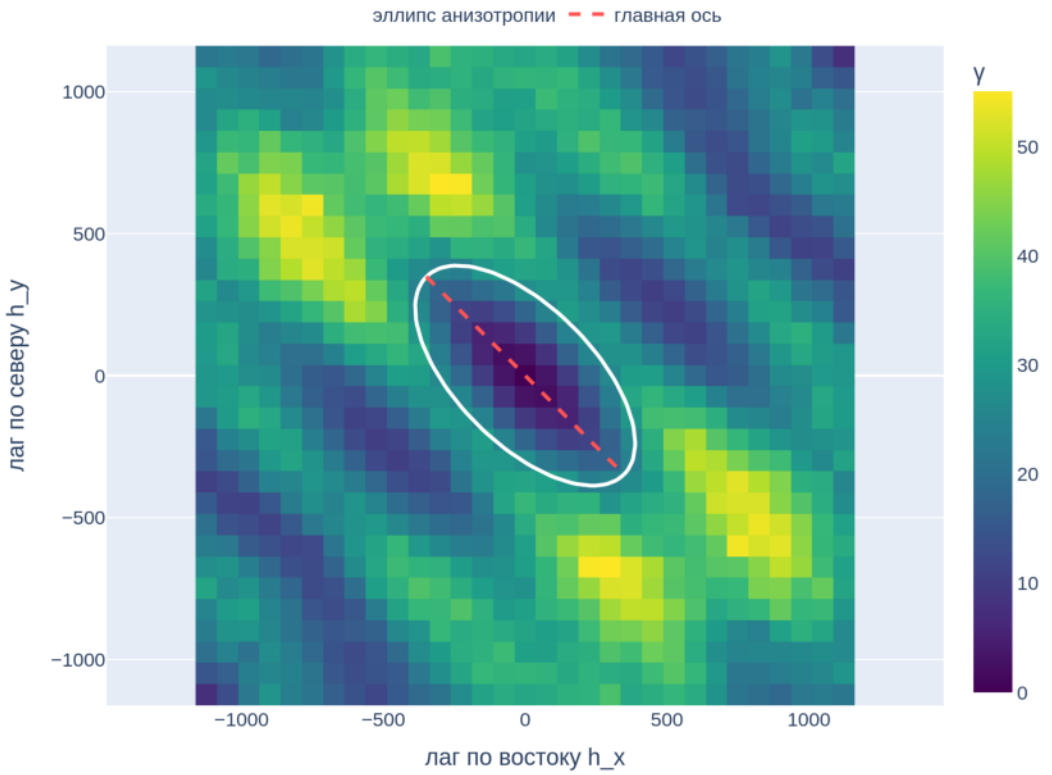



Figure 13: A variogram map: the dark (low γ, high continuity) area is elongated at an azimuth of about 135°. The white ellipse and the red dashed major axis show the estimated direction and anisotropy.

**Parameters**

| Parameter | What it sets | Default / advice |
|---|---|---|
| Points with values | A point layer of wells or samples. | - |
| Z value field | The numeric attribute to analyse: roof elevation, thickness, grade. | remembered between runs |
| Bins per half-axis (map detail) | Into how many cells each lag half-axis is split. The map comes out (2N+1)×(2N+1) in size. More bins - a more detailed map, but fewer pairs per cell and more noise. | 15 |
| Max. lag, in layer units | The map window size, in layer units (for metric coordinates - metres). 0 = half the extent diagonal. | 0 |

| Parameter | What it sets | Default / advice |
|---|---|---|
| Min. number of pairs per cell (adv.) | Cells with fewer pairs are left empty. Cuts off noisy far lags where pairs are few. | 5 |
| Report (HTML) | A report with the heatmap, the ellipse, the major axis and a summary of estimates. | temporary file |
| Surface raster (opt.) | The γ surface as a raster in lag coordinates (see below). Not created by default. | off |

The parameter marked “adv.” is in the collapsed **Advanced Parameters** section.

The **wt** weight field from **1.05 Declustering** works here too, the same way as in 1.06. A pair of points is taken with a weight equal to the product of its endpoints’ weights.

### Estimating the azimuth, anisotropy and range

Besides the map itself the tool outputs three numbers, to the log and to the HTML report.

The major-axis azimuth is geographic, zero is north and the count runs clockwise. The anisotropy coefficient is the ratio of the minor axis to the major, one means isotropy and a smaller value a stronger elongation. The third number is the major-axis range. The estimate works like this. Along each direction the lag at which γ reaches the plateau (close to the data variance) is found, and the ranges are smoothed over azimuth. The major axis is taken at the largest range, and the minor perpendicular to it.

These three numbers are substituted into the 2D Kriging variogram structure - azimuth, anisotropy (minor/major) and range a. This is exactly how anisotropy enters kriging. The estimate is indicative, and it should be checked against the shape of the heatmap itself, not transferred blindly. The azimuth the map determines most reliably, while the range and the coefficient are cruder, especially on a sparse network.

To avoid transferring the numbers by hand, the dialog has a **Write anisotropy to a profile** field. Pick a previously saved profile, and the azimuth, the coefficient and the major-axis range are written into it on top of the model and nugget set in **Variogram**. The next time the profile is loaded in **2D Kriging**, these values are applied automatically and appear in the caption under the profile list. If the range hit the window, it is left unchanged and only the azimuth and the coefficient are updated.

If the structure is close to isotropic or the major-axis range turns out smaller than a few map cells, anisotropy is not estimated and the report marks it as “not expressed”. The ranges lie at the grid level and the directionality is unreliable, and reporting that is better than giving a random azimuth. It helps to reduce the max. lag or increase the number of bins to resolve the near structure.

### When the range hits the window

If along the major axis γ does not manage to reach the plateau within the window, the range is returned equal to the max. lag. A warning then appears in the report and the Log, saying that the range hit the max. lag and is a lower bound. This is the same situation as for an ordinary variogram (see “Maximum distance and reaching the plateau”) - the curve did not reach the plateau, and the sill is obtained by extrapolation. On the map the sign is simple, the dark area along the major axis stretches to the very edge.

In this case the range a cannot be carried into kriging as is. The real correlation length is larger than the window, and the anisotropy coefficient is understated in strength - the field is in fact even

more anisotropic. The azimuth, meanwhile, is usually determined normally. The cure is to increase the max. lag so the map captures the plateau. And if γ does not reach the plateau even in a wide window, a trend dominates the data, so remove it before interpolation or account for it with the appropriate kriging type.

#### The surface raster

If desired, the map is also saved as a raster (the **Surface raster** field). It is the same γ surface but in lag coordinates - the origin at (0, 0), the pixel size equal to the lag cell. The raster is not georeferenced, it lies in the separation space rather than in the deposit plan. It is meant for those who want to spin the map on the QGIS canvas, apply their own colour scale or measure a lag with a ruler. The HTML report is enough for the anisotropy estimate itself.

## 1.08 Variogram cross-validation

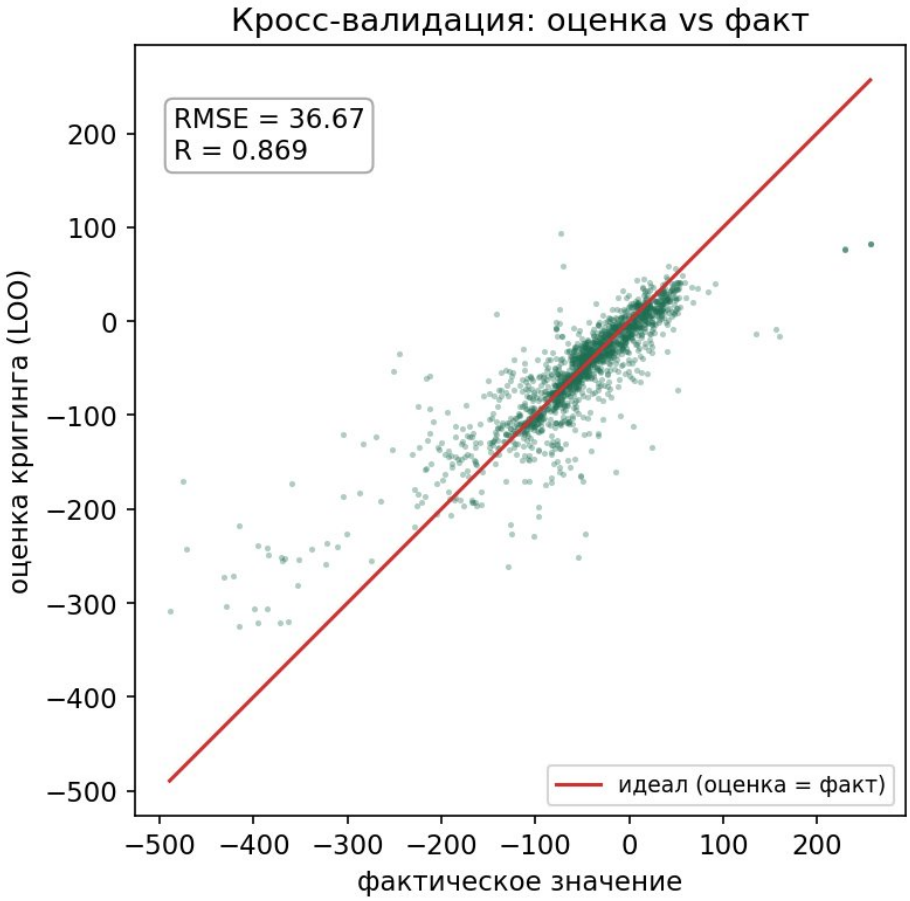



Figure 14: The idea of cross-validation: the kriging estimate from the remaining points (vertical) is compared with the actual value (horizontal). The tighter the cloud lies on the estimate = actual diagonal, the more accurate the prediction.

The tool checks how well the variogram is fitted, by the leave-one-out method. Each well in turn is excluded, its value is predicted by kriging from all the rest, and compared with the actual one. This way the parameters (nugget, range, model) are tuned by error rather than subjectively.

Parameters:

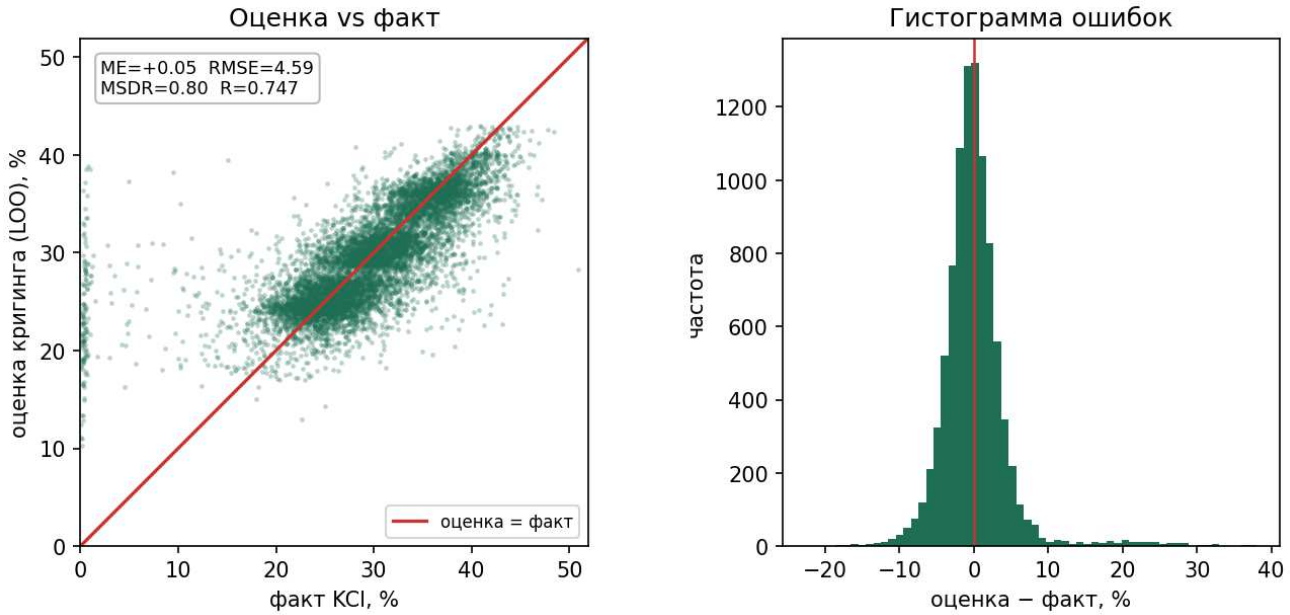



Figure 15: What the cross-validation HTML report looks like: on the left the “estimate vs actual” chart with the diagonal and metrics (example - KCl for the KrII seam), on the right the error histogram. A dense cloud along the diagonal - the model works. A band at an actual value near 0 - replacement zones. The histogram is symmetric about 0 - no bias.

| Parameter | What it sets | Default / advice |
|---|---|---|
| Points with values | Source points (wells). | - |
| Value field Z (empty = from the geometry) | The numeric attribute being checked. | - |
| Well-number field | An ID field for labels in the report and residuals layer. | optional |
| Kriging type, radius, min/max points, nugget, structures | Kriging and variogram settings. The check runs kriging with exactly these, so a good set carries into “2D Kriging” unchanged. | as in “2D Kriging” |
| Remove polynomial trend | Regression kriging: the trend is refit at each LOO step, the gain shows in the RMSE. | off |
| Trend degree | A plane or a quadratic surface. | 1 (plane) |
| Variogram models (table) | Table layer holding fitted models. | not set |
| Profile from the table | Name of the set in the table. | empty |
| Residuals layer (points) | Points with actual/estimate/error fields (see the field table below). | optional |
| Cross-validation report (HTML) | Interactive report: estimate vs actual, histogram, QQ-plot, metrics. | created by default |

The Log outputs the metrics:

**ME (mean error)** - the systematic error. Should be close to 0 (unbiasedness).

**MAE and RMSE** - the mean and root-mean-square prediction error. The smaller, the more accurate. But RMSE alone is not enough, it is minimal at a zero nugget (overfitting), although the uncertainty is then estimated wrongly.

**MSDR (standardized error)** - the mean square of the error divided by the kriging standard error. Should be close to 1. If MSDR is noticeably above 1 - the variance is underestimated (the nugget or sill are small). If below 1 - overestimated.

**R** - the “estimate - actual” correlation coefficient.

It is useful to distinguish two sides. The “estimate - actual” cloud and the RMSE speak of the **prediction accuracy**. Whether the **model** itself is correct - whether the variogram describes the uncertainty faithfully - is shown by the standardized errors, an MSDR near 1 and the QQ-plot. For kriging both sides are valuable. A small RMSE with an MSDR near 1 means the model both predicts well and does not deceive itself about its own accuracy. Chasing RMSE alone is not allowed, it is minimal at a zero nugget, where the uncertainty is understated.

In practice try several variogram variants and compare. A good model gives ME near 0, a small RMSE and an MSDR near 1. If the RMSE pulls toward a zero nugget while the MSDR is huge, that is a sign of overfitting, and a small nugget calibrates the uncertainty.

The residuals layer shows where the model misses. Large residuals by absolute value mark problem areas. One and the same residual sign across an area betrays a local trend. The layer is named automatically after the validated field and the source, and the fields carry aliases - readable names, visible in the attribute table and in the field properties. The field std_resid holds the standardized residual, that is (estimate - actual) divided by the kriging standard error. The sign matters, minus means kriging underestimated, plus means it overestimated. This is not a variance, a variance is always ≥ 0.

The residuals-layer fields:

| Field | Alias | Description |
|---|---|---|
| `<well number>` | Well number | The value of the chosen ID field (if **Well number field** is set). |
| `<field name>` | Actual (field name) | The actual value of the validated field. |
| `z_est` | Kriging estimate (LOO) | The estimate from the other points (leave-one-out). |
| `error` | Error (estimate - actual) | Estimate minus actual. Minus - underestimated, plus - overestimated. |
| `abs_error` | \|Error\| | The absolute value of the error, \|error\|. |
| `std_resid` | Std. residual (signed) | (estimate - actual) / the kriging standard error, signed. Not a variance (which is ≥ 0). |

Besides the residuals layer the tool produces an HTML report. It has four parts - an “estimate against actual” chart, an error histogram, a residuals QQ-plot and a metrics table.

Two lines are drawn on the chart, the grey diagonal is the ideal and the blue one is the Best-fit regression line. The Best-fit slope, intercept and angle are added to the metrics as a range-bias indicator. A slope near 1 means the method is equally accurate at low and high values. A slope below 1 means high values are underestimated and low ones overestimated, the regression to the mean that is the signature of smoothing methods. The data variance is added to the table as a reference for the total sill C0+C. Next to the metrics table a **Kriging parameters** block is shown. Only the settings that differ from the defaults are listed there (nugget, sill, range, outliers and so on), so you can see which parameters produced these metrics. On the “estimate vs actual” chart,

hovering over a point shows the well number and the values. The eight wells with the largest residuals by absolute value are labelled right on the chart - they are convenient to check first. The report opens in the QGIS result viewer or in a browser. If plotly is unavailable in the QGIS build, the report is still created, with the metrics table but without charts.

**The residuals QQ-plot.** Shows the shape of the error distribution. The errors are normalized to their own variance (a z-score) and compared with the normal distribution, so the chart reads by shape at any calibration. The uncertainty scale is handled separately by the MSDR in the metrics table. The horizontal axis is the normal-distribution quantiles, the vertical is the normalized error. If the errors are normal, the points lie on the red diagonal. Deviations read at once. Curled ends (S-shaped) - heavy tails, i.e. more large misses than under normality. An overall arc - skew, worth considering a value transform. A separate group broken off the line - an alien population in the data, for example barren samples from replacement zones (where the component is practically absent). Normality matters because the MSDR and the standard-error map rest on it.

**The main thing - what to do with the results.** The point of the tool is to settle the whole set of parameters before the grid is built, so that you approve or correct what you will then set in **2D Kriging**. This is both the variogram (nugget, sill, range, model, anisotropy) and the kriging settings themselves (search radius, min/max points, type - ordinary or simple). Cross-validation computes kriging with exactly the same settings, so a good set is carried into the **2D Kriging** tool unchanged. The order of decisions is this.

- ME near 0, MSDR near 1, the RMSE and R suit you - the set can be approved. Carry these same parameters, the variogram and the search settings, into **2D Kriging** and build the surface.
- MSDR noticeably above 1 - kriging is too “sure of itself”, the standard-error map will be understated. Increase the nugget C0 or the sill and check again.
- MSDR below 1 - the uncertainty is overstated, so reduce the nugget or the sill.
- ME noticeably different from 0 - a systematic shift. Check the data and the kriging type, and for simple kriging the specified mean as well.
- A large RMSE and a low R - the model predicts poorly. Try a different range, model or anisotropy (azimuth and axis ratio). If nothing helps, it is the data’s limit, short-range variability the network does not catch. Ore replacement zones are an example - on the chart above this is the vertical band at an actual value near 0.

The residuals layer prompts pointwise. Where the residuals are large, the network should be densified (add wells) or the samples checked. Where the residuals are systematically of one sign over the area, there is a local trend kriging did not account for.

In sum, this tool is the last step before the final kriging. First you calibrate the variogram here by error, then set the same parameters in **2D Kriging**. The surface together with the standard-error map then come out justified rather than fitted subjectively.

A note on speed. The check solves kriging as many times as there are points, so on large sets (tens of thousands of wells) it runs noticeably longer. Reduce the sample if needed.

The kriging itself runs fast on dense data. The measurements are laid out in cells the size of the search radius, and in every grid node only the neighbouring cells are scanned rather than the whole sample. On a hundred and sixty thousand stations this gave a fourfold speed-up. The selection stays exact and the estimate does not change.

If the data are clustered unevenly, set the optional **wt** weight field from tool **1.05 Declustering**. The ME, MAE, RMSE, MSDR and R metrics are then computed with weights, so a dense cluster of wells does not dominate the quality assessment. The leave-one-out estimate itself is unchanged, only the summary is weighted.

## 1.09 Method cross-validation (LOO)

Leave-one-out control for a gridding method, kriging or minimum curvature. Each validation point is removed in turn, its value is predicted by the method from the rest and compared with the fact.

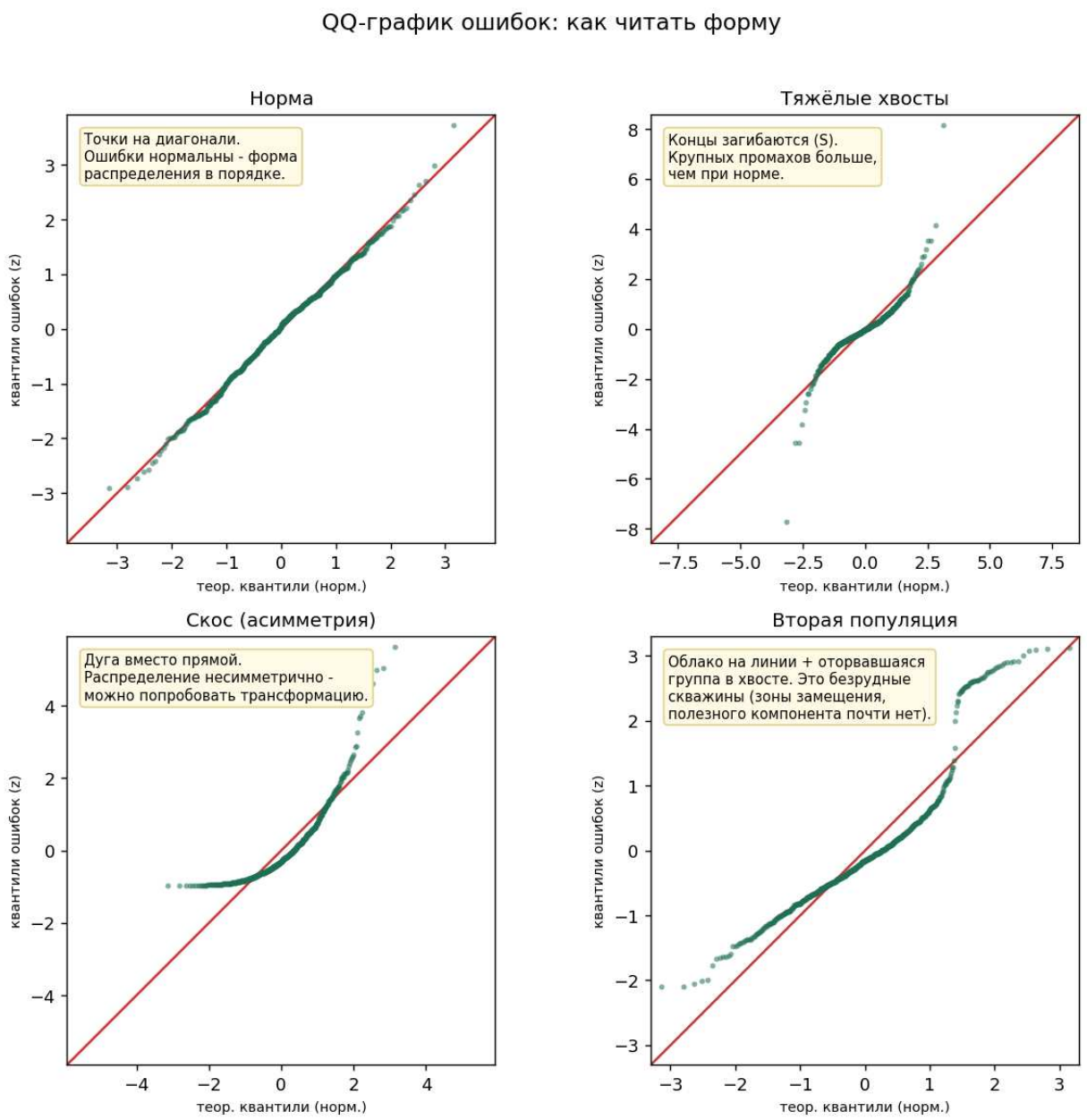



Figure 16: Four typical shapes of the residuals QQ-plot: normal (points on the diagonal), heavy tails, skew and a second population - a group broken off in the tail.

The errors give quality metrics - an objective measure of the method and a way to compare methods on your own data.

This differs from **1.08 Variogram cross-validation**. That one fits the variogram model for kriging, while this one compares gridding methods as such and works for minimum curvature too.

The metrics are **ME** (bias, closer to 0), **MAE** and **RMSE** (smaller is better) and **R** (correlation of estimate and fact). For kriging there is also **MSDR** (closer to 1 when the standard-error scale is adequate). The estimate-vs-fact chart has two lines, a grey 1:1 diagonal (the ideal) and a blue **Best-fit** regression line. Its slope, intercept and angle go into the metrics as a range-bias indicator. A slope near 1 means the method is equally accurate at low and high values. A slope below 1 means high values are underestimated and low ones overestimated, the regression to the mean that is the signature of smoothing methods.

Three Surfer-style options are available. A **random subset** of N points speeds control on large data, while the whole sample still takes part in each estimate. An **area filter** restricts validation to a subarea by extent and by Z value, useful to avoid control at known anomalies. An **exclusion buffer** in X and Y drops points in a rectangle around the validation point, needed for dense clusters, otherwise the estimate just repeats the nearest neighbour.

| Parameter | Purpose | Default |
|---|---|---|
| Points with values | Samples. | - |
| Value field Z (empty = from the geometry) | Numeric field. | - |
| Well id field | For labels in the report. | - |
| Method | Kriging or minimum curvature. | Kriging |
| Points to validate | 0 = auto, min(N, 100). | 0 |
| Exclusion buffer in X, Y | Rectangle around the point, neighbours in it are left out. | 0 |
| Kriging parameters | Variogram and search (as in 1.01). | - |
| Min curvature parameters (Adv.) | Extent, cell, tension, threshold, iterations. | auto |
| Validate only within the extent (Adv.) | Control area by X/Y. | everywhere |
| Validate where Z is in range (Adv.) | Control area by value. | none |
| RNG seed (Adv.) | For a reproducible subset. | 0 |
| Cross-validation errors | Point layer with fact, estimate and error fields. | - |
| HTML report | Estimate-vs-fact plot, histogram, metrics. | default |

For minimum curvature each point is re-estimated from a warm start off the full solution, so a single pass is fast. On very large samples reduce the number of validation points.

The **wt** weight field from **1.05 Declustering** works here the same way as in 1.08. The summary of metrics is weighted, while the leave-one-out estimate itself does not change.

## 1.10 Example wells (demo)

The **Example wells (demo)** tool builds a point layer with random coordinates and three structured fields. These are the absolute roof elevation (roof), the thickness (thick) and the grade of an abstract component X (%). The roof and thickness ranges are set after the model of an industrial seam (KrII). The tool is meant for learning and testing kriging, isolines and cross-validation without real data.

Parameters:

| Parameter | What it sets | Default / advice |
|---|---|---|
| Area (extent) | The generation rectangle (by layer, canvas, coordinates, drawing). | - |
| Number of wells | How many points to create. | 300 |
| Minimum / maximum of value X | The component grade range. | 0 / 50 |
| Smoothness (fraction of extent) | Correlation range as a fraction of the extent: larger - bigger “patches”. | 0.15 |
| Roof, thickness: min/max (Adv.) | Ranges of the roof and thick fields. | as for KrII |
| Nugget fraction (Adv.) | Share of variance on short-range noise: larger - less predictable. | 0.35 |
| Add a categorical mineral-type field | A mintype field (silvinite, replacement) for indicator kriging. | off |
| Add a head field | A head field with a regional slope for the flow gradient. | off |
| Add K and T fields and head | Head plus log-normal K (m/day) and T = K·thickness for the specific discharge (Darcy). | off |
| RNG seed (Adv.) | Reproducibility of the generation. 0 = random. | 0 |
| Declustering weight field (Adv.) | **wt** weights from tool **1.05**, optional. | none |
| Sample wells (demo) | The output point layer. | - |
| Drift surface (raster) + dz field | Enable the output to get an s raster and a dz field for external drift. | off (skipped) |

At start the Log outputs the starting variogram (total sill ≈ the data variance, nugget, range). The generated data have a recoverable variogram, so the whole cycle can be learned on them. Build a grid in **2D Kriging**, then isolines, and check the parameters with cross-validation.

Two checkboxes and a separate output add optional fields for learning the related tools.

**Add a categorical mineral-type field** adds a mintype field with a silvinite background and replacement spots for categorical indicator kriging.

**Add a head field** adds a head field with a pronounced regional slope for the hydraulic gradient. Krige head, feed the raster to the flow tool, and the arrows follow the head downhill.

Enabling the **Drift surface** output writes, as a separate raster, a smooth secondary surface s known everywhere, and adds a dz field linearly related to it. This pair is for learning external drift kriging. Krige dz with the s raster as the drift and compare it with plain kriging of dz without the drift. If the drift-surface output is skipped, the dz field is not added.

The **Add K and T fields and head** checkbox generates head, log-normal K and T = K·thickness. K is the hydraulic conductivity, spanning orders of magnitude as in real pumping tests. They are for learning the Darcy specific discharge. Krige K and T in **2D Kriging** with the **ln** transform (or ln fields by hand), and krige head as well. Then feed the rasters into the **Specific discharge** tool.

Result fields:

| Field | Type | Holds |
|---|---|---|
| well | text | Well number, format SK-0001. |
| roof | number | Absolute roof elevation of the seam, m. |
| thick | number | Seam thickness, m. |
| X | number | Grade of the abstract component, %. |
| head | number | Head (piezometric level), m. With the head checkbox or the K and T checkbox. |
| K | number | Hydraulic conductivity, m/day (log-normal). Only with the K and T checkbox. |
| T | number | Transmissivity T = K·thickness, $m^2$/day. Only with the K and T checkbox. |
| mintype | text | Mineral type (silvinite, partial replacement, rock salt). Only with the mineral-type checkbox. |
| dz | number | A value linearly related to the drift surface. Only when the drift-surface output is enabled. |

The field set covers all the plugin tools. Interpolate roof and thick with ordinary kriging, X with a trend, mintype with indicator kriging, and head, K and T with the hydrogeology tools.

The full list of output-layer fields is in the **Sample wells (demo)** appendix section (“Demo-layer fields” at the end of the manual).

## 1.11 Example geophysical profiles (demo)

The tool creates a point layer of geophysical profiles for learning and testing without real data. Several parallel profiles with pickets are built. There are two modes.

**Electrical prospecting.** Along the profiles apparent resistivity rho_k (Ohm*m), self-potential SP (mV) and induced polarisation IP (mV/V) are set. The data contains a low-resistivity anomaly - a water-bearing or replaced zone. There rho_k drops from a background of tens of Ohm*m to units, and SP shows a negative minimum. This matches field practice. The lowest resistivities are typical of sedimentary rocks, while salt, gypsum and anhydrite are high-resistivity, and water saturation or salinisation drives the resistivity down. The anomaly is set as a compact spot rather than a stripe, so the profiles are not synchronous and interpolation shows a local focus rather than a solid band.

**Subsidence (trough).** The value is settlement (mm) as a subsidence trough over a mined area, across several observation tours. The trough deepens from tour to tour and is capped at two metres in magnitude. The sign is uniform, down (negative) or magnitude (positive), by choice. The edges are strictly zero - away from the mined area there is no subsidence. At the same pickets you can compute the **settle** difference between tours to get the settlement rate.

The workflow repeats the main one. The **rho_k** (or **settle**) field is interpolated with **1.01 2D Kriging** or minimum curvature, isolines are built from the grid with **1.04**, and the anomaly is outlined. The **sp** field can be interpolated the same way and its SP minimum compared with the rho_k drop. The **rho_true** (or **settle_true**) field is the embedded noise-free value, a reference for checking interpolation accuracy.

**Kriging or minimum curvature**

Geophysical profiles are a typical case where the choice of interpolation method matters more than its tuning. The data are dense along the profiles and sparse between them, while the quantity itself (resistivity, potential) is physically smooth and continuous.

Kriging faithfully reflects the uneven network. Without tuning the variogram anisotropy it stretches the structure along the survey lines, and the field breaks into bands along the pickets. Minimum curvature (**1.02**) imposes a physically meaningful smoothness and stitches the separate profiles into a connected surface, so for profile surveys and potential fields it is usually preferable. The same holds for the **sp** field, where the SP minimum appears as a single body rather than columns. It is most vivid on subsidence. The trough is a compact axisymmetric bowl, kriging rolls it into a band along the profiles, and minimum curvature restores the bowl with a clear centre.

The practical takeaway is simple. Build fields from profiles with minimum curvature, and use kriging when the variogram anisotropy is tuned to the network geometry.

The area is set by an extent. The mode, ranges, number of tours and surface elevation can be changed under **Advanced**.

| Parameter | Purpose | Default |
|---|---|---|
| Extent | Generation bounds. | - |
| Mode | Electrical prospecting or subsidence. | Electrical prospecting |
| Number of profiles | How many parallel profiles. | 4 |
| Picket step, m | Distance between pickets. | 20 |
| Background rho_k, Ohm*m (Adv.) | Resistivity outside the anomaly. | 60 |
| Minimum rho_k, Ohm*m (Adv.) | Resistivity at the spot centre. | 10 |
| SP amplitude, mV (Adv.) | Depth of the SP minimum, usually below 0. | -100 |
| IP amplitude, mV/V (Adv.) | Height of the IP anomaly. | 15 |
| rho_k noise (Adv.) | Spread fraction in log scale. | 0.06 |
| Maximum subsidence, mm (Adv.) | Trough depth at the last tour. | 400 |
| Number of tours (Adv.) | How many cycles for subsidence mode. | 2 |
| Subsidence sign (Adv.) | Down (negative) or magnitude (positive). | Down |
| Elevation: base and amplitude, m (Adv.) | Smooth surface relief. | 120 and 15 |
| Geophysical profiles | Point layer with the fields. | - |

The electrical fields are **profile** (profile number), **picket_m** (picket in metres from the profile start), **pk** (a PK label, e.g. PK5+20), **z** (surface elevation, m), **rho_k** (rho_k, Ohm*m), **rho_true** (rho_k without noise), **sp** (SP, mV) and **vp** (IP, mV/V).

The subsidence fields are **profile**, **picket_m**, **pk**, **tour** (tour number), **z** (elevation, m), **settle** (subsidence, mm) and **settle_true** (subsidence without noise).

## 1.12 Graft a detailed surface into a regional one

The tool sews two surfaces of the same quantity into one, a detailed surface over a site and a regional one over the whole area around it. The task keeps coming back to surveyors and hydrologists - a detailed survey into an open DEM - but it does not depend on terrain. A cluster of exploration holes into a regional model of a seam roof, a detailed sampling area into a general grade map. The arrangement is the same, the quantity is any.

Two rasters and a polygon of the graft area go in. One raster at the given extent and resolution comes out.

| Parameter | What it sets | Default / advice |
|---|---|---|
| Detailed surface | The raster of the site. Its grid becomes the working one. | - |
| Regional surface | The raster of the surroundings. Brought onto the detailed grid. | - |
| Graft area (polygons) | Where the detailed surface stays. Has to be narrower than its coverage. | - |
| Width of the transition, cells of the regional surface | The ring in which one surface passes into the other. | 4 |
| Removing the systematic offset | Median over the ring, a tilted plane, or leave it. | median |
| Extent of the result | By the regional surface, by the detailed one, or set by hand. | by the regional |
| Cell of the result | 0 - same as the detailed surface. | 0 |
| Limit on the size of the result, millions of cells (adv.) | Beyond it the tool refuses to work. | 50 |
| Minimum cells of overlap (adv.) | Below this count the correction is treated as unreliable and reported in the log. | 200 |

**Two kinds of step, of different nature**

**A systematic offset.** For terrain these are different vertical systems. A detailed survey runs in a local one, open products in a geoid-based one - COP30 on EGM2008, SRTM on EGM96. The discrepancy is measured in metres and is the same across the whole site. In geology the same place is taken by different methods of adjustment, another drilling reference, a correction of core against logging.

It cannot be smoothed away. The edge would smooth out while the whole regional part stayed lifted or dropped, and the catchment would shift with it. So the offset is removed beforehand, over the ring of overlap around the graft area, where both surfaces exist.

The median is taken rather than a mean. Roofs, canopies and single holes fall into the ring, and in radar products those are outliers of tens of metres, so a mean would be shifted by them while a median holds. A tilted plane is needed where the site stretches for tens of kilometres and the correction changes across the area. Before the fit the outliers of the ring are cut off by percentiles.

On a constant offset the median and the plane give the **same** result, and that is not an error, because a plane fitted over a ring with an unchanging offset degenerates into a constant. The difference between the modes appears only where the offset varies across the area. The correction is reported in the log as a number, and that number shows what is happening.

The third mode, «do not remove», is for surfaces that are certainly in the same system, where a correction would only introduce an error. That happens when the detailed and the regional models are computed from the same holes. The median over the ring is still reported in the log, and if it is markedly different from zero, the choice should be reconsidered.

**A mismatch of shapes at the joint.** Here a weight does the work rather than a hole cut out and interpolated afterwards. Inside the area the weight of the detailed surface is one, beyond the buffer it is zero. In the ring it goes over smoothly, by a function with zero slope at both ends. The transition comes out smooth by construction rather than by the fact of smoothing, and no shape invented by interpolation gets into the seam. A linear transition would put a break of slope exactly on the border of the ring - just where hydrologists look for one.

### The extent and the resolution of the result

The extent is taken from the regional surface by default, because a graft is made for the whole of the surroundings rather than for the patch under the site. The extent of the detailed surface can be used instead, or a rectangle set by hand.

The cell is set separately, and on a large extent that is a must. A detailed survey may be centimetric while a regional one is thirty metres over tens of kilometres. A graft at the detailed step would give a raster that does not exist. Fifty kilometres at a cell of five centimetres is a trillion cells. At a cell of one metre two and a half billion are left, at five metres a hundred million, at thirty fewer than three million.

So the size of the result is worked out beforehand, and beyond the limit the tool refuses to work and names a fitting cell. The limit sits in the advanced parameters, fifty million cells by default.

The detail inside the graft area is bounded by the cell of the result rather than by the survey, so a centimetric survey in a five-metre grid gives five-metre detail. If the whole detail is needed, narrow the extent to the surroundings of the site.

When coarsened, the detailed surface is averaged rather than sampled at the nodes. Sampling takes the value at a node and loses everything between them, and on a detailed survey that is the detail itself.

Both surfaces are clipped to the frame of the result and brought onto its grid together with their holes. In a DEM a hole is marked by a service value, and that value has to be recognised. Otherwise it goes into the computation as a real elevation, and the ring of overlap finds data where there is none. The correction then loses its meaning, while to the eye the raster is simply inserted without a transition. The log reports the share of cells with data for each surface. A suspiciously high share for the detailed one means the holes in its file are not marked, which has to be fixed before grafting.

For the tilted plane the log reports its extreme values over the field. The plane is fitted over the ring around the site but applied to the whole extent, and on a large frame it can go far beyond what it saw. So if the spread of the correction is markedly larger than the offset itself, take the median.

