

A quick guide to meshmaker for flox

Coordinate systems:

- Meshmaker requires ESRI shapefiles in a cartesian coordinate system (CCS)
- No conversion between coordinate systems is performed, even if the information is present in the shapefile META data
- Unknown or not present coordinate systems are tolerated
- In short: All shapes you load for mesh creation need to be in the same CCS
- Meshmaker does not care about unit length. If you make a mesh for flox however, it is required that the unit length is 1 meter.

Topology:

- Meshmaker will load almost anything, the Triangulation process performed by "Triangle" however is delicate.
- Overlapping segments or lines cause failure
- Segments should not cross or touch, except at vertices
- Features that are close enough to each other to be beyond the floating point number precision, are to be avoided

Memory, Data volume:

- Meshmaker is a 32bit application, there are various limits to memory and feature numbers
- Upon loading of a shapefile a spatial index is created. If you load datasets several million units apart, this can fail.
- Point clouds with several million points can exhaust memory capabilities
- Especially unfiltered LIDAR clouds can cause problems. As a rule of thumb: roughly 4 million points are currently the limit

Troubleshooting:

- If you loaded all data without issues, but the 'make' process does not provide a result, troubleshooting can hard
- It is recommended to load shapes 1 by 1 to determine which one is causing the problem.
- You can do this by deleting the unwanted lines in the Table of Contents and hitting 'reload', then start the make process again.
- 9 out of 10 times problems are within topology of your data. If this is the case, alter your data to be non-overlapping
- Usage of smartmeshing can help when segments are too close.

The screenshot shows the 'meshmaker for flox 1.0' software interface. The main window displays a map with a green vector network and a white point cloud. The right-hand side contains a control panel with several sections:

- Input:** A 'Table Of Contents' list, a 'reload' button, a 'load log' button, a 'load shape' button, and a 'smartmeshing' checkbox.
- File Selection:** A text field for a file path (e.g., 'F:\01Int\1023\beispielnetze\tutorials\mesh_creation\vector\strickler') and a 'set kst shape' button.
- Output:** A text field for the output path (e.g., 'F:\01Int\1023\beispielnetze\tutorials\mesh_creation\result\mesh.tif') and a 'set path' button.
- Grid Settings:** A 'grid resolution [m]' input field (set to 1), a 'breakline enforcement' dropdown menu (set to 'Closest breakline'), and three checkboxes: 'convex triangulation', 'keep last triangulation', and 'bounding box'.
- Log:** A scrollable log window at the bottom right showing file loading status.

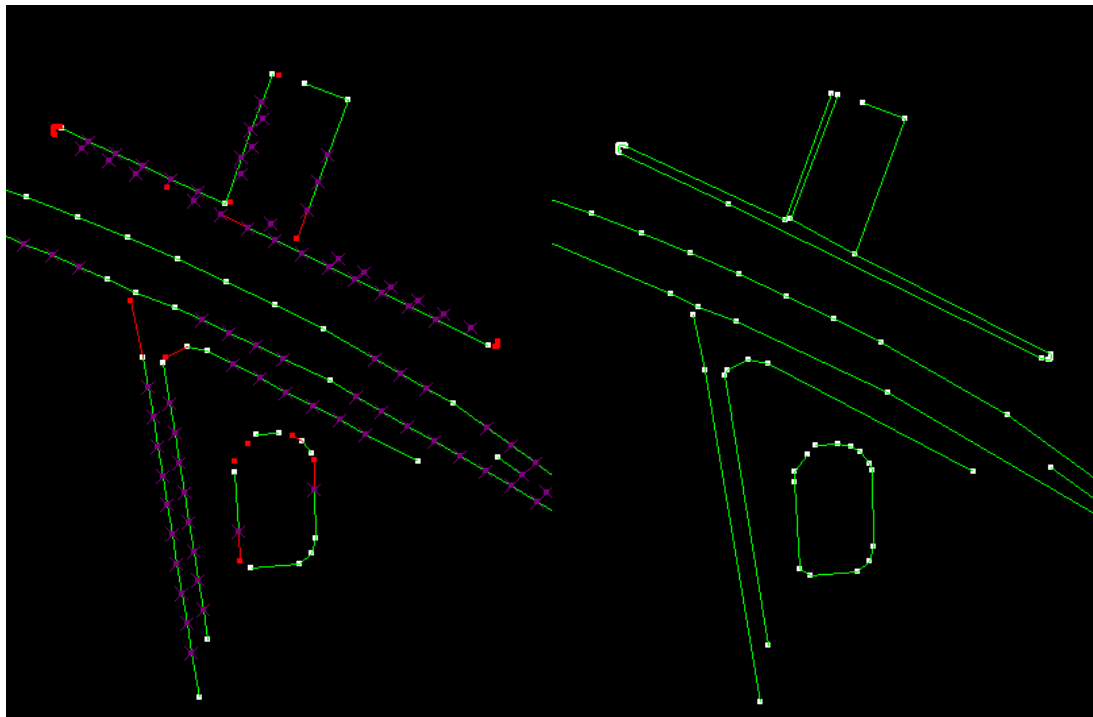
Orange callout boxes provide the following explanations:

- List of the loaded shapes:** Points to the 'Table Of Contents' in the Input section.
- Button to reload the already loaded shapes:** Points to the 'reload' button.
- Meshmaker creates a log file with the ending _mm.txt in the output directory upon hitting 'make'. This button can be used to reload it:** Points to the 'load log' button.
- Click to add shapes to your project:** Points to the 'load shape' button.
- Smartmeshing, see next page for details:** Points to the 'smartmeshing' checkbox.
- Output path:** Points to the output file path text field.
- Output cellsize:** Points to the 'grid resolution [m]' input field.
- Convex triangulation can be used if no boundary is present. Its use is not recommended though:** Points to the 'convex triangulation' checkbox.
- Keep the last triangulation step in Triangle format:** Points to the 'keep last triangulation' checkbox.
- Breakline enforcement is used to 'burn' breaklines into the resulting grid. Choose which breakline is used for this.:** Points to the 'breakline enforcement' dropdown menu.
- Set a bounding box for the final grid:** Points to the 'bounding box' checkbox.
- Create the grid:** Points to the 'make' button at the bottom right.
- Load Strickler polygon to create kst_grid for flox (this is optional):** Points to the file path text field.

Smartmeshing:

Smartmeshing provides a way to do a sort of filtering for your data. It uses a *hierarchical* approach, meaning that the shapefile you load with the smartmeshing option active, checks all data *already* in loaded (and itself). Data once loaded is never altered. This is one of the reasons you should load your boundary and other data of high importance first.

You can select a critical distance to be applied to the loading process. Points and lines closer to an already loaded dataset won't be loaded. Segments that come close to an already loaded dataset will be ignored as a whole. The support distance introduces a support points in segment, so that only the subsegments close to the existing data will be ignored and not the whole dataset. See this image for clarification:



Smartmeshing on.

Full dataset (smartmeshing off)

Purple points are support points. Red points and red lines are ignored. Green lines as well as purple and white points are in the mesh. Notice how in the top feature, all of the introduced support points are kept although the lines are very close to each other.