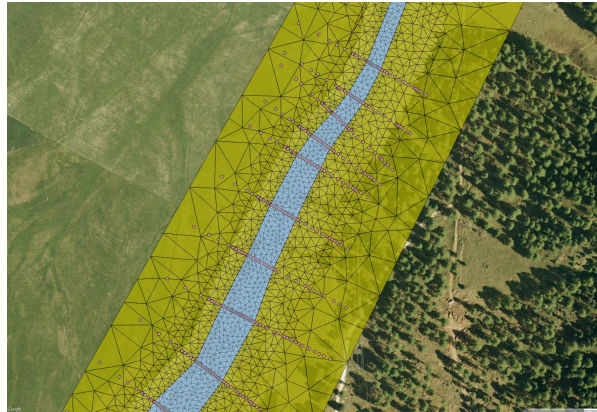


Manual

Pre- and Post-Processing in QGIS with BASEmesh

November 3, 2014



Contents

1	Introduction	1
1.1	Installation	1
1.2	Components of BASEmesh	2
1.3	Mesh representation in BASEmesh	4
2	Quality Meshing	6
3	Interpolation of Elevation Data	8
4	Elevation Meshing	9
5	Visualizing meshes in 3D	10
6	Renumbering Meshes	11
7	Finding Node IDs for <i>stringdef</i>	12
8	Import and Export of mesh - files	13
9	Visualization of Simulation Results (Post-Processing)	15
9.1	Visualization with the Crayfish plugin	15
9.2	Visualization with QGIS operations	16

1 Introduction

BASEmesh is a plugin to the free and open source geographic information system (GIS) software *Quantum GIS*¹ (QGIS). The plugin is a tool for creating computational meshes, based on the advanced mesh generator TRIANGLE by Jonathan R. Shewchuk². Furthermore, features for loading and editing existing meshes are provided. At the moment, BASEmesh only supports the automated creation of meshes consisting of triangular elements. Meshes containing quadrilateral elements, however, can be imported, manually edited and exported.

Beside 2D meshes, the plugin is also able to convert 1D meshes of different formats into each other (at the moment conversions are possible between BASEchain and HEC-RAS).

The meshes created by BASEmesh can directly be used for computations with the basic simulation environment software BASEMENT³. Post-processing, i.e. the visualization of simulation results, is not directly supported by BASEmesh. Still, there are various options for displaying and processing BASEMENT results in QGIS, which are presented in the last part of this manual.

The manual at hand introduces the features of BASEmesh and the basics of its usage. No information is given about the usage of GIS software in general. Some basic knowledge in applying GIS software might be acquired using QGIS' excellent documentation. To get in touch with BASEmesh and QGIS fast and easy, a step by step tutorial is provided as well and can be downloaded as separate document on the BASEmesh website.

This manual provides the reader with an overview of the capabilities of BASEmesh, the installation procedure, and a short description of the BASEmesh components that enable creating, importing, editing, and exporting computational meshes.

1.1 Installation

BASEmesh is at present available on a specific Plugin repository which has to be connected manually in the QGIS plugin manager. In contrast to other plugins, it is not available via the official QGIS plugin repository which is set as default in every QGIS installation.

To install BASEmesh, follow these steps:

1. Start QGIS
2. Load the QGIS plugin manager by choosing *Manage and Install Plugins...* in the menu *Plugins* in the QGIS main toolbar
3. Go to *Settings* (you should now see the connection to the official QGIS-plugin repository)
4. Click on *Add...* and give a name, e.g. 'BASEmesh repository'

¹<http://www.qgis.org/en>

²<http://www.cs.cmu.edu/~quake/triangle.html>

³<http://www.basement.ethz.ch/>

5. Enter the repository address: `http://people.ee.ethz.ch/~basement/qgis_plugins/qgis_plugins.xml`⁴
6. Press *OK*
7. The additional repository should now be visible (make sure that the *Status* is *connected*)
8. Go to *All* in the menu of the plugin manager and search for 'BASEmesh'
9. Choose the BASEmesh plugin (if several are available, choose the one with the highest version number) and press *Install plugin*

If everything worked, you should be able to access BASEmesh via the toolbar and via the menu *Plugins* (see Figure 1).