### The graft area has to be narrower than the survey

The ring of overlap is taken outside the area. If the detailed data end exactly along it, there is no overlap and nothing to measure the offset against. The tool says so plainly instead of sewing silently. A silent graft would leave a step of the whole datum, and it gets noticed when the flow stalls at the seam.

Hence the rule. Draw the graft area inside the coverage of the detailed survey, with a margin of at least the width of the transition.

### What goes into the log and what to look at

The tool reports four quantities. These are the number of cells in the overlap, the offset before the correction as a median and a spread by percentiles, the remainder after the correction, and the step between neighbouring cells. The last one is given separately at the seam and over the rest of the field.

That last pair is the measure of acceptance. If the graft worked, the step at the seam does not stand out against ordinary terrain. Take a test scene with an offset of 12.4 m. A manual graft left a step of 14.5 m at the seam, the tool leaves 2.6 m against a field background of 2.55 m. When the seam is markedly larger than the background, the log gets a warning, and usually the offset was removed wrongly or the transition is too narrow.

Gaps inside the detailed surface are closed by the regional one, no separate filling is needed.

Hence an important difference from the manual way this task is solved in QGIS. There you cut a hole in the regional surface by a buffer around the mask, glue the two together and fill the void by interpolation. Here no hole is cut at all and no empty cells appear, in the ring both surfaces are

taken with a weight. Interpolation in a hole would invent a shape that is not in the data, while a weight takes what already exists on both sides and changes their share.

And the main point. The step at the seam comes not from the way of gluing but from the systematic offset between the products. If it is not removed, the seam stays as high as the whole datum, whatever the hole is filled with. On a test scene, closing the hole with the regional surface without removing the offset gives a step of 14.4 m. The graft with the correction gives 1.62 m against a terrain background of 2.12 m.

If empty cells do end up in the result, their count is reported in the log. A cell can be empty only where neither of the two surfaces exists, and usually the regional one does not cover the given extent.

### Topography: terrain from open data

The **“2. Topography”** group answers a frequent request - the best possible terrain model from open data, out of the box. The front door to the group is the DEM downloader by extent. Next to it stand the vector base map from OpenStreetMap and Topo2Raster, which builds terrain from points and contours.

Then come terrain preparation with depression filling, slope with aspect, and peaks. Flow and accumulation, the river network, basins and catchments are in the **«6. Hydrology»** group, which continues this chain. Nothing has to be installed for it, no GRASS, no SAGA, no other third-party modules.

Next to it stands the **“2. Topography: diagnostics and repair”** group. It holds the tools that check a finished relief - splitting contours into sets, the residual against the DEM and the search for terracing. A separate group is needed because Processing has no subgroups, and keeping the checks inside the working chain is awkward, they break the sequence of building. The tool numbering stays continuous with topography.

All output layers of the group land in the **Topography** group of the layer tree, so they do not drown among the working layers of the project. The tools of the group chain together. Downloader 2.01 delivers a ready metric DEM that goes straight into isolines (1.04) and any computation of the group. Watercourses from 2.02 and the river network from 6.02 fit Topo2Raster (2.03) as streamlines as is, because their vertices run downstream.

The full list of output-layer fields is in the **Electrical-prospecting / Subsidence profiles** appendix section (“Demo-layer fields” at the end of the manual).

## 2. Topography

### 2.01 Download DEM by extent

Downloads a DEM by extent from an open store, no registration or keys, from one of three sources. Tiles are mosaicked seamlessly and reprojected into a metric coordinate system with cubic resampling. Raw degree tiles never enter the analysis, so the GLO-30 peculiarity north of latitude 50 (a coarser longitude step) is handled automatically.

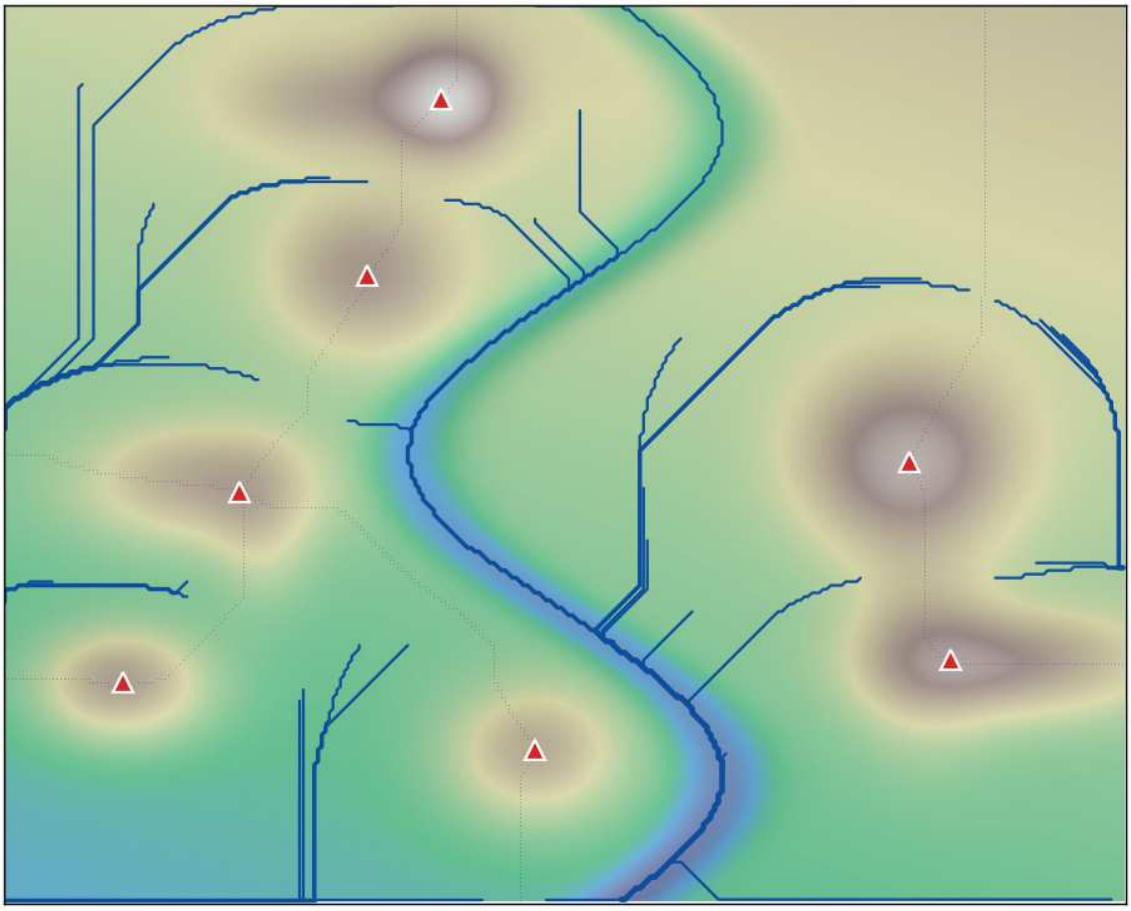


Figure 17: The full chain of the group on the demo relief: hillshade, the river network with width by Strahler order (6.02), basin boundaries (6.03) and peaks (2.06).

| Parameter | What it sets | Default / hint |
|---|---|---|
| Terrain source | GLO-30 (Copernicus, DSM: heights over canopy and roofs), GEDTM30 (DTM with forest and buildings removed, machine learning from ICESat-2/GEDI) or Mapzen Terrain Tiles (a compilation of SRTM, ArcticDEM and national DEMs). | GLO-30 |
| Download extent | Extent in any CRS, converted to degrees automatically. | - |
| Target CRS | Metric output CRS. Empty = project CRS if metric, otherwise UTM by extent center. | empty |
| Cell size, m | Output grid step after reprojection. | 30 |
| Smooth terrain (FPDEMS) | Edge-preserving smoothing (see below). off = raw source raster. | off |
| Hydrological correction | Filling of false depressions so water flows down. Uncheck for karst and subsidence troughs. | on |
| Smoothing: normals window, cells (adv.) | Side of the normals filter window, odd. | 11 |
| Smoothing: normals difference threshold, deg (adv.) | Edge-preservation strictness: lower = more aggressive preservation. | 15 |
| Smoothing: number of passes (adv.) | Outer elevation-rebuild steps. | 2 |
| Slope epsilon for filling, m (adv.) | Minimum slope on flats. 0 = raise pits to the outlet only. | 0.001 |
| Limit of 1x1 degree tiles (adv.) | Guard against an oversized extent for GLO-30 and Mapzen. | 25 |
| DEM (metric CRS) | Output float32 raster, layer in the Topography group. | - |

**Terrain source** is the key choice. **Copernicus GLO-30** is a digital surface model (DSM), heights are taken over the top of canopy and rooftops, and it is distributed as one-degree tiles. **GEDTM30** is a digital terrain model (DTM) by OpenGeoHub under CC BY 4.0, where forest and buildings are removed by machine learning from ICESat-2 and GEDI data. Under forest canopy GEDTM30 is markedly more accurate, and independent validation confirms it. It is distributed as a single global cloud-optimized GeoTIFF. For forested terrain the DTM is usually preferable, for open areas the difference is small.

**Mapzen Terrain Tiles** is a compilation of open DEMs that Mapzen assembled and published on AWS Open Data. Over Russia it holds SRTM 30 m south of latitude 60 and ArcticDEM to the north, so for the Kama region it is the same SRTM. Elsewhere the compilation includes national models: the USA from 1 to 10 m, Great Britain 2 m, Norway 10 m, Australia 5 m, New Zealand 8 m, and EU-DEM for Europe. The tiles form a pyramid of zoom levels, and the level is chosen from the cell size: the tile pixel is taken no coarser than the cell, at the latitude of Perm a 30 m cell gives about 20 m. Where detailed national data exist, a fine cell gives real detail. Where they do not, a fine cell only makes the file bigger. Most sources in the compilation are surface models, like GLO-30. If the

extent is too large for the given cell, the tool takes a coarser level and says so in the log. The source is useful as an independent check of GLO-30 and wherever it carries a national model.

Parameters:

- **Download extent** - the extent in any CRS, converted to degrees internally to pick the data.
- **Target CRS** - the metric CRS of the result. Leaving it empty enables the automatics, the project CRS is taken if it is metric, otherwise the UTM zone at the extent center. A degree target CRS is rejected with a clear message.
- **Cell size, m** - 30 by default, the native GLO-30 resolution. For Mapzen the cell also sets the tile level.
- **Hydrological correction** - on by default, spurious depressions are filled right away (see 2.04) so water flows downhill on the model. For tasks where closed basins matter (karst, subsidence troughs) uncheck the box.
- Under **Advanced** there are **Slope epsilon for filling** and **Tile limit** (a guard against an accidental extent covering half a country).

A network failure or an extent entirely in the ocean ends with a clear message rather than an empty raster. The data source is Copernicus DEM © ESA, and the open license allows use with attribution. GEDTM30 © OpenGeoHub, CC BY 4.0. Each part of Mapzen Terrain Tiles has its own authors, the download log gives a link to their list (github.com/tilezen/joerd, docs/attribution.md), and those are the ones to credit in a publication.

## 2.02 Download base topography by extent

The vector twin of the DEM downloader. For the same extent it fetches terrain-related layers from OpenStreetMap.

| Parameter | What it sets | Default / hint |
|---|---|---|
| Download extent | Extent of the OSM Overpass request. Keep it modest - large requests are cut by the server. | - |
| Extent area limit, square degrees | Guard against an oversized request. | 0.5 |
| Request timeout, s | Overpass response wait; on failure the tool switches to a mirror. | 90 |

The output layers are watercourses, water bodies (planes), peaks with the **ele** field and cliffs (barriers). All land in the Topography group.

- **Watercourses** - rivers, streams and canals. In OSM watercourses are drawn downstream, so the layer fits 2.03 as streamlines without preparation.
- **Water bodies** - closed outlines of lakes and ponds, constant-elevation planes for 2.03. Compound multipolygons (large lakes assembled from several ways) are skipped by the first version.
- **Peaks with elevations** - natural=peak points with the ele elevation, ready summit marks for 2.03 and a control for 2.06.
- **Cliffs and embankments** - terrain breaklines for 2.03.
- **Coastline** - off by default, needed on coastal territories.

Output is in the project CRS, lines are clipped to the extent. Public Overpass servers have limits, so on a failure of the main server the request goes to a mirror, and for large territories you shrink the extent or raise the area limit under **Advanced**. The data are © OpenStreetMap contributors, ODbL license.

## 2.03 Topo2Raster (terrain from vectors)

Builds terrain from vector data by multigrid interpolation from a coarse grid to a fine one, in the spirit of ANUDEM. The tool covers the classic task. Digitized contour lines of a topographic plan, spot elevations, rivers and lakes are at hand, and a correct grid is needed.

| Parameter | What it sets | Default / hint |
|---|---|---|
| Elevation points | Layer of spot heights (e.g. peaks from 2.06). | - |
| Point elevation field | Z attribute of the points. | - |
| Isolines | Contour layer (digitized plan or 1.04 over a DEM). | - |
| Contour elevation field | Level attribute of the contours. | ELEV |
| Streamlines (downstream) | River lines, vertices pointing downstream. The 6.02 network and OSM fit as is. | - |
| Cliffs (smoothing barriers) | Lines along which the drop is not smeared. | - |
| Weight of measured elevations (adv.) | In a shared cell a measurement outweighs a digitized contour vertex. 1 = as before. | 1 |
| Top of forms, Bottom of forms | The sides of forms with elevations: crests and toes, ridges and bases, a summit as a point. See the section on the surface between structural lines. | - |
| Form link field | The same value on every object of one form. | link |
| Elevation field of form sides (adv.) | When the lines carry no Z. | - |
| Shape function across (adv.) | Linear is a design slope, smooth rounds the edges. | linear |
| Lakes and water edge | Water-body planes. | - |
| Water-edge elevation field | Water-edge elevation. Empty = node Z (slope) or shore minimum. | empty |
| Extent | Output extent. Empty = from input layers. | empty |
| Cell size, m | Output grid step. | 30 |
| Fill depressions in the result | Hydrological correction of the result. | on |
| Smoothing iterations per level (adv.) | Passes of internal field smoothing. | 60 |
| Minimum streamline drop, m/cell (adv.) | Guaranteed slope along rivers. | 0.01 |
| Slope epsilon for filling, m (adv.) | Minimum slope on flats during filling. | 0.001 |
| Terrain | Output float32 raster. | - |

Points or isolines are required (at least one elevation source). The water-edge priority is this - node Z (river slope) over the elevation field (plane), field over shore minimum.

Every input type works as its own constraint:

- **Elevation points** and **contours** - hard nodes, the surface passes through them exactly. At least one of these layers is required, each with a numeric elevation field.

- **Streamlines** - a forced monotonic drop downstream. Line vertices must run downstream, and OSM watercourses (2.02) and the river network (6.02) fit as is. The minimum drop per cell is set under **Advanced**.
- **Cliffs** - smoothing barriers. The surfaces on the two sides of a cliff are independent, the step is not smeared. The line itself, one cell wide, gets an intermediate elevation.
- **Water edge** - three height priorities per feature, and a layer may mix types. A polygon may have three-dimensional vertices, a Z at the nodes. Then the edge is interpolated from their heights and sloped along the channel. This is how a river's falling level from source to mouth is set. If there is no Z but an elevation field is filled, the feature is held as a horizontal plane. With neither, the level is taken automatically from the minimum of the adjacent shore. Mixed feature types in one layer are handled each by its own branch, a lake as a plane, a river as a slope.

Inside, a two-stroke cycle runs at every grid level. Membrane smoothing sets the frame and holds the constraints, then minimum-curvature polishing (the Briggs stencil) removes the membrane bias between curved contours. On a round-trip test (demo relief, contours every 4 m, reconstruction, comparison with the original) the polishing cuts the error by roughly a third. The residual maximum error lives on summits above the last contour, so summit marks in the input visibly improve the tops - exactly why topographers label them on maps.

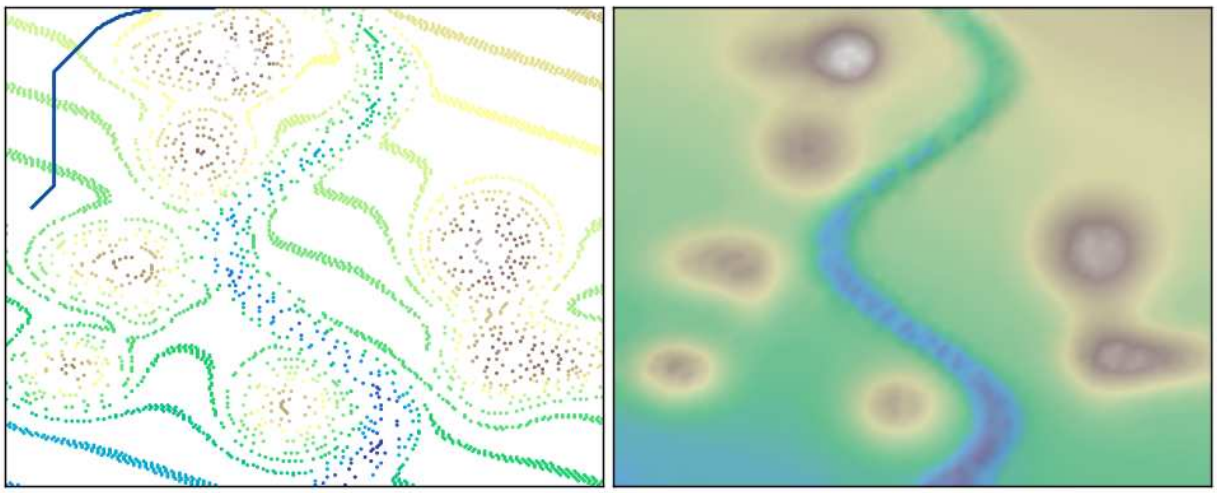


Figure 18: Left: the input constraints, densified contours (color by elevation) and the main streamline. Right: the reconstructed terrain.

The default extent is taken from the layers with a two-cell margin. All layers are brought to the CRS of the first given layer, which must be metric. The final **depression filling** is on by default, its logic is described in 2.04.

### Three-dimensional thalwegs

A thalweg without elevations works as a condition. A downstream fall is maintained along it, while the actual height is decided by the interpolator from the surrounding data. When the line carries vertex elevations, they become hard nodes, and a survey along the channel starts setting the bed rather than hinting at the direction.

The elevations are then brought once to a downstream fall. The reason is not neatness but the way the computation works. A channel survey is noisy, some vertices go uphill, and such a node would work against the fall enforcement on every iteration. Relaxation pins the cell to the elevation, the enforcement pushes it lower, and the cycle repeats. After the correction the enforcement becomes a no-op and the conflict disappears.

The correction only goes downwards, the tool never invents an elevation above the measured one. Its largest value is printed to the log, and it shows how noisy the survey was.

Cells where a thalweg elevation disagrees with another node by more than five centimetres are counted separately. Usually this is a channel crossing a contour, and the thalweg elevation is taken there. The number of such places is reported as a warning, such a disagreement must not be resolved silently. A line without elevations behaves as before, and a mixed layer is handled object by object.

### The surface between structural lines

The **Cliffs** input sets a barrier, the drop along a line is not smeared, but the line does not set the drop either, because it carries no elevations. The pair of inputs **Top of forms** and **Bottom of forms** solves the opposite problem - to place exactly the surface that the two sides with known elevations define.

Why this is needed is best put in the industry requirements for digital plans. Terrain created automatically has to be corrected by hand, agreeing it with the heights of retaining walls, slopes and fills. The tool replaces that prescribed manual correction with a rule. The second addressee is areal quarries, cuts, fills and dumps. Inside them contours are not described at all by the standard, so there is nothing to build the terrain from except crests and toes.

**A side is a set.** Any number of lines and points with one value of the link field goes into the top or the bottom, and the distance is measured to the union of the set. All the cases follow from this single rule. A slope is a crest and a toe. A pit is a closed crest and a point on the floor, a ditch is two crests and a floor line. A ring dam is two closed lines, and a river bank is a crest with part of the water edge. Lines and points may be mixed within one side.

**Elevations** follow the priorities - vertices with Z, then the elevation field, then the object stays a barrier and is not assembled into a form. There is no significance threshold in the tool. Supply two sides and a surface is built, supply one and it works as a barrier. The decision is made by the person who draws the second line.

**How it is computed.** No correspondence of points between the sides is sought. For every cell the exact distances to both sets are computed together with the elevation of the nearest source. The weight is the ratio of the distances, and the elevation is linear in the weight. An overlap of the geometry is impossible by construction. The body of the form goes into the multigrid as hard nodes. Its border additionally works as a barrier, so that outside data do not drag the surface across the bench. The relaxation scheme is not touched.

**The price of the method.** The nearest point and the corresponding point are different things. While the elevations are constant there is no difference. Once the crest elevation varies along a curved form, the surface departs slightly from the ruled one. On concave corners a medial axis appears, where the nearest source switches abruptly and the surface gains a kink. Both numbers are measured on synthetic data, because on a real survey there is nothing to measure them against, no analytic reference exists there.

**What goes into the log.** For every form there go in the number of body cells, the median width in cells, the elevation mismatch where the sides converge, and the number of objects skipped for want of elevations. Separately there is a warning about forms narrower than two cells, since such a form does not exist in a raster at any scale and the cell has to be refined. Lone sides and objects without a link go into the log with a reason.

### The boundary of the build area

A polygon limits the surface the same way an outer boundary does in design systems. Beyond it no raster is output, there is nodata.

What matters is not the clipping itself but the moment it is applied. The mask goes on **after** the interpolation rather than by clipping the input data. Points, contours and thalwegs beyond the

boundary keep shaping the surface right at it, so the edges do not curl. Had we clipped the input instead, an artificial break would appear at the boundary, because the interpolator would have nothing to lean on from the outside.

Supply the boundary layer in any coordinate system, the geometry is transformed automatically. The log prints how many cells fell inside and what share of the area that is. If none did, the run stops with a note to check the coordinate system, which is almost always the reason.

## 2.04 Terrain preparation

Prepares a DEM for analysis with two independent modifications, each toggled by its own checkbox, in a fixed order - smoothing first, then filling.

| Parameter | What it sets | Default / hint |
|---|---|---|
| Input DEM | Terrain raster to prepare. | - |
| Smooth terrain (FPDEMS) | Edge-preserving smoothing (see below). | off |
| Fill depressions | Filling of false pits (Planchon-Darboux). | on |
| Slope epsilon, m (0 = pits only) | Minimum slope on flats. 0 = raise pits to the outlet, above zero = a through slope for D8. | 0.001 |
| Smoothing: normals window, cells (adv.) | Side of the normals filter window, odd. | 11 |
| Smoothing: normals difference threshold, deg (adv.) | Edge-preservation strictness: lower = more aggressive. | 15 |
| Smoothing: number of passes (adv.) | Outer elevation-rebuild steps. | 2 |
| Prepared DEM | Output float32 raster. | - |

**Smooth terrain (FPDEMS)** removes the excessive roughness of satellite models. Satellite DEMs are noisy, and ordinary filters - mean, median, Gaussian - cut the noise together with edges, terrace walls and banks, the breaks get flattened. FPDEMS (Lindsay, Francioni, Cockburn, 2019) works differently. It operates on the field of surface normals rather than on heights directly. Each cell's normal is computed first. Then the normal field is smoothed, and the closer a neighbour's normal is to the central one, the larger its weight in the average. At an edge the neighbours' normals diverge sharply, the weight drops, the edge is preserved. After that the heights are pulled towards the smoothed normal field. As a result flat areas are smoothed while structural lines stand. The **normals-difference threshold** (in **Advanced**) controls the strictness. A smaller value preserves edges more aggressively, a larger one smooths more overall. The method was originally proposed for lidar DEMs but is equally useful for satellite ones.

**Fill depressions** with the Planchon-Darboux method raises spurious pits so flow does not stop. Depressions in raster models are most often interpolation and noise artifacts, and hydrological analysis without filling breaks at the first pit. **Slope epsilon** controls the mode. With zero only true pits are raised exactly to the spill level, flat areas stay flat. With a positive value (0.001 m by default) a through slope is additionally built across flats, and D8 becomes defined on them. Flow and accumulation need a positive epsilon. Cells on the grid border and next to nodata are treated as outlets. The report prints the number of raised cells and the maximum raise - a handy indicator of the source DEM quality.

The same smoothing checkbox is present in DEM download (2.01) for a quick path right at download time. The standalone tool 2.04 is needed when the terrain came not from 2.01 but from your own data.

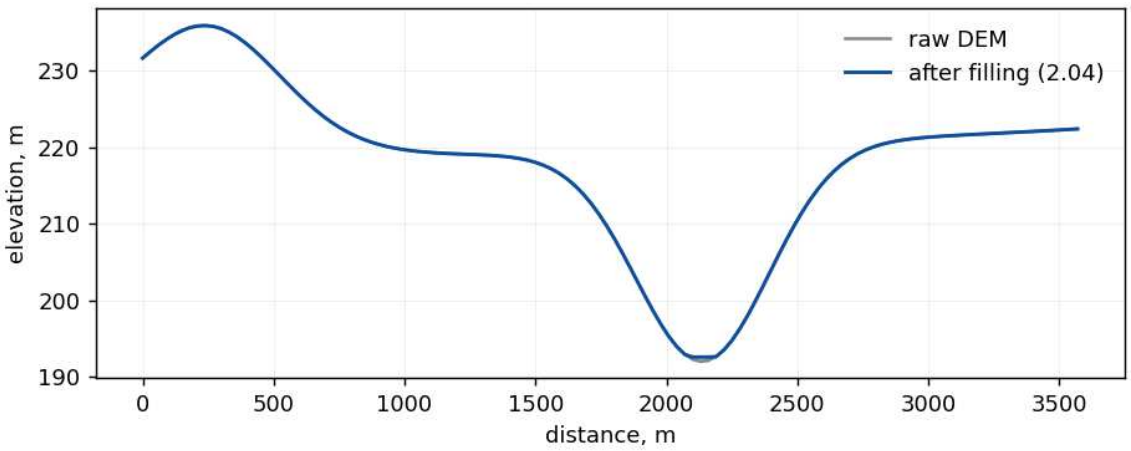



Figure 19: A profile through a depression: the raw surface and the filling result. The raised part is shaded.

## 2.05 Slope and aspect

Slope in degrees and aspect with the Horn 3×3 kernel, as in gdaldem. Aspect is the downslope azimuth in degrees from north clockwise, north 0, east 90. Flat cells get an aspect of -1 so they are not confused with north-facing ones. Nodata cells and their neighbors get nodata, the 3×3 kernel is not computed across gaps.

| Parameter | What it sets | Default / hint |
|---|---|---|
| Input DEM | Terrain raster. | - |
| Slope, degrees | Output: slope raster (Horn 3x3, as gdaldem). | - |
| Aspect, degrees | Output: downslope azimuth, flats = -1. | - |

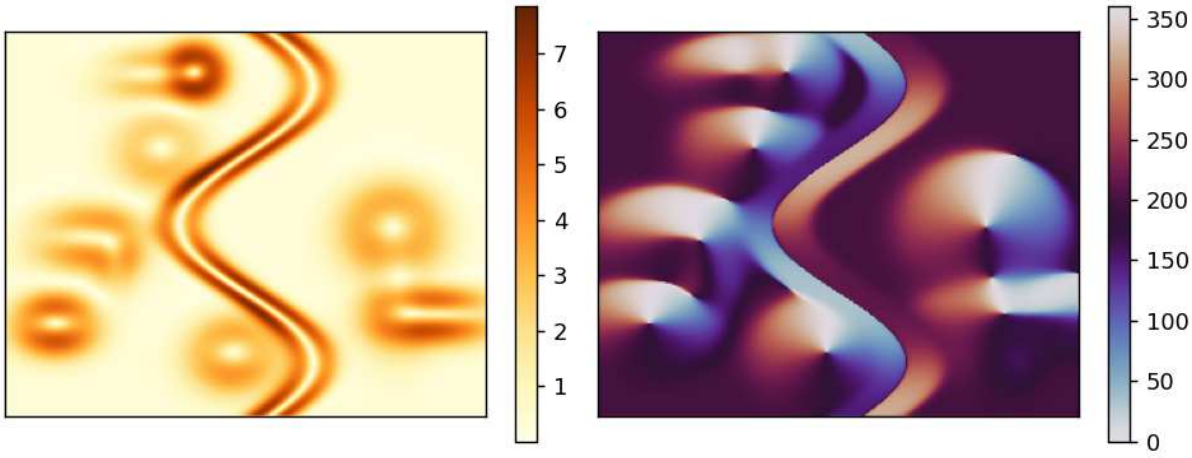



Figure 20: Slope (left) and aspect (right) of the demo relief. The cyclic aspect palette stitches 0 and 360 degrees.

## 2.06 Peaks and pits

Finds peaks - cells that are the highest in a square window of the given radius, with a drop over the window minimum at or above the threshold.

| Parameter | What it sets | Default / hint |
|---|---|---|
| Input DEM | Terrain raster. | - |
| Window radius, m | Suppression of secondary peaks within this radius. | 500 |

The output fields are **z** (the elevation) and **drop** (the drop over the window minimum). The layer compares against OSM peaks from 2.02, and matching the ele marks with the DEM elevations is one more quick data check.

## 2.07 Demo relief

A utility generator of synthetic terrain from a tilted plain, hills and a winding valley with a constant fall. The relief is deterministic by seed, and local depressions are left between the hills on purpose so the filling tool has something to show. All figures of this chapter are built on it.

| Parameter | What it sets | Default / hint |
|---|---|---|
| Width, cells | Demo grid size along X. | 300 |
| Height, cells | Demo grid size along Y. | 300 |
| Cell size, m | Demo grid step. | 30 |
| RNG seed (Adv.) | Reproducibility of the random relief. | 42 |
| Output CRS (metric) | Coordinate system of the demo raster. | EPSG:32640 |
| Compact int16 (adv.) | Integer output for demo shipping. | off |
| Demo relief | Output raster. | - |
| Gauge points (demo) | Three points near the thalwegs for tool 6.04. | created |
| Ditch traces (demo) | Two polylines across the thalwegs for tool 6.05. | created |
| Fraction of the area for the works (Adv.) | Size of the area under the design pad. | 0.4 |
| Pad elevation offset, m (Adv.) | Moves the balance into imported or exported soil. | 0 |
| Design surface (demo) | The pair to the relief for tool 2.12. | off |
| Work areas (demo) | Three polygons for the 2.12 breakdown. | off |

The tool is meant for teaching examples and for work without a network. Live data comes from 2.01. The **Compact int16** checkbox outputs the raster in whole meters for shipping demo fragments.

The fields of the source feature are carried into the output. Pick the ones you need in the advanced parameters, and the catchment will carry the river name, the gauge number or whatever else is there. Without them only a number remains in the output, and whose catchment it is has to be looked up in the source layer. If a name clashes, the incoming field gets the src_ prefix.

### A pair of surfaces for volumes

Two outputs, off by default, give a ready pair for tool **2.12 Cut and fill** - a design pad and work area polygons. The pad is horizontal, and outside the work area the natural relief remains.

The pad elevation is the mean of the relief inside the area, and that is not a matter of taste but the exact answer. Volume is a sum of differences multiplied by the cell area, and the net turns to zero exactly when the elevation equals the mean. The median splits cells in half, not cubic metres, and the balance does not close with it. So the demo comes out with a balance that closes. The elevation offset in the advanced parameters moves it into imported or exported soil, which checks all three verdicts of 2.12 in two runs.

Three demo gauge points and two demo ditch traces are produced along with the relief. The ditches are a hillside one above the strongest thalweg and a gutter further downslope. Both are polylines

with a bend, so that the rasterisation of a turning trace is exercised as well. The gauges are placed at the strongest thalwegs and spread across the grid. Each is deliberately shifted a few cells aside from the stream, so that the snapping in tool 6.04 can be seen bringing the gauge back. The points are deterministic from the same seed as the relief.

#### The gully and ravine network

The **Gully and ravine network** tick cuts thalwegs with steep sides into the relief, with tributaries entering at an acute angle. The cut deepens downstream, as in a real gully, shallow at the head and deep at the mouth.

This mode exists for validation sets. A narrow cut between adjacent contours is the hardest place for any interpolation. The contours barely describe it, and a surface built from them shaves the gully off. On a profile across it this shows at once, and tools 2.08 and 2.09 put a number on it.

#### Where the demo lands

When no extent is given, the demo lands next to the project layers. Their combined extent is brought into the chosen coordinate system and used as the placement. In an empty project there is nothing to go by, so the demo is created at a conventional spot, always the same one, so that the examples in the manual reproduce. In a local coordinate system, a mine grid for instance, that conventional spot turns out far from the working data.

So in an empty project set the **Where to place it (extent)** parameter. The relief will land there, and the grid size is computed from the extent and the cell size.

# 2. Topography: diagnostics and repair

## 2.08 Split contours for validation

The tool splits a set of contours into two. One is used to build the relief, the other to check the result. Its purpose is to produce a numerical estimate fit for a report.

#### Why split at all

The simplest way to check a relief looks like this - take the source contours, read the built DEM at their points and compute the residual. The result will look good, but it measures something other than it seems. The interpolator has seen those points, they were the input data, and it is almost bound to reproduce them. That is a check of input reproduction, not of predictive accuracy.

A real check needs data the model has not seen. This tool creates them.

#### Why the split is by elevation, not by feature

Holding out individual pieces of a single contour is pointless, the neighbouring pieces of the same level give the answer away and the check comes out flattering. So a held-out level disappears entirely, with all of its pieces. The interpolator can restore it only from the neighbouring levels, and that is what prediction means.

The extreme levels of the set always stay in the building set. Beyond the range of the set the interpolator extrapolates, and a residual there would measure something other than what the check is for.

#### Outputs

Two layers, **Contours for building** and **Contours for validation**. Both get a **hold** field, zero for building and one for validation. The **Contour residuals against the DEM** tool recognises this field

by itself and prints the two figures separately. Feeding it the combined set is easier than running it twice.

### Working order

Split the contours, build the relief from the building set, with **Topo2Raster** for instance, measure the residual over both sets at once and compare the two figures.

### Parameters

| Parameter | What it sets | Default / advice |
|---|---|---|
| Contours (lines) | The source set. | - |
| Elevation field | Numeric field of the contour elevation. | - |
| Hold out every Nth elevation | Thinning step over levels. 4 sends about a quarter to validation. | 4 |
| Selection offset (Adv.) | Shifts the choice of levels so the check can be run over different subsets. | 0 |

## 2.09 Contour residuals against the DEM

The tool measures how well the built DEM reproduces the source contours. At points along a contour the raster value is taken and compared with the elevation of the contour itself. The residual is positive where the DEM lies below the contour.

### The numbers it reports

The mean is the bias, and a non-zero value means the surface as a whole is shifted in elevation. SD and RMSE are the spread. The median absolute value resists single outliers, the maximum shows the worst place on the area.

Separately it reports the share of points that miss by more than half the contour interval. This is a practical quantity. If it is noticeable, a contour drawn from such a DEM will not sit where the original one was, and the map stops agreeing with itself.

The tool detects the contour interval from the set of elevations as the smallest difference between adjacent levels. If the set is assembled from different sources, set the interval by hand.

### Two figures instead of one

If the layer carries a **hold** field from tool 2.08, the residual is computed separately for building and for validation, and both lines go to the log. The first says how the model reproduces the input, the second how it predicts. The second is always worse than the first, and that is normal. What matters is the gap between them. A large gap means the model memorises well and generalises poorly, and the shape of the relief between the contours is restored wrongly. Without a **hold** field the tool reports a single figure and warns in the log that this is reproduction of the input.

### The report

The HTML report has four parts. These are a table of numbers per set, a histogram of residuals with the half-interval bounds marked, the spread of the mean residual by elevation, and a short reading of the result. The reading answers four questions. Is there a systematic bias, is the spread large relative to the interval, is the share of misses noticeable, and is there a gap between reproduction and prediction.

Look closely at the breakdown by elevation. If the residual grows towards the summits or towards the thalwegs, it speaks of forms being cut off rather than of random noise.

### The point layer

A point layer of residuals is produced with the fields **fid_src** (the source contour feature), **elev** (elevation), **z_dem** (the DEM value), **resid** and **abs_resid**, **hold**. Colour it by **resid** with a diverging ramp and the places where the surface systematically runs low or high show up at once, without any statistics.

### Parameters

| Parameter | What it sets | Default / advice |
|---|---|---|
| Contours (lines) | The layer with elevations. A combined set with a hold field works. | - |
| Elevation field | Numeric elevation field. | - |
| DEM (the built relief) | The raster being checked. | - |
| DEM band (Adv.) | Raster band. | 1 |
| Sampling step along the contour | 0 means vertices only. Above zero it adds points on long straight legs. | 0 |
| Raster sampling (Adv.) | Bilinear or nearest. Bilinear is fairer for a smooth surface. | bilinear |
| Contour interval (Adv.) | 0 means detect from the elevations. | 0 |
| Residual points | The output point layer. | created |
| Residual report (HTML) | Table, histogram, reading. | created |

## 2.10 Terracing check of a DEM

The tool looks for terracing, the characteristic defect of a relief built from contours. The slope goes in steps, with a bench near a contour level and an abrupt drop between levels. On a hillshade it shows as a system of concentric benches, on a profile as steps. Slopes in such a relief are wrong, and flow computations over it break down.

### Two independent signs

**Vertical curvature** is the second derivative along the slope. On a stepped surface it spikes at the drops and is close to zero on the benches, and the whole picture repeats the pattern of the contours in bands. The curvature raster is produced as an output, and terracing is visible on it by eye, without any statistics.

**Attraction of elevations to the levels** is a direct check on the values themselves, without derivatives. On an undistorted surface the elevations between adjacent levels are spread more or less evenly. The share of cells in a narrow band around a level is then close to the width of that band, and the ratio comes out near one. On a terraced surface the elevations gather at the levels and the ratio grows.

The ratio reads like this. Near one means no signs, one and a half is a reason to look at the curvature raster by eye, two and above means terracing.

The two signs are worth looking at together. Curvature is visually convincing, but its spikes also come from real landforms, from breaks of slope for instance. Attraction to the levels gives a number but does not show where the trouble is. Together they answer both “is there any” and “where”.

### The contour interval

The tool computes the phase of an elevation within the interval, so the interval is required. It can be set by hand or taken from a contour layer, and then both the interval and the base elevation come from the real set. The second way is safer if the elevations do not start at a round number.

### Flat areas are kept out of the count

Cells with a near-zero slope are excluded from the attraction index. The threshold is a fraction of the interval, so a cell is ignored when the elevation drop across it is smaller than a hundredth of the interval. At an interval of 0.5 m that is 5 mm per cell. The **Ignore cells with a drop below, fraction of the interval** parameter sits in the advanced ones, zero turns it off.

Without this screening the index lies on any real matrix that holds a water body. The elevations of a water surface stand still and all fall into one phase within the interval, which skews the distribution, the index is understated and reports a false result. On a matrix where a reservoir covered 44.5 percent of the area the index without screening gave 0.50 instead of 0.97, while over land alone it came out at 1.00.

The share of excluded cells is printed to the log and goes into the report. This number matters. If the tool has thrown away half the area, you should know about it rather than wonder why the figure changed.

### The report

The HTML report holds a table of numbers and a histogram of the phase, that is, the distribution of elevations within the interval. A flat histogram means there is no terracing. A peak at zero means the elevations gather at the levels and the surface is stepped.

### How to remove it

Terracing is removed by breaklines (thalwegs, breaks of slope, ridges) and by denser source data, not by smoothing. Smoothing removes the steps together with the landforms, and the numbers improve while the map gets worse.

### Parameters

| Parameter | What it sets | Default / advice |
|---|---|---|
| DEM (the relief being checked) | The raster being checked. | - |
| DEM band (Adv.) | Raster band. | 1 |
| Contour interval | 0 means take it from a contour layer. | 0 |
| Contours to detect the interval | Optional layer, the interval and base elevation are taken from it. | - |
| Contour elevation field | Numeric elevation field. | - |
| Base elevation of the levels (Adv.) | The datum of the levels when the interval is set by hand. | 0 |
| Half-width of the band around a level (Adv.) | Fraction of the interval. The expected share equals twice this value. | 0.1 |
| Vertical curvature | The output raster. | created |
| Terracing report (HTML) | Table, phase histogram, reading. | created |

## 2.11 Remove steps (clamped smoothing)

The tool treats terracing, that is it removes the steps from slopes without moving the contours themselves.

### How that is possible

The surface is smoothed iteratively, but every point is forbidden to move away from its original value by more than a set fraction of the interval. By default that is half the interval, which is exactly the quantisation error, since a surface built from contours of that spacing is known no better anyway.

Two properties follow. The method cannot invent forms finer than the source interval allows, because the amplitude of the correction is bounded from above. And it cannot throw a point across a contour, since the shift is smaller than half the step between levels. On a reference stepped relief the attraction index falls from 5.00 to 1.13, and the mean error against the true surface from 1.25 m to 0.12 m.

### The treatment checks itself

The index of attraction to the levels is computed before and after the correction, and both figures go to the log together with the largest actual shift of the surface. If the index stays above two after the treatment, the tool says outright that the steps have not gone and advises adding iterations or checking the interval.

The HTML report holds a table of numbers before and after, two phase histograms side by side and a reading. It is a ready document for a client or a reviewer, showing what was, what became and at what cost.

### What the tool does not do

It does not bring back what is not in the data. If a narrow cut was shaved off when the relief was built, smoothing will not restore it, the correction is bounded by half the interval and the cut is deeper. Nor does it help on a water surface, where a mask is needed rather than smoothing.

### Parameters

| Parameter | What it sets | Default / advice |
|---|---|---|
| DEM with steps | The raster being treated. | - |
| DEM band (Adv.) | Raster band. | 1 |
| Contour interval | 0 means take it from a contour layer. | 0 |
| Contours for the interval | Optional layer the step is taken from. | - |
| Contour elevation field | Numeric elevation field. | - |
| Smoothing iterations | More means smoother. Fifty is almost always enough. | 50 |
| Allowed shift, fraction of the interval (Adv.) | Zero forbids any correction. | 0.5 |
| Relief without steps | The output raster. | created |
| Before and after report (HTML) | Table, histograms, reading. | created |

# 2. Topography (continued)

## 2.12 Cut and fill (earthwork volumes)

The tool computes earthwork volumes between two surfaces - what was filled, what was removed and whether the balance closes. It is needed wherever there is a before-and-after survey or a design surface, be it a pad to be graded, a spoil heap, an open pit, the silting of a pond or ground subsidence.

The formula is as plain as can be - the difference of elevations per cell multiplied by the cell area. All the difficulty lies not in it but around it, which is what the rest of this section is about.

### The sign and the reference surface

The difference is taken as “after minus before”. A positive difference is fill, material was added. A negative one is cut, material was removed. ArcGIS uses the opposite sign in its Cut/Fill tool, and that has to be allowed for when cross-checking figures.

The reference surface is given either as a raster or, when there is none, as a single elevation. An elevation is handy for a pad to be graded and for counting from a water line, and there is no need to make a raster of constant height for that.

### Bringing both to one grid

Two matrices almost never sit on the same grid. The first surface owns the grid and the second is resampled onto it bilinearly. Nearest neighbour will not do here, it brings back the very steps that the terracing check looks for. Beyond the data no volume is computed and elevations are not extrapolated.

The log prints the origin, the step and the cell count of both matrices, and whether any resampling took place. Reconciling volumes with another program starts from those lines, as the next section describes.

### Why figures differ from other programs

Almost never because of the formula. Bilinear resampling preserves the volume, the bilinear weights sum to one, so a grid shift on its own changes nothing.

The difference comes from which cells took part - a slightly different clip, another mask, half a step at the boundary of a work area. So when reconciling with Civil, Credo or any other program, start with the grid description from the log and the number of cells counted. Compare the volumes only after that. Nine times out of ten that explains the discrepancy.

### The dead band

Two surfaces produced by different means always rustle by centimetres. Without a cut-off all that background lands in fill or in cut and inflates both figures without changing the net.

The dead band is set in metres, and cells whose absolute difference is smaller count as unchanged. By default there is no cut-off. Set it deliberately and state it in the report, it appears in the statement as a separate row precisely for that.

### Work areas and the balance

Work area polygons are counted separately, each with its own figures. The area name field may be numeric or text. The per-area table has a total row, and it must agree with the header. That is a simple and reliable check that the rasterisation of the areas neither lost nor duplicated cells.

The balance verdict looks at the share of the net in the turnover, not at its magnitude. A hundred cubic metres of imbalance against a turnover of a hundred thousand is a balance. Against a turnover

of two hundred it is hauling away half of it. The tolerance is set as a fraction of the turnover, five percent by default.

### Outputs

A difference raster in metres, positive is fill. An HTML statement comes with the total, the per-area breakdown and the grid description. The **Work areas with volumes** layer holds the same polygons that were supplied, with the volumes added to the attributes. These are `fill_vol`, `cut_vol`, `net_vol`, the areas, the largest elevations, the cell count and the verdict. The statement is for approval, while labelling the areas straight on the map is only possible from attributes, so it is a separate output.

The **Clip the difference raster to the work areas** checkbox blanks the difference outside the outline of the works. Outside it the difference is made of survey noise, and a colour ramp stretched over it hides the very thing the raster is looked at for. Clipping does not affect the figures, the statistics are computed before it, and both totals stay in the statement, over the whole area and per work area.

The line of zero works is not built by a separate tool, and that is deliberate. It is the zero contour over the difference raster, built with **1.04 Isolines from raster**.

### Parameters

| Parameter | What it sets | Default / advice |
|---|---|---|
| Surface “after” | The design surface or the new survey. Owns the grid. | required |
| Surface “before” | The original survey. Empty - compare against an elevation. | empty |
| Reference elevation, m | Used when there is no “before” raster. | 0 |
| Work areas (polygons) | Splits the statement by area. | empty |
| Area name field | Numeric or text. | empty |
| Dead band in elevation, m | Cuts the background noise off. | 0 |
| Clip the difference raster to the work areas | Blanks the difference outside the outline. Does not affect the figures. | off |
| Work areas with volumes (polygons) | The area polygons with volumes in the attributes. | created |
| Balance tolerance (Adv.) | Fraction of the turnover for the verdict. | 0.05 |
| Difference raster | The output raster, fill is positive. | created |
| Earthwork volume report (HTML) | The report with the total and the areas. | created |

### A quick check on the demo

**2.07 Demo relief** produces a ready pair: switch on the **Design surface (demo)** and **Work areas (demo)** outputs. Feed the pad as “after”, the relief as “before” and the areas as polygons. Fill and cut must come out equal figure for figure, the net zero and the verdict about a balance that closes. The pad elevation offset in the advanced parameters of 2.07 moves the balance into imported or exported soil, so all three verdicts are checked in two runs.

Fill matching cut at a zero offset is also an independent check of the tool itself. An error in the formula or in the grid alignment would show up right here.

**Scope**

The volume is geometric. Bulking, compaction and layered soils are not applied here, that is the designer's work.

The accuracy at the boundary of the works is set by the cell size. The volume along the outline is always biased by about half a cell per running metre of boundary. On a small outline with a coarse cell that bias can exceed the volume itself, so compute small pads on a grid commensurate with their size.

The tool does not build side slopes, benches or layered statements, and it does not fit a design surface onto the terrain. That is the work of computer-aided design systems.

## 2.13 Crest and toe candidates

Finds the places where the slope changes fastest and traces them into lines - the crests and toes of benches, pit walls, the edges of fills and cuts.

| Parameter | What it sets | Default / advice |
|---|---|---|
| Input DEM | A ground raster, not a DSM. The cell follows the survey density: 20-25 cm at 20 points per square metre. | - |
| Minimum drop, m | The noise cut-off: a ridge of the evidence must gain this relief drop within the probe base. | 0.5 |
| Minimum line length, cells | Removes fragments. | 10 |
| Drop probe base, cells (Adv.) | The half-width of the window in which the drop across the line is measured. | 8 |

**The evidence of a break is the gradient of the slope, not the slope.** On an even face, however steep, the slope is constant and the evidence is small. The evidence is large where the slope changes, that is on the crest itself and on the toe itself. The sign of the profile curvature splits the lines found. A convex break gives a crest, a concave one a toe, and this goes into the **kind** field as brow and toe.

**A break without a drop is not a break.** A ridge of the evidence is discarded before any thresholds if its neighbourhood holds no relief drop of the given size. Centimetre noise of a dense survey never gains it, the neighbourhood of a real crest always does. This is the main filter, and it is physical rather than statistical. The thresholds of the internal mathematics are fitted automatically and are not exposed.

**The drop is measured within the probe base, not across the whole width of the bench.** This is the first of the common mistakes. On a ten-metre bench with a three-cell base the **drop** field will read three metres rather than ten. The base is 8 cells to each side by default, which on a metre cell is ±8 m and covers a quarry bench. The width of the face is known to you and not to the tool, so set the base by it. The base is printed to the log both in cells and in metres.

The output fields are **kind**, **drop** (m), **length_m** and **slope_deg** (the mean side slope). The layer arrives coloured by the drop, crests warm and toes cool. Classes appear only when the drop really varies. On a pit with equal benches the spread is a few per cent, splitting it into classes would cut noise, and one class per kind remains there.

**The significance threshold stays with the human.** A formal definition of a crest does not exist, recognition rests on the surveyor's experience. What exists is the drop you are prepared to call a bench, and it differs on a quarry, a road embankment and a river bank. The tool therefore deliberately returns more than needed, together with the numbers for the selection, and the percentiles of the drop go to the log as a ready hint where to cut. Select with a layer filter over the drop field while watching the map, nothing has to be recomputed.

## 2.14 Crests and toes into work

Turns crest and toe lines into working structural lines. It takes the elevations off the DEM, assembles crest-toe forms and lays them into two layers.

**There are two scenarios, and the second matters more.** The first is obvious, the lines came from 2.13 over a dense survey. In the second the lines are already in a topographic deliverable, where crests are described as a matter of course and carry their own classifier codes. No detector is needed there, only what this tool does. The kind of a line is coded, but which crest goes with which toe is not, and the link has to be assembled. There are many such deliverables and few dense surveys.

| Parameter | What it sets | Default / advice |
|---|---|---|
| Candidates (output of 2.13) | A line layer with a kind field. A layer of crests and toes from a topographic deliverable will do as well. | - |
| DEM | The same one supplied to 2.13. | optional |
| Kind field | The field holding brow and toe. | kind |
| Drop cut-off, m | Repeats the filter of 2.13 so that no separate filtered layer is needed. | 0 (take everything) |
| Descent path limit, m | Beyond this the descent stops and the crest goes to the unpaired. | 50 |
| Share of agreeing probes (Adv.) | Below this share a form is not assembled. | 0.4 |

**Forms are assembled by descending the slope, not by proximity.** From the probe vertices of a crest a descent follows the flow directions until a toe is met, and the toes vote. This is how a bench works physically, water from the crest runs down the face exactly to its toe. On a curved wall with narrow berms the nearest toe by distance often belongs to the neighbouring bench, so the choice by proximity errs where the descent is right. Keep the path limit close to the width of the face. Too large a limit lets a crest run to a foreign toe and form a plausible false pair with it.

**The DEM is optional, and that matters more than it seems.** The descent answers a single question - which way is down. When the lines carry elevations of their own (the output of 2.16), the answer is already in the data. No terrain is needed, and the toe is the nearest line lying below the crest.

Without this the topographic scenario closed into a circle. To build the terrain from forms you need pairs, and to assemble pairs by descent you need terrain, which does not exist yet and is the very thing being built. A draft pass of 2.03 over the contours alone had to be made, knowing it was wrong inside the quarry. Now the order is straight. 2.16 gives the elevations, 2.14 without a DEM assembles the pairs, 2.03 builds the surface.

With a DEM the former descent is used. On a curved wall with narrow berms it is more accurate, because there the nearest toe by distance often belongs to the neighbouring bench.

**A form is one toe with a set of crests at it.** The tracing cuts a long crest into pieces, a ramp breaks the outline, the evidence is interrupted on gentle stretches. All the pieces descend to the same toe, so the result is grouped by the toe rather than written out as pairs. The toe goes into the output once, and its crests share the **link** field. This is the same view as in building a surface between structural lines, where the sides are sets rather than single lines.

**Unpaired lines do not vanish silently.** A third layer gathers them with the reason in an attribute. Either the descent did not reach a toe, when the path limit is too small or the crest is false, or it scattered over different toes, when the line has glued two benches together. The lowest share of agreeing probes is printed to the log and points at such a gluing.

The **Top** and **Bottom** outputs are LineStringZ with the kind and link fields, the elevations taken off the DEM into the geometry. These are ready inputs for surface building and for an export as 3D lines into AutoCAD and Credo.

## 2.15 Example open pit (demo)

Builds a demo pit and, more importantly, the true structural lines for it. The raster-lines pair serves as a reference for 2.13, as an input for surface building and as a teaching example, all without closed data.

| Parameter | What it sets | Default / advice |
|---|---|---|
| Width, Height, cells | The size of the grid. | 400 x 300 |
| Cell size, m | The grid step. | 1 |
| RNG seed (Adv.) | Repeatability. One seed gives one and the same pit. | 7 |
| Output CRS | Metric. With a local project CRS choose the same one. | EPSG:32640 |
| Number of benches | How many benches the wall holds. | 3 |
| Bench height, m (Adv.) | The rise of a single bench. | 10 |
| Survey noise, m (Adv.) | An imitation of the error of a dense survey. | 0.03 |
| Dump, Converging ditch (Adv.) | Additional shapes. | on |
| Where to place (extent) | Places the demo, does not change its size. | by the project layers |

The composition of the terrain is chosen so that every shape tests its own side of the detector. The elliptical pit with benches and berms gives correct pairs. The ramp cutting through the benches gives a **genuine break** - on its arc the true lines stop, and the candidates must stop there too. The flat-top dump gives a closed pair. The converging perimeter ditch gives three lines meeting at a point, the case where the weight of a surface between lines degenerates while the elevation stays correct.

**The extent places the demo and does not change its size.** The shapes of the pit are physical, a 10 m bench on a 7 m face. A pit stretched to kilometres turns into a flat surface with nothing for the detector to find, so the size is set by the width and the height alone. And separately about the coordinate system. If the project uses a local or an unknown one, choose the same in the output CRS. Reprojecting an extent out of an unknown system into UTM gives a wrong result, and the demo lands far from where it is needed.

The second output, the **true lines**, carries the kind (brow, toe, thalweg) and link fields, with the elevations in the Z geometry. Against it the completeness and the precision of the detector are measured as numbers rather than by eye. Put the candidates of 2.13 over the true lines and see where they diverge.

## 2.16 Profiling of slopes

Gives mute lines a profile from the contours that adjoin them.

| Parameter | What it sets | Default / advice |
|---|---|---|
| Lines without elevations | Crests and toes from a deliverable or from 2.14. | - |
| Contours | The contour layer of the same deliverable. | - |

| Parameter | What it sets | Default / advice |
|---|---|---|
| Contour elevation field | The elevation attribute. Empty - Z of the vertices. | ELEV |
| Adjoining tolerance, m | How far a contour end may stop short of the line. | 0.5 |

**Where the elevations come from.** A crest in a topographic drawing carries no elevation of its own, and for a long time that looked like a dead end. But the standard requires contours to be brought up to the object line **with node points formed**, and every such point carries the elevation of its contour. From them comes the profile of the whole line - a varying elevation out of the data itself rather than one value per object.

An important correction. A contour **does not run along the crest**. Contours are cut by the slope, and the single one that lies along a crest does so only when its level happens to match the crest, that is by chance. What works is the adjoining, not the coincidence.

**What counts as a meeting.** A through intersection of the line with a contour, and a contour end within the tolerance. Between the meetings the elevation is interpolated along the arc of the line, beyond the extreme ones it is held constant. Extrapolating the gradient along a crest is unsafe, it often changes at the turns.

**The limits of the method.** The method works exactly as far as the deliverable is topologically consistent. If the contours are not brought up to the line, the line stays mute and goes into a separate layer with the reason. The number of support points is written into the `n_samples` attribute, and into the log as a median and a minimum. One point means a constant elevation along the whole line, and that is visible at once.

**Place in the pipeline.** The output is LineStringZ, a ready form side for the **Top of forms** and **Bottom of forms** inputs of 2.03. Together with 2.14, which assembles the pairs and fills the link field, this closes the topographic scenario. Areal quarries, cuts, fills and dumps, which have no contours inside them described by the standard, receive a surface out of crests and toes alone.

#### Spot heights as a second source

A plan coming out of AutoCAD often carries no Z on the breaks and slopes. The surveyor sets the elevations as separate points and leaves the lines flat, so spot heights are taken as a second source.

A point does not cross the line, so a meeting is a matter of proximity. A point no farther than the abutment tolerance gives a control elevation at its arc position. From there everything goes as with contours, the profile is built over the whole series of control points.

#### Where the vertices go

An elevation lives in a vertex while meetings happen wherever they happen, so vertices have to be added. The rule is as follows. If a vertex already stands near a meeting, closer than the snapping tolerance, no new one is added and the meeting itself moves onto the existing vertex. The elevation lands in it exactly, and the line gains no extra nodes. If the nearest vertex is farther away, a vertex is inserted at the meeting.

The insertions are thinned by the smallest step, two in a row are no closer than the given distance. On dense contours the line would otherwise grow more nodes than the relief needs. A control point that gets no vertex is not lost, the profile is computed over the whole series, and the neighbouring vertices are drawn towards it.

## 2.17 Remove bridges from a DEM

Removes bridges from a DEM and puts the river surface in their place. A satellite surface model takes the height from the top of the deck, so a bridge lies across the river like a dam. Flow over

such a model stops at the embankment, a false lake spreads upstream of the bridge, and the river network from 6.02 breaks at the crossing.

| Parameter | What it sets | Default / hint |
|---|---|---|
| Input DEM | Terrain in metres in a metric coordinate system. | - |
| Bridges: deck outlines or axes | Polygons along the deck or lines along the bridge axis. Both may share one layer. | - |
| Bridge width for axes, m | The width an axis is widened to. Does not affect outlines. | 20 |
| Margin around the bridge, cells | By how many cells a bridge is widened before recomputing. | 1 |
| DEM without bridges | Output float32 raster, a layer in the Topography group. | - |

**How the surface is recomputed.** The cells under a bridge are recomputed from the cells around it, by spans. Each cell takes a linear interpolation between the nearest surrounding cells along the row, the column and the two diagonals. Among the directions with the shortest spans the lowest value is chosen. A bridge is narrow along the river and long along the road, so the short spans of a cell under the deck run across the road. A span from the water upstream to the water downstream gives the water level with the same fall as the river, and a span that touches a bank gives a higher one. The low one is taken, and the surface settles at the water level. If the river runs at an angle to the grid, the shortest span may hit the valley side. Then a span up to twice as long is also considered, but only if both of its ends stand at the bottom of the surroundings, that is on the water. This finds the channel without cutting into the valley side. At the abutments all short spans run along the bank, and there the surface meets the bank.

The surface is only lowered. A cell that is already below the interpolation, such as water seen between spans, stays as it was. Cells outside the bridge outline do not change.

**Margin.** The edge of the deck is blurred over a cell in a surface model. Without a margin a cell on the slope of the deck becomes the surroundings, and the water under the bridge settles higher than the real one. On a DEM with a 30 m cell a bridge covers one or two cells, and a margin of one cell is usually needed. Approach embankments distort the terrain as much as the span itself, and they are outlined together with the bridge.

**How to outline.** The outline follows the deck from bank to bank, as a narrow strip across the river. There is no need to outline a bridge with a wide square: the wider the patch along the river, the longer the span from water to water and the more of the nearby banks gets into the surface under the bridge. If bridges already exist as lines, for example from OpenStreetMap, they are given as axes with a width.

**Log.** For each bridge the log reports how many cells were recomputed, how far the surface was lowered and the heights of the surroundings. A bridge under which nothing was lowered gets a warning, usually the outline is not where it belongs.

The tool does not fill depressions. For flow calculations the corrected DEM goes into **2.04 Terrain preparation**, then into 6.01 and 6.02. The easiest check is the river network: it no longer breaks at the crossing.

# 3. Additional analysis tools

## 3.01 Categorical indicator kriging

The **Categorical indicator kriging** tool builds a probability map from a categorical field - mineral type, lithotype, any text class. Unlike ordinary kriging, which interpolates a number, here it esti-

mates how likely each class is at every point of the area. This is what you need where the type matters rather than the magnitude - where to expect replacement, where the seam composition changes, where the boundary between varieties runs.

Parameters:

| Parameter | What it sets | Default / advice |
|---|---|---|
| Point layer | Source points. | - |
| Categorical field (class) | The class field (mineral type, lithotype). Empty and NULL are excluded. | - |
| Search radius, min/max points, cell size, extent | Search and grid - as in “2D Kriging”. | as in “2D Kriging” |
| Nugget share | The share of the nugget in the fitted variogram, from 0 to 1. Empty means as fitted. | empty |
| Class probabilities (multiband) | Raster: one band per class, the class name in the band description. | - |
| Zone map (most likely class) | Raster of the most-likely class code; the code mapping goes to the Log. | - |
| Confidence (max probability) | Raster of the maximum probability: where the class is firm, where it is contested. | optional |
| Probability levels | Levels for the vector boundaries, fractions from 0 to 1. | 0.25 0.5 0.75 |
| Class for the contours | The class name exactly as in the class field. Empty = all classes. | empty |
| Bicubic smoothing of the boundaries (adv.) | Grid densification before contouring, as in 1.04. | ×2 |
| Boundary rounding (Chaikin), iterations (adv.) | An extra light rounding. | 2 |
| Probability level boundaries (lines) | Level lines carrying the class and the level. | optional |
| Probability bands (polygons) | Bands between the levels with a ready colouring. | optional |

Clipping by a mask is the same as in 1.01 - a polygon from the project or the convex hull of the points with a buffer. The mask is applied to all three outputs, the probabilities, the zones and the confidence.

**How it is computed**

Coding the classes as numbers 1, 2, 3 and interpolating that code is not allowed. Categories have no order, class 3 is not “farther” than class 1, and a mean between them is meaningless. So the tool takes the indicator route. For each class an indicator is built - one where the borehole is of that class, zero everywhere else. Each indicator is kriged separately by ordinary kriging, like an ordinary field, and yields a surface from zero to one, which is the class probability. The indicator variogram is fitted automatically with a spherical model from the experimental one.

Separate indicators do not sum to exactly one and may go slightly out of range, a known property of the method. So the estimate of each class is clipped to zero-one, and then the class probabilities are normalised so that in every cell they sum to one.

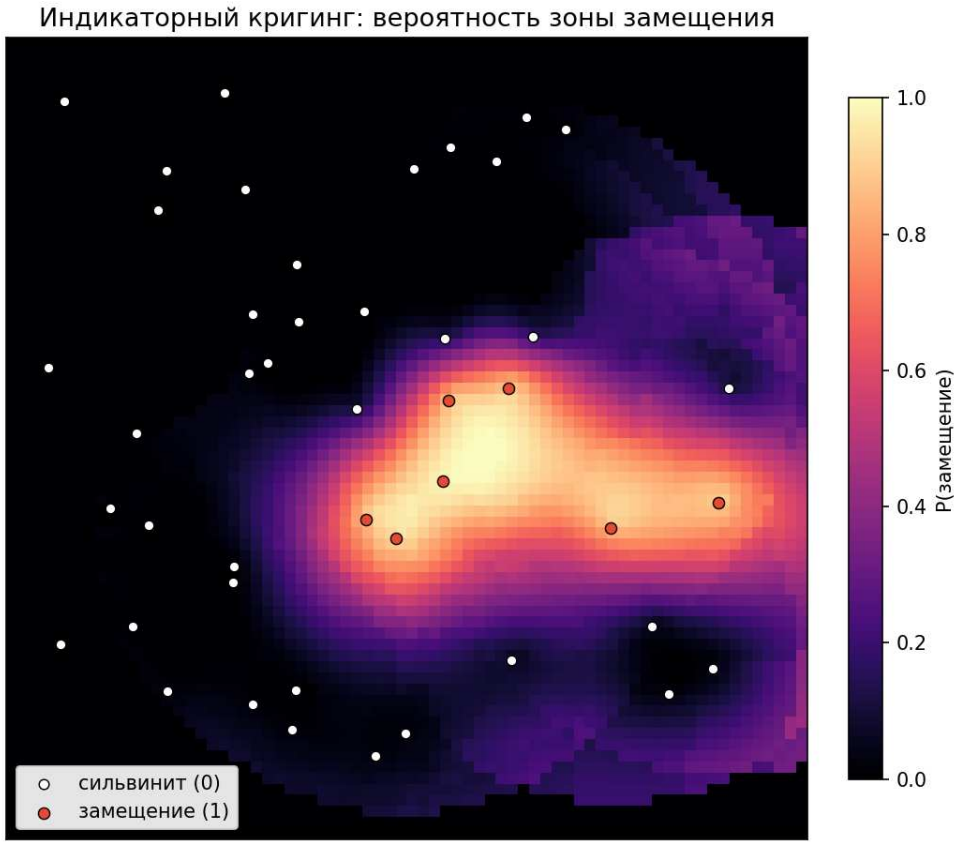



Figure 21: Indicator kriging on synthetics: categorised wells (red - replacement, white - sylvinite) turn into a class-probability map. A 0.5 threshold cuts the domain outline from it.

### When a borehole misses its own zone

The complaint is a common one. A borehole of the hazardous class is drawn outside the hazardous zone. This is not a failure of the fit but a property of the nugget, and it has to be understood before touching the other parameters.

Exactly at the measured point the kriging is exact whatever the nugget, and the estimate at a node that lands on the collar equals the indicator itself. But a grid node almost never lands on a collar. With a non-zero nugget the surface has a discontinuity around the point, and a few metres away the estimate already drops to the local mean. Take a synthetic example with a nugget of 0.5. Five metres from a lone borehole of the hazardous class the probability falls to 0.47, below the 0.5 threshold, while with the nugget set to zero it stays at 0.97.

Hence the **Nugget share** parameter. An empty field leaves the automatic fit as it is. Zero makes the surface smooth next to the data, and a borehole keeps its class in its own cell. The total variance is preserved when the share is moved, only the smoothness changes, so the scale of the probabilities does not drift. The fitted shares per class are printed to the Log, and it makes sense to look at them first. A large nugget means the classes are mixed at short distances, and that is a meaningful fact about the data rather than an obstacle.

The nugget should be zeroed with open eyes. It is not an invention of the fit but a measure of how much neighbouring boreholes disagree on the class. Zero means a decision to treat every measurement as exact and binding. For hazard maps that is often right, because the cost of a miss and the cost of a false alarm are not symmetric, but the map becomes harsher and patchier afterwards.

The second thing the hit depends on is the cell. The estimate is computed at the cell centre, and if boreholes of different classes fall into one cell, no nugget will separate them. Set the cell finer than the spacing between neighbouring boreholes. The third is the polygon boundary. The probability bands are built from the raster with smoothing and rounding, and both move the level line slightly relative to the cells. If a point is outside a band by a hair, check this first by turning both smoothings off.

### What you get

Three results. A multiband probability raster, one band per class, the class name written into the band description. A zone map, the code of the most likely class in the cell, with the code to class mapping printed to the log. An optional confidence raster, the maximum probability in the cell, which shows where the class is firm and where zones compete and the boundary runs.

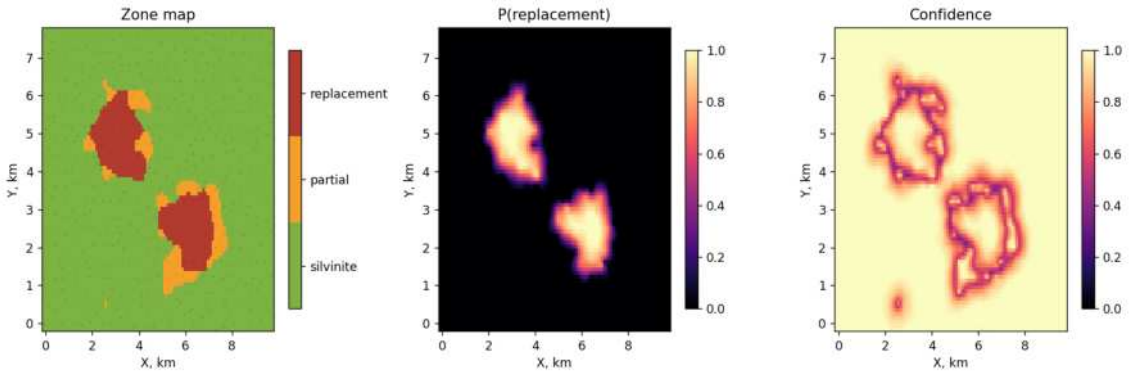



Figure 22: Categorical indicator kriging result: a map of the most likely mineral type, a silvinite background with replacement spots, boreholes drawn on top.

Another virtue shows itself on data with outliers. Ordinary kriging has to interpolate a magnitude, and a single anomalous sample breaks the variogram - a pair of nearby points with incomparable values lifts the nugget, and the map degenerates into the mean. An indicator works not with the

magnitude but with the fact of belonging to a class, so an anomalous value in it is indistinguishable from any other on the same side of the threshold. Where the parameter changes abruptly and irregularly and the question is binary in essence, the indicator path is more reliable than direct interpolation.

The categorical approach is convenient because it needs no boundary drawn in advance. There is no need to decide whether partial replacement counts as dangerous. All types are mapped as they are, and the required combination of classes is assembled later from the probabilities. Rare classes with few boreholes give a noisy variogram, the tool warns about this in the log, so read the probability of such a class with caution.

To learn the tool without real data, switch on **Add a categorical mineral-type field** in **Example wells (demo)**. A mintype field is added to the layer with a silvinite background and replacement spots after a mine, ready to run the tool on.

Each class indicator is kriged toward the proportion of that class in the wells, so far from the data the probability tends to that proportion rather than to zero. This matters for narrow elongated replacement zones, whose variogram range is often shorter than the well spacing. Before 5.13.40 the indicator was kriged toward zero when no weights were given, such a class vanished between wells, and replacement zones came out two to three times smaller than they are.

With an uneven network set the optional **wt** weight field from tool **1.05 Declustering**. The class proportion is then weighted, and far from the data the probability tends to the representative proportion rather than to the proportion in the cluster.

#### Vector boundaries from the probabilities

The zone map answers the question of who wins in a cell, and that is often not enough. Planning needs the transition band instead - where the class is firm, where it is contested, where the other class is firm. Two optional vector outputs give exactly that, both off by default.

The levels are set by a parameter, by default 0.25, 0.5 and 0.75, which gives four bands - firmly no, two contested ones and firmly yes. The lines carry the class and the level. The polygons carry the class and the band bounds, in the **P_MIN** and **P_MAX** fields. The **band** field holds a ready label of the form “0.25 - 0.5”. That label colours the layer by categories from green to red straight away, with no legend to set up by hand.

One band may arrive as several features if it is split into separate patches of area. This is normal and convenient, the areas are computed patch by patch. The legend still has exactly as many rows as there are bands, because the colouring follows the band rather than one of its bounds.

The boundaries are built from the probability channel rather than from the zone map, and this is significant. The zone map holds only the winner in a cell, and the position of the boundary inside the cell is already lost there. A contour of such a map therefore runs in steps along the cell edges. Smoothing that stepping means inventing the position of the boundary. The probability field keeps this information, and a contour of it falls exactly where the model itself puts it.

With two classes the level 0.5 coincides with the zone boundary, a class wins exactly where its probability exceeds one half. A single level of 0.5 therefore gives the usual binary map, only with a smooth boundary instead of a stepped one. With three or more classes these are different things, because a class can win with 0.4. What is built here is the probability of being that class, not the boundary of the winner.

With two classes enter the one you care about into **Class for the contours**. The probabilities complement each other to one, so the second set would be a mirror duplicate of the first.

## 3.02 External Drift Kriging

The **External Drift Kriging** tool estimates a field from points when that field is systematically related to a quantity already known everywhere as a raster. Such a raster is called the drift. It can be the structural surface of an adjacent seam, a coarse regional model, a surface built on a sparse

grid, or a seismic attribute. Ordinary kriging sees only the wells themselves. Here knowledge of the shape of the field between them is added, and the estimate leans on that shape where there are no wells.

The tool sits in the **Additional analysis tools** group and computes with the same kriging as **2D Kriging**. The kriging mathematics does not change. What changes is only what the regional component is removed against.

Parameters:

| Parameter | What it sets | Default / advice |
|---|---|---|
| Point layer | Source points. | - |
| Value field Z (empty = from the geometry) | The attribute being interpolated. | - |
| External drift raster | A secondary surface s known everywhere. Same CRS as the points, covers the area. | - |
| Drift raster band (Adv.) | The band of a multiband drift raster. | 1 |
| Drift degree | A linear (a0+a1·s) or quadratic relation. | 1 (linear) |
| Kriging type, radius, min/max, nugget, structures | Kriging of the residuals - as in “2D Kriging”. | as in “2D Kriging” |
| Smooth grid (Gaussian), smoothing radius | Optional smoothing of the result. | off / 1 |
| Drift kriging raster | The output estimate (drift + kriged residuals). | - |
| Kriging standard error | An optional raster of the residual standard error. | skipped |

**How it differs from trend removal**

The **Remove polynomial trend** option of **2D Kriging** describes the regional component with a polynomial in the coordinates, that is with a tilted or curved plane. This works when the dip of the seam is uniform and its shape is simple. But if the field has a pronounced structure that follows a known surface, a plane will not describe it.

External drift removes the regional component not against the coordinates but against an external raster. Suppose the roof of the seam of interest follows the relief of the underlying one, for which a surface already exists. That relation is removed by regressing on the underlying surface. What gets kriged is the departures from it. The drift here is not a function of the position on the map but a function of the external raster value at the same point. Everything else matches trend removal, it is the same regression-kriging scheme.

**How it is computed**

First the drift raster is sampled at each well, bilinearly over the four neighbouring cells. Then the field value is regressed on this sampled value by least squares. The **Drift degree** sets the form of the relation. Degree 1 is the linear drift, value equals a0 plus a1 times the drift, the usual choice for external drift. Degree 2 describes a curved relation with the square of the drift. It may absorb part of the real structure, so after using it look at the residual variogram.

Next the regression residuals are kriged, exactly like an ordinary field in **2D Kriging**, with their own variogram, search and anisotropy. At the last step the drift raster is resampled onto the kriging grid and the drift is added back to the kriged residual estimate. The final estimate in each cell equals the drift plus the kriged residual. Because the drift is known everywhere, between the wells the estimate is drawn not towards a local mean but towards the shape of the external surface.

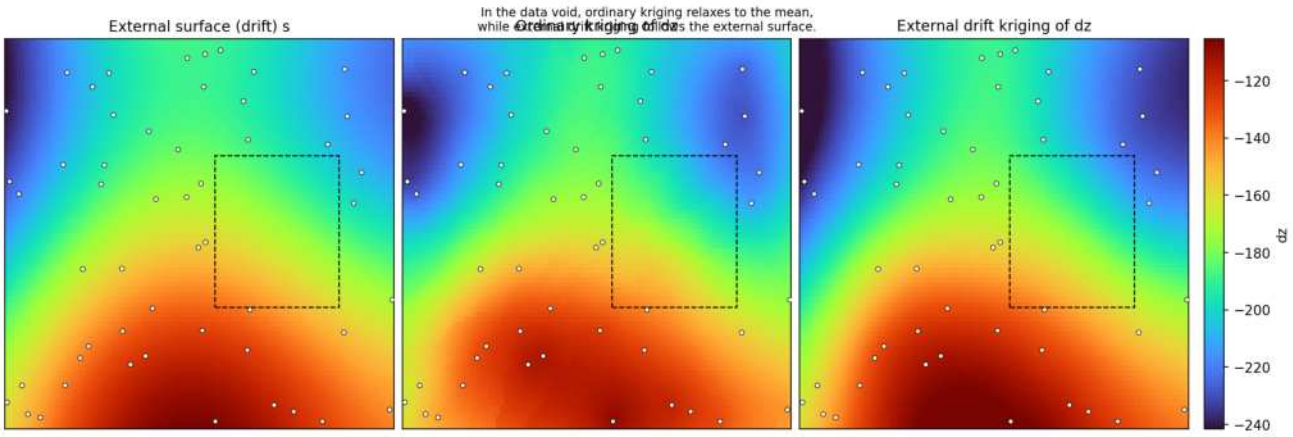



Figure 23: External drift kriging on sparse wells (computed with the Isoliner core). Left - the external surface s (drift). In the centre, ordinary kriging of dz relaxes to the mean in the data void. On the right, external drift kriging of dz follows the shape of the external surface where there are no wells. The dashed box marks the data void.

Wells that fall outside the drift raster do not enter the fit, and the tool reports to the Log how many were dropped. Grid cells not covered by the drift raster cannot be completed, so they are left empty together with the standard error in them.

### Parameters

The **point layer** and the **value field Z** are set as in **2D Kriging**. The **External drift raster** parameter is the secondary surface known everywhere. The optional **drift raster band** selects the band of a multi-band raster. Search, cell size, extent, clipping to the well hull, the nugget and variogram structures, outlier removal and grid smoothing all work as in **2D Kriging**. The description and the defaults are the same.

An important condition. The drift raster and the point layer must share the coordinate system, otherwise the drift value will be sampled at the wrong point. When the CRS does not match the tool warns in the Log. The drift raster must cover the whole estimation area, otherwise empty cells will appear along the edges.

### The variogram on residuals

As with trend removal, the variogram here is fitted on the regression residuals, not on the raw value. After the drift is removed the residual variogram returns to its normal form and reaches a sill with a nugget. The range reflects the true scale of the local correlation. The standard-error raster in this mode is the error of kriging the residuals. The drift is treated as deterministic and adds no error of its own.