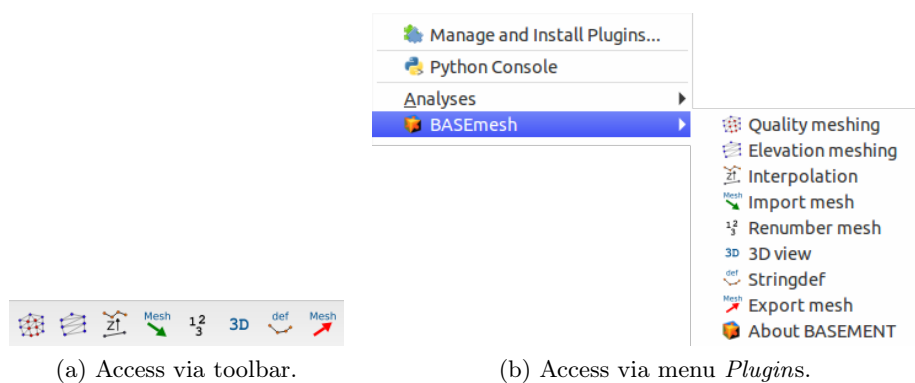


Figure 1: BASEmesh representation in QGIS after successful installation.

1.2 Components of BASEmesh

The philosophy of creating a computational mesh within BASEmesh is as follows (see Figure 2):

Generate a quality mesh Here you produce the actual computational grid ('quality mesh') with all its attributes you define, i.e. domain area, element sizes, mesh holes, break-lines or enforced vertices. This mesh does not contain any topographic information. The quality of this mesh influences the results of your numerical analysis, e.g. stability, computation time, accuracy etc.

Provide an elevation model The topography of your computational domain is usually described by points in a text file containing the (x,y,z)-coordinates or by a raster dataset. In case of xyz-data the elevation model has to be formulated as triangulated irregular network (TIN), called *elevation mesh* within BASEmesh.

⁴If you copy this adress, the 'tilde (~)' must be set correctly. Otherwise the plugin will not be found.

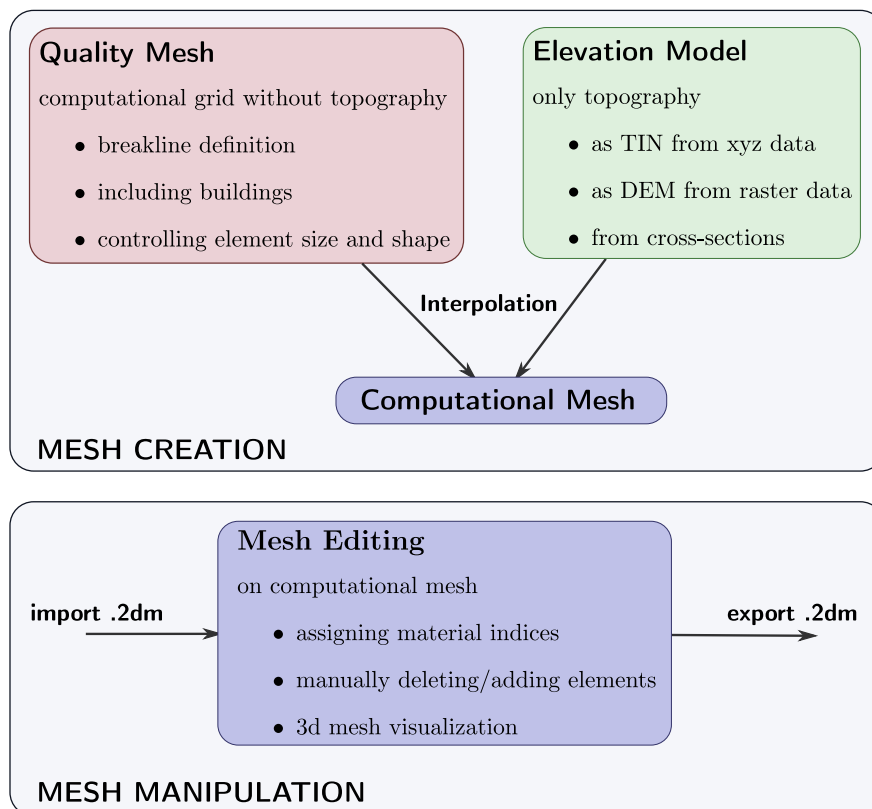


Figure 2: Pre-processing components of BASEmesh

Interpolation The topographical information contained in the elevation model will be interpolated on the quality mesh, i.e. an elevation value is assigned to each node of the computational grid.

In addition to mesh creation the possibility of manipulating an existing mesh is provided as well in BASEmesh (see Figure 2). Starting from a mesh you have created previously or you have imported to QGIS, you can edit the following:

Material indices This attribute is used in BASEMENT to distinguish between different materials which can be used to define friction, soil composition etc.

Manual meshing Using basic QGIS functions you can manually move nodes, add elements, remove elements, or change the triangulation.

Combine meshes Multiple meshes can be combined to one big mesh, by carefully applying manual meshing.

Renumber your mesh After applying manual changes to your mesh, you need to make sure that the internal representation of the mesh is correct. To this end BASEmesh

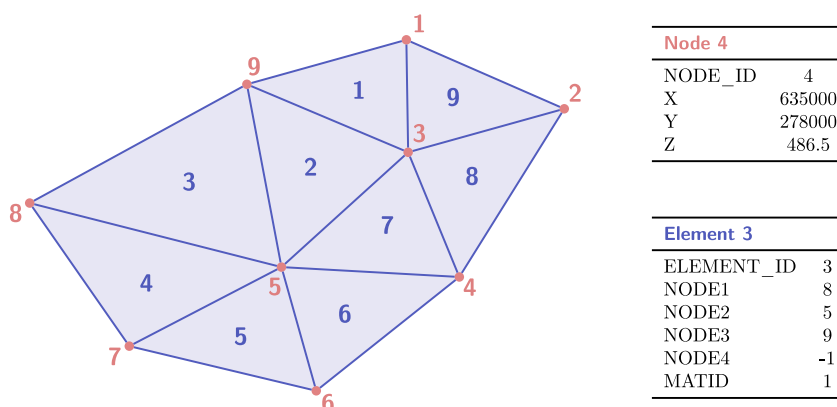


Figure 3: Example of internal mesh representation with nodes as point layer (red) and elements as polygon layer (blue).

provides a tool called *Renumber Mesh*, where all the manual changes are incorporated into a valid mesh structure (see 1.3).

Find *stringdef* nodes After changes of the computational mesh the node IDs may change as well, and all *stringdef* definitions of BASEMENT have to be updated. To this end BASEmesh provides a tool called *Stringdef*.

On top of that, at any step in the mesh creation process you can visualize your generated mesh using the **3D viewer**.

1.3 Mesh representation in BASEmesh

Meshes generated by BASEmesh follow a specific structure. When manually editing a mesh it is of high importance to maintain this structure, otherwise it is not possible to export the mesh to a 2dm - grid which can be used for computational purposes. The mesh structure of BASEmesh is defined as follows (see Figure 3):

- In BASEmesh, meshes are represented by two vector layers: (1) '*meshName_nodes*' containing only the nodes of the mesh and (2) '*meshName_elements*' containing only the elements of the mesh.
- '*meshName_nodes*' contains point features (2-dimensional geome). The attributes for each feature are *NODE_ID*, *X*, *Y*, and *Z*. Please note that the elevation information is only contained in the attribute table of the vector file.
- '*meshName_elements*' contains polygon features (2-dimensional). The attributes for each feature are *ELEMENT_ID*, *NODE1*, *NODE2*, *NODE3*, *NODE4*, and *MATID*. The attributes *NODE x* stand for the *NODE_ID*'s within the layer '*meshName_nodes*' and define the mesh connectivity.

- The *MATID* attribute is used to set material indices for each element within the mesh. BASEMENT uses these indices to set up zones with different characteristics in the mesh, e.g. to set different friction values in the river bed and in the flood plain of the domain.
- In case of triangular elements the attribute *NODE4* is -1. In case of quadrilateral elements, however, the *NODE4* attribute has to be specified.

Within QGIS, it is possible to create elements without defining the attributes *NODEx*. In this case, the tool *Renumber Mesh* must be used to update the connectivity of the mesh before exporting to a 2dm - grid.

2 Quality Meshing

The quality mesh defines the computational elements and nodes as they are used for the numerical computations. In contrast to DTM-models or TIN of elevation data, it has high demands concerning the quality of the elements (element size, element angles, shape, etc.), which directly influence the quality, stability and performance of the computations. It is important to have smooth transitions between elements with different sizes and to prevent distorted elements with small angles or very small element sizes. In general, a high-quality mesh is one which is nice to look at! The quality mesh contains all geometric information (x-y plane), but has no topographical information (z plane). Quality mesh generation involves mainly two steps:

1. Specification of all geometric information about the computational domain:
 - *Model boundary*: extent of the computational domain.
 - *Break lines*: distinct interruptions of the the surface slopes (dyke crest, river side walls, ...) which shall be preserved in the computational mesh.
 - *Holes*: parts within the mesh which are excluded from modelling (e.g. buildings). These parts are defined by special points surrounded by breaklines.
 - *Vertices*: enforced geometric points in the mesh (e.g. measurement points).
2. Generation of triangles respecting specific quality criteria using the free program TRIANGLE. Parameters of major importance are:
 - *Area points*: definition of the mesh density using maximum area constraints for the triangular mesh elements. These maximum areas are defined by points with given area constraints surrounded by breaklines.
 - *Minimum triangle angle*: no elements with angles smaller than the minimum angle specified are generated (smaller angles lead to less elements, while larger angles lead to more elements).