A convenient way to fit the residual variogram without leaving the tool is not yet provided, so the residuals are judged by the share of variance removed, which the tool prints to the Log. If the drift took out a noticeable part of the spread, the relation with the external surface is real and the drift is appropriate. If it took out almost nothing, the field is not related to that raster, and plain **2D Kriging** will give the same result more simply.

## 3.03 Exceedance probability map

The **Exceedance probability map** tool answers not “how much” but “how likely the value exceeds a threshold”. From the kriging estimate raster and its standard-error raster it builds a probability raster from 0 to 1. Each cell holds the probability that the true value is above a given threshold.

The tool sits in the **Additional analysis tools** group and works as a post-processing step, like the hydraulic gradient. It runs no kriging of its own and does not touch the **2D Kriging** window, it takes ready rasters. So it works equally with the output of ordinary kriging and of external drift kriging.

### How it is computed

Kriging gives, in each cell, an estimate and its standard error, and the local distribution of the value is taken as normal. The value in the cell is treated as normal with the mean equal to the estimate and the standard deviation equal to the kriging error.

Under that assumption the exceedance probability is one formula through the normal distribution function. Where the estimate is well above the threshold the probability is close to one, where it is below it is close to zero, and at the threshold itself it equals one half. The larger the standard error, the smoother the transition, and away from the wells there is less certainty and the probability is drawn towards 0.5.

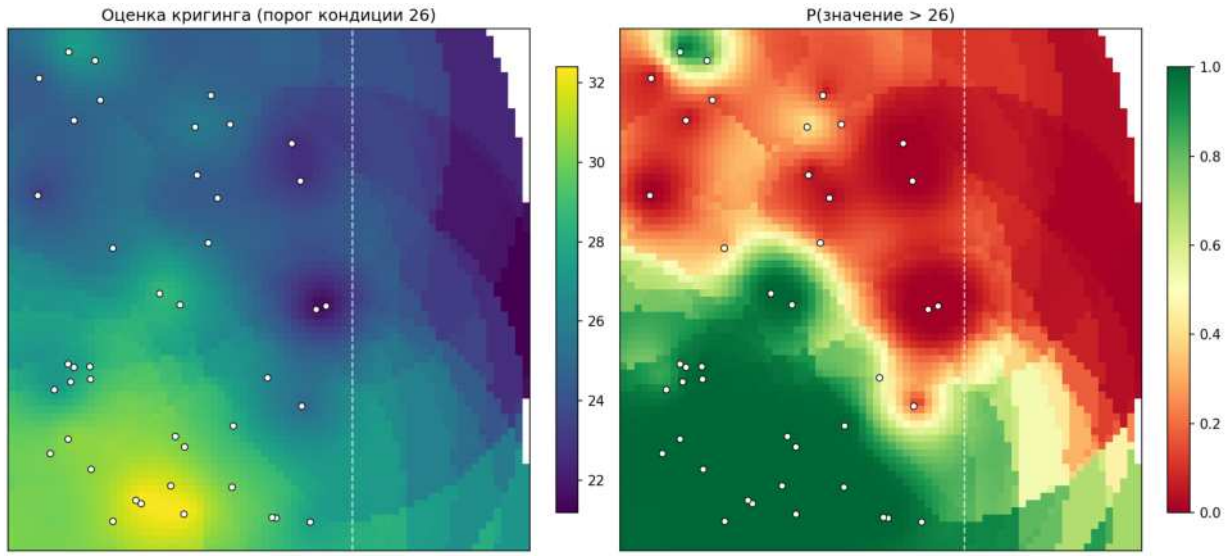



Figure 24: The kriging estimate with a cut-off threshold on the left, the exceedance-probability map on the right. Green - take confidently, red - confidently do not, beyond the drilling boundary the map converges to 0.5.

No separate kriging is needed for this, so the map is built instantly. The normality assumption is rough in places, especially for strongly skewed fields such as grades with a long right tail. Where that matters, indicator kriging by thresholds, which does not rely on the shape of the distribution, is more accurate.

### How to get the inputs

Run **2D Kriging** (or **External Drift Kriging**) on your field and enable the optional **Kriging standard error** output. You get two rasters, the estimate and the error, and you feed them here. Their grids match, since they come from one run, but if rasters with different grids are supplied, the error is resampled onto the estimate grid bilinearly.

### Parameters

| Parameter | What it sets | Default / advice |
|---|---|---|
| Estimate raster (kriging) | The field estimate raster (a kriging result). | - |
| Kriging standard-error raster | The standard-error raster of the same run. | - |
| Side | Probability above the threshold P(Z>t) or below P(Z<t). | above |
| Threshold | The value the probability is computed against. | 0 |
| Estimate raster band, error raster band (Adv.) | Bands of multiband rasters. | 1 |
| Probability raster (0…1) | The output probability raster. | - |

**Use**

The first use is cut-off grades. The threshold is the cut-off, and the map shows the probability that the grade is above it. This says more than a single line drawn on the estimate, because near the edge of the ore body the certainty drops and the probability map shows it. The second is risk zones for any threshold, a thickness below a critical value or an elevation above or below a hazardous one. The probability map complements the estimate map where not only the value matters but the confidence in it.

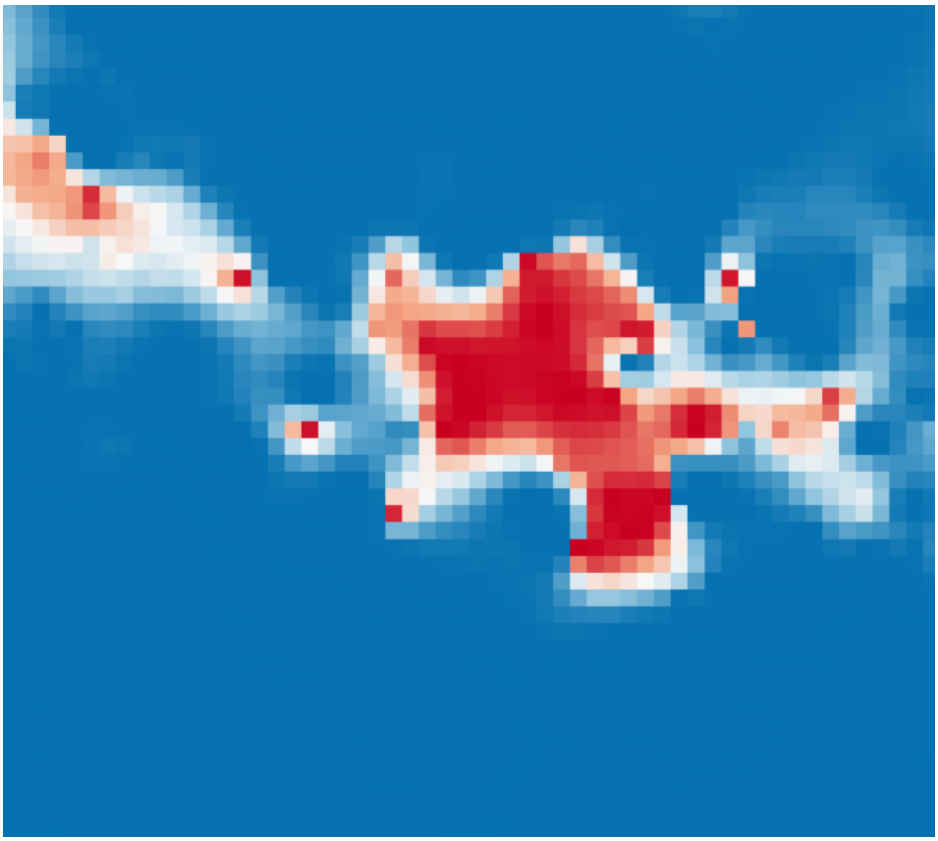


Figure 25: An exceedance probability map with a diverging colour ramp broken at 0.5. Red is where the value is confidently above the threshold, blue confidently below, and the white band along the P=0.5 line is the zone of uncertainty (contested values). The further from the wells, the wider the band.

## 3.04 Hydraulic gradient and flow direction

The **Hydraulic gradient and flow direction** tool works with the head field, that is the piezometric surface, and shows where and how steeply groundwater flows. The input is a head raster, usually the result of **2D Kriging** on borehole water levels. For a hydrogeologist this is as natural a step after building the head surface as isolines are after kriging.

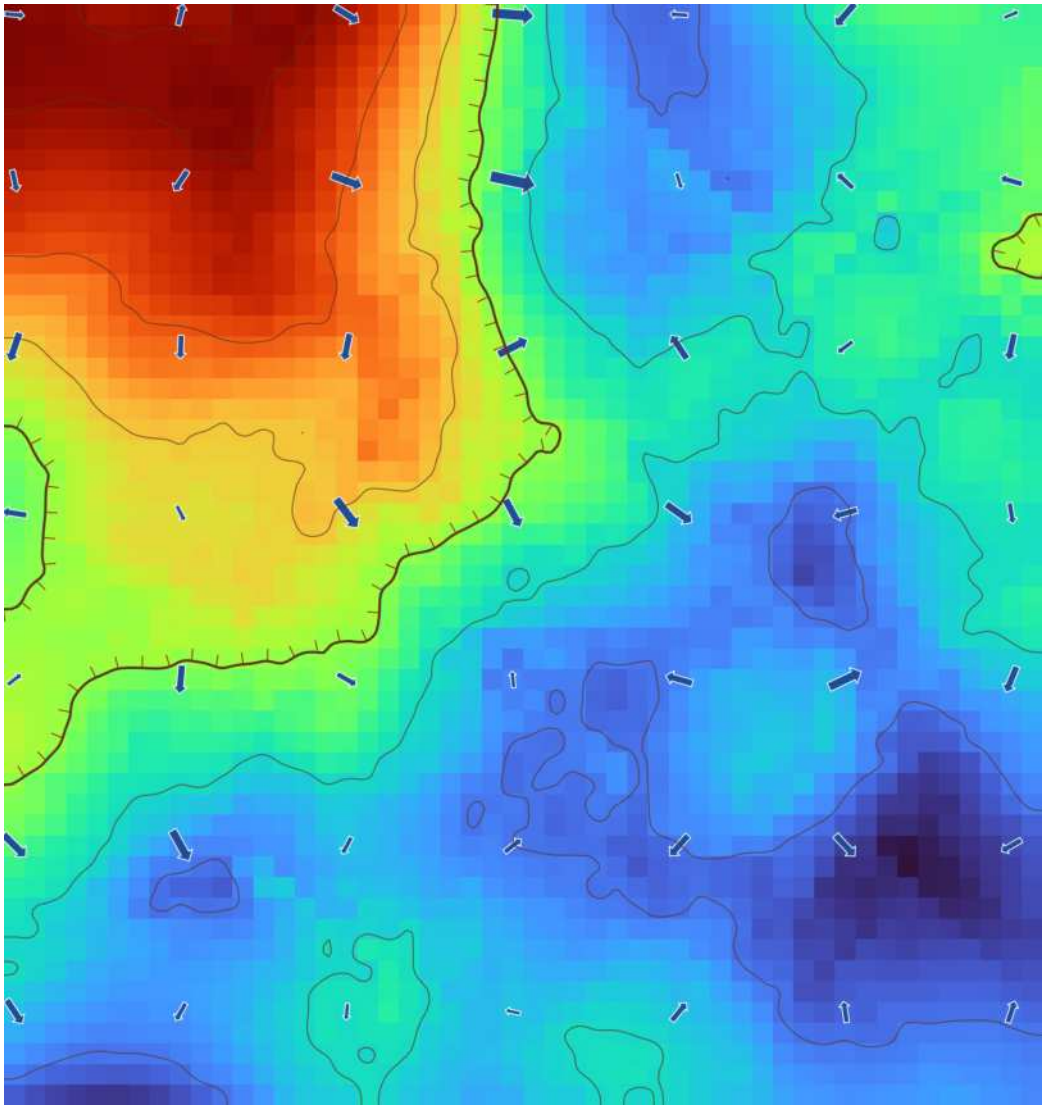


Figure 26: Flow vectors over the head surface: the arrows go down-gradient, from high head (warm tones) to low (cool), on top of the head isolines. The arrow length grows with the steepness of the gradient.

There are three outputs. The **gradient-magnitude raster** shows the steepness of the head surface, the hydraulic gradient i equals the magnitude of ∇h and is dimensionless. The **azimuth raster** holds the flow direction in degrees, where zero is north and the count goes clockwise. The point layer of **flow vectors** is thinned over the grid and styled as arrows right away, so the flow pattern is visible without touching the symbology.

The direction is computed strictly. Water flows down-gradient, from higher head to lower, so the arrow points towards the falling surface. On flat areas, where the head is almost constant, the direction is undefined and the azimuth there is left empty.

### Without permeability

The tool describes the geometry of the head field, not the flow velocity. The Darcy filtration velocity equals minus the hydraulic conductivity K times the gradient, and it needs K itself, which the tool neither asks for nor computes. In other words, the map answers where and how steeply, but not how fast. Once K (or transmissivity T) is available over the area, the specific discharge and the flow are computed by the neighbouring tool **Specific discharge (Darcy law)**. It multiplies this gradient by the aquifer properties.

### Parameters and smoothing

The input is the **head raster** and its **band**. The **flow vectors, thinning step** parameter sets how many cells apart to place an arrow so they do not merge, eight by default. The **smooth head before computing** parameter removes fine grid ripple, set in cells, off by default.

| Parameter | What it sets | Default / advice |
|---|---|---|
| Head raster | The input piezometric surface (usually a “2D Kriging” result). | - |
| Band (Adv.) | The band of the input raster. | 1 |
| Smooth head before computing, cells | Damps grid noise before differentiation. 0 = none. | 0 |
| Flow vectors: thinning step, cells | How many cells apart to place an arrow. | 8 |
| Hydraulic gradient (magnitude) | The output \|∇h\| raster. | - |
| Flow direction (azimuth) | The output azimuth raster (down-gradient). | created by default |
| Flow vectors (points) | The point layer of arrows (fields az, grad). | created by default |

Smoothing is switched on for substance, not for looks. Differentiation amplifies noise, so even a clean kriging grid can give a patchy gradient field with jittery arrows. A light smoothing brings the picture back to a readable form. The same effect can be had by smoothing the head itself back in **2D Kriging**.

### Arrows from points

The vector layer is points, and the arrows are drawn by the symbology. The preset is applied automatically. The arrow marker is rotated by the **az** field, so it shows the flow direction. Its size is scaled by the **grad** field, so the arrow is longer where the gradient is steeper. The size is set in millimetres and does not depend on the map scale. The symbology can be changed in the layer properties. A classic quiver diagram lays the arrow length out in map units, for which the marker is replaced with a geometry generator, and the template is in the styles folder next to the preset.

Fields of the flow-vector layer:

| Field | Type | Holds |
|---|---|---|
| az | number | Flow-direction azimuth, degrees (0 = north, clockwise, down-gradient). |
| grad | number | Magnitude of the hydraulic gradient \|∇h\| at the point, dimensionless. |

### The learning cycle

To walk the whole path without real data, switch on **Add a head field** in **Example wells (demo)**. A head field with a pronounced regional slope is added to the layer. Build a grid from it in **2D Kriging**, feed the raster here, and the arrows follow the head downhill. The same end-to-end scenario as for the other tools, only about hydrogeology.

## 3.05 Specific discharge (Darcy law)

The **Specific discharge** tool adds permeability to the flow geometry. The hydraulic gradient shows where and how steeply the head falls, but not how much water flows. Darcy's law links these through the aquifer properties - the higher the permeability and the steeper the gradient, the larger the flux. From a head raster and aquifer-property rasters the tool builds a physical flux rather than a dimensionless gradient.

The tool sits in the **Additional analysis tools** group and works as a post-processing step. It runs no kriging of its own, the property rasters are prepared separately by kriging from test points.

### What is computed

The specific discharge (Darcy flux) equals the hydraulic conductivity times the hydraulic gradient, $q = K \cdot |\nabla h|$, in metres per day. It is the volume of water through a unit cross-section area per unit time. If a transmissivity raster is supplied instead of conductivity, the tool computes the flow per unit width of the flow $Q = T \cdot |\nabla h|$, in square metres per day. Transmissivity is conductivity times thickness, so the flow per width already accounts for the aquifer thickness and does not need it separately. The direction of both fluxes is the same as the gradient direction, down the head slope.

The true water velocity is the specific discharge divided by the effective porosity, $v = q/n$. Porosity is usually absent from the data, so the tool does not ask for it and does not compute the true velocity. If you need it, divide the q raster by the porosity in the raster calculator.

### How to get the K and T rasters

The aquifer properties are known at the test points (pumping, injection) but are needed everywhere. They are interpolated by kriging, like any field. There is an important subtlety. Hydraulic conductivity and transmissivity are almost always log-normal, their values span orders of magnitude. Kriging the raw values distorts the result, so the logarithm is kriged. The simplest way is to enable the **ln** transform in **2D Kriging**, and then ln is kriged, the raster is returned already in the original units, and the ln checkbox here is not needed. If instead you krige an already-logged field, tick **K and T rasters are given as ln** in this tool and the values are recovered by exponentiation. Confined and unconfined aquifers are better kriged separately, their thickness physics differs.

### Parameters

| Parameter | What it sets | Default / advice |
|---|---|---|
| Head raster | The piezometric surface (a kriging result over the levels). | - |
| Hydraulic conductivity raster K (m/day) | The aquifer property for the specific discharge. Optional, but at least one of K, T is needed. | - |
| Transmissivity raster T ($m^2$/day) | The aquifer property for the flow per width. | - |
| K and T rasters are given as ln | Apply exp to the input rasters (for log-kriged K and T). | off |

| Parameter | What it sets | Default / advice |
|---|---|---|
| Smooth head, cells | Damps head noise before differentiation. | 0 |
| Flow vectors: thinning step | How many cells apart to place an arrow. | 8 |
| Raster bands (Adv.) | Bands of the multiband head, K, T rasters. | 1 |
| Specific discharge q (m/day) | The output raster q = K·\|∇h\|. | created if K is given |
| Flow per width Q (m²/day) | The output raster Q = T·\|∇h\|. | created if T is given |
| Flow direction (azimuth) | The output azimuth raster. | optional |
| Flow vectors (points) | The arrow layer: rotated by the `az` field, sized by the `q` field. | created by default |

**What the arrow layer holds**

Every arrow carries four fields. `az` is the flow azimuth in degrees, by which the arrow is rotated. `q` is the discharge value itself, by which the arrow is sized. `kind` says which quantity was computed, `flux` is the specific discharge and `flux_width` is the discharge per width. `units` repeats the unit, `m/day` or `m2/day`.

The kind and the unit are in the layer for a reason. With K set, the `q` field holds the specific discharge in metres per day, with T set it holds the discharge per width in square metres per day. The numbers are not comparable while the layers look identical, and without these two fields one run cannot be told from another. The field values are not translated, they are data, and a project opened with a Russian interface reads the same.

If neither K nor T is set, the tool refuses to compute and says so. By Darcy law the discharge is the product of the gradient and one of them. For the head gradient with the flow direction alone there is a separate tool, **Hydraulic gradient**.

**Use**

Where water moves faster and where slower, estimating inflows to workings, zones of higher seepage along permeable beds. Together with the exceedance probability map you can show not only the expected flux but also the confidence in it where test points are sparse.

## 3.06 Gaussian simulation (SGS)

Kriging gives a single smoothed surface and an estimation variance. Sequential Gaussian simulation answers a different question - how large is the uncertainty. It builds an ensemble of equally probable realizations, and each one reproduces the data histogram and variogram, passes through the boreholes and therefore stays rough rather than smoothed. Across the realizations every node accumulates a distribution of values, which shows where the estimate is reliable and where the data are silent.

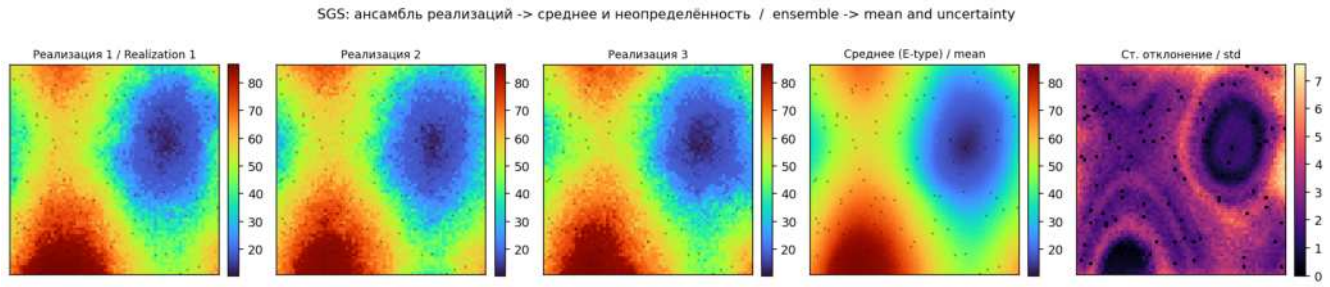



Figure 27: An SGS ensemble of realizations and the mean and uncertainty derived from it.

How it works. The values are mapped to normal scores and the simulation runs in Gaussian space. The grid nodes are visited in random order. At each one simple kriging on the neighbours and already-simulated points gives a local mean and variance, a value is drawn from that normal distribution, and it immediately becomes conditioning for the next nodes. Boreholes are snapped to the nearest nodes and frozen across all realizations. At the end each realization is back-transformed to the original units. The normal-score variogram is fitted automatically with a sill close to one.

Clipping by a mask is the same as in 1.01 and is applied to every realisation. On a large ensemble this takes noticeable time, and if it gets in the way, clip the stack separately after the run.

**Parameters**

| Parameter | What it sets | Default / hint |
|---|---|---|
| Points (boreholes) | The point layer with the data. | - |
| Value field | The numeric attribute to simulate. | - |
| Cell size | The grid step. 0 means auto, min(extent)/50. | 0 |
| Number of realizations | How many runs to average. | 60 (50-100) |
| Cut-off threshold | Enables the exceedance-probability map. | not set |
| Probability ABOVE the threshold | The probability direction. | yes |
| Raster extent | The output rectangle. | by layer |
| Score variogram model (Adv.) | The model for the normal scores. | auto |
| Max neighbours per node (Adv.) | How many points in the node's simple kriging. | 16 |
| Search radius (Adv.) | 0 means auto, 3 variogram ranges. | 0 |
| RNG seed (Adv.) | For reproducibility. 0 means random. | 0 |

The outputs are ensemble rasters. **Mean (E-type)** resembles kriging. **Standard deviation** shows the uncertainty, small at the boreholes and large away from them. The **P10**, **P50**, **P90** quantiles outline the likely range, and **Exceedance probability** for a given threshold offers a non-parametric alternative to the map from the probability tool. Runtime grows with grid size and the number of realizations, so start with a coarse cell.

If the data are clustered unevenly, supply the **wt** weight field from tool **1.05 Declustering**. The normal-score transform then builds the distribution with weights, and the ensemble histogram is not skewed toward over-sampled rich areas.

## Kriging kinds: which one to pick

Behind the word "kriging" the plugin holds a family of methods, and the choice between them affects the result more than fine-tuning the variogram. All the kinds solve the same system of equations with covariances from the variogram. They differ in what is assumed known about the field mean, and in what exactly is estimated - a point, a block or a probability. This chapter is an overview, while the parameters of each tool are described in their own chapters.

**Simple kriging (SK)** assumes the mean of the field is known in advance and constant over the area. Near the wells the estimate follows the data, away from them it is pulled to the given mean. Take it when the mean is backed by statistics over a representative sample of the same domain, a mean set by eye drags all the underdrilled margins towards the error. Switched by the type in **2D Kriging**.

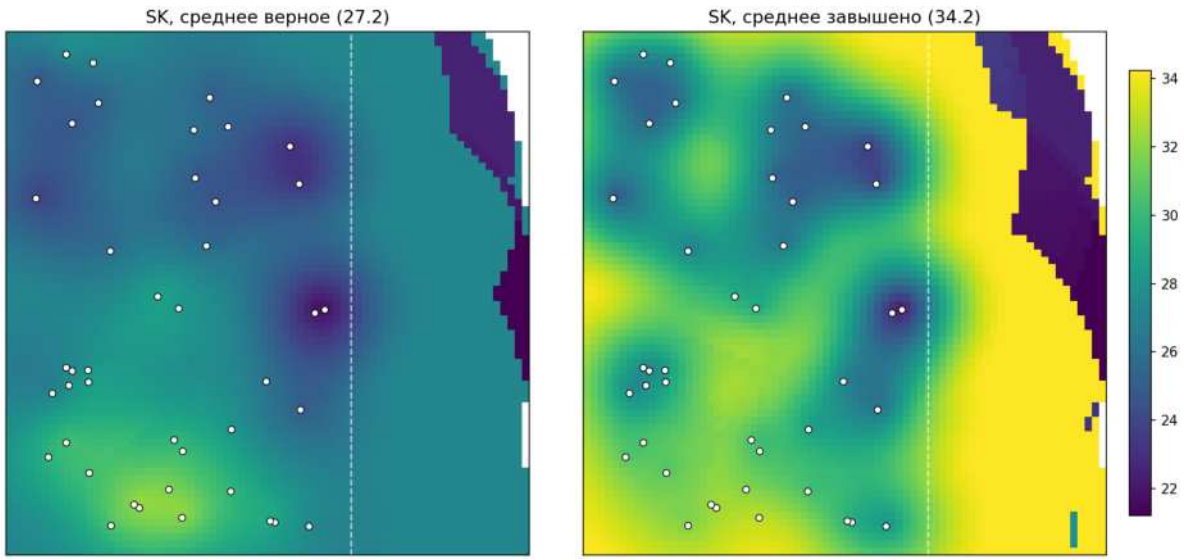



Figure 28: Simple kriging: the mean set from the data on the left, inflated by seven on the right. There are no wells east of the dashed line, and the whole underdrilled east “floats up” to the false mean.

**Ordinary kriging (OK)** does not know the mean and estimates it locally in every neighbourhood. An extra equation with the “weights sum to one” condition takes care of that. Away from the wells the estimate tends to the mean of the nearest neighbourhood, not to the global one. This is the default choice, and if unsure where to start, start with OK.

**Kriging with a trend** (the detrend checkbox in **2D Kriging**) is for fields with a regular regional slope, a roof on a monocline or a fold limb. A 1st- or 2nd-degree polynomial is removed by least squares, the residuals are kriged, the trend is added back. There are two rules. Define the variogram over the residuals, and if the share of the removed variance that the plugin prints is small, the trend is not needed. And do not extrapolate a quadratic trend far beyond the well cloud.

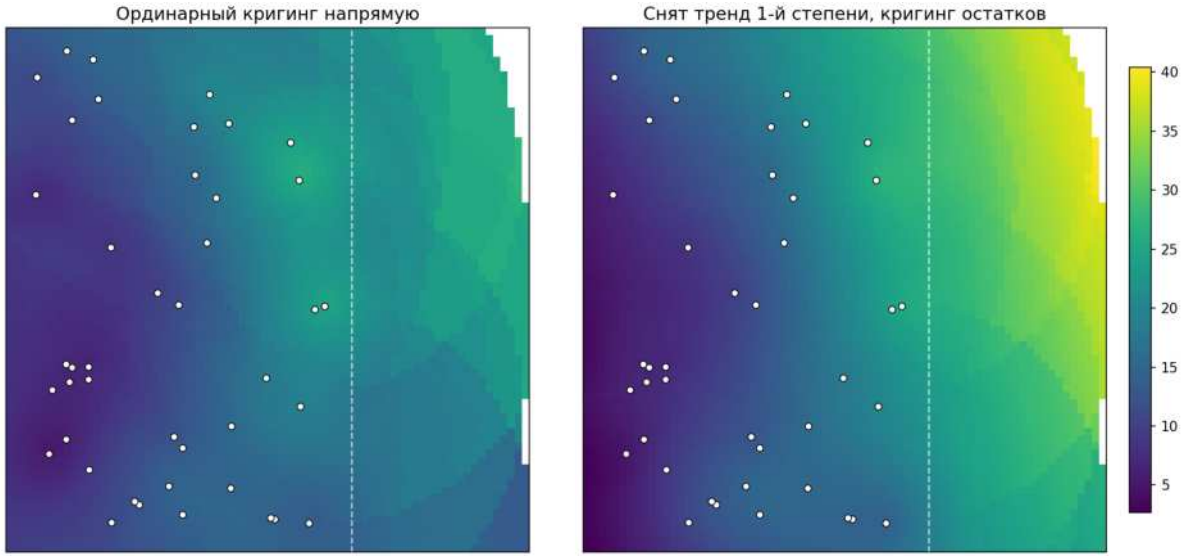



Figure 29: A field with a regional slope: plain ordinary kriging stalls at the local mean beyond the wells, regression kriging continues the slope regularly.

**Kriging with an external drift** (chapter 3.02) is for a trend known not as a formula but as a field. That is a structural surface of a neighbouring bed, a regional model, a seismic attribute. The

scheme is the same - a regression on the drift, kriging of the residuals, the regression returned.

**Block kriging** (the discretisation parameter in **2D Kriging**) estimates the mean over a block rather than a point value. The right-hand side of the system is averaged over the discretisation, the error variance drops, outliers are damped. Take it for reserves over a block grid, and mind the support effect. A block-kriging grid is regularly smoother than a point one, and a sample grade cannot be compared with a block grade directly.

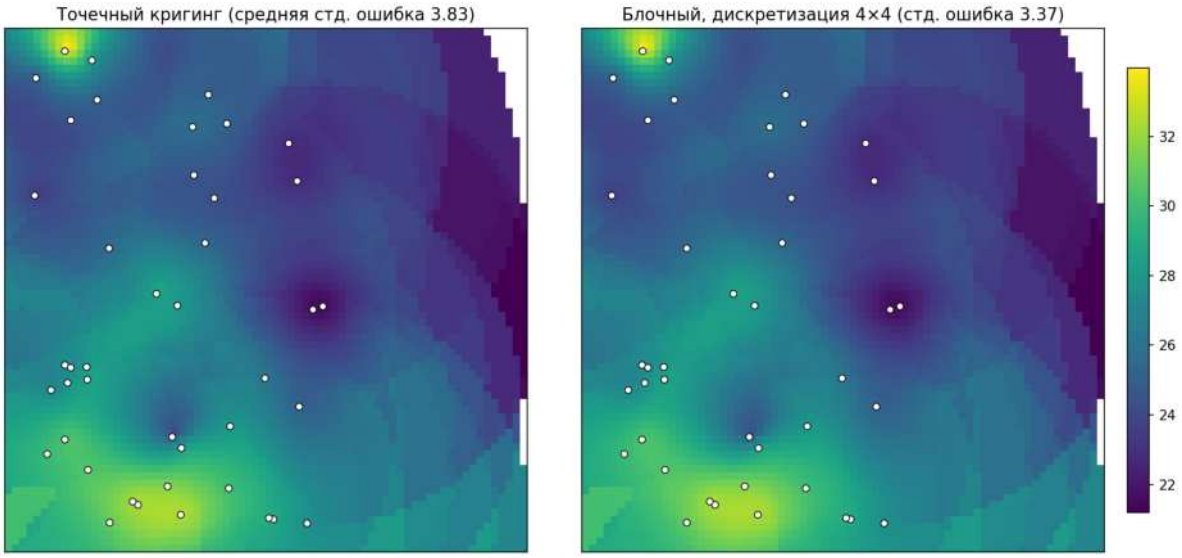



Figure 30: The same wells with two deliberate outliers: the cones on the block grid are damped, the mean standard error is lower.

**Indicator kriging** (chapter 3.01) is for categories, that is mineral type, facies or a replacement zone. The category becomes a 0/1 indicator and is kriged with plain OK. The result is the class probability at a point, and domains are cut from it by a threshold. The indicator variogram is its own and usually shorter than the grade one.

**Gaussian simulation** (chapter 3.06) is not kriging but its complement. Instead of one smooth surface it gives an ensemble of equally probable rough realisations, from which the uncertainty is seen directly.

**Cheat sheet**

| Task | Kind | Where it lives |
|---|---|---|
| The universal case, the start of any task | Ordinary (OK) | 2D Kriging, the default type |
| Plenty of data, the domain mean is justified | Simple (SK) | 2D Kriging, the SK type + mean |
| A roof or a bottom with a regional slope | With a trend | 2D Kriging, detrend |
| The trend is known as a raster | With an external drift | chapter 3.02 |
| Reserves over a block grid | Block | 2D Kriging, discretisation |
| Mineral type, replacement, categories | Indicator | chapter 3.01 |
| Uncertainty assessment | SGS simulation | chapter 3.06 |

The search neighbourhood is common to all the kinds, and three rules remove most problems. Take the radius of the order of the variogram range, take 12-16 neighbours at most, and keep the neighbourhood anisotropy consistent with the variogram anisotropy from the variogram map.

### Equal distances at the edge of the selection

On a regular sampling grid the estimated node is often equidistant from several samples at once. As many neighbours are taken as the maximum allows, and the edge of the selection cuts through a group of equal distances, so there is one place and several candidates.

The choice is made from the data itself - the smaller X coordinate, at equal X the smaller Y, and if those coincide as well, the smaller value. The key is chosen deliberately, because it does not depend on the order in which the samples are written in the input layer. The same layer saved in a different order gives the same map. The order of the equations in the system is brought to one form by the same key and for the same reason. A permutation of equations changes nothing mathematically, but floating-point arithmetic is not associative, and the last digits would diverge.

The practical meaning is simple. Export the same set from a database with a different sort order, merge layers in a different order, export again - none of that changes the elevations on the map any more.

### When the system does not solve

The kriging system can be degenerate. The usual causes are coincident samples with different values, a zero nugget, and a variogram at which the matrix loses its conditioning. In such a node the estimate is taken by the fallback method, by inverse distances to the same neighbours, and the variance is set to the largest one possible.

That is a fallback rather than kriging, and the standard-error map is overstated in those cells. The tool therefore lists them in the log, the number of cells and their share of the estimated area. Single cells go as a message, a share of a percent and above goes as a warning. If the warning appeared, what needs looking at is not the map but the variogram and the coincident samples.

## 3.07 Density from measurements (variable support)

The tool builds a density map where a measurement is given not by a point but by a finite-size support. That is a point with an uncertainty sigma, a line segment (a corridor of half-width) or a polygon. The unit mass of a measurement is spread over its support. Mass is conserved and density is inverse to the support area, so coarse georeferencing self-attenuates geometrically, without thresholds or filters. This is density estimation (how much and where), not value interpolation - kriging remains for values.

One geometry type per run. Points, lines and polygons are mixed by a series of runs into one raster (append mode). Each type spreads its mass in its own way:

- **Point** - a Gaussian spot with a sigma from the precision field, truncated at three sigmas. A sigma below the half-cell is raised to the half-cell.
- **Line** - a soft-edged corridor. The polyline is densified, mass is split by length, and each subpoint is a Gaussian profile with the half-width sigma. The from_m/to_m fields cut an interval by linear referencing.
- **Polygon** - mass is split by area uniformly or, in dasymetric mode, proportionally to an auxiliary raster (population, built-up area). If the raster is empty inside the polygon, it falls back to uniform.

### Output and invariant

The main output is a three-band raster. Band 1 is density in mass per km2 (independent of cell size). Bands 2 and 3 are service (sum m*sigma and sum m) so that append series and the effective-sigma map stay exact. The optional second raster is the mass-weighted sigma per cell, an effective-precision map. It separates density backed by precise georeferences from the smeared one, and it is an analogue of kriging variance for the density floor.

The invariant is that the density integral over the raster equals the sum of input masses. It is always computed and written to the log. A discrepancy means supports left the area, and the behaviour is set by the edge switch (renormalise inside or lose mass with a warning).

### How to read

Density shows where measurements cluster, weighted by their reliability. A precise georeference gives a compact spot, a coarse one a diffuse and low one. The effective-sigma map shows where density is gathered from precise supports and where from smeared ones - there the trust is lower.

| Parameter | Purpose | Default |
|---|---|---|
| Measurements | Points, lines or polygons (one type). | - |
| Mass field | Object mass. | 1 |
| Precision field | Point sigma or line half-width. | default sigma |
| Cell size, m | Grid step. | 50 |
| Area | Extent. | by layer |
| Support beyond edge | Renormalise inside or lose mass. | renormalise |
| Default sigma (Adv.) | When the precision field is empty. | half-cell |
| from_m / to_m (Adv.) | Line interval cut. | whole line |
| Auxiliary raster (Adv.) | Dasymetry for polygons. | - |
| Append to raster (Adv.) | An existing three-band raster. | - |
| Density | Three-band raster (density, sum m*sigma, sum m). | - |
| Effective sigma | Effective-precision map (optional). | - |

### The “Density map” window (live preview)

The face of the tool is a separate **Density map** window on the **Isoliner** toolbar. The layer and fields are set on the left, a preview on a coarse grid runs on the right. It takes milliseconds, so the sigma and the cell size change the image at once rather than after a run. The invariant is always shown at the bottom - input mass, mass on the grid and the share lost at the edge.

The **Demo** button creates a training set with tool 3.08. The **Write raster** button runs the full computation with the same algorithm 3.07 and puts the result on the map already dressed. The styling is pseudocolour with transparent zeros, density isolines and an effective-sigma layer as a trust map. The Processing form remains for models and batch runs, and the result is the same.

## 3.08 Example for density (demo)

The tool creates a synthetic set for 3.07 with a known total mass, so that the invariant can be checked by the numbers. Ten points with different sigmas, from fractions of a cell to large, carry mass 500. Two lines carry mass 200 with an interval cut on one, and two polygons carry mass 300 with one of them for dasymetry. An auxiliary raster comes with them. Total mass 1000.

Run 3.07 on the point layer, and the density integral in the log should give 500, on lines 200, on polygons 300. The layer fields are **mass**, **prec**, **from_m** and **to_m**.

| Parameter | Purpose | Default |
|---|---|---|
| Extent | Generation bounds. | - |
| Auxiliary raster cell, m | Step of the auxiliary raster. | 50 |
| RNG seed (Adv.) | Example reproducibility. | 1 |

| Parameter | Purpose | Default |
|---|---|---|
| Demo points / lines / polygons | Three measurement layers. | - |
| Auxiliary raster | Raster for dasymetry. | - |

The full list of output-layer fields is in the **Density (demo)** appendix section (“Demo-layer fields” at the end of the manual).

### 3.09 Example for 3.01 and 3.02 (demo)

The tool creates a synthetic potash bed block whose true fields are known in advance. The error of **3.01** and **3.02** need not be judged by eye here, it can be counted by subtracting the true field from the result.

The demo outputs five layers. The wells carry the fields **well**, **roof** (bed roof elevation, m) and **mintype** (sylvinite, partial replacement, full replacement). The network is uneven, with a denser patch like a mined panel. The marker horizon raster is known everywhere and serves as the drift for **3.02**. The true roof lies 25 m below the horizon and departs from it by local deviations of about a metre. The true replacement zones come as polygons. The zones are elongated, and full replacement lies inside partial replacement. The last layer is the outline of the gap in the network. There are no wells in the gap, and a fold of the horizon about 9 m high lies beneath it.