Since the quality mesh has no topographical information, it is usually necessary to interpolate data from an elevation model on the mesh nodes before using it for computations.

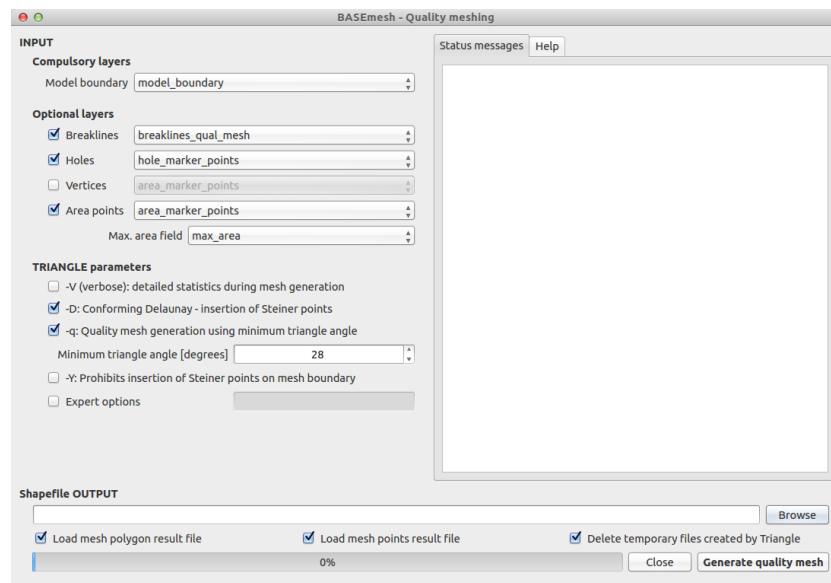


Figure 4: User interface for the creation of quality meshes.

3 Interpolation of Elevation Data

An important task in mesh generation is the interpolation of elevation data onto the nodes of the quality mesh. As a result, the final computational mesh is obtained, which then can be exported and used for simulations. The elevation data serving as input can be provided in two different elevation model types:

1. *Elevation mesh* triangulated from pointwise elevation data (TIN). The routine identifies the coordinates of each quality mesh node and determines any underlying elevation mesh element. Two methods are used for data interpolation:
 - a) If an underlying elevation mesh element is found, the elevation of the quality mesh node is interpolated at its x-y-coordinates. This is the normal case, since the elevation mesh usually covers the whole computational domain. Nodes interpolated with this method are marked by a '1' in the 'element' - field of the node attribute table⁵.
 - b) If no underlying elevation mesh element is found, the quality node elevation is set to that of the nearest node of the elevation mesh. This is the case if quality mesh nodes lie outside the domain covered by the elevation mesh or when holes are present in the elevation mesh. It may lead to incorrect quality mesh node elevations. Hence, it is recommended to choose a bigger domain for elevation meshing than for quality meshing. Nodes interpolated this way are marked by a '0' in the 'element' - field of the result attribute table⁶.
2. *Digital elevation map* as raster data which contains the topography as DTM. The raster elevation data is directly mapped on the computational mesh nodes without interpolation. If no corresponding raster cell is found, the elevation is set to '-999'.

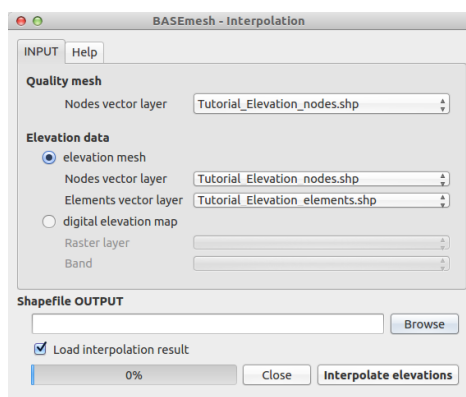


Figure 5: User interface for the interpolation of topographical data onto quality meshes.

⁵If the quality mesh node is located at the exact coordinates of an elevation mesh node, its height value is preserved exactly

⁶In the QGIS status messages, those nodes are named 'with special treatment'.

4 Elevation Meshing

The elevation mesh consists of triangulated elevation points and corresponding breaklines. It represents the topography of the computational domain and organizes the elevation data into a geometric structure which can be used for future elevation interpolations. The elevation mesh needs to be distinguished from the quality mesh, since it does not consider any quality aspects of the mesh elements and therefore should not be used for computations. The following data sources are combined:

1. *Elevation points* with x,y,z-coordinates. These elevation points may be taken from DTMs, cross section profiles, field measurements and other data sources. As input, all elevation points need to be combined into one single point shapefile.
2. *Breaklines* defining distinct interruptions of the the surface slopes (e.g. dyke crests or river side walls). Breaklines are given as line shapefile and connect elevation points with each other. During mesh generation, no triangles crossing the breaklines will be created. Be aware that all end points or vertices of a breakline must have corresponding elevation points!
3. *Model boundary* defining the extents of the elevation mesh. It must be given as polygon shapefile, which connects elevation points with each other. Be aware that all vertices of the model boundary polygon must have corresponding elevation points!

The elevation mesh is generated automatically with the program Triangle, without any parameters to be set.

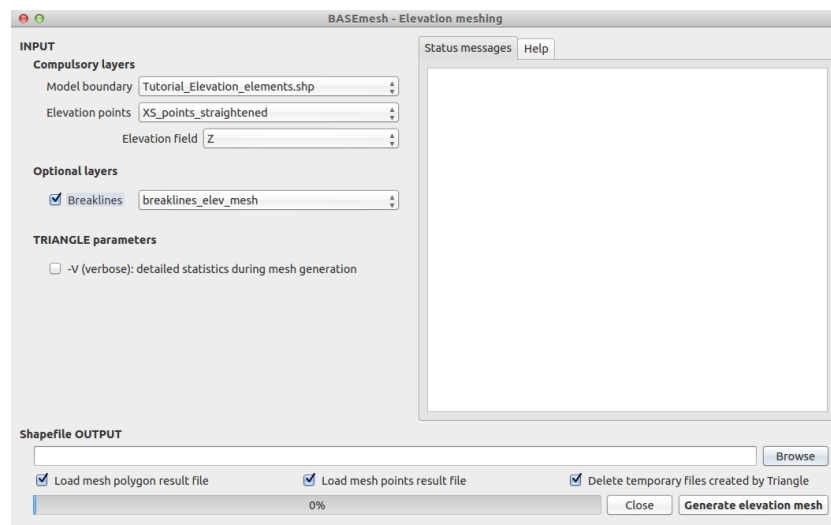


Figure 6: User interface for the creation of elevation meshes.

5 Visualizing meshes in 3D

During the process of mesh generation you can have a look at your mesh using the 3D viewer. The intention is to only have a quick glance to the third dimension of your mesh to detect potential errors or shortcomings in your mesh visually.

The 3D view tool requires the specification of a polygon shapefile with mesh elements and a point shapefile with the corresponding mesh nodes. The mesh nodes need to have elevation attributes and the mesh elements need to have material indices specified in their attribute table. Changing the *Elevation scaling factor* will stretch (>1.0) or squeeze (<1.0) the third dimension of your mesh view.

The window popping up shows your mesh (if only parts of your mesh is visible, zoom out). You can change the view as follows:

- zooming in and out: scroll using the wheel of your mouse
- shifting: drag and drop using the wheel button of your mouse
- rotating: drag and drop using left-click of your mouse (the orbit centre is located in the centre of your mesh)

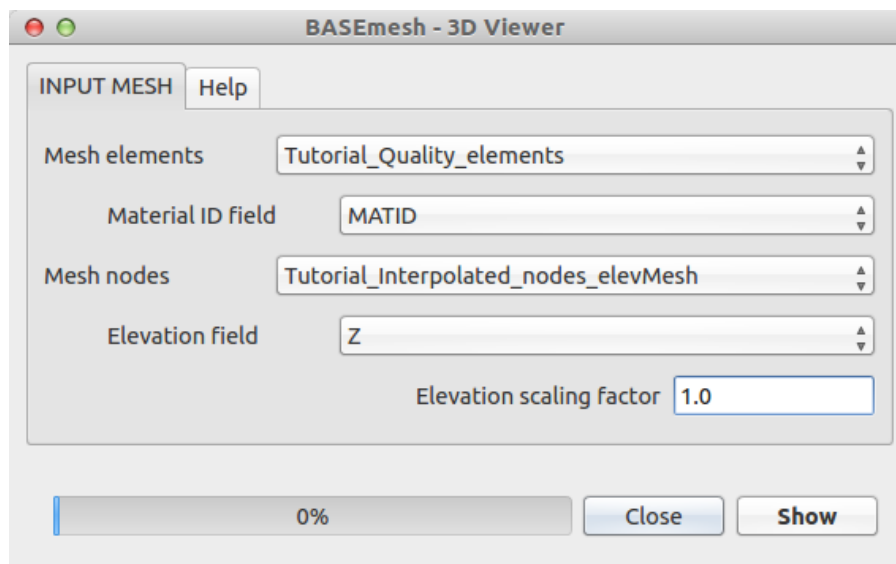


Figure 7: User interface for 3D view.