Show it in this order. Build **1.01** on the **roof** field and **3.02** with the marker horizon as the drift, with the same cell and extent. Ordinary kriging smooths the fold away in the gap, kriging with drift reproduces it. On a 6 × 4 km extent with seed 1 and a 30 m cell, the RMS roof error inside the gap is about 2.3 m for **1.01** and about 0.6 m for **3.02**. Then build **3.01** on the **mintype** field and compare the zone map with the true zones, and the confidence raster with the places of error. Errors gather at zone boundaries and where confidence is low.

The grid of the true fields is built like the grids of groups 1 and 3, from the lower left corner of the extent. With the same cell and extent in **1.01**, **3.01** and **3.02**, the results fall on the true fields cell for cell, and the difference is taken with the raster calculator.

| Parameter | Purpose | Default |
|---|---|---|
| Extent | Block bounds. In degrees the block is built in UTM. | - |
| Number of wells | How many wells to place before the gap is cut out. | 80 |
| Leave a gap in the well network | Remove the wells above the horizon fold. | yes |
| RNG seed (Adv.) | Example reproducibility, 0 gives a random set. | 1 |

If the extent is given in degrees, for example with the project in WGS 84, the demo is built in the UTM system at the extent centre, and all layers come out in it. The zone number and block size in kilometres go to the log. **3.01** and **3.02** run on these layers as they are.

## 4. Cross-sections

### 4.01 Cross-section along a line

The **Cross-section along a line** tool builds a geological section from a set of surfaces. It is not just a profile curve but beds as filled bands between a roof and a floor. The surfaces are usually obtained by kriging, and the tool assembles them into a section along a given line.

The tool sits in the **Cross-sections** group and works as a post-processing step over ready rasters. It runs no kriging of its own.

### How beds are defined

The surfaces are supplied as a list and ordered top to bottom - roof, floor, then the next roof, and so on. Beds are built as bands between adjacent surfaces, so N surfaces give N minus one beds. Two surfaces, a roof and a floor, are enough for one bed. For a sequence of beds, add the surfaces in stratigraphic order. A single surface is allowed as well, that is a section over the terrain without beds, see the next section.

### A section over a single surface

A single surface is a legitimate case, not an error. There will be no beds by definition, but the terrain line, the frame, the axes with ticks, the corner points and the section definition are all built. A geological section usually starts exactly there. First a profile over the DEM, then intersections with the mapped geology plotted on it by tool **4.05**, and the geology drawn downwards by hand.

Bed bands and the 3D fence are not built in such a run, and no layers are created for them at all. An empty layer in a project looks like a breakage even though everything was built correctly, so it is better not to create one than to create it and hide it.

The terrain comes out as the **Surface lines on the drawing** layer. Give it your own line style and labels, the drawing is finished on top of that layer.

Set the frame bottom by elevation in the advanced parameters for such a section. Without it the frame follows the data closely, with a small margin, and there is no room to draw below the terrain. The elevation is taken into account before the vertical scale is computed, so in the aspect ratio mode the drawing comes out at the scale of exactly the frame you will see. An elevation above the data is ignored, this is the bottom of the frame, not a clip from above.

The top of the frame is set the same way and for the same reason. A drawing is often extended upwards so that collars, water levels or labels fit in, and those are not among the surfaces themselves. A top below the bottom is rejected - the frame would be inside out.

Along with the frame, set the **step of the elevation ticks**. By default the labels are chosen as rounded numbers by their count, which is convenient while the frame follows the data. Once the frame is set in numbers, the labels are wanted in the same numbers. A step of two with a frame from a hundred and fifty to a hundred and seventy gives exactly the ticks one expects. Zero returns the automatic choice.

The pair is useful across a series of sections. The same frame and the same labels let the drawings be laid side by side on a sheet without adjustment.

### The layer tree sets the order of the surfaces

This is the main thing to know about the tool. The order of the surfaces is not a detail of presentation but input data on a par with the grids themselves, and it is what decides which band belongs to which bed.

By default the order is taken from the project layer tree, top to bottom, exactly as shown in the Layers panel. What you see in the panel is what you get on the drawing. The ticks in the multi-select decide which surfaces take part, not in what order they go.

Hence a simple working rule. Before building a section, arrange the surfaces in the Layers panel by stratigraphy, the roof of the upper host bed at the top and the bottom of the lower one at the bottom. Drag the layers with the mouse if they arrived in an arbitrary order - it takes half a minute and saves working the result out afterwards.

If the order is broken, the tool raises no error. It builds bands between the pairs you gave it, and some pairs turn out inverted, with the floor above the roof, so such bands are dropped silently. What

you see is fewer beds on the drawing than expected, or beds in an implausible sequence. When that happens, check the order in the Layers panel first, not the kriging parameters.

When the tree order is not suitable for some reason, uncheck **Surface order from the project layer tree**. The order is then taken from the sequence of ticks in the list, as in earlier versions. Layers absent from the project tree (added as files right in the widget) go to the end of the list, keeping their relative order.

The demo generator (4.10) arranges its six surfaces in the tree by stratigraphy itself, so nothing needs rearranging on the demo data.

### Outputs

The section drawing is polygons in axes of distance along the line and elevation. The elevation can be stretched by a vertical exaggeration so thin beds read well. This layer goes into a print layout as a ready section. Its coordinate system is conventional, with distance and elevation in map units.

Surface lines is every supplied surface as a separate line in the drawing axes, with the **sec**, **sec_id**, **num** and **name** fields. Breaks over missing data are preserved. When working with beds the layer is optional, since the roof and the floor are already visible as band edges, but it lets you label and style the roof line separately from the fill. With a single surface it is the only layer carrying the terrain.

The 3D fence is the same bands but as vertical PolygonZ walls in real coordinates. They are viewed in the 3D Map View next to the kriging surfaces, where the grid is set as terrain and the section walls show the beds in space.

### Vertical scale

The horizontal extent of a section is the line length, the vertical one is tens of metres of beds. The two are not comparable, so without a vertical stretch the drawing looks flat. The scale is set in three ways.

The **scale ratio H:V (1:N)** mode is the usual drawing notation. A value of 50 means H:V = 1:50, that is, the vertical scale is fifty times larger than the horizontal one. This is how the scale is set on potash sections, where the horizontal runs for kilometres while thicknesses are measured in metres. The stretch factor equals N.

The **exaggeration** mode is the same number given directly, without the drawing notation.

The **H:V ratio (drawing width:height)** mode works from the extents. You set the desired ratio of sheet width to height, say 10, and the tool computes the factor itself from the line length and the elevation span. It is handy when fitting the section to a sheet matters more than holding a given scale.

The effective exaggeration is printed to the log. For an exact overlay of layers it must match across the section, the boreholes and the composition. In H:V mode the section (4.01) spans the whole section in height. The boreholes (4.02) take the factor from the definition and line up by themselves, while the composition (4.03) computes the ratio over a single bed. To overlay it, take the exaggeration printed by 4.01 and set it in 4.03 in the **exaggeration** mode.

### Several sections in one run

By default the tool builds a section for every line of the layer, and this is the normal mode of work. A line layer of profiles is processed as a whole, and the tool need not be run once per line. If you do not need all of them, select the lines you want on the map and tick **Selected features only** on the line parameter. Unchecking **A section for every line of the layer** restores the earlier behaviour, a section along the first line.

All the sections go into one set of layers and are told apart by attributes rather than by a separate layer per trace. That way they are easy to label and filter by an expression, and styles do not have

to be reassigned. The section name comes from a field of the line layer if one is given in **Section name field**, otherwise the sections are numbered in order.

On the drawing the sections are separated by a layout in the common drawing coordinate system. **Stacked top to bottom** places them one under another with a shared zero of distance, so the left edge is the same for all. **In a row, left to right** places them side by side with a shared elevation datum, so the elevation scale runs through the whole sheet. **In a grid** combines both rules, a row keeps the elevation datum and a column keeps the zero of distance. The number of columns is set in the advanced parameters. The lattice pitch comes from the largest drawing plus the gap, so the drawings never overlap however much the trace lengths differ. The gap is a fraction of the extent, 0.15 by default.

The vertical scale is common to the whole run. Otherwise a short trace and a long one would come out at different stretches and would not be comparable on one sheet. In the scale ratio and exaggeration modes it is simply the number you set. In the extent ratio mode the factor is computed from the longest line and the elevation span of the whole set. The widest drawing gets exactly the requested proportion, and the others come out taller but at the same scale.

The 3D fence is not moved by the layout. It stands in real coordinates, each wall on its own trace. In the 3D Map View a set of sections looks like a real fence across the area.

The first section always gets a zero offset, so a run over a single line gives exactly the same drawing as before.

### Band colours: the bed reference

The bed bands are coloured by the roof name. Without a reference the colour is computed from the name itself, so it does not change between runs, but it is arbitrary.

The optional **Bed reference (table)** input makes the colour meaningful. The reference is an ordinary project layer, a GeoPackage or an Excel table, with one row per bed and per interbed, top to bottom down the section. Three columns are required - the body code, the number from the top, and the body kind. The colour is optional, but it is usually the very reason the reference is supplied.

The **body** column holds the word bed or interbed, and this is a law rather than a guess, because the body cannot be derived from the code. The bed **АБ** looks like “А plus Б” by its spelling, yet it is a single bed containing А, Б, the А-А’ parting and А’. The interbed **Б-В** cannot be derived from the code at all, because there is no bed Б in the list, there is АБ. The body is decided by the geologist, and the machine does not recompute it.

The top-to-bottom order gives the reference a second ability, finding the body between two boundaries. If the section runs the roofs of KpII and KpIIIa, then by order the interbed KpII-KpIII lies between them, and the band takes its name and colour. Beds that did not make it onto the section are marked grey and listed in the log. That shows at once what is missing from the set of surfaces.

Layer names rarely match the codes word for word, so the lookup proceeds in stages from strict to tolerant.

The exact code is tried first, then the same code ignoring case and outer spaces. Next comes a normalised form in which the role suffix `_top`, `_bottom`, `_кровля`, `_подошва` is stripped and Latin look-alike letters are folded into Cyrillic. The same form without apostrophes is tried last. This is how `KpII_top` finds the code `KpII` and `A'Б_top` finds `АБ`. The strict steps come first, so if the reference holds both `АБ` and `A'Б` as different bodies, each keeps its own.

The same reference goes into 4.02, and then the bands and the borehole columns speak one language of codes and match in colour.

A ready sample for the Verkhnekamskoye deposit lives in the **templates** folder inside the plugin directory. It is 36 rows from the cover deposits down to the lower rock salt, with colours and bodies. The file comes in two forms, `plast_reference_vkmks.xlsx` and `plast_reference_vkmks.csv`. Take it as a starting point and edit it for your own deposit. The **strata** and **note** columns are optional, they are read but not used yet, and are left for future development.

**Attributes of the output layers**

All the section layers carry two common fields - **sec** with the section name and **sec_id** with the feature id of the source line. Use them to label the drawings, filter the layer down to one trace and colour the sections differently.

**Section definition** - one line per section, with the original trace geometry and the fields read by the downstream tools of the group:

| Field | What it holds |
|---|---|
| sec, sec_id | Section name and source line id. |
| vex | Vertical exaggeration of the drawing. |
| step | Stationing step, m (the polyline vertices are always included). |
| zmin, zmax | The elevation range of the drawing. |
| ox, oy | Offset of the drawing in the layout. Zero for a single section. |

**Section (3D fence)** - vertical MultiPolygonZ walls along the trace, one per bed:

| Field | What it holds |
|---|---|
| sec, sec_id | Section name and source line id. |
| bed | Bed number from top to bottom. |
| top, bot | Names of the roof and bottom surfaces. |
| t_mean | Mean bed thickness along the trace, m. |
| seclen | Trace length, m. |
| color | The colour of the bed as `#RRGGBB`, the same as on the drawing. |

A wall is split by the links of the trace, one panel per link, all of them collected into a single multipart feature. A wall must not be given as one ring. A ring of “top one way, bottom back” comes out long and strongly non-convex, 3D scenes triangulate it by a fan from the first vertex, and the section then looks full of holes. The quadrilateral of a link is planar by construction, because two neighbouring nodes define a vertical plane and both elevations lie in it.

The colour of a bed travels by two ways at once, as categories on the layer, like on the drawing, and as the `color` field in the attributes. Symbology is not read by every consumer, a 3D scene colours by its own rule, and a contractor has no access to QGIS symbology at all, so the colour is duplicated in the data.

**Section corner points carry the following fields: sec and sec_id, num (corner number), name (УГ-1, УГ-2, …) and pos (top or bottom), then d (station, m), x and y (map coordinates), az (azimuth of the next leg) and label (a ready-made label).** Horizontal axes** carry sec, sec_id, elev (axis elevation, m) and label. **Corner table** carries sec, sec_id, kind (row type) and text (cell content).

**Parameters**

| Parameter | What it sets | Default / advice |
|---|---|---|
| Section line | A line layer. The **Selected features only** tick works here. | - |
| Surfaces top to bottom | A list of surface rasters. At least two are needed. | - |

| Parameter | What it sets | Default / advice |
|---|---|---|
| Surface order from the project layer tree | The order comes from the **Layers** panel, top to bottom. Unchecked, it comes from the ticks in the list. | on |
| A section for every line of the layer | Batch mode. Unchecked, a section is built along the first line. | on |
| Section name field | The field of the line layer the name comes from. Empty means numbering in order. | - |
| Layout of multiple sections | Stacked, in a row or in a grid. | stacked |
| Sampling step along the line | How many map units between samples. 0 means by cell size. | 0 |
| Vertical scale | Mode: extent ratio, exaggeration or scale ratio H:V. | extent ratio |
| Scale value | Width:height ratio, exaggeration, or N from the 1:N notation. | 10 |
| Columns in the grid (Adv.) | Number of columns for the grid layout. | 2 |
| Gap between sections (Adv.) | A fraction of the drawing extent. | 0.15 |
| Raster sampling (Adv.) | Bilinear or nearest. | bilinear |
| Bed reference (table) | Colour, body and order from data. | empty |
| Frame bottom, elevation (Adv.) | Lowers the frame so there is room to draw. Empty - from the data. | empty |
| Frame top, elevation (Adv.) | Raises the frame: collars, water levels, labels. Empty - from the data. | empty |
| Step of the elevation ticks (Adv.) | Labels with a given step instead of rounded by count. 0 - automatic. | 0 |
| Surface lines on the drawing | The output line layer, one line per surface. | created |
| Section drawing (distance × elevation) | The output polygon layer for a layout. Not created with a single surface. | created |
| 3D fence (PolygonZ) | The output layer of vertical walls in real coordinates. Not created with a single surface. | created |

Each bed gets attributes - a number, the roof and floor names, the mean thickness and the section length. Colour the layer by bed number or by thickness. Where a surface is undefined (nodata), the band breaks and the bed splits into several polygons.

### Trying it on a demo

A ready training set is produced by the **Example for a section (demo)** tool (4.10). It gives six surfaces top to bottom, three section lines, a pair of drilling-model layers collar and interval, and multiband bed grids. It arranges the surfaces in the tree by stratigraphy itself. The three lines differ in length and in the number of bends, which is handy for looking at the layout and the common

vertical scale. Run it, then feed the surfaces and the line here. The collar and interval pair goes with the definition into **Boreholes on the section**, and the bed grid (bands 1/2/3) goes into **Bed composition on the section**. The full contents of the set are in section 4.10.

### Relation to QGIS

A plain profile curve over a single grid is built by the native **Elevation Profile** panel, no separate tool is needed for that. The section instead shows the beds between surfaces, which the native tools do not do. A kriging surface can also be viewed in 3D without a section, for which the grid is set as terrain in the 3D Map View.

## 4.02 Boreholes on sections (drilling model)

The **Boreholes on the section** tool places boreholes onto section drawings from a pair of drilling-model layers. It sits in the **Cross-sections** group and works in batch, one run serves every section of the definition.

### The drilling model

Boreholes are described by two tables following the minimal model of the mining packages (Leapfrog, Micromine, Datamine, Surpac). **collar** is a point layer of collars with the **hole_id** (identifier, string), **z** (collar elevation) and **eoh** (end-of-hole depth downhole) fields. **interval** is a plain interval table with the **hole_id**, **from**, **to** and **code** fields, where the code is what we colour by, a bed index, a lithotype or a class. Depths are measured downhole from the collar, positive downwards, not as elevations, so inclined holes do not break the model. Any other columns of the interval table travel into the drawing attributes as they are.

Such a pair is produced by **Example for a section (demo)** (4.10) and by a corporate export. The tool finds the fields of both tables by the contract names and common synonyms (Hole_ID, elev, depth_from, litho) itself, case does not matter. The field pickers are hidden under the advanced parameters and are needed only for non-standard layers. What was found is printed to the log in one line.

### Inclined holes

A third table, **inclinometry**, is optional. Without it a hole is treated as vertical, the downhole depth becomes an elevation by subtraction from the collar, and the column stands at a single point of the section.

Inclinometry defines the axis by survey stations - hole, measured depth downhole, azimuth from north clockwise, zenith angle from the vertical. A zenith of zero is a vertical hole and ninety a horizontal one, which is the convention of drilling records.

The axis is built by the minimum curvature method, and between two stations the hole is taken as an arc of a circle. The tangential method, which computes the whole leg from the lower station, carries the bottom tens of metres away when the stations are sparse.

Inclinometry changes three things. The intervals are carried along the axis, so the roof and the floor of a bed stand at different distances along the section. The hole comes out as a polyline rather than a segment. The corridor selection runs over the whole axis rather than over the collar, since otherwise a hole with a distant collar but a bottom close to the line would not reach the drawing.

On a drawing with vertical exaggeration the inclination looks weaker than it is. Elevations are stretched vex times while the distance along the line is not, so the visible angle is compressed by the same factor. At a vex of about sixty a hole inclined at forty-five degrees looks vertical. The tool reports the number of inclined holes and the largest horizontal displacement of the bottom, so the inclination is checked by numbers rather than by eye.

### Holes without intervals

The interval table is optional as well. Without it the drawing receives the holes and the labels, because the position and the depth of the holes matter at the design stage too, before the geology is described.

The length of a hole is then set by the end-of-hole depth alone. A hole without it is skipped and their number goes to the log, because inventing a depth would be wrong.

### When collars and intervals do not match

The most frequent fault of a data set is a mismatch of hole_id between the tables. The reader reports not only the number of orphaned intervals but the keys themselves from both sides. When the cause is recognised it is named outright - only the case differs, the strings differ by spaces alone, or they are typed in different keyboard layouts. The last matters in particular, because a Cyrillic H and a Latin H are indistinguishable on screen.

When the keys differ in substance, no cause is invented, and the hole numbers are compared between the two tables by hand.

### The tolerant reader

The data is read without prior cleaning. Empty and non-numeric depths are skipped, and swapped from and to are exchanged. Intervals beyond the end of hole are drawn as they are, overlaps are neither resolved nor hidden, and gaps between neighbouring intervals are not filled. Intervals without a collar are skipped. Everything skipped or accepted with a note is counted and reported to the log as a short summary. On clean data the summary is a single line.

### Batch operation and alignment

The lines, the vertical scale and the layout come from the section definition produced by **Cross-section along a line**. Every borehole is projected onto every line and lands on the drawings it is closer to than the corridor (0 means all). Depths become elevations by subtraction from z, the vex factor is shared from the definition, so the columns sit on the beds by height without fitting.

The collar layer may live in a different coordinate system than the definition, for example when exported from a corporate database in the working system of the enterprise. The tool reprojects the collars into the definition system itself and prints a line about it to the log together with the coordinates of the first collar. If the corridor turns out empty, the log reports the distance of the nearest collar from the line. That number tells at once whether the corridor is narrow or the layers live in different places.

### Colours and the legend

The interval layer comes out coloured right away, a category per code in the order of first appearance top to bottom. The colour is deterministic from the code itself and does not change between runs and machines. The code-to-colour legend appears in the layer tree by itself. The last entry is the grey **other** category, everything that did not match a known code falls into it, so nothing disappears silently. The columns carry a thin black outline and read on top of the section bands. The bed bands in **Cross-section along a line** are coloured by the same mechanism, from the roof name. A bed named with the same code as in the interval table therefore matches the borehole columns in colour by itself. The traces are drawn as a thin grey line from the collar to the end of hole, the collars as points with a short label. The label comes from the **number** field of the collar layer, where name and label are synonyms. Without it the label comes from hole_id. The composite identifier stays in the attributes for joins.

### Clipping by the drawing and the frame

By default the columns are clipped by the drawing frame - the zmin and zmax elevation range from the definition. An interval on the edge is trimmed to it, an interval entirely beyond the frame is skipped. The trace and the label are clamped by the frame, and a borehole entirely beyond the frame drops out of the drawing. The checkbox turns clipping off entirely.

The frame is one for the whole drawing, while the roof of the uppermost sequence differs from point to point. For a cut along the beds, feed the optional **Section drawing** input - the band polygons from **Cross-section along a line**. The columns are cut by the upper and lower envelope of the bands at their own position, so boreholes do not stick out of the drawing, as in the mining packages. Beyond the bands this input has no effect, and the frame keeps working there.

The tolerance under the advanced parameters widens the frame and the bands outwards, in elevation units. Only the geometry is clipped, the ztop and zbot attributes keep the true interval elevations. A clipping summary is printed to the log per section.

### Column colours: the bed reference

By default the interval colour is computed from its code and does not change between runs. The optional **Bed reference (table)** input replaces those colours with the reference ones. Codes found in it are painted from it, and the order of the legend categories follows the bedding from top to bottom. Codes outside it keep their previous colour. The log gets the reading summary and a line on how many codes were found.

The reference for 4.01 and 4.02 may be one and the same, and that is its strength. The drawing bands carry a bed code while the borehole intervals often carry a rock type. If the interval table does have a bed index field, set it as the code field and supply the same reference as in 4.01. The columns will then merge with the bands in colour, and only the disagreement between the borehole and the built surface will stand out. If the intervals carry lithology instead, a bed reference will not help there, and that needs a legend of its own.

### Parameters

| Parameter | What it sets | Default |
|---|---|---|
| Section definition | The lines, vex and layout from **Cross-section along a line**. | - |
| Borehole collars (collar) | The collar point layer. | - |
| Borehole intervals (interval) | The interval table. | - |
| Corridor from the line | A buffer, map units. 0 puts every borehole on every section. | 0 |
| Clip intervals by the drawing frame | The zmin and zmax frame from the definition. | on |
| Section drawing (for clipping) | The band polygons from 4.01, a cut by the envelopes. | empty |
| collar and interval fields (Adv.) | Overrides of the automatic field search. | found automatically |
| Collar label field (Adv.) | Where the short collar label comes from. | number |
| Clipping tolerance (Adv.) | Widening of the frame and the bands, elevation units. | 0 |
| Bed reference (table) | Colour and code order from data. | empty |
| Borehole intervals (drawing) | Vertical segments coloured by code. | created |

| Parameter | What it sets | Default |
|---|---|---|
| Borehole traces (drawing) | Lines from the collar to the end of hole. | created |
| Collars on the drawing | Points labelled from number. | created |
| Borehole intervals (3D) | LineStringZ in real coordinates, depth goes into Z. | on request |

The chosen layers and the corridor are remembered, the next run in the same project opens with the inputs already in place.

#### Colouring by grade

A fourth table, the **chemistry**, is optional. With it a column is coloured by the grade rather than by the bed code. Choose a field, and the bands get a continuous scale from pale to dense. The range of values, the median and the number of samples carrying a value go to the log, because without those numbers a dense colour cannot be interpreted.

With the chemistry set, the column is sliced by **samples** rather than by beds. This is a different slicing, and the two do not combine in one layer. A sample is taken from the core while the bed boundaries are drawn from the description, and one band cannot be both at once. The lithology intervals do not go to the drawing in that case, and the tool says so in the log.

A sample without a value stays a grey band rather than disappearing from the drawing. This is not cosmetics. A band that vanished reads as an absence of core, which is not what the data say.

The ready colour of every band is written into the `ccolor` attribute, as with the colouring by code. The print layout and other styles then see the same thing as the screen.

## 4.03 Bed composition on the section

The **Bed composition on the section** tool colours the band of one bed by a composition grid along the line. It takes a roof, a floor and a composition grid, runs no kriging of its own, and works one bed at a time. It sits in the **Cross-sections** group.

This is how the lithological composition change inside an industrial bed is shown along the section. The composition grid is prepared separately, the content by ordinary kriging and the mineral type by indicator kriging (the **Categorical indicator kriging** tool).

#### Two modes

Continuous content (KCl, insoluble residue) goes like this. The band is cut into thin vertical slices, each with a mean value. Set a graduated style for the layer (by the **value** field), and a smooth content transition is visible along the band.

A categorical mineral type or facies (sylvinite, replacement, halite) goes differently. Adjacent slices of the same class merge into facies zones. Set a categorized style (by the **class** field). Replacement zones show as a colour change along the line.

#### Parameters

| Parameter | What it sets | Default |
|---|---|---|
| Section line | The same line as for the section. | - |
| Bed roof | The roof raster. | - |
| Bed floor | The floor raster. | - |

| Parameter | What it sets | Default |
|---|---|---|
| 1st industrial bed | A multiband grid: roof, bottom, content, mineral type. | on request |
| 2nd industrial bed | The same for the second industrial bed, independent fields. | on request |
| Sampling step along the line | How many units between samples. 0 means by cell. | 0 |
| Vertical scale | Mode: H:V ratio or exaggeration. | H:V |
| Scale value | H:V ratio or exaggeration. | 10 |
| Raster sampling (Adv.) | Bilinear or nearest (always nearest for a class). | bilinear |
| Bed composition (drawing) | Output polygons in distance × elevation axes. | created |
| Bed composition (3D) | PolygonZ polygons in real coordinates. | on request |

Run the tool for each industrial bed separately, with its own composition grid. Place the composition band on top of the section drawing. For an exact overlay take the exaggeration printed by **Cross-section along a line** and set it here in the **exaggeration** mode. The H:V ratio is computed over a single bed and is not suitable for overlay.

## The section definition and shared parameters

Geometrically a section is set by two things - a line in the real coordinate system and a vertical scale vex. The **Cross-section along a line** tool outputs them together as a **Section definition** layer. There is one line per section, with the sec, sec_id, vex, step, zmin, zmax, ox and oy fields, and this is the shared source of truth.

The intersect, project and unproject tools read the line and vex from this definition, so their results match the section without manual scale fitting. Build the section once, the definition travels with the project and feeds the other tools of the group.

When there are several sections, the definition carries them all. Tools 4.04, 4.05 and 4.06 handle the whole set in one run.

Every result lands on its own drawing by the offset from the ox and oy fields and gets the sec and sec_id fields. If you need a single section, filter the definition layer or select the line you want, the feature order is preserved. The definition may hold sections with different vex, which happens when definitions from separate runs are merged. The tool then warns that the drawings are not comparable and advises rebuilding them in one run.

The **Bed composition on the section** (4.03) tool and the beta tools 4.07 and 4.08 still work over a single section and take the first one from the definition. **Boreholes on the section** (4.02) handles the whole set in batch. With several sections, feed them a definition filtered down to the trace you need.

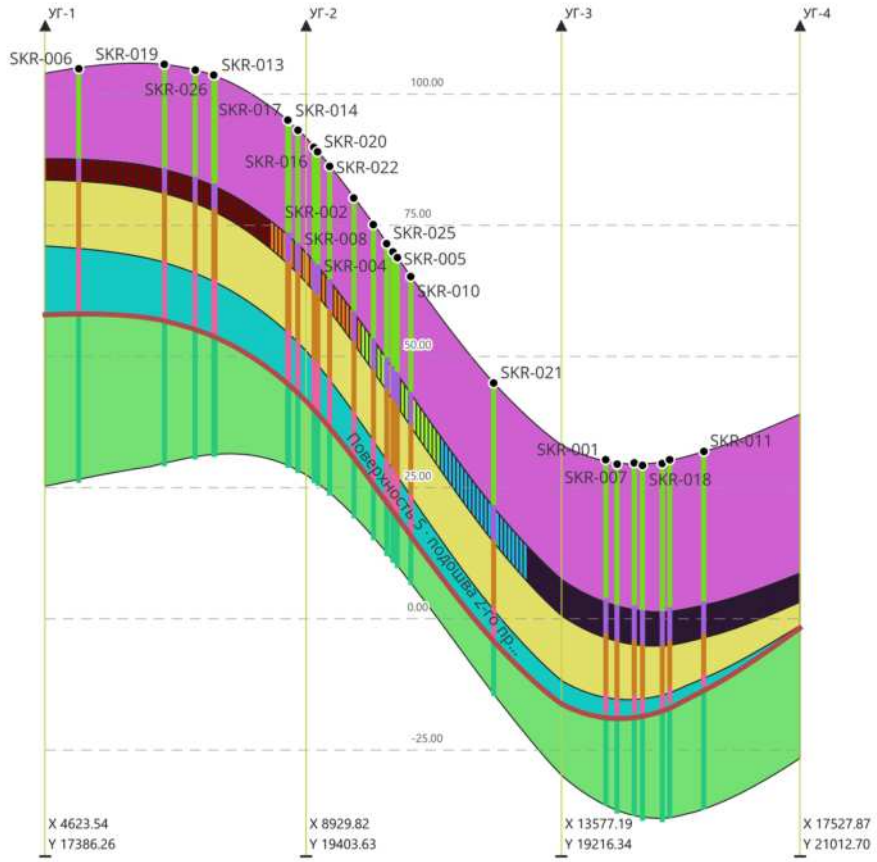
The **Bed composition on the section** tool accepts the section definition as an optional input. When given, the vertical scale is taken from it, so the composition band sits exactly on the beds by height. **Boreholes on the section** works from the definition only, it has no scale choice of its own.

Definition layers made by earlier versions of the plugin are read as before. They have no sec or ox fields, so the name comes out empty and the offset zero, and the result lands exactly as it used to.

The section also clips pinch-outs. Where the roof drops to the floor, the bed disappears and no band is built. In the demo the second industrial bed pinches out to the east.

For a polyline the Cross-section along a line tool optionally outputs three helper layers in the drawing axes. Corner points are placed at every polyline node, at the top and at the bottom of the section. A point carries the fields of number, name (УГ-1, УГ-2 …), side (top or bottom), distance along the line, plan X and Y, segment azimuth and a ready label. The top is labelled with the name, the bottom with the plan coordinates X and Y, rounded to two decimals. The azimuth and distance stay as layer fields - handy to place into a layout table. A style is supplied, an upward triangle on top and a shelf at the bottom.

A corner table is produced optionally - a polygon layer below the section. The cells lie between the corner verticals with borders under them, in two rows. They hold the length and azimuth of the segment between adjacent corners, with a centred label and a white fill. It renders on the canvas and is carried into a layout with the section. Corner verticals are lines at the nodes spanning the full section height. Horizontal axes are equal-elevation lines with ticks (five by default, with nice rounding) for an elevation scale. The drawing margins are extended by five percent up and down, and the corner points lie on these edges.




| | | | |
|---|---|---|---|
| d | 4755.39 | 4651.14 | 4339.91 |
| Аз | 64.90 | 92.31 | 65.55 |

Figure 31: Section decoration: the frame with corner verticals and triangles, horizontal axes with ticks on the left, and the corner table below.

### The model manifest instead of a list of layers

The **Model manifest** tool (5.05) records in the project properties what role every layer plays. The roles are the datum surface, a contact of a body, the collars, the intervals, the axis surveys, the

reference. The section and the boreholes read this markup.

If no surfaces are given in 4.01, every layer with the role of a body contact is taken, in the order of the layer tree. If the collars, the intervals or the inclinometry are not given in 4.02, each of these inputs is looked up by its own role. Whatever is taken from the manifest is named in the log, so that the substitution does not go unnoticed.

There is one rule and it outranks convenience. The manifest shortens the path but does not become a condition of work. If there are no roles, because the project is foreign or inherited, everything works by explicit choice as before.

## 4.04 Intersect surfaces with the section

The **Intersect surfaces with the section** tool places surface grids onto the section as lines in distance-elevation axes. Each grid is sampled along the definition line, and its trace lies on the drawing next to the beds. The line and vex come from the section definition, so the match with the section is automatic.

This is how water tables, marker surfaces, the salt roof and anomaly surfaces are placed on the section. The inputs are the section definition and a list of grids, the output is lines in the section axes (and optionally 3D lines in real coordinates).

**Parameters**

| Parameter | What it sets | Default / hint |
|---|---|---|
| Section definition | The definition layer from 4.01: a line with vex and step fields. Sets the line, the scale and the frame height. | - |
| Surface grids | The list of rasters whose traces are drawn on the section as lines. | - |
| Sampling step along the line (Adv.) | How many map units between samples. 0 means by cell size. | 0 |
| Raster sampling (Adv.) | Bilinear or nearest. | bilinear |
| Surface lines on the section (drawing) | The output lines in distance × elevation axes. | created |
| Surface lines (3D) | The output 3D lines in real coordinates. | on request |

## 4.05 Vector intersection with the section

While 4.04 places surfaces as grids, this tool places **vector** objects on the section by exact intersection with the section line. The result type depends on the object.

A line **without an elevation** (flat in plan - a fault, a boundary, a contour) gives a **full-height vertical** at the crossing station. Where the section crosses it horizontally is known, the depth is not, so the mark spans the whole frame. A line **with a Z elevation** gives a **point** at the real elevation of the crossing, which covers a 3D object, an inclined one, a surface contour. A roof contour with an elevation, for instance, lands as a point exactly on the bed. A polygon (a plan zone - replacement, a mine field, a licence) gives a **vertical band** over the interval where the section runs through the zone.

The line, vex and frame height come from the section definition - written by **Cross-section along a line**, which now stores the vertical extent, so nothing needs to be supplied for objects without Z. For older definitions without the height a fallback remains, the **section drawing** as the optional input or a Z range in the advanced parameters. When the object has Z, no height is needed, the point is

placed at the elevation. Empty outputs are not created, a fault yields only verticals, a marker only points, a zone only bands.

Unlike **Project objects onto the section** (approximate, corridor-based) this is an exact intersection - a mark appears only where the geometry truly cuts the section line. Several layers can be fed at once, lines and polygons in one list, and all are processed in a single run, like the list of surfaces in 4.04. In the outputs the **src** field keeps the source layer of each mark. The demo generator outputs a fault, a Z marker and a replacement zone that cross the demo section, so the tool can be tried at once.

### Parameters

| Parameter | What it sets | Default / hint |
|---|---|---|
| Section definition | The definition layer from 4.01: the line, vex and frame height. Enough for objects without Z. | - |
| Layers to intersect | Lines and polygons in one list, in a single run. The src field in the outputs keeps the source layer. | - |
| Section drawing | A fallback source of frame height for older definitions without zmin/zmax. | optional |
| Carry the feature attributes onto the section | The fields of the source features go into the outputs, so the marks can be coloured and labelled by your own data. | on |
| Keep the name of the source layer | The output is named after the source layer when a single one is supplied. | off |
| Keep the style of the source layer | The appearance is taken from the source layer rather than from the standard style of the module. | off |
| Terrain line on the drawing | The surface line layer from 4.01. The top of zones and faults follows the terrain rather than the frame. | optional |
| Bottom line on the drawing | The same layer or a part of it. The bottom of zones and faults follows the sole rather than the frame. | optional |
| Field with the bottom elevation of bands and verticals | The bottom of each feature from its own attribute. | optional |
| The bottom value is a depth from the top (Adv.) | Switches the field from an absolute elevation to a depth. | off |
| Bottom of Z range (Adv.) | The lower frame elevation when neither a definition with height nor a drawing is given. | 0 |
| Top of Z range (Adv.) | The upper frame elevation in the same case. | 0 |
| Verticals on the section | Output for lines without Z (a fault, a boundary): a full-height vertical. | created |

| Parameter | What it sets | Default / hint |
|---|---|---|
| Intersection points | Output for lines with Z (a contour with elevation): a point at the real elevation. | created |
| Zone bands on the section | Output for polygons (a plan zone): a vertical band over the interval. | created |

Empty outputs are not created, each object type goes only into its own layer.

### The name and the style of the source layer

Two checkboxes for the case of a single supplied layer. Both are off by default, so earlier runs reproduce unchanged.

**Keep the name of the source layer.** The output is named after the source, so a geology layer gives “Geology on the section” instead of the generic “Zone bands on the section”. If a single layer produced several kinds of features at once, the kind is appended to the name, since otherwise three layers in the tree would carry the same one.

**Keep the style of the source layer.** The appearance is taken from the layer itself rather than from the standard style of the module. Together with carrying the attributes this removes the main piece of manual work on the section. A categorised geology colouring by a field lands on the bands as it is, with no need to repeat the palette by hand. The style is carried over when the geometry type matches, polygons into bands and lines without an elevation into verticals. A line with a Z elevation gives a point, a line style will not fit it, and the standard one stays there.

If several layers are supplied, both checkboxes stay silent, because the outputs merge all the sources into one layer, and a line explaining why goes to the log.

### Clipping by the terrain and by the bottom line

Zones and faults are drawn over the full height of the frame by default. Where the feature crosses the line is known, how deep it goes is not. Three optional parameters remove that limitation at the edges.

**The terrain line on the drawing** clips the features from above. Supply the terrain line layer that **4.01 Cross-section along a line** outputs. The top of zones and faults will follow the terrain, and the upper edge of a band will repeat its breaks rather than stay straight.

The clipping follows the line from the drawing rather than the DEM raster, and that is deliberate. The section may have been built over a different surface, and then the raster and the drawing would diverge, while clipping must follow what the person has in front of them. Sections are matched by the **sec_id** field, so in a batch run each drawing is clipped by its own profile.

**The bottom line on the drawing** clips the features from below and works the same way. Supply the sole, the floor of a seam or any lower surface from the same drawing. With no line supplied the bottom stays on the frame.

If the layer holds several surfaces, clipping follows the envelope, the highest one above and the lowest one below. The **Surface lines on the drawing** output therefore fits as it is and can go into both inputs at once. When one particular surface is needed, supply the layer filtered by the **name** field.

Both edges are computed at the same stations, the nodes of both lines being merged into one set. Otherwise the top and the bottom, each computed at its own stations, would cross between the nodes and the band would come out inverted. The bottom is never raised above the top, and where the bottom line runs over the terrain, the band collapses and the feature is not output at all. The

count of features cut away entirely is printed to the log, and these are usually zones lying above the terrain.

**The bottom elevation field** sets the bottom of bands and verticals per feature, from the attribute of the feature itself. By default the value is read as an absolute elevation. A checkbox in the advanced parameters switches it to a depth from the top of the feature, which is handy when the data holds the thickness of a zone rather than its floor. The bottom is never taken below the frame.

#### Feature attributes on the section

The **Carry the feature attributes onto the section** checkbox is on by default. The point is simple. The bands and verticals on the drawing are coloured and labelled by your own fields, without joining back to the source layer by hand.

Several layers are supplied and their schemas differ, so the columns are merged into one common set. A field a layer does not have stays empty. The same name in different layers counts as one column, and the type is taken from the first layer where it occurred.

Names that clash with the service ones (**sec**, **src**, **label**, **d**, **z**, **d1**, **d2**) are renamed with a suffix. This is not cosmetic. The **d** column carries the distance along the section, and a feature attribute with the same name would silently replace the coordinate, which you would notice only on the drawing. The number of merged columns is printed to the log.

**Dip and dip direction.** A line without an elevation gives a vertical by default, since the position is known and the angle is not. The dip and the dip direction can be given by field name or as constants for the layer. A dip trace is then drawn from the surface downwards instead of the vertical, while a zone band becomes a parallelogram.

What reaches the drawing is not the true angle but the **apparent** one:

```
tan(apparent) = tan(true) · cos(dip direction - section azimuth)
```

A section across the strike gives the true angle, a section along the strike gives zero and the object lies flat. The azimuth of the section is taken from the segment carrying the intersection, so on a bent profile the inclination differs from place to place.

The dip direction is required and does not follow from the geometry, one and the same object may dip either way. Without it the objects stay vertical - the silent assumption that the plane is perpendicular to the section is a common error of construction. The side of the inclination needs no parameter, the sign of the cosine decides it.

The trace length is set horizontally in metres, zero means down to the frame. A short pointer trace is usually taken as one and a half to two centimetres on the sheet, which at 1:2000 is 30-40 m.

Three numbers go into the attributes - **dip** (true), **dip_az** and **app_dip** (apparent). The angle must not be measured with a protractor on the drawing, the vertical exaggeration distorts the drawn inclination, as it distorts everything else on a section.

## 4.06 Intersect a TIN with the section

A raster grid (4.04) is `z = f(x, y)`, one elevation per plan point. It cannot represent an overturned fold at all, because above one point such a fold has several elevations of the same surface. This tool cuts the section through a **TIN** - a surface of true 3D triangles that can overhang.

The mechanics are pure geometry. The section is a vertical curtain along the polyline, and each TIN triangle is intersected with the vertical plane of its segment. That gives a segment in station along the line and real elevation, and all the segments are assembled into the surface trace. Overhang is reproduced without extra steps. Several segments lie at different elevations above one station, and the trace folds - the limbs of an overturned fold come out as they are.

The inputs are layers of **3D polygons** (PolygonZ, TIN faces, with non-triangles fan-split into triangles) and optionally a **mesh layer**. The line and vex come from the section definition, the height

from the faces themselves, so nothing needs to be set for a TIN. Besides the drawing trace you can also get it in real 3D coordinates.

There is an important limit. **A QGIS mesh is 2.5D**, its height is a scalar per vertex, one value above a point again. Overturning is not preserved in a mesh, so overhangs come only from true 3D faces from a geomodeller (Leapfrog, Micromine and the like). A mesh is accepted for generality, on single-valued surfaces. The demo generator outputs an overturned TIN fold - the folding trace is visible on it at once.

### Parameters

| Parameter | What it sets | Default / hint |
|---|---|---|
| Section definition | The definition layer from 4.01: the line and vex. The height comes from the faces themselves. | - |
| TIN faces (PolygonZ) | Layers of 3D polygons - TIN faces from a geomodeller. Overhang and overturning are reproduced. | optional |
| Mesh layer (2.5D) | A QGIS mesh, for generality. One value above a point, overturning is not preserved in a mesh. | optional |
| TIN trace on the section (drawing) | The output 2D trace in the section axes, may fold. | created |
| TIN trace (3D) | The output trace in real 3D coordinates. | on request |
| Closed sections (polygons) | Rings of the section assembled from the segments. | on request |
| Carry over fields of the source object (adv.) | Names separated by commas. They go into the rings. | - |
| Tolerance for closing the rings (adv.) | In drawing units. The ends of neighbouring triangles coincide only approximately. | 0.001 |

Supply at least one of the two inputs - TIN faces or a mesh.

### Chains and closed sections

Cutting gives one segment per triangle, and a shell has tens of thousands of them. Such segments are merged into chains, and one polyline per chain goes onto the drawing, because a layer of separate segments can be neither labelled, nor selected, nor styled.

The chains are assembled **per source**. Every shell is cut by its own triangles, and combining them into one set is not allowed, because neighbouring solids would merge into a single line.

Closed chains additionally come out as polygons. For a shell that is the solid in section, with an area that can be computed and labelled.

A ring appears **by itself** if the chain closes. Closing everything by force is not allowed, and here is why. A closed contour is guaranteed only for a closed shell, while the section of a surface is open by construction and its ends lie on the edge of the surface. That holds for a TIN of terrain, the roof of a seam and any open set of faces. Joining the ends means drawing a chord that is not in the data and computing an area over it that does not exist.

So a chain that did not close by itself stays a line and does not go into the polygons. The log reports the number of chains, how many closed and how many stayed open, and the largest gap between the ends. If a closed shell was cut and the chains came out open, that gap is the hint - raise the tolerance for closing. The ends of segments from neighbouring triangles coincide only approximately, and the tolerance merges them into one node.

An inner ring is assembled separately from the outer one, so the section of a solid with a cavity gives two polygons rather than one.

Fields of the source object are carried into the rings if their names are listed, separated by commas. There may be several layers of faces with different fields, and a field missing from a layer stays empty. The values are written as text so that types need not be unpicked on the output side. Besides them every chain carries a flag of closure `closed` and the gap `gap` between its ends, and the polygons also carry an area in drawing units. The fields of the source are carried into both the lines and the polygons.

## 4.07 Project objects onto the section

The **Project objects onto the section** tool projects points, lines and polygons onto the section line. For each vertex the horizontal coordinate is the distance along the line to its projection, and the height is the elevation from the 3D geometry or from a chosen field. Distant objects are cut off by a corridor.

| Parameter | What it sets | Default / hint |
|---|---|---|
| Section definition | Section line with the vex field (vertical exaggeration). | - |
| Objects to project | Layer whose objects are projected onto the section plane. | - |
| Elevation field | Z attribute if the object geometry has no Z. | - |
| Corridor from the line | Width of the capture band from the section line. 0 = all objects. | 0 |
| Keep the name of the source layer | The output name is built from the name of the source. | off |
| Keep the style of the source layer | The presentation is taken from the layer itself. | off |
| Objects on the section (drawing) | Output: objects in section-drawing coordinates. | - |

This generalises the borehole projection to any objects - anomalies, sampling points, traces, outlines. The result is in the section axes, placed on top of the drawing.

### Every section of the definition

The tool works over all sections of the definition rather than the first one. Every object is projected onto the drawing whose corridor its line fell within. The name of the section and its number stay in the attributes, and the result is split into drawings by them. The drawings are spread apart by the offset from the definition, as in 4.01.

The **Project onto every section** checkbox removes the corridor selection and an object lands on every drawing. This is needed for the section lines themselves and for objects common to the whole area, since by distance they would reach only their own section.

#### The name and the style of the source layer

Two checkboxes for those who have already set up the presentation in plan. The output name is built from the name of the source, and the presentation is taken from the layer itself. A categorised style lands on the section as it is, and the legend stays the same. The same is done in **Vector intersection with the section** (4.05).

The style is carried over when the geometry type of the output matches that of the source. Points are projected as points and polygons as bands, so the presentation fits exactly. A line carrying a Z elevation gives a point on the section, and a line style will not fit it, so the standard style remains and the log reports it.

## 4.08 Unproject from the section

The **Unproject from the section** tool does the reverse. Objects drawn on the section drawing are returned to real coordinates. The horizontal coordinate of a vertex is read as the distance along the line (giving the plan), the height as the elevation Z = height / vex. The line and vex come from the same definition the drawing was built with.

| Parameter | What it sets | Default / hint |
|---|---|---|
| Section definition | Section line with the vex field. | - |
| Objects from the section drawing | Objects in drawing coordinates to return to plan. | - |
| Objects in plan (with Z elevation) | Output: objects in plan coordinates with restored elevation. | - |

So an object drawn by hand on the section gets back into the plan and into 3D with a Z elevation, be it an ore outline, a fault or a boundary.

#### The round trip: correcting a contact on the section

The cycle closes as follows. A geologist sees on the drawing that the contact of a bed is drawn wrongly and draws it correctly as a line in the section axes. The tool returns that line to plan with a Z elevation at every vertex.

The **Return as vertices** checkbox turns the result into points. This is needed because the elevations then go into the building of a surface, where observations are taken as points. In kriging such a point pins the surface exactly at a zero nugget, and in Topo2Raster it becomes a hard node.

The **src** field marks the origin. A contact taken off a section is an interpretation rather than a measurement, and in a common sample with boreholes it has to be told apart. By this field the point is given a smaller weight through the declustering weight field, and a drawing stops weighing as much as a borehole.

#### A pit: every wall at once

At a pit or a quarry a section is built along every wall. The drawings lie side by side on one canvas, spread apart by the offset from the definition. The geologist draws the layers on every drawing from the orthophoto of the wall.

The round trip returns the drawings to plan, and every object lands on the line of its own section. The tool picks the section by the band of the canvas the object was drawn in. The distance along the axis is measured from the start of its own section, and on another line it would point to the wrong place.

The geometry type is preserved, points return as points, lines as lines, polygons as vertical faces. A vertical face looks right in 3D, while in plan it is a band on the section line with zero area, so computations over it are done on the vertices.

The contacts from all the walls are then gathered into one sample. The layer-top points surveyed inside the pit are added to them, and a surface is built over the whole. A contact off a section is an interpretation from a photograph, while a surveyed point is an instrumental measurement. Tag the origin in the **src** field and give it a different weight in the kriging.

## 4.09 Shaft wall unwrap (beta)

The **Shaft wall unwrap** tool builds a cylindrical section. Around the shaft axis at a given radius a circle is taken with an angular step (1 degree by default), and the surface grids are sampled along it. The unwrap lies in axes of arc length along the circle and elevation.

| Parameter | What it sets | Default / hint |
|---|---|---|
| Shaft axis (collar point) | Collar point of the shaft. | - |
| Shaft radius, map units | Radius of the wall unwrap. | 4 |
| Surface grids (markers) | Surface grids whose elevations are drawn on the unwrap. | - |
| Angular step, degrees | Step of the wall traversal by azimuth. | 1 |
| Vertical scale | Vertical-exaggeration mode (H:V ratio or multiplier). | ratio |
| Scale value | Exaggeration number for the chosen mode. | 10 |
| Raster sampling (adv.) | How the value is sampled from the surface grid. | - |
| Wall unwrap (arc × elevation) | Output: shaft-wall unwrap. | - |

Each marker surface gives the line of its intersection with the shaft wall - where the beds dip the lines are tilted and wavy. The axis is set by a collar point layer, the radius is in map units, the vertical scale is as in the section.

## 4.10 Example for a section (demo)

The **Example for a section (demo)** tool prepares a complete training set for the **Cross-sections** group, so its tools can be tried without kriging real data. In the panel it stands last in the **Cross-sections** group.

A single run outputs the whole data set for the section group:

- six stacked surfaces with a dip and variable thickness, five interbedded beds, the second and the fourth industrial and thin;
- three section lines across the area, a polyline with two bends, a short straight one and a slanted one;
- a ready pair of drilling-model layers and an inclinometry table;
- one multiband grid per industrial bed.

Every third hole of the set is inclined, with a zenith angle from 45 to 70 degrees building up downhole and the azimuth along the section lines. A hole inclined across the section would project almost into a point, leaving nothing to check the carrying of intervals along the axis against. The remaining holes are vertical, so that both behaviours are visible on one set. The bed-grid band convention is this. Band 1 is the roof, band 2 the bottom, bands 3 and further the parameters, here the content and the mineral type with a replacement zone. The content fields of the beds are independent and stochastic. One file describes the whole bed - like a block model where new parameters are added

as bands. For the intersection tools it adds demo vectors - a fault without an elevation, a marker contour with Z, a replacement zone, and an overturned TIN fold from PolygonZ 3D faces.

### Boreholes with sampling intervals

The pair of drilling-model layers is produced ready, by the same contract Geoconstructor exports by. Along with it a **demo bed reference** is produced, holding the very codes that stand in the intervals, with the bedding order, the body kind and the colour. The bundled reference from 4.11 is built for the Verkhnekamskoye deposit and does not know the demo codes, so without its own reference the demo columns had no colour.

The demo reference does not colour the bed bands in 4.01, and that is how the lookup works rather than an omission. There a bed is determined by the names of the roof and floor layers, the role is stripped from a layer named **KpII_top**, the code **KpII** remains, and the reference is queried by it. The demo surfaces are named by plain numbers and carry no bed code, so the bands keep their default colours. On your own data, where the layers are named by bed codes, the same reference colours both the bands and the columns. **Collars** is a point layer with the collar elevation and the end of hole. **Intervals** is an ordinary table with depths along the hole from the collar and the bed code.

So tool **4.02 Boreholes on sections** can be tried without your own data and without touching a database. Run 4.10, then 4.01 along any of the three lines, then 4.02 with that pair, the section definition and the demo reference, and the borehole columns get their colour and legend order. The boreholes stand along all three lines, so the columns land on any drawing. The same pair suits three-dimensional viewing unchanged, where the depth goes straight into Z, without a vertical scale or a layout offset.

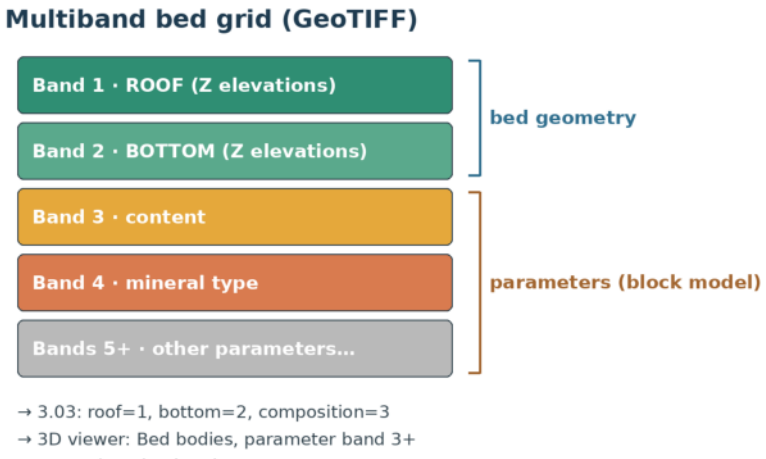



Figure 32: The multiband bed-grid convention: bands 1-2 carry the geometry (roof and bottom), bands 3+ the parameters; one file feeds tool 4.03.

### Demo layers and their attributes

| Layer | Geometry | Attributes |
|---|---|---|
| Section lines (demo) | line | name (Section 1, 2, 3). Three traces of different length: a polyline with two bends (the vertices test the stationing), a short straight one and a slanted one with a single bend. |

| Layer | Geometry | Attributes |
|---|---|---|
| Collars (demo) | points | hole_id, z (collar elevation), eoh (end of hole), number for the label. |
| Intervals (demo) | table | hole_id, from, to (depths along the hole from the collar), code (bed index). |
| Bed reference (demo) | table | code, order, body, color, strata. Matching the demo interval codes. |
| Zone (demo, polygon) | polygon | name. For the vector-intersection tool 4.05. |
| Fault (demo, 2D) | line | name. Crosses the trace, moved off the line bend. |
| Marker with Z (demo, 3D) | line Z | name. Tests 3D geometries in 4.05. |
| Overturned TIN (demo) | polygons Z | name. An overturned fold for 4.06. |
| Surface 1…6 | raster | A single elevation band. |
| 1st/2nd industrial bed | raster | Bands: 1 roof, 2 bottom, 3 content, 4 mineral type. |

The workflow is shown in section 4.01. The surfaces go into **Cross-section along a line**, and the collar and interval pair into **Boreholes on the section**. The composition grid goes into **Bed composition on the section**, and the demo vectors and the TIN into the intersection tools 4.05 and 4.06. The whole cross-section group then runs on consistent data.

**Parameters**

| Parameter | What it sets | Default / hint |
|---|---|---|
| Area (extent) | The rectangle in which the set is generated. | set per project |
| RNG seed (Adv.) | Reproducibility of the set. 0 means random on each run. | 0 |
| Surface 1…6 | Six rasters top to bottom: the roofs and floors of the host beds and the two industrial beds. | created |
| Section lines (3) | Three traces across the area to feed into 4.01. By default the tool builds a section along each of them. | created |
| Borehole collars collar | The drilling-model point layer (hole_id, z, eoh, PointZ). | created |
| Borehole intervals interval | The drilling-model table (hole_id, from, to, code, kcl). | created |
| Composition: content | A content grid of the industrial beds for 4.03. | on request |
| Composition: type/facies | A mineral-type grid (1 sylvinite, 2 replacement) for 4.03. | on request |
| Fault, Z marker, zone | Demo vectors for 4.05: a line without Z, a contour with Z, a zone polygon. | on request |

| Parameter | What it sets | Default / hint |
|---|---|---|
| Overturned TIN | 3D faces of an overturned fold for 4.06. | on request |

**Structural elements and outcrop traces.** The demo also gives two layers with an analytic reference for checking angles.

The structural elements are arranged like this. Three lines cross the straight Section 2 at one true dip of 25 degrees and different dip directions, so the apparent angles come out as 25, 18.25 and zero. The expected value is written into the **app_exp** field, and the result of 4.05 is checked by a number rather than by eye. The straight section is chosen deliberately, because on a bent one the azimuth of the segment changes along the way and the reference stops being unambiguous.

The outcrop traces for 4.12 are a gentle one (20 degrees), a steep one (72) and one deliberately straight in plan. On the first two the tool must return the numbers from the dip_true and az_true fields. The third one checks the refusal, since an attitude is not defined by a straight trace. The steep one is needed separately, because a fit by z would drift on it.

The full list of output-layer fields is in the **Section data** appendix section (“Demo-layer fields” at the end of the manual).

## 4.11 Bed reference template

The tool adds the bundled bed reference template to the project - the very one read by **4.01** and **4.02**. The reason is simple. The template lives as a file inside the plugin directory, where a user does not normally look. Without this step the reference has to be searched for manually, or an arbitrary table gets fed into the section and the drawing comes out without colour.

The tool has no parameters. One run, and the **Bed reference (template)** layer appears in the project and is immediately visible in the drop-down of the section tools.

### What is inside

A table without geometry, one row per bed or interbed, top to bottom down the section. The columns are the body code, the number from the top, the body kind (bed or interbed) and the colour. The optional strata and note columns are read but not used yet.

The template is built for the Verkhnekamskoye deposit, 37 rows from the cover deposits down to the lower rock salt. For another deposit it serves as a skeleton. Save the layer to a file by the standard QGIS means and edit the codes, the order and the colours for your own stratigraphy.

### The body is filled in by the geologist

Let us repeat the main point about the reference. The body cannot be derived from the code. The bed **АБ** looks like “А plus Б” by its spelling, yet it is a single bed containing А, Б, the А-А’ parting and А’. The interbed **Б-В** cannot be derived from the code at all, because there is no bed Б in the list, there is АБ. The tool does not recompute the body values and in that sense is no wiser than the person who entered them.

### Composite bodies are grey

The template holds **КрIIIа+б**, a composite body, the sum of КрIIIа, the КрIIIа-б interbed and КрIIIб. Such conglomerates are painted grey in the reference, the same shade the plugin uses to mark the unrecognised. The reason is that the colour of a composite body does not follow from the colours of its parts. They may differ, and any choice would be arbitrary. Grey reads as “colour not set”, and it is the person who sets it.

Holding the whole next to its parts in one list has a price. The top-to-bottom order is single, so between **KpIIIб** and **KpIIIв** there are now two bodies rather than one. Such a pair of boundaries on a section yields a grey band with a list instead of the interbed name. At the coarse granularity, between **KpIIIa+б** and **KpIIIв**, everything works out as usual.

### The same file outside QGIS

The template lives in the **templates** folder inside the plugin directory in two forms, `plast_reference_vkmks.xlsx` and `plast_reference_vkmks.csv`. The Excel workbook has a second sheet with the filling rules. Edit it any way you find convenient, the section tools accept a project table, a GeoPackage and a CSV alike.

## 4.12 Attitude from an outcrop trace

Computes the dip and the dip direction from a three-dimensional trace of an outcrop.

| Parameter | What it sets | Default / advice |
|---|---|---|
| Outcrop traces | Lines with elevations (Z in the vertices). | - |
| Window, vertices | 0 - one attitude per trace, otherwise a sliding window. | 0 |
| Minimum curvature (adv.) | The conditioning threshold below which a refusal follows. | 0.15 |

**Where the attitude comes from.** When the boundary of a bed or a fault is digitized with elevations, the plane through it is defined uniquely. Both elements of the attitude follow from that plane. The three-point rule is the minimal case, here the computation runs over all the vertices at once.

The method follows Allmendinger. The normal to the plane is the eigenvector of the orientation matrix belonging to the smallest eigenvalue. A fit of the form `z = a·x + b·y + c` is simpler but falls apart on steep attitudes, where the coefficients run to infinity. Through the eigenvectors steepness stops being a special case.

**A trace straight in plan does not define an attitude.** Infinitely many planes pass through one straight line in space, they rotate about it like pages about a spine. The tool sees this through the ratio of the eigenvalues and refuses with a reason instead of a confident number out of rounding noise. The measure goes into the **planar** field, where zero is points on a line and one a spread in two directions.

**A fold is caught by the residual.** The method assumes the trace lies on one plane, and a long boundary rarely does, there is a flexure, a displacement by a fault, an inflection. The mean residual goes into the **rms** field, and a large value means that no single attitude exists for the whole trace. Then turn on the window, the attitude is computed over a sliding stretch and the output carries its change along the boundary.

**Place in the chain.** The output feeds straight into 4.05, which expects exactly the dip and dip_az fields. If the terrain is given by contours and the trace is flat, put it through 2.16 first - the elevations will be taken off the adjoining contours.

## 4.13 Bed grades at the collars

Reduces the assay samples to a single number per hole and puts it into the point of the collar. This is the input for kriging, and from there the grade map is built by 1.01, like any other map, with no interpolation of its own needed here.

The tool closes the chain from the database to the map. The samples come out of the database as the `chem` table, and the section is coloured by them in 4.02. Here the same samples turn into points from which the areal distribution is computed.

### Why the join goes by depths

The sampling grid and the lithology grid rarely coincide. A sample may lie inside a bed, cross the boundary of two, or span several. A sample is taken from the core, while the boundaries are drawn from the description. Joining by record number is therefore impossible, and the join goes by depths from the collar.

The weight of a sample is the length of its overlap with the bed. A sample that falls into the bed by half weighs half of its length. If the bed is met by the hole twice, both intervals are reduced to one number, and the weight of each is the length backed by data.

### A missing grade is not a zero

This is the rule the tool is shaped around. Companies differ in the set of components, some have bromine and thickness, others do not. An empty cell means the component was not determined, not that it is absent in the rock.

A sample without a value for a component therefore drops out of the average together with its weight. Substituting zero would give a knowingly understated average, the more so the rarer the component is. Feeding such a number into a discriminant function means getting a confident and wrong answer.

Along with the average, the coverage goes into the layer for each component separately, that is which share of the bed thickness is backed by data for that particular one. For potassium chloride the coverage may be full and for bromine only half, and a single common number would hide that.

### Parameters

| Parameter | What it sets | Default / advice |
|---|---|---|
| Collars (collar, points) | The point layer of the collars. | - |
| Intervals along the hole (interval, table) | The lithology with bed codes. | - |
| Assay samples (chem, table) | The samples with depths and grades. | - |
| Grade fields | Which components to compute. | chosen from the fields of the sample table |
| Bed code | One code from the code field of the intervals. | for example KrII |
| Smallest sample coverage of the bed (Adv.) | The threshold for marking the unreliable ones. | 0.5 |
| Grades at the collars | The output point layer. | - |

The identifier and depth fields are found by their expected names automatically, and have to be set by hand only on non-standard data.

### What the output layer holds

The layer holds the collar point, the bed code, the thickness of the bed along the hole, the sample coverage and the number of samples that touched it. Then come two fields per component, the grade itself and its coverage (`cov_`).

Values with low coverage are not thrown out of the layer, because deciding for the user which coverage is enough would be wrong. Filter by the **cover** field before feeding the layer into kriging, and the tool writes to the log how many such holes there are.

Overlaps between samples are counted and go into the summary. Two samples at one depth are a dispute in the data, and keeping quiet about it is not an option.

## Geological model

The group brings together tools that work not with a single surface but with a stack. The roofs and floors of neighbouring bodies are tied to one another, and that tie is either kept or broken. Checking it is the first task. The second is to have available material whose answer is known in advance, on which you see where the grid representation works and where it stops working.

# 5. Geological model (beta)

## 5.01 Consistency of a bed stack

Roofs and floors are usually built separately, every surface is interpolated from its own measurements and knows nothing about the neighbouring ones. While the beds are persistent this goes unnoticed. In a pinch-out zone, where the thickness goes to zero, two independently built surfaces almost inevitably intersect. After that the arithmetic over such a stack gives negative thicknesses and volumes, and the section shows inverted bands.

The tool answers whether such places exist and where exactly.

### What is counted

**Pinch-out** - a thickness within the tolerance of zero. This is geology rather than a defect, and it goes into a separate count.

**Negative thickness** - the roof of a bed lies below its floor.

**Overlap of neighbours** - the floor of the upper bed has dropped below the roof of the lower one. This check is separate and does not reduce to the previous one, since every bed on its own may be sound while together they overlap.

**The smallest gap** between neighbours is produced as a map, not only as a number. The number says how much, the map says where, and the second matters more, because that is where the overlap will appear at the next recomputation of the surfaces.

**The sign of a reversed order** fires when the overlap has taken almost the whole area while the beds themselves are clean. Geology does not look like that, a real overlap is local and sits next to a pinch-out. Almost certainly the beds have been supplied bottom up.

The pinch-out tolerance is set by a parameter and by meaning equals the accuracy of the surfaces. With a zero tolerance the numerical noise in the zone of convergence will spill into errors.

### Where the order of beds comes from

The order of occurrence from top to bottom is taken from the layer tree of the project, and if a bed reference is supplied, from it. The order in which the files were picked in the dialog means nothing, QGIS does not preserve it and it cannot be relied upon. Where the order came from is printed to the log.

The reference is applied only if it describes the supplied beds. A foreign reference that matches no code is ignored with a message instead of imposing a random order.

### What comes out

Out come a raster of zones with codes - consistent, pinch-out, negative thickness, overlap - and a map of the smallest gap to the neighbouring bed. For every bed the log gets the range of thickness and the areas of the zones, and for every pair of neighbours the smallest gap and the area of the overlap.

Areas are printed in hectares rather than in cells, because a thousand cells on a one-metre grid and on a thirty-metre one differ by a factor of nine hundred.

### What the tool does not settle

A negative thickness comes from two origins, and arithmetic cannot tell them apart, an error of interpolation or an overturned bed on a fold. The tool marks the places and prints the numbers but passes no verdict. It is the drilling data that tells them apart. Codes along the hole that do not follow the stratigraphic order are the sign of overturning, and that is read straight from the drilling model.

Telling an erosional truncation from a pinch-out is likewise impossible without the dissolution surface on the input. At a truncation the thickness breaks off on the mirror, while at a pinch-out the roof and the floor converge on each other, and on the map of zones both look the same.

## 5.02 Example section (demo)

The tool builds a teaching section of the Verkhnekamskoye type with all the cases for whose sake the rest of the group exists. It is useful because the answer is known in advance, and any disagreement with it is an error of the tool rather than of the data.

### What is in the example

**The column** is taken from the bed reference in full, thirty-six bodies from top to bottom. First the cover deposits, the variegated and the terrigenous-carbonate sequences, the salt-marl sequence. Then the salt with the transition unit, the cover rock salt, the carnallite zone with its interbeds. Then the sylvinite beds, the underlying salt, the marker clay and the lower salt. Interbeds here are bodies just like beds.

**A recumbent fold** in the middle of the area. At its hinge a vertical borehole crosses one and the same body three times, in the horizontal limb, in the upper arc and in the lower one. The neighbouring body is penetrated twice. Above a point in plan there are several roofs, and the elevation ceases to be a function of two coordinates. No density of the network will fix that.

**A pinch-out** brings the thickness of one bed to zero by a smoothed step between two boundaries. The boundaries are produced as lines - the very input needed to build a thickness trend.

**A salt dome with a dissolved top** stands aside from the fold. The dome lifts the salt as a whole, and the dissolution surface cuts away what has been lifted. It is not one body that is cut but the whole column down to the level of dissolution. The mirror is the roof not of an appointed body but of the one that survived. Aside from the dome that is the transition unit, over the crest the cover salt.

The salt-marl sequence settled on the dissolved salt and fills everything up to the base of the terrigenous-carbonate one. Over the dome it is thin, aside from it full.

### The grids in the fold zone are wrong on purpose

The surfaces are built from the first penetration from the top, that is exactly as a person would assemble them from borehole measurements without noticing the overturning. In the fold zone they are invalid by construction, and this is deliberate, since 5.01 must show overlaps and negative thicknesses on such grids. The outline of the zone is produced as a separate layer so that it is not mistaken for an error of the tool.

### What comes out

The surfaces are written into a folder, one file per contact, and are loaded into a group in the order of the column. Thirty-seven separate rows in the dialog would be unopenable, so the parameters carry a single folder.

Besides the surfaces the tool produces everything needed to work with a stack. These are the dissolution mirror and the map of what comes out under it. Next comes the thickness of the water-protective sequence from the roof of bed B to the mirror. With it come the collars and intervals of

the drilling model, with the number of the penetration in the entry field. Then come the pinch-out boundaries as lines, the outline of the fold zone and the bed reference with codes, order, kind of body and colour.

Everything is deterministic, the same parameters give the same section.

#### How to use it

Build the example, feed its surfaces into **5.01** and make sure that the overlap is found in the fold zone and the pinch-out where the boundaries are set. Then run a section line across the crest of the dome with **4.01**, supplying the bed reference. The bands break off on the mirror, over the crest the upper body is absent altogether and the next one is truncated. This is the picture an erosional contact gives, and it is convenient for checking how sections behave on real data.

## 5.03 Assembly of a stack from the relief

The column is built from the top down, body after body along the reference. For every body the thickness is interpolated from the boreholes, the floor is obtained by subtracting the thickness from the overlying surface, and it serves as the top for the next one.

#### Why this is reliable

If the thicknesses are non-negative, the surfaces cannot intersect, they are obtained by subtracting one from another. Consistency comes out by construction rather than by a check afterwards. This is the main merit of the scheme, and it matters more than its simplicity.

#### What is paid for it

The error of every thickness adds up with the previous ones, and for the lower bodies the accumulated deviation can exceed the thickness itself. The scheme suits shallow stacks and engineering tasks, where the upper contact is the daylight surface itself.

For deep stacks supply a traced contact as the datum and say whose roof it serves as. Then the assembly goes from it in both directions, downwards by subtracting and upwards by adding, and the error is halved.

#### What counts as thickness

The thickness is taken along the hole and reduced to the vertical by the dip. In an inclined borehole the penetrated thickness is greater than the true one, and without a correction the stack swells. The angle is read from a field at the collar, without it the hole is treated as vertical.

The absence of a body in a borehole is a zero rather than a gap. Between such a borehole and its neighbour the thickness goes to zero, and the body pinches out on its own.

#### Contact lines

Where a body reaches the surface, the thickness of everything above it is zero by the definition of the boundary, and this value is known rather than estimated. The vertices of the line are thinned with a step of about the cell, because the density of the digitizing must not turn into weight.

## 5.04 Correction of a stack by the statistics of thicknesses

Fixes an inherited stack that nobody is going to reassemble from scratch. The statistics of the thickness of every body is computed from the measurements, and a confidence interval is taken. The thicknesses on the grid are cut by its bounds, and everything beyond them is replaced by the bound itself. From the corrected thicknesses the stack is reassembled.

The number of cells does not change, unlike with discarding, - the place remains, only the value is disarmed. In statistics the move is called winsorizing.

The cutting separates an artefact of interpolation from geology. A thickness outside the interval on a persistent body is almost certainly an excursion of the method between boreholes rather than a real thickening. Where a body changes its thickness in a regular way, the interval is wide and the correction does not fire.

The statistics come from the measurements if they are supplied. Without them the interval is computed from the grid itself, and that is weaker, because the grid already holds the artefact we are looking for. The tool warns about it.

## 5.05 Model manifest

Records what role every layer of the project plays. The roles are the datum surface, a contact of a body, the mirror, the collars, the intervals and the axis surveys, then the reference, the ground observations, a gauging section and a cross-section.

After the assembly the project turns into a set of rasters and tables with no explicit structure. What is what is known to the user, but that knowledge does not travel with the project. The manifest lives in the custom properties of the QGIS project - the regular place, which is saved in the project file and travels with it.

There is a rule that matters more than the others. The tools read the manifest but do not require it. If the roles are there, the inputs are found on their own. If they are not, everything works by explicit choice. The manifest shortens the path rather than becoming a condition of work.

The roles are guessed from the names of the layers and produced as a table for checking, because a name guarantees nothing. Roles written by other modules are kept, since the manifest is shared and removing the unfamiliar would break the work of neighbouring modules.

Surfaces built by the bed assembly (5.03) are recognized without guessing. The assembly writes them out as files and marks every file with its role, so the mark travels with the file. The layer can be renamed, moved to another project or handed to a colleague, and the role stays with it. Previously such surfaces were named “01_CODE” and matched no hint by name, which meant that the module left its own outputs unmarked.

The sources of a role are taken in order of seniority - first the project manifest, then the mark in the file, then the guess by name. A decision made by a person and written into the manifest is overridden neither by a file nor by a guess.

## 5.06 Folding of a surface

Computes how crumpled a surface is, from the spread of elevations around the local slope.

Folding is measured on the surface rather than on the thickness, because a cylindrical fold with a constant true thickness exists and is invisible in the thickness altogether.

A plain variance of elevations will not do. On a general slope it is large everywhere, and a quiet hillside gets the same score as a crumpled zone. Therefore a plane is fitted in every window and the spread of the residuals around it is computed, so the slope goes and the sinuosity remains. The trend is linear on purpose, a quadratic one would absorb part of the folding itself.

One window decides nothing. The variance depends on scale, and the proper measure is the rate of its growth with the size of the window. The slope in a double logarithm is that rate. On a quiet surface it is about zero, on a crumpled one noticeably larger.

The residual of the surface is produced separately and serves as an input for the fractal dimension. On a detrended surface the slope of the variogram describes the sinuosity rather than the general dip of the bed.

The thickness from the database stays a second, independent sign of folding. It is the penetrated thickness rather than the true one, and on a limb it grows the more steeply the steeper the dip.

### Hydrology

The group follows the order of the task. Tools 6.01-6.07 work on the terrain model. They build flow directions and accumulation, the river network, basins and the catchment of a gauge point, a ditch or an open pit, and compute the catchment morphometry for a report. The terrain model for them is prepared in the «Topography» group, by DEM download (2.01) and terrain preparation (2.04).

Tools 6.08-6.11 answer the question by which flooding is justified. How high does the water rise at a given discharge, and what discharge passes at a given level. The link between discharge and level is built for a cross-section and is called a rating curve.

The task came from practice. Hydrologists build such curves in programs with manual entry. The distance and elevation pairs are entered as a table for every section, and the coefficients and slopes are set manually. Meanwhile all of that is data already present in GIS. The distances are held in the geometry of the section, the elevations in the soundings, and the slope between sections is computed from the chainage.

The branch stands on its own and does not overlap with the geological model, the only thing they share is the machinery of profiles.

# 6. Hydrology

## 6.01 Flow and accumulation (D8)

Computes flow directions over eight neighbors (D8, Jenson-Domingue) and accumulation, that is how many cells drain into each one, itself included. Directions are coded as in ArcGIS - E=1, SE=2, S=4, SW=8, W=16, NW=32, N=64, NE=128, sink=0, nodata=255.

| Parameter | What it sets | Default / hint |
|---|---|---|
| Input DEM | Terrain raster. | - |
| Fill depressions before computing | Hydrological correction before D8, otherwise flow stalls in pits. | on |
| Slope epsilon for filling, m (adv.) | Minimum slope on flats. | 0.001 |
| Flow directions (ESRI codes) | Output: byte direction raster. E=1, SE=2, S=4, SW=8, W=16, NW=32, N=64, NE=128, sink=0. | - |
| Accumulation, cells | Output: float32, number of upstream cells. | - |

The border semantics are deliberate. A cell on the grid frame leaves the grid only when it has no lower neighbor inside, otherwise flows running along the edge would break. A shore cell, on the contrary, pours into an adjacent nodata (the sea, a cutout) even when a lower land neighbor exists.

The **Fill depressions before computing** checkbox is on by default, since on a raw DEM flow stops in pits and accumulation breaks. The computation is fully vectorized, a 2000×2000 grid takes seconds.

## 6.02 River network

Extracts the river network from a DEM. Cells with accumulation at or above the threshold are linked from heads and junctions downstream.

| Parameter | What it sets | Default / hint |
|---|---|---|
| Input DEM | Terrain raster. | - |
| Accumulation threshold, cells | Minimum catchment cells to start a channel. A denser network = a lower threshold. 1000 at 30 m is about 0.9 sq km. | 1000 |
| Fill depressions before computing | Hydrological correction before tracing. | on |
| Slope epsilon for filling, m (adv.) | Minimum slope on flats. | 0.001 |
| River network | Output: lines with order (Strahler), acc_out, length_m. Vertices point downstream. | - |

The **accumulation threshold** is set in cells and means the catchment area where a river starts, that is the head catchment area divided by the cell area. For a 30 m DEM a threshold of 1000 starts rivers at a catchment of about 0.9 sq. km. A smaller threshold gives a denser network.

The output fields are **order** (the Strahler order, 1 at heads, growing where two equal orders merge), **acc_out** (accumulation at the link outlet) and **length_m** (the length). Line vertices run downstream, so the layer fits 2.03 as streamlines without preparation and compares against OSM watercourses from 2.02. Overlaying the extracted network on the real one is a quick DEM quality check.

## 6.03 Basins and watersheds

Divides the territory into drainage basins, polygon boundaries are the watersheds.

| Parameter | What it sets | Default / hint |
|---|---|---|
| Input DEM | Terrain raster. | - |
| Pour points | Basin outlet points. Empty = basins from mouths by threshold. | empty |
| Mouth accumulation threshold, cells | Threshold for mouths when no points are given. | 1000 |
| Fill depressions before computing | Hydrological correction before the computation. | on |
| Point snap radius, m (adv.) | An outlet point snaps to the accumulation maximum within this radius. | 150 |
| Slope epsilon for filling, m (adv.) | Minimum slope on flats. | 0.001 |
| Basins (polygons) | Output: polygons with basin, area_m2. | - |
| Basins (label raster) | Output: integer basin-label raster. | - |

Two modes. With **pour points** every point snaps to the cell with the highest accumulation within the snap radius and then gathers the whole catchment above itself. Without the snap, a point

placed by eye next to a river would collect a tiny hillslope basin. Without points the basins are built automatically from mouths, that is from cells where flow leaves the grid with accumulation at or above the threshold. Cells outside every basin get label 0 and are not exported to polygons.