6 Renumbering Meshes

An unstructured mesh generally consists of a list of nodes, a list of elements and the mesh connectivity logic behind. The elements are formed by three or four nodes and dispose of a certain orientation (here: clockwise orientation). If a mesh is generated via BASEmesh using *quality meshing* or *elevation meshing*, then the mesh connectivity is always in a valid state. However, if the mesh is modified afterwards, e.g. by deleting or adding elements/nodes or swapping edges, then the mesh connectivity becomes invalid. Therefore, after modifications, the mesh connectivity has to be updated. This task is performed using the *Renumber Mesh* tool:

1. The mesh connectivity is checked and updated. For modified or added elements, the corresponding node ID-numbers are searched by comparisons of x-y coordinates. Furthermore, the clockwise element orientation is established by re-ordering the element nodes if necessary.
2. Nodes and elements are renumbered in a way that their ID-numbers begin ascending from '1' without gaps.

Renumbering requires a mesh as input data, i.e. a polygon shapefile containing the mesh elements and a point shapefile containing the mesh nodes. In addition, all elements must have material indices specified in their attribute table and all nodes must have elevation attributes. After renumbering, the mesh is in a valid state and can be exported for computations.

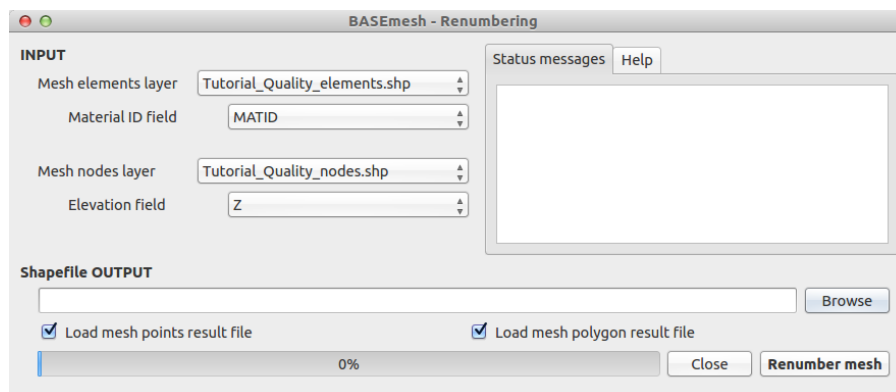


Figure 8: User interface for renumbering meshes.

7 Finding Node IDs for *stringdef*

In BASEMENT a list of node IDs is defined as *stringdef*. They can be used to define a boundary condition or an output along these nodes. The IDs correspond to the node IDs of the computational mesh. As soon as there are changes in this mesh, or only in its connectivity, the IDs will change as well and the list of node IDs inside of BASEMENT have to be updated. This task is automated with the tool *Stringdef* of BASEmesh:

1. For each line feature with a non-empty stringdef-field, all nodes that are located exactly on that line, are listed in a text file (BASEMENT-like stringdef block). The content of the stringdef-field represents the stringdef name.
2. The stringdef line features are favorably included in the layer where the breaklines of the quality mesh are defined. To distinguish between regular breaklines and stringdef lines, the stringdef-field can be used (empty vs. non-empty).
3. The upstream direction of the generated stringdefs is *right*. Therefore the line feature has to be drawn from the left riverbank to the right riverbank.

The Stringdef tool requires a point layer (i.e. the computational nodes) and a line layer (i.e. the breaklines with at least one stringdef line feature) as input. The resulting list(s) of node IDs is written as text file output that is user defined.

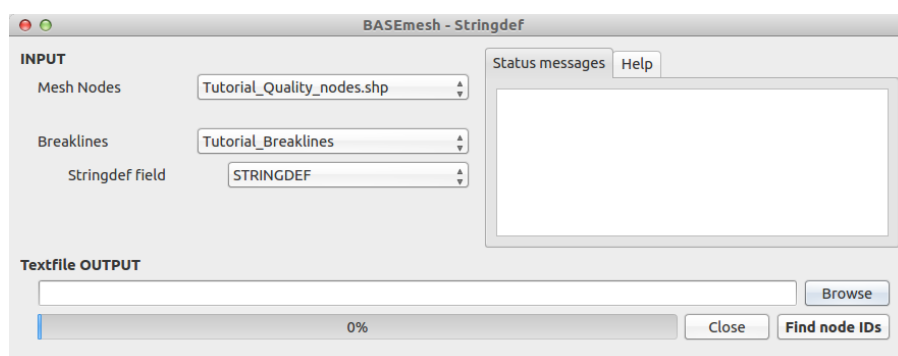


Figure 9: User interface for Stringdef.