The output fields are **basin** (the basin number) and **area_m2** (the area by cell count). A label raster can be written additionally. Labeling runs by pointer jumping over the flow graph, so even long winding catchments take a fraction of a second.

## 6.04 Gauge point report

The tool computes watershed morphometry from a **gauge** - a closure point on a stream. This is a classic task of engineering hydrology and site surveys, basin characteristics from a given point. Tools 6.01 - 6.03 give flow, the river network and basins over the territory as a whole, while 6.04 answers the question about one specific gauge.

The fields of the source feature are carried into the output in the same way as in 6.03. Pick the ones you need in the advanced parameters, and on a name clash the incoming field gets the src_ prefix.

### How it works

Every gauge point is snapped to the cell of highest accumulation within the snapping radius. The gauge can therefore be placed by eye next to the thalweg, without hitting a stream cell with the mouse. The full watershed is collected from the snapped cell, and zonal statistics are computed over it. The main stream is traced upstream cell by cell towards the highest accumulation, until a cell without inflows.

Watersheds of neighbouring gauges on one stream nest into each other, and every gauge gets its full basin rather than a remainder below the upper one. This is what sets 6.04 apart from 6.03, where the territory is split into non-overlapping basins.

### What is computed

| Value | Field | Units |
|---|---|---|
| Basin area | area_km2 | sq. km |
| Mean elevation | z_mean | m |
| Minimum elevation | z_min | m |
| Maximum elevation | z_max | m |
| Mean basin slope | slope_deg | degrees (Horn 3x3) |
| Gauge elevation | z_gauge | m |
| Main stream length | stream_km | km |
| Stream fall | fall_m | m |
| Mean stream slope | slope_ppm | permille |
| Mean slope of the hillsides Isk (SP) | sp_slope_ppm | permille |
| Total length of the **contours** | sp_iso_km | km |
| Weighted mean slope of the stream (SP) | sp_stream_ppm | permille |
| Length of the stream **network** in the catchment | net_km | km |
| Stream network density | net_dens | km per sq. km |
| Cells in the basin | cells | count |

The five fields at the bottom of the table are filled only when the matching parameters are on. The three **sp_** fields need a contour interval, the two **net_** fields need an accumulation threshold, and otherwise they stay empty.

A value that cannot be computed is written as null rather than zero, because zero is a measurement and null is the absence of one.

### Three lengths that are easy to confuse

Three length fields sit next to each other in the attribute table, and the short names do not tell them apart. The table headers carry the names in words, but an export to CSV and any expression keep the field names, so the difference is spelled out here.

**stream_km** is the length of the **main** stream, from the gauge upstream by the highest accumulation to the source. One line, tributaries not included.

**net_km** is the length of the **whole** stream network inside the catchment, the main stream together with its tributaries. Filled when an accumulation threshold is given.

**sp_iso_km** is the length of the **contours**, not of the streams. It is the Σli factor of the SP formula for the slope of the hillsides. It is two orders of magnitude larger than the network, because contours run over the whole area at every step in elevation. For a catchment of 54 sq. km at a 5 m interval it comes out around 2800 km, and that is correct.

### Parameters

| Parameter | What it sets | Default / hint |
|---|---|---|
| Input DEM | Relief in metres in a metric CRS. | - |
| Gauge points | A point layer, one point per gauge. | - |
| Contour interval for the SP computation, m | Switches the sp_ fields on. Zero switches them off. | 10 |
| Accumulation threshold for the stream network length, cells | Switches the net_ fields on. Zero switches them off. | 0 |
| Snapping radius, m (adv.) | Search window for maximum accumulation. | 150 |
| Fill depressions | Relief preparation before the flow routing. | on |
| Slope epsilon, m (adv.) | Fill slope, as in 2.04. | as in 2.04 |
| Gauge watersheds (polygons) | Polygons with the attributes listed above. | created |
| Gauge report (HTML) | A table per gauge. | created |

The key numbers for every gauge are echoed to the Processing log in two lines, so the result can be read without opening the attribute table.

### A quick check on the demo

Run **2.07 Demo relief**. Besides the raster it outputs the **Gauge points (demo)** layer, three points near the thalwegs shifted aside from the stream. Feed the relief and those points into 6.04 with the default snapping radius. The polygons will follow the watersheds, and in the report the gauges on one stream will share the fall while the slope grows upstream.

### Scope

The tool computes basin morphometry and nothing else. Discharges, runoff moduli, hydraulics and snowmelt are deliberately out of scope, that is computational hydrology by the codes of practice, a separate topic. Units are assumed metric, a DEM in metres in a metric coordinate system.

Catchments are built from the topology of the relief. On terrain without clear flow boundaries - flat floodplains, hydraulic transfers and backwater - the result should be verified by hydrodynamic modelling.

### Computation to SP 33-101-2003

Hydrologists compute runoff by a normative method, and two of its quantities are defined by formulas that do not match the physical ones. The **contour interval** parameter switches their computation on, zero switches it off.

**The mean slope of the hillsides Isk** is computed by

Isk = h · Σli / (2A),

where h is the contour interval, Σli the total length of the contours within the catchment and A its area. No contours are built for this, the length is counted from the crossings of a level between neighbouring cells and goes into the **sp_iso_km** field.

Isk is not the physical slope. On a plane the formula gives half the tangent, because the two in the denominator accounts for the two-sidedness of the hillsides, the water runs to the stream from both flanks of the valley. The quantity makes sense only within the SP method, so it sits in a separate field beside the physical slope rather than instead of it.

**The weighted mean slope of the stream** is a product over the reaches between the inflection points of the profile. Each partial slope is raised to the share of its length. On a straight profile it equals the plain fall over length. On a broken one it comes out lower, since a river with a steep head and a gentle lower course is not described by a single ratio.

For comparison the physical quantities stay in the output as well - **slope_deg** is the mean slope of the cells by Horn’s method, **stream_ppm** the fall of the stream divided by its length.

### The length of the stream network

The **accumulation threshold for the stream network length** parameter switches on the **net_km** and **net_dens** fields, zero switches them off.

The network is the set of cells where the accumulation has reached the threshold, that is the same definition of a stream as in **6.02 River network**. The length is counted over the links of the flow grid, with a diagonal link weighing the square root of two cell sizes. The figure therefore matches the length of the lines 6.02 would build at the same threshold.

A link counts only when both of its cells are inside the catchment. A link that leaves the boundary already belongs to the neighbouring catchment, and charging it here would count one river twice.

The threshold decides what counts as a stream, and the length of the network changes with it severalfold. At a small threshold the network grows into every hollow, at a large one only the major streams remain. A figure without the threshold beside it therefore means nothing, and the log prints the threshold next to the length.

**The network density net_dens** is the length of the network divided by the area of the catchment, in kilometres per square kilometre. It is comparable between basins, unlike the length itself.

## 6.05 Catchment of a line or an outline (ditches, open pits)

The tool computes the catchment area of an intake - a hillside ditch, a chute, a road gutter or the outline of an open pit. The question is how much area the intake intercepts when the intake itself is not on the DEM yet.

The fields of the source feature are carried into the output in the same way as in 6.03. Pick the ones you need in the advanced parameters, and on a name clash the incoming field gets the src_ prefix.

### How it works

Burning the trace into the relief is not needed for that. The trace is rasterised into grid cells, all of them are taken as intakes, and the catchment is the set of cells whose flow path arrives at any cell

of the trace. The trace may be a polyline, may cross a divide and may run outside the DEM frame - the outside part simply does not take part.

Rasterisation steps along the segments by half a cell, so there are no gaps at bends and diagonals through which water could slip past the intake.

Catchments of neighbouring traces nest into each other, so the ditch further downslope also gets what the upper one intercepts. This is the same behaviour as for gauges in 6.04, and it is the right one - every trace gets its full catchment rather than a remainder.

### An outline instead of a line

Both lines and polygons are accepted as input. A polygon is treated as an intake in its entirety, the outline and the whole area inside it.

For an open pit this is essential. Inside it the pit is a depression, and once the depressions are filled the flow directions there become arbitrary - the traversal order of a flat filled surface assigns them at random. Take only the line of the pit wall as the intake, and some of the inner cells will flow past it along those arbitrary directions, dragging the outer ones with them. Relying on the whole interior removes the question. Whatever happens to the directions inside the pit, its entire area is already in the catchment, while outside the water is traced along the real relief.

The holes of a polygon enter the intake on equal terms with the rest of the area - there is no external relief inside the outline. A multipolygon is processed part by part. There is no longer any need to trace the wall with a line, the ready outline is supplied instead.

### Burning the trace

The separate **Burn the trace into the relief** checkbox answers a different question - will the ditch hold the flow if it is shallower than the local landforms. Burning lowers the relief along the trace by a given depth, and the flow is then computed on the changed relief.

Burning changes the hydrology deliberately, so it is off by default and the result depends on the depth. The fact of burning and the depth are printed to the log and go into the HTML report, so the figure cannot be taken for a computation on the original relief.

### What is computed

| Value | Field | Units |
|---|---|---|
| Catchment area | area_km2 | sq. km |
| Mean elevation | z_mean | m |
| Minimum elevation | z_min | m |
| Maximum elevation | z_max | m |
| Mean catchment slope | slope_deg | degrees (Horn 3x3) |
| Trace or outline length | trace_km | km |
| Intake area | seed_km2 | sq. km |
| Intake cells | trace_cells | count |
| Cells in the catchment | cells | count |

### Parameters

| Parameter | What it sets | Default / hint |
|---|---|---|
| Input DEM | Relief in metres in a metric CRS. | - |
| Traces and outlines | Lines or polygons, one feature per intake. | - |

| Parameter | What it sets | Default / hint |
|---|---|---|
| All features as one catchment | A joint catchment instead of separate ones. | off |
| Burn the trace into the relief | The flow-holding check. | off |
| Burn depth, m (adv.) | How far to lower the relief along the trace. | 2.0 |
| Fill depressions | Relief preparation before the flow routing. | on |
| Slope epsilon, m (adv.) | Fill slope, as in 2.04. | as in 2.04 |
| Trace catchments (polygons) | Polygons with the attributes listed above. | created |
| Trace report (HTML) | A table per trace. | created |

**A quick check on the demo**

**2.07 Demo relief** outputs the **Ditch traces (demo)** layer: a hillside ditch above a strong thalweg and a gutter further downslope, both polylines with a bend. Feed the relief and that layer into 6.05. The catchment should lie upslope of the traces and stop at the divides rather than at the grid frame.

**Scope**

The tool answers the question about area, not about discharge. Discharges, runoff moduli and the capacity of the ditch belong to computational hydrology by the codes of practice and are out of scope. Units are assumed metric.

Catchments are built from the topology of the relief. On terrain without clear flow boundaries - flat floodplains, hydraulic transfers and backwater - the result should be verified by hydrodynamic modelling.

## 6.06 Report on an existing catchment

The tool computes the morphometry over **existing** catchment polygons. It gives the area, the mean and extreme elevations, the mean slope of the basin, the length of the main stream and the fall.

It is needed where a catchment already exists, drawn manually, taken from somebody else’s project or computed by another program. The boundaries are not recomputed - the figures are for the polygon that was supplied, otherwise the answer would be about a different catchment.

| Parameter | What it sets | Default / advice |
|---|---|---|
| Input DEM | The elevation raster the characteristics are computed over. | - |
| Catchment polygons | The existing outlines to compute over. | - |
| Outlet points | Gauges. Empty - the point is taken by the highest accumulation inside the polygon. | empty |
| Compute the stream length and the fall | Requires tracing over the DEM and costs noticeably more than the rest. | on |
| Contour interval for the SP computation, m | Switches the sp_ fields on. Zero switches them off. | 10 |
| Accumulation threshold for the stream network length, cells | Switches the net_ fields on. Zero switches them off. | 0 |

| Parameter | What it sets | Default / advice |
|---|---|---|
| Snapping radius (adv.) | Moves a gauge to the cell with the highest accumulation nearby. | 0 |
| Fill the depressions (adv.) | Without filling the water stops in the local pits of the DEM. | on |
| Catchments with characteristics | Output: the same polygons with the computed fields appended. | - |

### The fields of the output layer

| Value | Field | Units |
|---|---|---|
| Catchment area | area_km2 | sq. km |
| Mean elevation | z_mean | m |
| Minimum elevation | z_min | m |
| Maximum elevation | z_max | m |
| Mean slope of the catchment | slope_mean | degrees (Horn 3x3) |
| Elevation of the outlet | z_gauge | m |
| Main stream length | stream_km | km |
| Stream fall | stream_fall_m | m |
| Mean stream slope | stream_ppm | permille |
| Mean slope of the hillsides Isk (SP) | sp_slope_ppm | permille |
| Total length of the **contours** | sp_iso_km | km |
| Weighted mean slope of the stream (SP) | sp_stream_ppm | permille |
| Length of the stream **network** in the catchment | net_km | km |
| Stream network density | net_dens | km per sq. km |
| How the outlet was chosen | outlet | - |
| Cells in the catchment | cells | count |

The **sp_** fields are filled when a contour interval is given, the **net_** fields need an accumulation threshold, and the stream fields need the length checkbox on. Otherwise they stay empty, and empty is the absence of a measurement, not a zero.

Three lengths in this table are easy to confuse, and the difference is large. **stream_km** is the length of the main stream, one line from the outlet to the source. **net_km** is the length of the whole network including tributaries. **sp_iso_km** is the length of the **contours**, the Σli factor of the SP formula for the slope of the hillsides. It is two orders of magnitude larger than the network, because contours run over the whole area at every step in elevation. For a catchment of 54 sq. km at a 5 m interval it comes out around 2800 km, and that figure is right rather than broken.

The attribute table headers carry the names in words, but an export to CSV and any expression keep the field names.

### The outlet

The stream length and the fall are measured from the point the water flows to. When a catchment is built that point is known from the gauge, an existing polygon has none.

If points are given they are used, and a gauge is looked for inside every polygon. If not, the cell with the highest accumulation inside the outline is taken.

The highest accumulation rather than the lowest elevation. The lowest point may lie in a pit inside the area, or on an edge that caught a neighbouring valley, and the stream would then run the wrong way. The **outlet** field records which way it went for every polygon.

The stream is cut by the boundary of the polygon, and the length is measured inside the given catchment rather than along the whole trace, which may run beyond it.

The fields of the source polygons are kept, the computed ones are appended. If a name clashes, a suffix is added to the computed field.

#### Computation to SP 33-101-2003 and the stream network length

The same quantities are computed by the same formulas as in **6.04 Gauge point report**. The **contour interval** parameter switches on the mean slope of the hillsides Isk and the weighted mean slope of the stream. The **accumulation threshold** parameter switches on the length of the stream network and its density. The formulas and the meaning of every field are explained there.

There is one difference. The physical slope of the basin lives here in the **slope_mean** field rather than **slope_deg**.

## 6.07 Flow lines from points, lines and outlines

Where the water goes from a given place. From every seed cell the tool walks downhill along the D8 flow directions and draws the path travelled as a line.

| Parameter | What it sets | Default / advice |
|---|---|---|
| Input DEM | Terrain in metres in a metric coordinate system. | - |
| Where it flows from | Points, lines or outlines. A point gives one cell, a line the cells under it, a polygon every cell inside. | - |
| Receivers | Ponds, rivers, ditches. A trace stops when it reaches them. | - |
| Stop at a watercourse: threshold, cells | Stop in a cell that more than the given number of cells drain into. The same threshold as in 6.02. 0 = off. | 0 |
| Stop at flattening: slope, degrees | Stop where the flow loses its force. 0 = off. | 0 |
| Length of the flat stretch, m | A shelf shorter than this is skipped. | 50 |
| Release zone: slope, degrees | Keeps only traces that have a steep enough stretch. 0 = do not check. | 0 |
| Length of the release zone, m | How long the steep stretch has to be. | 100 |
| Break a trace at a merge | A trace stops once it joins one already walked. | off |
| Trace smoothing, iterations (adv.) | Rounds the D8 steps for the drawing. Elevations do not change. | 0 |
| Carry over fields of the source object (adv.) | Which attributes of the source go into the trace. | - |
| Fill depressions | Planchon-Darboux before tracing: otherwise the path drowns in the first pit. | on |

**Areal sources.** A ditch of eight cells gives eight lines, the outline of a dump as many as it covers. Downhill the traces converge into one channel, and without a break every one of them walks it, so

a thousand cells of a dump give a thousand copies of the same channel. The **Break a trace at a merge** checkbox turns the bundle into a tree. Every cell is then walked once, connectivity is kept, and the trace is carried up to the point of the merge.

**The release zone** keeps the traces that have a stretch steeper than the given angle and no shorter than the given length. That is how an avalanche release is looked for - it starts where the slope holds its steepness over a distance rather than in a single cell. The usual thresholds are 25 degrees and a hundred metres, the length found is written into the `steep_m` field.

**Flattening** stops a trace at the foot of a slope, on a terrace, on a floodplain. The slope is measured as an average over the last metres of the path, and the length of that stretch is set next to it. A short shelf inside a steep slope will not pull the average down and is skipped, while real flattening will. The cut is made at the start of the stretch, and that is the place where the flow spreads out.

**Why degrees.** The thresholds are set by an angle rather than a ratio. Avalanche and geotechnical criteria are written in degrees, and a surveyor’s slope raster is in degrees too.

**The reason for stopping** is written into the `reason` field and tells four cases apart. **sink** - nowhere lower to go. **sheet edge** - the terrain ran out. **receiver** - a given pond or watercourse was reached. **merge** - the trace joined one already walked.

Output fields. `src_id` is the number of the source object, `cells` the number of cells, `length_m` the length of the path. `drop_m` is the fall, `slope` the mean slope, `z_start` and `z_end` the elevations of the ends, `reason` the reason for stopping.

A D8 trace runs along eight directions and therefore looks like a series of steps at 45 degrees. That is a property of the method rather than an error, and smoothing rounds the polyline for the drawing while elevations do not change. A downhill trace shows the path by the topology of the terrain and accounts for neither infiltration nor backwater nor pipes.

## 6.08 Cross-sections and rating curves

The main tool of the group. From a section and its elevations it builds a table and a graph of the dependence of discharge on level.

### How it is computed

The profile is taken along the vertices of the section, the elevation from the vertex Z. A surveyed elevation is more accurate than any terrain model, so the source is the geometry of the section itself rather than a DEM sample.

The profile is divided into the left bank, the channel and the right bank. Counting them separately is mandatory, because the roughness of the floodplain and of the channel differ several times over, and a single count understates the channel part. On every part the flow area, the top width, the wetted perimeter and the hydraulic radius are computed, then the discharge by the Manning formula. The total discharge is the sum over the parts.

The wetted perimeter is taken along solid boundaries, that is along the bed and the slopes, without the vertical planes of the division. The methodologies differ here, and the simplest of the conventions has been adopted.

### The division

It is set in one of two ways. Either the section arrives as a single line with the boundaries standing in fields as distances along the profile, or as three lines with a role field and a common name. In the second case the division lives in the geometry, the way hydrologists draw profiles. The second way is preferable, no distances by hand.

**Roughness and slope**

They are taken from the fields of the section rather than from the parameters of the tool. The reason is not convenience. There are many sections, each with its own values, and the parameters cannot express that, while with fields the calibration comes down to editing a table. The parameters serve as a default for sections without fields, the accepted values are printed to the log.

The slope can be computed **from the chain of sections**. With the chainage and the bed elevations set, the slope of each is taken to its neighbour, from the difference of elevations and the distance along the river. In existing practice it is written out relative to the previous section by hand, here the same value comes out of the attributes on its own.

**Fields and units** Sections on input:

| field | what | units |
|---|---|---|
| `sec` | section name, the lines are grouped by it | text |
| `km` | chainage from the mouth | km |
| `role` | role of the line when supplied as three lines | text: left, channel, right |
| `div_l`, `div_r` | part boundaries, distances along the profile from the start of the line | m |
| `n_left`, `n_channel`, `n_right` | Manning roughness by part | dimensionless |
| `slope` | slope of the water surface | m/m |

The table of soundings on input of **6.10** holds `sec`, `dist` and `elev`. `dist` is the distance along the section in metres, `elev` the absolute elevation in metres. Then `km` and the same computation fields, if they are present.

The table of probability discharges carries `prob` in percent and `q` in cubic metres per second. The table of observed levels carries `level` in metres and `label` as text.

The curve on output carries `level` m, `area` m2, `width` m, `perim` m, `radius` m, `v` m/s, `q` m3/s, `n` dimensionless and `slope` m/m.

The section drawing carries the fields `sec`, `km`, `kind` (the part of the drawing), `part`, `row`, `text` and `value`. A part line in it also carries the properties at the footer level:

- `level`, `width`, `area`, `perim`, `radius`, `v`, `q`, `n` and `slope` in the same units as the curve;
- `depth_avg` - the mean depth in metres;
- `q_pct` - the share of the total discharge in percent;
- `part_no` - the number of the part;
- `soil` and `cover` - soil and cover, they come empty and are filled in manually.

Ground elevations do not come out as a separate layer. The elevation lies in the Z of the profile vertices, and the labels are taken by a style from the geometry.

Levels at the sections on the map carry `sec`, `level` m, `q` m3/s, `prob` percent, `label` (the caption) and `kind` (the kind of level, `prob` computed, `obs` observed).

**In what units** The slope is dimensionless, the ratio of the fall to the length, metre per metre. In the fields and the parameters it is set exactly so, 0.0004 rather than in per mille.

The roughness is set as the Manning coefficient, 0.030 for a clean channel and 0.070 for an overgrown floodplain. Strictly speaking the coefficient is not dimensionless, the dimension hides inside the formula, and that is why the formula is metric. The discharge comes out in cubic metres per second at an area in square metres and a radius in metres, while for feet the formula would require its own factor.

Fields with the names of the contract are picked up without the user - sec, km, div_l, div_r, n_left, n_channel, n_right, slope. The picked ones are printed to the log. An explicit choice is always senior to what has been found.

### Levels

Probability discharges - 1, 5, 10 percent - are supplied as a table of probability and discharge pairs. The tool does not compute them from observation series, that is hydrological statistics. For every discharge a level is found along the curve. The levels come out as lines in drawing coordinates, with ready labels of the UVV1% kind, the way they are put on a gauging section. A discharge above the curve gives a warning rather than an extrapolation.

Observed levels are supplied by their own table, an elevation and a label. This is a measurement rather than a computation, it does not rely on the curve and lies next to the computed ones, in the manner of UV 472.90 X/2021. In the layer of levels on the map such rows are marked kind=obs, the computed ones kind=prob. In the section drawing they correspond to kind=level_obs and kind=level, and the field gives them different styling.

### The section drawing as one layer

The whole drawing of a gauging section comes out as one layer. It holds the profile by parts, the level lines and the vertical elevation scale, and a style draws the footer from the fields of the part lines. The parts are told apart by the **kind** field, and the presentation is attached to it by a style. The sheet is set up once, and after that only the data change.

The ground elevation lies in the Z of the profile vertices, so a separate point layer for the labels is not needed. The vertical exaggeration is set by the **Ratio of the scales, vertical to horizontal** parameter, ten by default, as gauging sections are drawn. The drawing is built in its own engineering coordinate system and is not tied to the terrain.

### The drawing footer

The footer is not assembled from separate cells. The characteristics of a part at the footer level hang in the attributes of the part line itself. These are the width, mean depth, flow area, wetted perimeter and hydraulic radius. Then come the slope, the roughness coefficient, the velocity, the discharge and the share of the total. The footer rows are drawn by a style over those fields, so editing the data needs no reassembly of the sheet.

The level is set by the **Level for the drawing footer** parameter. Without it the highest of the computed levels is taken, since a footer is computed on high water. The accepted level is printed to the log.

The ground elevations are labelled by a style from the Z of the profile vertices. The sheet is assembled by a print layout, the tool gives the data and the design lives in the template.

### What matters about the method

The Manning formula describes steady uniform flow. The curve is a hydraulic characteristic of the section rather than a computation of a release wave, and the tool promises no more.

The slope enters the discharge under a square root, so an error in it tells directly. On lowland rivers it is small, and the value is always printed to the log so that the accepted one is visible.

### What comes out

Out comes a table of the curve by parts and in total, with the area, width, perimeter, radius, velocity and discharge at every level. With it come **Section drawing as one layer** in drawing coordinates and **Levels at the sections on the map (for 6.09)** in map coordinates. **Rating curve plot (drawing)** is switched on separately.

An HTML report is made for every section. It holds the profile with the levels and the division boundaries drawn on it. Then come a graph of discharge against level with the probability lines, a table of levels and the table of the curve. The pictures are embedded into the page itself, so the report stays one file that can be forwarded.

#### The rating curve plot

The curve is given as a vector drawing - the line, the axes, the scales with ticks and labels, the dashed marks of the exceedance levels. It is the very plot that goes into a report, except that this one can be edited like any other layer.

What goes along the horizontal is chosen by a parameter - discharge Q, flow area W or mean velocity v. Velocity is needed where the curve is read for scour rather than for discharge, because the permissible non-scouring velocity of the soil is compared straight off the plot.

The plot goes as a separate layer and in its own axes, the chosen quantity along the horizontal and elevation along the vertical. It cannot share a layer with the profile of the gauge line, because metres of distance and cubic metres per second are not comparable and no common scale for them exists. Both are put on a sheet through the layout.

The step of the elevation scale is set to a metre for a normative drawing. The step of the discharge scale at zero means round numbers by the range. The gauge lines are laid out in a row, each in its own block.

The exceedance levels arrive as discharges, and the elevation for a mark is obtained by the reverse pass over the same curve. The mark lies on the curve exactly rather than beside it. A level outside the range of the curve is skipped.

The **kind** field in the attributes tells the parts of the drawing apart - curve, axis, tick, mark, label. The presentation is set up by it.

## 6.09 Flood extent polygon

Cuts the surface by a water level and produces the flood extent and a raster of depth.

The level can be set directly or by a discharge. In the second case it is taken backwards along the curve from 6.08 - the very move drawn as a red arrow on manual constructions. The curve is supplied as a table, the discharge as a number, the accepted level is printed to the log.

The tool is deliberately separate from 6.08. It has a different input and a different consumer, and there is no point in performing the whole curve computation for the sake of one polygon.

#### The level is not constant along the river

Upstream it is higher. Cutting the whole area by a single elevation floods the valley wherever the bed lies below that elevation, even though the water does not reach there. Therefore the levels are supplied at the sections, a water surface rises from them, and the depth is computed from it.

There are two paths. Ready levels as lines - take the output **Levels at the sections on the map** from 6.08. Or a desired discharge over a curve already computed. You supply the sections on the map, the table of the curve and a number, and for every section the level is found backwards along the curve.

The second path answers the frequent question about a catastrophic scenario. The curve does not depend on the discharge, it is a characteristic of the section. So any discharge, not only the design one, is cut without running 6.08 again. A section whose discharge went above the top of the curve is skipped with a warning. The curve is built up to the top of the profile, and above it there is nowhere to take an elevation from. If such scenarios are needed, raise the upper elevation in 6.08 and the curve will continue with a margin.

The drawing layer of levels will not do for the map, because along its horizontal axis runs the distance along the section rather than a coordinate. The tool recognizes this and refuses with an explanation of which output to take.

Small patches are dropped by area, since on a flat floodplain single cells below the water line give speckle that has nothing to do with the flood. Connectivity with the channel is not checked, and the log says so. Closed depressions away from the river will stay in the extent, and that is visible on the map.

## 6.10 Import section tables

Turns a table of soundings into sections with elevations in the vertices.

Existing programs keep the profile as pairs of a distance from the start of the section and an absolute elevation. Over years of work many such tables accumulate. There is no point in entering them again for the sake of moving into GIS.

If the sections are already digitized on the map, supply them as a layer. The soundings will lie along the real lines, and the distance along the section becomes the distance along the line. A section for which no line was found is built by the scheme, with a warning. Soundings longer than the line are pressed to its end, also with a warning.

Without a line layer the geometry is built as straight sections by an azimuth and the chainage. This is a scheme rather than a survey, and it does not affect the computation of the curve at all, since the curve needs the distances along the section and the elevations.

Computation properties are carried along with the profile if the table holds them - the division boundaries, the roughness values and the slope. Without them the geometry would come back but not the computation, and the curve over the restored sections would part from the original one.

## 6.11 Example river (demo)

A teaching chain of sections with a known answer. It is a valley with a channel and two floodplains, elevations in the vertex Z. The fields of division, roughness, slope and chainage are filled in.

The channel and the floodplains have different roughness, so the curve shows how the floodplain comes in. Up to the banks the discharge grows steeply, while above the break the floodplain adds area but adds little discharge.

Besides the sections it produces a table of soundings, the input for 6.10, and a valley surface as a raster, the input for 6.09. While the valley exists as lines only, there is nothing for the flooding to cut. The surface is built from the same sections as the curves, so the flood extent and the rating curve speak of one valley rather than of two similar ones.

A separate output is a reference curve computed directly from the gauge line itself. A discrepancy with what 6.08 gives on the same sections is an error of the tool rather than of the data, because the answer is known in advance.

### How to check the group

Build the example, feed the sections into 6.08 and compare the table of the curve with the reference one - they must coincide. Then feed the table of soundings into 6.10 together with the layer of the demo sections. The restored sections will lie over the originals and give the same curve. Finally feed the valley surface, the rating curve and a discharge into 6.09 - you get the flood extent and the depth.

## What the group does not have yet

Soundings as a separate point layer and sampling of the profile from a DEM for sections without Z. Bathymetry is a separate task, the restoration of the bed between surveyed sections along the

channel has its own anisotropy. Until it is done, the curves are computed over surfaces where the bed is present.

# 7. Fractal analysis

## 7.01 Fractal dimension

The tool computes a fractal-dimension map of a surface by the variogram method, native to the plugin.

A log-log variogram over lags of one to N cells is built in a sliding window. Its slope gives the Hurst exponent H, and the dimension is D = 3 - H. Smooth differentiable areas give D near 2, rugged and noisy ones tend to 3. The D values give the level of ruggedness, while their steps show the boundaries, bringing out zones of tectonic disturbance, block boundaries and changes of the roof relief character.

The output is a D grid that feeds straight into **1.04 Isolines from raster** for dimension isolines, and an advanced checkbox adds H as band 2. The global D and H over the whole surface are printed to the log.

### Reading the map

The absolute D values matter less than their steps. A linear step across the area is a lineament, a candidate tectonic disturbance. A patch of a raised D is a zone of intense folding or a rugged roof relief. Wide even fields of a low D are quiet blocks. For reading, apply a singleband pseudocolour symbology with a contrast palette and quantile classification. For a report plan build isolines with belts over the D grid with tool 1.04. The disturbance zones then get outlined like contour lines.

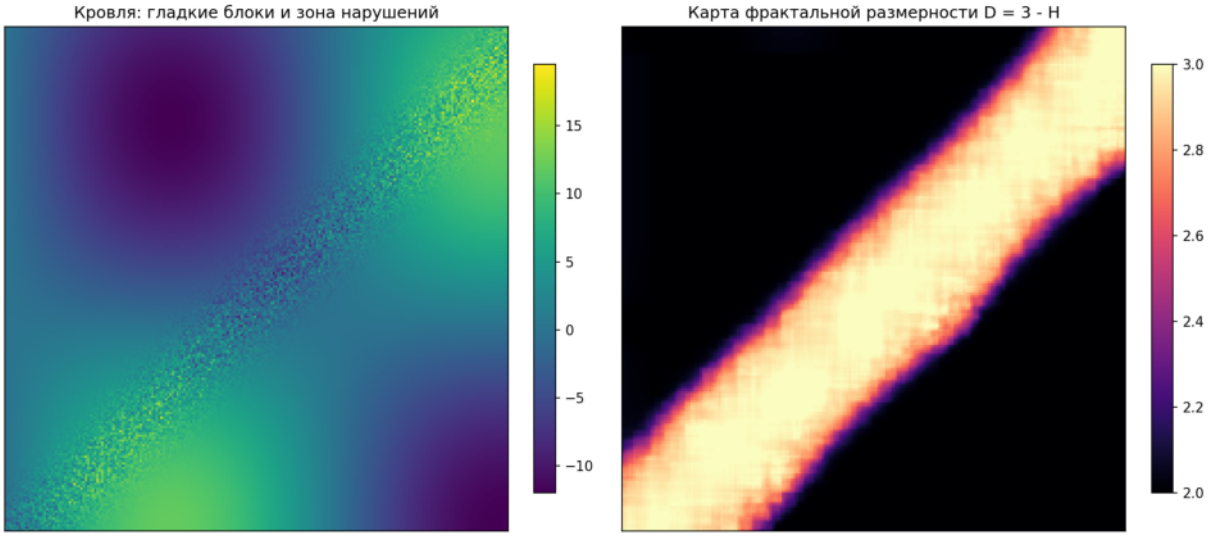



Figure 33: A synthetic roof with a diagonal crushing zone and its D map: quiet blocks near 2, the disturbance zone shows up as a bright lineament.

### Picking the window and the lags

A small window (5-8 cells) reveals the microstructure and local disturbances, a large one (12-20) the regional zones, and in doubt compute both and compare. Four lags fit almost always, and more lags give a steadier slope but a coarser minimal scale the method can resolve. The window and the lags are limited by the grid size, the tool checks that itself.

### Workflow

A bed roof from kriging → **7.01** with a window of 8 → the D grid. Then **1.04 Isolines from raster** (band 1) → dimension isolines with belts over the structural plan. The global D from the log is one number per surface to compare areas or beds with each other. The raster must be in a metric CRS, and the demo surfaces fit as they are.

### Parameters

| Parameter | What it sets | Default / advice |
|---|---|---|
| Surface (raster) | A relief grid or any surface. | - |
| Window half-radius, cells | The sliding-window size. | 8 |
| Number of lags (Adv.) | Variogram lags 1..N cells. | 4 |
| Elevation band (Adv.) | The band with elevations. | 1 |
| Write H (Adv.) | Add H as band 2. | off |
| Fractal dimension | A D grid (and H if checked). | - |

## 7.02 Box-counting of masks

Classic box-counting for binary masks. The raster is binarised by a threshold, the object being the values above it. The mask is then covered by cells of a decreasing size, and the slope of log N versus log(1/size) gives one dimension D for the whole mask. A linear object gives D near 1, a blob - near 2, rugged outlines of replacement zones or mined-out areas fall in between. The accuracy on finite masks is about ±0.1, so the method is good for comparing masks with each other rather than as an absolute measure. The result is printed to the log with a table of sizes and counts and returned as the number D - usable further in Processing models.

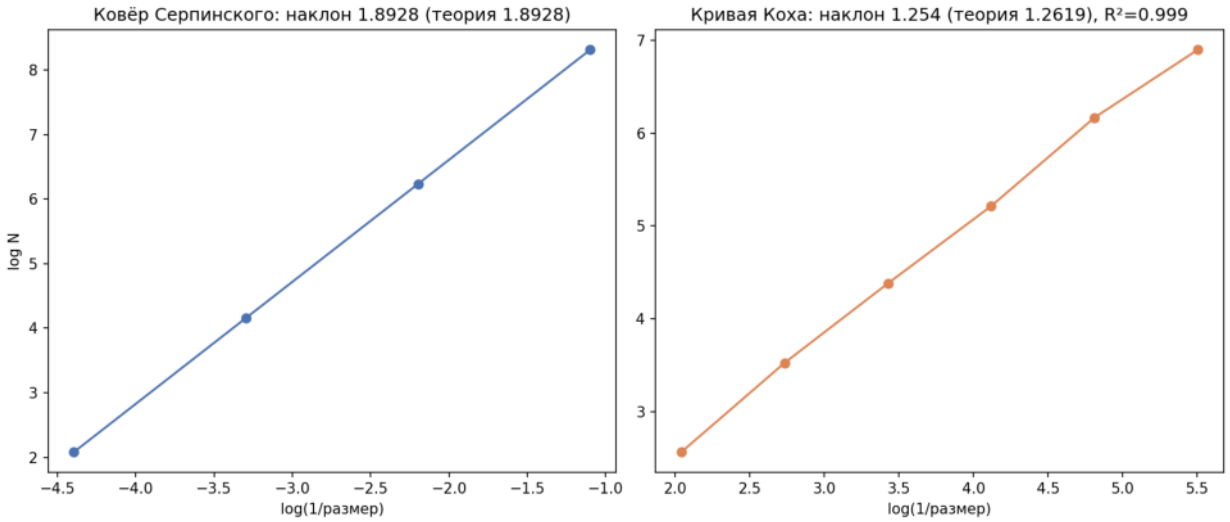



Figure 34: Checking the estimators on the references: the Sierpinski carpet gives a slope of 1.8928 against the theoretical 1.8928, the Koch curve - 1.254 against 1.2619. Points on a line - the power law holds.

### Where the mask comes from

A mask comes from one of four sources. The first three are the mineral-type band of a bed grid with a threshold between the class codes, an indicator-kriging probability grid with a 0.5 threshold and an exceedance-probability map with a cut-off threshold. The fourth is vector outlines of workings or zones, rasterised beforehand with the standard “Rasterize (vector to raster)”. Compare the D of

masks of the same nature on the same grid. A growth of the replacement-outline ruggedness from bed to bed or from year to year is a meaningful signal.

| Parameter | What it sets | Default |
|---|---|---|
| Mask raster | Any raster; the mask - values above the threshold. | - |
| Threshold | The object/background boundary. | 0.5 |
| Band (Adv.) | The raster band. | 1 |

## 7.03 Dimension of lines and boundaries

The dimension of every line by the divider (Richardson) method. The line is walked with chords of a decreasing span, and the slope of log N versus log r gives D. A straight line gives one, a rugged line - more. Polygons are accepted alongside lines - the exterior ring of the boundary is measured, so the ruggedness of zone and basin outlines is computed without a prior conversion. The output is the same features with the D and steps fields, the mean D is printed to the log, and short lines get an empty D. The method is checked on references, and the Koch curve gives 1.262 against the theoretical 1.2619.

### An isoline-smoothing diagnostic

Oversmoothed isolines lose their ruggedness and D drops towards one. The workflow is this. Build the isolines twice, without smoothing and with the working parameters, run both layers through the tool and compare the mean D from the log. A drop by hundredths is cosmetics, the shape is kept. A drop by tenths means the smoothing destroys the field geometry, so weaken the rounding or keep the densification only. The D field in the attributes lets you find the specific lines that suffered most.

### Other uses

The ruggedness of zone outlines in plan, the digitising detail of boundaries from different sources, generalisation control when preparing small-scale plans - anywhere “how winding the line is” must become a number.

| Parameter | What it sets | Default |
|---|---|---|
| Lines | A line layer (isolines, outlines). | - |
| Lines with the dimension | The same lines with the D and steps fields. | - |

## 7.04 Minkowski dimension (vectors)

Box-counting directly over vectors, no rasterisation. Lines and polygon boundaries are covered by a grid of a decreasing size. The slope of log N versus log(1/size) gives the Minkowski dimension. A straight line and a smooth boundary give D near one, a river network - 1.1-1.5, a heavily rugged coastline - up to 1.3 and above. Every feature gets the D_mink and D_r2 fields. D_r2 is the quality of the log-log fit, and below 0.85 the estimate cannot be trusted.