8 Import and Export of mesh - files

2D mesh files:

The *.2dm*-format is a special ASCII data format for unstructured meshes specified by the SMS software⁷. Due to its simplicity, this data format is used by a variety of surface water modelling programs, as e.g. BASEMENT. A *.2dm*-file contains the node list with all coordinates and the elements with the mesh connectivity.

The main task of BASEmesh is the generation of a computational mesh in *.2dm*-format, which can be directly used for numerical simulations with BASEMENT. Therefore it is necessary to export the generated mesh to the *.2dm*-format. Furthermore, it is also often useful to visualize an existing *.2dm*-mesh, to change its material indices, to extract node or element numbers or to modify other mesh properties. For these purposes, BASEmesh provides options for importing and exporting *.2dm* - meshes.

1D mesh files:

Beside 2D meshes, BASEmesh is also able to import and export 1D mesh files. These file formats can be converted into each other. At the moment, two file formats are supported:

- BASEchain geometry files (*.bmg*), and
- HEC-RAS⁸ geometry files (*.g01*).

These 1D mesh functions are useful for visualization purposes of 1D meshes, as for example the lateral coupling of 1D and 2D models. Furthermore, the conversion utility allows for using the HEC-RAS software for pre-processing of 1D geometry files (e.g. extracting cross sections from digital elevation models (DEM) or interpolations of cross sections).

Import mesh After specification of the locations of the input and output files, the 2D / 1D mesh-file can be imported into QGIS. The mesh data are stored in shape-format within QGIS.

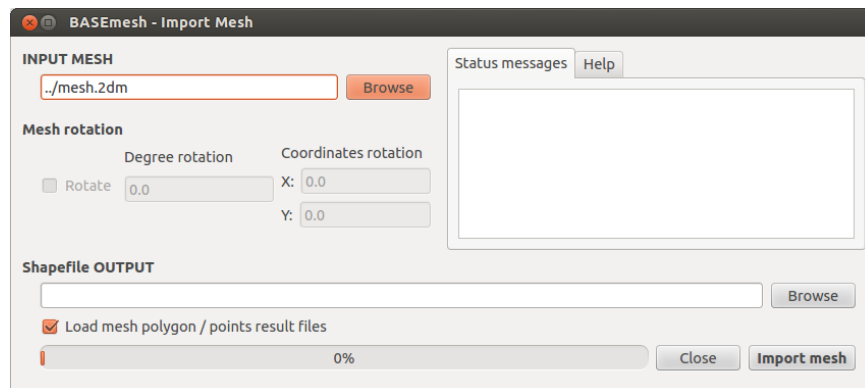
- 2D: Two shapefiles are generated with (a) element data (suffix: *'_elements.shp'*) and (b) node data (*'_nodes.shp'*), containing the mesh connectivity information and material indices.
- 1D: Two shapefiles are generate with (a) cross section data (suffix: *'_crossSections.shp'*) and (b) point data (*'_points.shp'*), containing the cross section data and the cross section points.

⁷SMS (Surface-water Modelling Software) is a commercial pre- and post-processor for unstrucutred 2D grids from Aquaveo, <http://www.aquaveo.com/>.

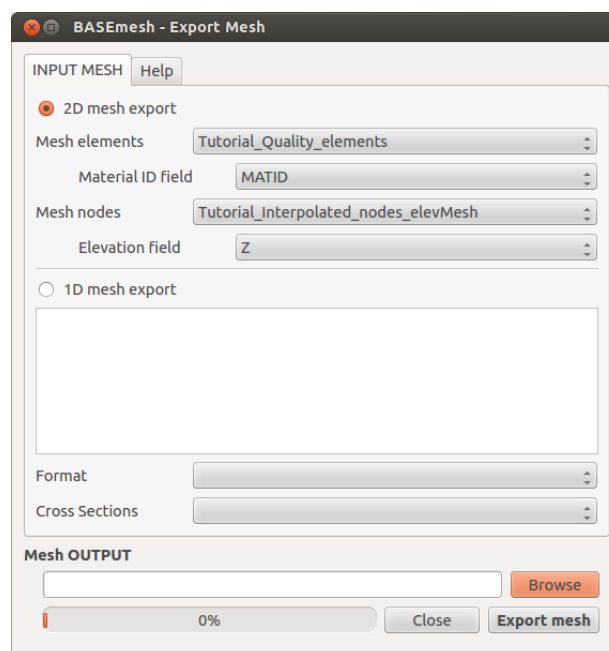
⁸HEC-RAS (= River Analysis Software) is a richly featured 1D simulation software with graphical pre- and post-processor developed by the US Army Corps of Engineers. It is available for free at <http://www.hec.usace.army.mil/software/hec-ras/>.