The D of the layer as one set is computed separately and printed to the log. For a river network that is the dimension of the network as a whole, regularly higher than that of the individual branches.

The method complements the divider of 7.03. The divider measures the sinuosity of one line, Minkowski the plane filling by a set of features. The dimension is also returned as a number output for Processing models.

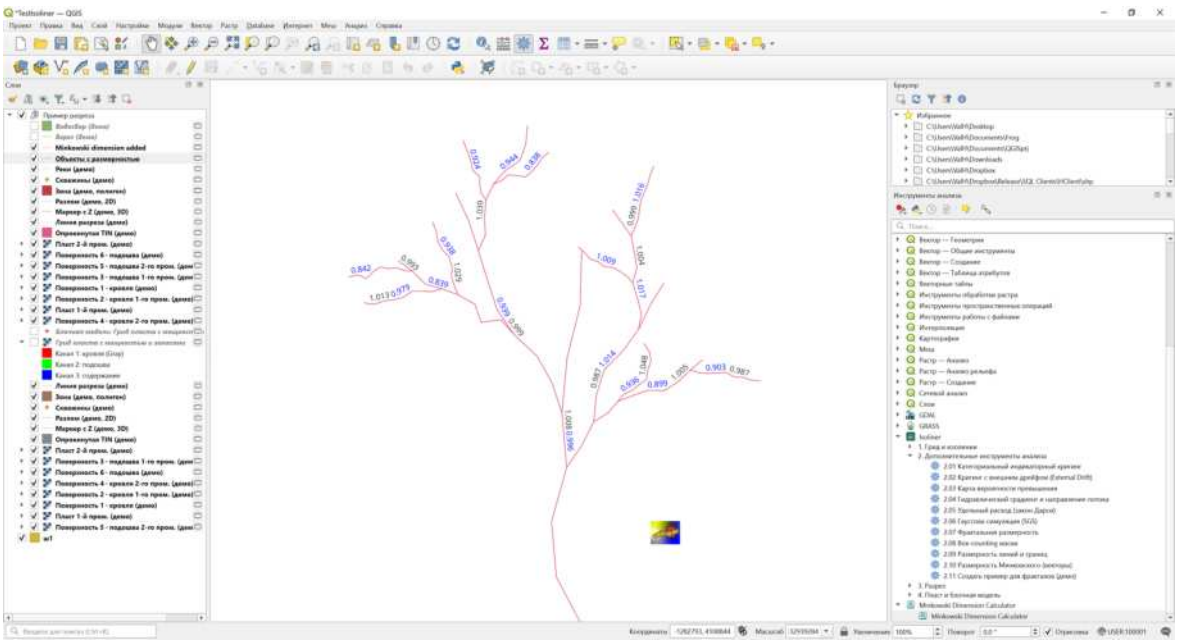


Figure 35: Demo rivers labelled by the per-branch D_mink: nearly smooth branches give values around one, the network as a whole - higher.

| Parameter | What it sets | Default |
|---|---|---|
| Lines or polygons | A vector layer; for polygons the boundary rings are taken. | - |
| Number of grid sizes, K (Adv.) | The number of steps; a too large K takes the cells below the line detail and lowers D. | 8 |
| Grid offsets per size (Adv.) | Random shifts, the minimal cover is taken - removes the grid alignment. | 3 |
| Densify factor (Adv.) | The sampling step along segments as a cell fraction; 0 - vertices only. | 0.5 |

### 7.05 Example for fractals (demo)

A generator of study features for the whole fractal five. It gives a branching river network with an order field, a basin polygon with a rugged boundary, and a separate coastline built by midpoint displacements. Feed the rivers into 7.04 and you get the network dimension. Feed the coast and the basin boundary into 7.03 and 7.04, and compare the divider with Minkowski. Rasterise the basin with the standard tool, and it doubles as an example for 7.02.

| Parameter | What it sets | Default |
|---|---|---|
| Extent | The generation area. | - |
| RNG seed (Adv.) | The example repeatability. | 1 |

## 8. Data exchange

The group gathers the tools that exchange data with programs working outside GIS. At present this is LandXML, the interchange format of total-station processing and road design software. Schema

1.2 has stood since 2008 and has barely changed, so a file written by one program is read by the others.

The tools of the group are neutral to the content. They carry points, lines, a surface and an alignment with cross sections, and what those objects mean is decided by the user.

## 8.01 Read LandXML

The tool reads a LandXML exchange file. The programs that process total station surveys give their result in this format - Credo, Trimble Business Center, Topcon Magnet, Leica Infinity, Civil 3D, Bentley. Schema 1.2 has held since 2008 and has hardly changed.

Until this tool the data from such programs had to be reduced to tables by hand. Nothing in QGIS read cross sections with stations.

### What is read

| Section of the file | What it is | Output |
|---|---|---|
| CgPoints | survey points with a name, a code and a description | a point layer with elevations |
| PlanFeatures | named lines with codes | a line layer with elevations |
| Surfaces | a triangulation | a layer of 3D faces |
| Alignments | an alignment with stationing | a line layer with the start station and the length |
| Profile | a longitudinal profile | the alignment line with elevations from the profile |
| CrossSects | cross sections | lines across the alignment with elevations |

The outputs are optional. Switch off the ones you do not need, the file is parsed once either way.

The surface comes out as 3D faces rather than as a mesh, and that is deliberate - such a layer is taken straight away by **4.06 Intersect a TIN with the section**. Cross sections are placed on the ground across the alignment, and a line with elevations like that is what the tools of the **6. Hydrology** group take.

Cadastral parcels, pipe networks, roadway objects, field observations and monuments are deliberately not read. These are other people’s subject areas, and supporting them halfway is worse than not supporting them at all. When found in the file they are listed in the log.

### Parameters

| Parameter | What it sets | Default / hint |
|---|---|---|
| LandXML file | The input file. | - |
| The file gives north first, then east | The order of the numbers in a point record. | on |
| Coordinate reference system | Empty - taken from the file, and from the project when the file has none. | empty |
| Curve chording tolerance, m (adv.) | The sagitta used when splitting arcs. | 0.05 |

### The standard elements of the schema

A LandXML file is made of top-level sections. The tool reads the ones listed below and takes the following from them.

**Units** sets the units of measurement. `linearUnit` is read and the value is converted to metres. The metre, kilometre, centimetre, millimetre, international foot, US survey foot, inch, yard and mile differ by their factors. Without the section the numbers are taken as they are, with a warning.

**CoordinateSystem** sets the coordinate system. `name` and `epsgCode` are read. The code is overridden by the system given in the parameters of the tool.

**CgPoints** holds the survey points. Every `CgPoint` gives its `name`, `code` and `desc` attributes and the coordinates in the text of the element. The text holds two or three numbers - north, east and, when present, the elevation.

**PlanFeatures** holds the named lines. `PlanFeature` gives `name` and `desc`, and the geometry is assembled from `CoordGeom` - the `Line`, `Curve`, `Spiral` and `IrregularLine` elements with `PntList2D` and `PntList3D`.

**Surfaces** holds the triangulation. Inside `Definition` the tool reads `Pnts` and `Faces`. `Pnts` holds `P` elements, each with an `id` attribute and coordinates in the text. `Faces` holds `F` elements with three references to an `id`. A face with the attribute `i="1"` is taken as hidden and skipped, with a count.

**Alignments** holds the alignments. `Alignment` gives `name`, `desc`, `staStart` and `length`. Inside lie `CoordGeom` with the same geometry as the lines, `Profile` with the longitudinal profile and `CrossSects` with the cross sections.

**Profile** gives the longitudinal profile. `ProfAlign` is read with its `PVI` elements, each holding a station and an elevation. Without `ProfAlign` the tool reads `ProfSurf` with `PntList2D`, where the pairs run one after another.

**CrossSects** gives the cross sections. `CrossSect` gives `sta` and `name`, inside lie `CrossSectSurf` with the name of the section surface, and in them `CrossSectPnt` with a pair of offset and elevation. That pair is not swapped by the “north first” rule, because the rule concerns points on the ground, while here it is an offset and an elevation.

The `Parcels`, `PipeNetworks`, `Roadways`, `Survey`, `Monuments` and other sections are not read. When found in the file they are listed in the log with the number of occurrences.

### The fields of the output layers

| Layer | Field | What it holds |
|---|---|---|
| Survey points | name | the name of the point from the file |
| | code | the survey code |
| | desc | the description |
| | z | the elevation, empty when the file has none |
| Lines | name | the name of the line |
| | desc | the description, or the name of the set of lines when there is none |
| | npts | the number of vertices |
| Surface | surface | the name of the surface from the file |
| | tri | the number of the face within the surface, from one |
| Alignments | name | the name of the alignment |
| | desc | the description |
| | sta_start | the start station |

| Layer | Field | What it holds |
|---|---|---|
| | length_m | the length declared in the file |
| Profile | align | the name of its alignment |
| | npts | the number of elevations |
| Cross sections | align | the name of the alignment |
| | sta | the station of the cross section |
| | name | the name from the file |
| | surf | the name of the section surface |
| | npts | the number of points |

The length of the alignment is the one declared in the file rather than measured along the polyline, since on chorded curves the two differ.

For a point without an elevation the **z** field stays empty while the geometry carries a zero. An empty field is the absence of a measurement, the zero in the geometry is what the layer format requires.

Every feature of the surface layer is one triangle. One cross section in the file often holds several section surfaces, the ground and the design, and each gives its own line with its own value in the **surf** field.

### The coordinate order

In the schema a point is written as “north, east”, that is Y before X. Some programs write it the other way round, and the wrong order gives a mirrored turn of the site. On a hundred metres of ground such a turn is harder to notice than it sounds.

The tool therefore prints the extent of what was read to the log. Check it against what you expect. If the coordinates changed places, it shows in the very first line.

### Units

The units are taken from the header of the file and converted to metres. A file in feet reads without a single error, and without conversion it would give elevations three times smaller. So the absence of units in the file goes into the warnings, and an unknown unit name is not interpreted by guesswork.

The international foot and the US survey foot differ in the sixth digit and are named differently in the file. Each is converted by its own factor.

### Curves

An alignment in the file is made of straight lines, circular curves and transition curves. A straight line is carried over as it is. A circular curve is split into a polyline by the given sagitta, and the ends of the arc are placed exactly. A transition curve is replaced by a chord, and the number of such replacements is printed to the log, since the length of the alignment is understated on them.

A curve without a centre is also replaced by a chord. An arc is not recovered from the coordinates of its start and end alone, and there is no point in guessing it.

### Cross sections on the ground

A cross section in the file is given by a station and by pairs of “offset, elevation”. The tool places it on the ground - it finds the point of that station on the alignment and lays the offsets across the direction of travel, positive to the right. The vertices of the line carry the elevations.

When the alignment has no geometry in the file, the cross sections are skipped and their number is reported. Giving them out in local coordinates beside the other layers would be worse, one layer would end up in two coordinate systems at once.

## 8.02 Write LandXML

The tool assembles a LandXML file from the layers of the project. Credo, Trimble Business Center, Topcon Magnet, Leica Infinity and Civil 3D open such a file.

### What is written

| Input | What goes into the file |
|---|---|
| Survey points | CgPoints with the name and the code from the chosen fields |
| Lines | PlanFeatures with the name from the chosen field |
| Surface: a layer of 3D faces | Surfaces, as triangles |
| Alignment (a single line) | Alignment with stationing and a longitudinal profile from the elevations of its vertices |
| Cross sections (lines with elevations) | CrossSects, when an alignment is given |

The inputs are optional, whatever is given is taken. The file is always written in metres and as the schema says, that is north first, then east. The coordinate system is taken from the first layer given and written as an EPSG code.

### Cross sections and the alignment

Cross sections are written only together with an alignment. The offset of a cross section point is measured from the axis, and without an axis there is nowhere to take it from. The offset comes out positive to the right along the alignment and negative to the left, as road practice has it.

The station of a cross section is taken from the chosen field. Without a field it is computed as the mean station of the vertices of the line projected onto the alignment.

### Limits

Only a triangle counts as a face of a surface. Polygons with another vertex count are skipped and counted in the log, they have to be triangulated beforehand.

Layers in degrees are refused. LandXML holds plane coordinates, and degrees in it turn into nonsense.

## 8.03 Example LandXML (demo)

The tool writes a set of LandXML files holding one and the same terrain written in different ways. It serves two purposes. To check the reading against material whose answer is known in advance, and to see what a file coming from somebody else’s program looks like.

The geometry is the same in every file. Twelve survey points, two lines, a surface of twenty points, an alignment 240 metres long with a circular curve, a profile and three cross sections. Only the way of writing differs, so the reading results are comparable. Read any two files and compare them, there should be no discrepancy.

### The variants

| File | What differs |
|---|---|
| `landxml_base.xml` | Metres, northing-easting order, the surface as points and faces. One face is marked invisible and is not read. |
| `landxml_east_first.xml` | Coordinates written easting-northing. Read correctly only with the order checkbox cleared. |
| `landxml_imperial.xml` | Feet, the US survey foot. Exercises the conversion to metres. |
| `landxml_no_units.xml` | No units section. The numbers are taken as they are and a warning goes to the log. |
| `landxml_no_crs.xml` | No coordinate system. The layers get the project CRS. |
| `landxml_points_only.xml` | The surface as points without faces. There is no triangulation in the file. |
| `landxml_breaklines.xml` | The surface given by breaklines alone. Read as lines, no surface is created. |
| `landxml_spiral.xml` | An alignment with a transition curve. The curve is replaced by a chord and the count goes to the log. |
| `landxml_survey_points.xml` | Points without an elevation, with the oID and pntSurv attributes instead of a name and a code. |
| `landxml_plain_align.xml` | An alignment of one straight line, without a profile and without cross sections. |

### Where the shapes come from

The variants follow what the programs actually write rather than being invented. Feet and the surface as points without faces are what Civil 3D exports, its surface export mode is named after it. The surface by breaklines alone is what Trimble Business Center writes when it hands over an alignment as a surface along breaklines. The coordinate order is not fixed by the schema, so a file with the reverse order turns up regularly. The units section and the coordinate system are optional by the schema.

What the set does not hold is an exact reproduction of any one program’s export. That needs real files from each of them, and the set will grow once they appear.

### How to work with it

Write the files into a folder, feed them one by one into **8.01 Read LandXML** and check the log against the table above. For the file with the reverse coordinate order clear the order checkbox, otherwise the points land beyond the equator.

# 9. Subsidence

The group computes deformations of the ground surface from the subsidence of benchmarks. The formulas are taken from the “Instructions on protecting mines from flooding and protecting undermined objects at the Verkhnekamskoye deposit” (Mining Institute, Ural Branch of the Russian Academy of Sciences), section 4. The sixth edition for Uralkali and the 2024 Instructions of the Talitsky mining complex agree in this part.

Subsidence is taken as the elevation difference at a benchmark over the chosen period, not by subtracting two grids. The difference at a benchmark is exact, while each grid carries its own interpolation error, and subtraction adds them up. Besides, the set of benchmarks changes from

cycle to cycle. So the subsidence at the benchmarks comes first, a grid is built from it (**1.01**, **1.02** or **1.03**), and the group works on that grid.

The units are those of the observation database. Tilt and horizontal strain go in mm/m, curvature in $10^{-6}$ 1/m. The radius of curvature in kilometres equals 1000 divided by the curvature in $10^{-6}$ 1/m.

## 9.01 Tilt and curvature from subsidence

The tool builds, from a subsidence grid, rasters of the largest tilt, its azimuth, the curvature along the tilt and the principal curvatures.

### How it is computed

Tilt by formula 4.22 of the Instructions is the subsidence difference divided by the distance. Curvature by formula 4.24 is the difference of the tilts of neighbouring intervals divided by the mean interval length.

On a grid the differences are taken not between neighbouring cells but over the **Difference base**, as between benchmarks. Tilt uses two points half a base to either side, curvature three points a base apart. A cell smaller than the base only refines the position and does not affect the value. This matters. Curvature over a 5 m cell and over a 15 m interval differs several times, and the tolerances are set for a 15-metre interval.

The **Reduce to a 15-metre interval** box multiplies the tilt by $q_i$ and the curvature by $q_i \cdot q_k$. By clause 4.24 curvature is taken from tilts already reduced to 15 m and is reduced once more itself. With a base of 15 m or less both coefficients equal one.

The tool does not apply overload factors. For existing objects the actual deformations along profile lines are compared with the tolerances without them (clause 5.4).

### Sign

In a levelling log subsidence goes as an elevation difference with a minus sign, in deformation statements with a plus sign. **The subsidence sign** is detected from the data or given explicitly, and the accepted choice is printed to the log.

Curvature is computed as the second difference of subsidence taken positive downwards. In the central part of the trough it is negative, and the surface is compressed there. At the edge of the trough curvature is positive and the surface is stretched. The tilt azimuth points to where subsidence grows, towards the trough centre.

### Check against profile lines

When **Benchmarks for the check** are given, the tool tests the grid along the profile lines. **The profile field** splits the benchmarks into profiles, **the benchmark order field** orders them along the line, and without it the benchmarks go by position.

For every profile the tilt and curvature are computed twice, from the measurements at the benchmarks and from the grid at the same points. Two layers come out. The interval lines carry the tilt with a sign along the profile, as in the statement of the observation database. The benchmark points carry the curvature and the radius. Both layers hold the grid value and the difference next to it, and the overall difference is printed to the log.

A large difference means that the grid cuts the trough between profiles or that the cell is too coarse. Then take another interpolation model or a smaller cell.

The UKK observation database computes tilt and curvature without the reduction to 15 m. To match its numbers, clear the reduction box.

| Parameter | What it sets | Default |
|---|---|---|
| Subsidence grid | A raster of subsidence over the period, built from benchmarks. | - |
| Subsidence units | Millimetres or metres. | millimetres |
| Subsidence sign | Detect from the data, plus or minus. | from the data |
| Difference base, m | The distance over which differences are taken. | 15 |
| Reduce to a 15-metre interval ($q_i$, $q_k$) | Reduction coefficients of the Instructions. | yes |
| Benchmarks for the check (points) | Benchmarks of profile lines. | - |
| Benchmark subsidence field | Subsidence in the same units and with the same sign as the grid. | - |
| Profile field | Splitting benchmarks into profiles. | one profile |
| Benchmark order field | Order along the profile. | by position |
| Benchmark name field | Benchmark name for the output. | - |

## 9.02 Horizontal strain

The tool computes horizontal strain, extension and compression, in two ways. Extension goes with a plus sign, compression with a minus sign, as in the observation database.

### Estimate from curvature

Where intervals are not measured, the strain is estimated by formula 4.34 of the Instructions, $\varepsilon = m_e \cdot K \cdot L$. The curvature K is taken from the output of **9.01**. The coefficient $m_e$ depends on the magnitude of the curvature and is computed by clause 4.27.

The half-trough length L is given as a number or computed from the **Mining depth** H, $L = (\operatorname{ctg} \delta_0 + \operatorname{ctg} \psi) \cdot H$. The angle of full subsidence $\psi$ is 55°. The limit angle $\delta_0$ is 55° at permanent boundaries of the mined-out space and 65° at temporary and long stopped ones.

In the 2014 edition of the Instructions the interval $l_0$ between the points of the half-trough stood in place of L, and the same curvature gave a strain about an order of magnitude smaller. The current editions write L. The $l_0$ option is kept in the **Estimation formula** parameter for recomputing old reports.

### From measured intervals

When **Measured intervals** are given, the strain of each is computed by formula 4.27 as the relative change of length. The reduction box multiplies it by q_ε. When the curvature raster is given too, the estimate at the middle of each interval is written as well, and the mean difference and RMS are printed to the log. This shows how well the curvature estimate works on this site.

| Parameter | What it sets | Default |
|---|---|---|
| Curvature (output of 9.01) | A curvature raster in $10^{-6}$ 1/m. | - |
| Estimation formula | Current Instructions (L) or the 2014 edition ($l_0$). | L |
| Half-trough length L, m | Half-trough length, 0 - from the depth. | 0 |
| Mining depth H, m | Depth for computing L. | 300 |

| Parameter | What it sets | Default |
|---|---|---|
| Boundary of the mined-out space | Permanent (55°) or temporary (65°). | permanent |
| Interval $l_0$, m (Adv.) | The interval for the 2014 formula. | 15 |
| Measured intervals (lines) | Lines between neighbouring benchmarks. | - |
| Initial length field | Initial length, empty - line length. | - |
| Current length field | Length in the current cycle. | - |
| Reduce to a 15-metre interval (q_ε) | Reduction coefficient. | yes |

## 9.03 Example subsidence trough (demo)

The tool builds a subsidence trough over a rectangular working from the typical function S(z) of table 1 of the Instructions. The tilt and curvature of such a trough are known in advance, and **9.01** and **9.02** are checked on it.

The working stands in the centre, its long side running west to east. The example assumes full undermining, when the working is at least 1.4H in size. Then the trough has a flat bottom, and a half-trough of length L runs from its edge. The Instructions treat incomplete undermining differently, and the demo does not model it, it only warns. Subsidence in the raster is written in millimetres with a minus sign, as an elevation difference, so that the sign choice in **9.01** is checked too.

On top of subsidence the example adds rates. The previous survey is taken as the same trough multiplied by the **Share of the previous survey**, and over the **Interval between surveys** the trough deepened by the remaining part. The subsidence rate goes out as a second raster in millimetres per year, with the same sign as subsidence. That raster is fed into **9.01** just like subsidence and gives the rates of tilt and curvature.

The third output is the benchmarks of two profile lines along the main sections I-I and II-II through the trough centre. The benchmark spacing is L/10 by default, as in clause 4.26.2 of the Instructions. Each benchmark carries the subsidence of the current survey **eta_mm**, of the previous one **eta_prev_mm** and the rate **rate_mm_y**. With this spacing the tilt and curvature by benchmarks match the formulas of the Instructions exactly. With a depth of about 357 m (L = 500 m) and a subsidence of 1 m they repeat table 2 of the 2014 edition, except the row z = 0.20. There the curvature is misprinted, the formula gives $-0.450 \cdot 10^{-4}$ 1/m instead of -0.630.

### How to check

Build the example and feed the raster into **9.01**, and the demo benchmarks into the check with the eta_mm field. The difference of tilt and curvature between the grid and the benchmarks must be small. Then feed the curvature into **9.02** and compare the estimate with the numbers in the log.

| Parameter | What it sets | Default |
|---|---|---|
| Mining depth H, m | The depth, L depends on it. | 350 |
| Working length D11, m | Size from west to east. | 1200 |
| Working width D12, m | Size from south to north. | 900 |
| Maximum subsidence, m | Subsidence at the trough bottom. | 1.0 |
| Boundary of the mined-out space | Limit angle 55° or 65°. | permanent |
| Benchmark spacing, m | Spacing of benchmarks on the profiles, 0 - L/10. | 0 |

| Parameter | What it sets | Default |
|---|---|---|
| Interval between surveys, months | Time between the previous and the current survey. | 12 |
| Share of the previous survey (Adv.) | Which part of the subsidence was already there at the previous survey. | 0.9 |
| Cell size, m (Adv.) | Raster cell, smaller than the benchmark spacing. | 5 |
| Output CRS (metric) | Coordinate system of the example. | EPSG:32640 |
| Where to place it (extent, optional) | Shifts the example without changing its size. | - |

## Typical situations and solutions

| What you see | Cause | Solution |
|---|---|---|
| Concentric “bull’s eyes”, cones | Kriging pulls the value exactly through outlier wells (nugget 0). | Set a nugget C0 (0.2-0.4 of the sill, in absolute variance units). And/or enable grid smoothing in **2D Kriging**. |
| Angular isolines (“octagons”) | A coarse grid: vertices are placed at cell edges. | Increase **Line rounding** to 3 or reduce the cell size in kriging. |
| Radial/fan lines in empty corners | Extrapolation beyond the data. | Enable **Clip to well hull** or set a clip mask. |
| Isolines cross in dense areas | Formerly - a consequence of smoothing each line. | Smoothing is done over the field (in **2D Kriging**). Increase the grid-smoothing radius there. |
| Polygons of one colour | By default the layer is created with a single symbol. | Set graduated symbology by ELEV_MIN. |

## Appendix. Isoliner in 15 minutes

You need no data of your own. In every scenario the first step creates or downloads the data itself, so the route can be walked on an empty project.

The tool list is long, and that is daunting at first sight. Start with five - **1.01**, **1.04**, **2.03**, **2.10** and **4.01**. The rest will find you when a task calls for it.

After every step there is a line saying what should come out. If something else came out, that is the place to stop and look into it rather than to move on.

---

### Scenario 1. I have points

Five minutes. Boreholes, samples, measurements, any irregular network of observations.

**Step 1. 1.10 Example wells (demo)** Minimum - nothing to set, defaults are fine. If you have points of your own, skip the step. *You get* - a point layer in the project.

**Step 2. 1.01 2D Kriging (points → raster)** Minimum - **Points**, **Value field**, **Cell size**. The rest by default. *You get* - a raster and a layer of standard error. The error grows where the points are sparse, and that is the first thing to look at.

**Step 3. 1.04 Isolines from raster** Minimum - **Raster**, **Interval**, **Isoline style** = Structure / hypsometry. Choose the interval so that the lines are visible but do not merge into solid hatching. *You get* - isolines with labels and contour bands whose borders coincide with the lines.

Further at will. **1.06** and **1.07** show whether the data hold anisotropy, and **1.09** rates the method by cross-validation.

---

## Scenario 2. I need a relief

Fifteen minutes. The main scenario - we build a relief, check it with numbers and treat what is found.

**Step 1. 2.01 Download DEM by extent** Minimum - **Relief source** = GEDTM30, **Download extent** over the area of interest. GEDTM30 is a terrain model without forest and buildings, and for building contours it suits better than Copernicus GLO-30, where the height is taken off the treetops. *You get* - a relief raster in the project coordinate system.

**Step 2. 1.04 Isolines from raster** Minimum - **Raster**, **Interval** of 5 or 10 m. The rest by default. *You get* - a contour layer. From here on it plays the part of source data, as if it had been handed to you.

**Step 3. 6.01 Flow and accumulation (D8)** Minimum - the **DEM** from step 1. Leave the filling of depressions on. *You get* - rasters of flow direction and accumulated area.

**Step 4. 6.02 River network** Minimum - the accumulation raster from step 3, **Catchment threshold** in cells. Start with a thousand and tune it until the network looks plausible. *You get* - watercourse lines directed downstream, that is, ready-made streamlines.

**Step 5. 2.06 Peaks and pits** Minimum - the **DEM** from step 1, **Search radius** in cells. *You get* - points of local maxima with elevations.

**Step 6. 2.02 Download base topography by extent** Minimum - the same **Extent**. From OSM take areal water bodies and dry channels, but not rivers and peaks. Those are already computed from the relief in steps 4 and 5 and agree with it, while from OSM they come from another source and may not match the matrix. *You get* - up to five layers in the Topography group.

**Step 7. 2.03 Topo2Raster (terrain from vectors)** Minimum - **Contours** from step 2 and the **elevation field**, **Streamlines** from step 4. Then **Spot heights** from step 5 and the elevation field, **Lakes and shoreline** from step 6 and the shoreline elevation field, and **Cell size**. Supply cliffs if the topographic base holds any. *You get* - a new relief raster built from typed vectors rather than from contours alone.

**Step 8. 2.09 Contour residuals against the DEM** Minimum - **Contours** from step 2, the **elevation field**, the **DEM** from step 7. *You get* - a point layer of residuals and an HTML report. Look at the mean offset and at the share of points that missed by more than half the interval.

**Step 9. 2.10 Terracing check of a DEM** Minimum - the **DEM** from step 7 and the **Contour interval**, the same as in step 2. *You get* - the index of attraction of elevations to the levels. Near one means there are no steps, two and above means terracing.

**Step 10. 2.11 Remove steps (clamped smoothing)** Needed only if the previous step found terracing. Minimum - **DEM with steps**, **Contour interval**, **Iterations** 50. *You get* - a corrected relief and a before and after report. The index should come down towards one.

If you want to test the method rather than your data, replace the first step with **2.07 Demo relief** and its **Gully network** tick. Then the true surface is known in advance and it is plain to see what gets lost in the building. Gullies are lost first.

There are two ways to check against the truth. By eye - build isolines over the restored relief with the same interval and lay them over the contours of step 2 in another colour. The lines should coincide and part only on the summits above the last contour. By number - the raster calculator, the expression “restored minus truth”, and the statistics of the difference in the layer properties. At an interval of 5 m expect a mean near zero and a spread of about a metre, with the largest discrepancies on the peaks. Adding the peak points from step 5 into step 7 shrinks those summit discrepancies noticeably. That is exactly the answer to why topographers write peak elevations on maps.

---

### Scenario 3. I need a section

Three minutes.

**Step 1. 4.10 Example for a section (demo)** Minimum - nothing to set, defaults are fine. *You get* - a set of layers in the Section sample group.

**Step 2. 4.01 Cross-section along a line** Minimum - **Section line**, **Surfaces top to bottom**, **Vertical scale**. *You get* - a section drawing in engineering coordinates, with axes, elevations and bands of the beds.

---

### Where to go next

The manual describes every tool in detail, with parameters and caveats. This page holds neither options nor theory on purpose. Its job is to walk you through the module once, not to replace the documentation.

OpenStreetMap data are © OpenStreetMap contributors, ODbL. The Copernicus GLO-30, GEDTM30 and Mapzen Terrain Tiles matrices are distributed freely.

## Appendix. Demo layer fields

A summary of the fields of all demo-data generators with units. The values are demonstrational, and where a quantity is abstract the units are nominal.

### Topography and Hydrology group outputs

Line layer **River network** (tool 6.02):

| Field | Type | Meaning, units |
|---|---|---|
| order | integer | Strahler order |
| acc_out | real | accumulation at the link outlet, cells |
| length_m | real | link length, m |

Polygon layer **Basins** (6.03) carries **basin** (the basin number, integer) and **area_m2** (the area, m²). Point layer **Peaks** (2.06) carries **z** (the elevation, m) and **drop** (the drop over the window minimum, m). Base topography layers (2.02) carry **name** and **osm_id**, watercourses also **waterway**, water bodies **water**, peaks **ele** (elevation, m), cliffs **kind**. The demo relief (2.07) is a raster without fields.

## Sample wells (demo) - tool 1.10

Point layer **Sample wells (demo)**:

| Field | Type | Meaning, units |
|---|---|---|
| well | string | well name or number |
| roof | double | absolute bed roof elevation, m |
| thick | double | bed thickness, m |
| X | double | content of a nominal component, percent (demo) |
| head | double | head, m (optional) |
| K | double | hydraulic conductivity, m/day (demo, optional) |
| T | double | transmissivity, m²/day, T = K·thickness (optional) |
| mintype | string | mineral type, category (demo replacement, optional) |
| dz | double | value linearly related to the drift, nominal (optional) |

Optional raster **Drift surface (demo)** - an external surface for external-drift kriging, nominal units.

## Electrical-prospecting profiles - tool 1.11, electrical mode

Point layer **Electrical-prospecting profiles (rho_k, SP, IP)**:

| Field | Type | Meaning, units |
|---|---|---|
| profile | integer | profile number |
| picket_m | double | picket from the profile start, m |
| pk | string | picket as a PK label (e.g. PK5+20) |
| z | double | surface elevation, m |
| rho_k | double | apparent resistivity rho_k, Ohm*m |
| rho_true | double | rho_k without noise, Ohm*m (reference) |
| sp | double | self-potential SP, mV |
| vp | double | induced polarisation IP, mV/V |

## Subsidence profiles - tool 1.11, subsidence mode

Point layer **Subsidence profiles (trough, tours: N)**:

| Field | Type | Meaning, units |
|---|---|---|
| profile | integer | profile number |
| picket_m | double | picket from the profile start, m |
| pk | string | picket as a PK label |
| tour | integer | observation tour number |
| z | double | surface elevation, m |
| settle | double | subsidence, mm |
| settle_true | double | subsidence without noise, mm (reference) |

## Section data - tool 4.10

- **Surface 1…6** (rasters) - elevations of six stacked surfaces, m. Top to bottom they are 1 roof of the upper host, 2 roof and 3 floor of the 1st productive, then 4 roof and 5 floor of the 2nd productive, 6 floor of the lower host.

- **Section lines (demo)**, line, field **name** - the line name (Section 1, 2, 3).
- **Collars (demo)**, points, fields **hole_id**, **z** (collar elevation), **eoh** (end of hole) and **number** (label).
- **Intervals (demo)**, table, fields **hole_id**, **from**, **to** (depths along the hole from the collar) and **code** (bed index).
- **1st productive bed (demo)** and **2nd productive bed (demo)** are multiband rasters. Band 1 is the roof (m), band 2 the floor (m), band 3 the content (percent), band 4 the mineral type (category 1 or 2).
- **Fault (demo, 2D)**, line, field **name**.
- **Marker with Z (demo, 3D)**, line with Z, field **name**.
- **Zone (demo, polygon)**, polygon, field **name**.
- **Overturned TIN (demo)**, 3D faces, field **name**.

### Fractal data - tool 7.05

- **Rivers (demo)**, lines, field **order** (integer) - tributary order in the network hierarchy.
- **Basin (demo)**, polygon, field **name**.
- **Coast (demo)**, line, field **name**.

### Density (demo) - tool 3.08

Point layer **Demo points**:

| Field | Type | Meaning, units |
|---|---|---|
| mass | double | measurement mass (demo, 50 per point) |
| prec | double | uncertainty sigma, m |

Line layer **Demo lines**:

| Field | Type | Meaning, units |
|---|---|---|
| mass | double | measurement mass (demo, 100 per line) |
| prec | double | corridor half-width, m |
| from_m | double | interval start along the line, m (may be empty) |
| to_m | double | interval end along the line, m (may be empty) |

Polygon layer **Demo polygons**:

| Field | Type | Meaning, units |
|---|---|---|
| mass | double | measurement mass (demo, 150 per polygon) |
| dasy | integer | 1 - polygon for dasymetry, 0 - uniform |

Auxiliary raster **Auxiliary raster (dasymetry)** - a value gradient for the dasymetric mode of 3.07, nominal units.

## Appendix. Variable-support density: step by step

A walkthrough of tools 3.07 and 3.08 from demo generation to a finished result, with an explanation of each parameter. The example shows how to check the mass invariant by the numbers.

## Step 1. Generate the demo (3.08)

Run **3.08 Example for density (demo)**.

- **Extent** - the generation bounds. Set a rectangle on the map or by a layer. Any metric extent works, for example a square a few kilometres across.
- **Auxiliary raster cell, m** - the step of the auxiliary raster for dasymetry. Default 50, enough for the demo.
- **RNG seed** (Adv.) - example reproducibility. With one seed the set is identical.

The output is four layers - **Demo points**, **Demo lines**, **Demo polygons** and **Auxiliary raster**. The log states the embedded mass, points 500, lines 200, polygons 300, total 1000.

## Step 2. Density from points (3.07)

Run **3.07 Density from measurements** on the **Demo points** layer.

- **Measurements** - the Demo points layer.
- **Mass field** - **mass**. If left empty, each object mass is 1.
- **Precision field** - **prec**. For points this is the Gaussian spot sigma in metres. The larger the sigma, the more diffuse and lower the point contribution.
- **Cell size, m** - the result grid step. A smaller cell means a more detailed map and a longer computation. The density in mass per km2 does not depend on the cell size, only the detail does.
- **Area** - leave empty, taken by the layer.
- **Support beyond edge** - **Renormalise inside** keeps all mass in the area, **Lose mass** discards the part beyond the edge with a warning. To check the invariant use renormalise.
- **Default sigma** (Adv.) - used when the precision field is empty. Zero means the half-cell.

Check the log. The line **Input mass 500, mass on grid 500, discrepancy 0** is the invariant, the density integral equals the sum of masses.

## Step 3. Density from lines

Run 3.07 on the **Demo lines** layer.

- **Precision field** - **prec** - here it is the corridor half-width in metres. The line is spread into a soft-edged strip.
- **from_m / to_m** (Adv.) - the **from_m** and **to_m** fields. One demo line has an interval set, and only its part is spread in the result. Empty fields mean the whole line.

In the log the mass on grid is 200. If the lose-mass mode is on and the corridor left the edge, the discrepancy shows how much mass was lost.

## Step 4. Density from polygons and dasymetry

Run 3.07 on the **Demo polygons** layer.

- Without an auxiliary raster the polygon mass is spread uniformly over its area.
- **Auxiliary raster** (Adv.) - feed the demo **Auxiliary raster**. Then dasymetry turns on for polygons, mass is distributed proportionally to the raster values inside the polygon rather than evenly. If the raster is empty inside the polygon, the tool falls back to uniform and writes this to the log.

The mass on grid is 300 in both modes, only the distribution shape inside the polygons changes.

## Step 5. Mix types by appending

To gather points, lines and polygons into one raster, run 3.07 three times in a row.

- The first run is as usual, you get a three-band raster.

- On the second and third, in the **Append to an existing raster** parameter (Adv.) point to the raster from the previous run. The tool reads the three bands, adds the new mass and returns the updated raster.

After three appends the total mass on grid is 1000. Bands 2 and 3 (sum m*sigma and sum m) are service for exactly this - they let you append in series and keep the effective-sigma map exact.

### Step 6. Effective sigma

Set the optional **Effective sigma** output. This is the mass-weighted sigma per cell, an effective-precision map. Where density is gathered from precise supports the sigma is small, and where it comes from smeared coarse georeferences it is large. This is an analogue of kriging variance for the density floor, and it shows where the result can be trusted and where it rests on coarse supports.

### Reading the result

Band 1 of the result is density in mass per km2. Open it with a single-band pseudocolour style. Density spots are where reliable measurements cluster. The invariant in the log confirms that mass is conserved and the result is correct. The layer must be in a metric coordinate system, otherwise the tool refuses to run.

## For enterprises

Isoliner grows on the tasks of real mining operations. We implement custom features to match production regulations, provide guaranteed technical support contracts and integrate the module into the production cycle, including corporate database connections. Details at https://www.informpp.ru/главная-страница/предприятиям

### Log and toolbar

Isoliner keeps a work log in the **isoliner.log** file next to the QGIS profile. At the start of a session it records the versions of the plugin and QGIS. Then come the name of the launched tool, its parameters and the run time, and on a failure the description of the error. The computation window closes but the file remains, so it is enough to attach it when reporting.

Open the log via **Plugins - Isoliner - Log** or with the **Log** button in the **About** window. The **Isoliner** toolbar holds the density map and the About window.

## License and support

The plugin is distributed under the GNU GPL v2 or later (GPL-2.0-or-later) - the same as QGIS itself. The full text is in the bundled LICENSE file. © Inform++ LLC, www.informpp.ru.