Export mesh

- 2D: Export into the 2D mesh-format requires the specification of a polygon shapefile with mesh elements and a point shapefile with the corresponding mesh nodes. The mesh elements need to have material indices specified in their attribute table and the mesh nodes need to have elevation attributes.
- 1D: Export into the 1D mesh-formats requires the specification of the file format and the cross section shapefile.



(a) Importing an existing mesh from a *.2dm*-file.



(b) Exporting a mesh from QGIS to file.

Figure 10: User interfaces for importing and exporting mesh files.

9 Visualization of Simulation Results (Post-Processing)

9.1 Visualization with the Crayfish plugin

Fast and easy visualizations of time dependent BASEMENT simulation results can be achieved using the Crayfish plugin⁹. The plugin is based on a fast engine which can deal with large and time dependent data sets in an efficient way. The plugin allows for plotting simulations results as adjustable color maps and offers additional visualization options like velocity vectors. To visualize BASEMENT results, two files need to be stored in the same folder:

1. The *.2dm* mesh file, containing all mesh information,
2. The *.sol* data file, containing time dependent flow data (e.g. water depths).

However, some special naming conventions need to be respected:

- Crayfish expects the data file name to contain the mesh name as prefix. After the prefix an underscore '_' is expected followed by the data name. Hence, the BASEMENT *.sol* files need to be renamed to fit this naming convention, e.g.
 - REGION_NDS_DEPTH.SOL → MESHNAME_DEPTH.SOL
- Crayfish does not allow for an underscore '_' within the flow variable name. Such underscores in the *.sol* files therefore need to be removed, e.g.
 - REGION_NDS_DEPTH_MAX.SOL → MESHNAME_DEPTHMAX.SOL

If these naming conventions are met, the data can be visualized as new Crayfish layer.

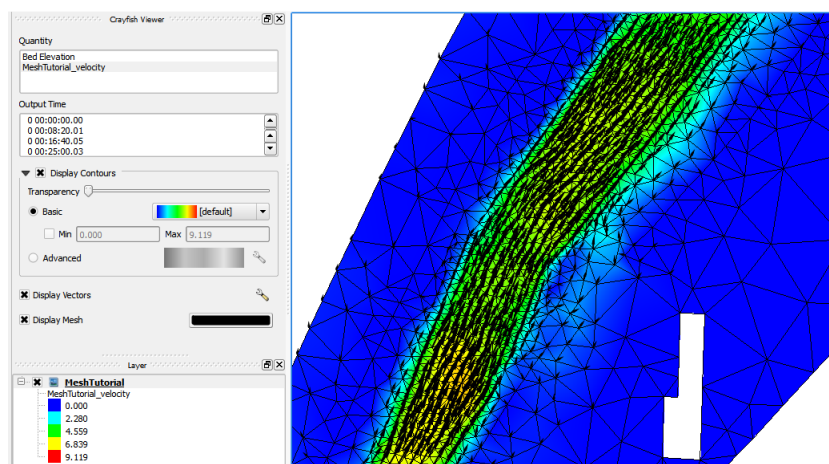


Figure 11: Screenshot of data visualization (velocity vectors) with the Crayfish plugin.

⁹The Crayfish plugin is developed by Lutra Consulting (www.lutraconsulting.co.uk/resources/crayfish) and can be downloaded from the official QGIS repository.

9.2 Visualization with QGIS operations

The BASEMENT simulation results can also be visualized using QGIS core functions, mainly via two approaches:

1. Shapefiles. BASEMENT has the option to write results files in the shape-format with each time step as column in the attribute table. These data files can be directly loaded into QGIS and visualized/colorized. A disadvantage of this visualization method is, however, that there is no smooth color transition between the discrete elements.
2. Raster data. Output data of BASEMENT (either as shapefiles or as ASCII-x-y-z data files¹⁰) can be rasterized in QGIS. Hereby, the pointwise data values are interpolated on a raster¹¹. Raster data are generally nice to work with, since GIS-applications have many and powerful tools to cope with raster data in various ways, e.g. to create contour lines or cross section profiles.

Post-processing with Shapefiles or Raster data is usually more time consuming than using the Crayfish plugin, especially if time dependent data shall be visualized. However, it may be a good choice especially if high-quality plots and graphics of a specific simulation timestep are needed and the powerful mapping tools of QGIS shall be used.

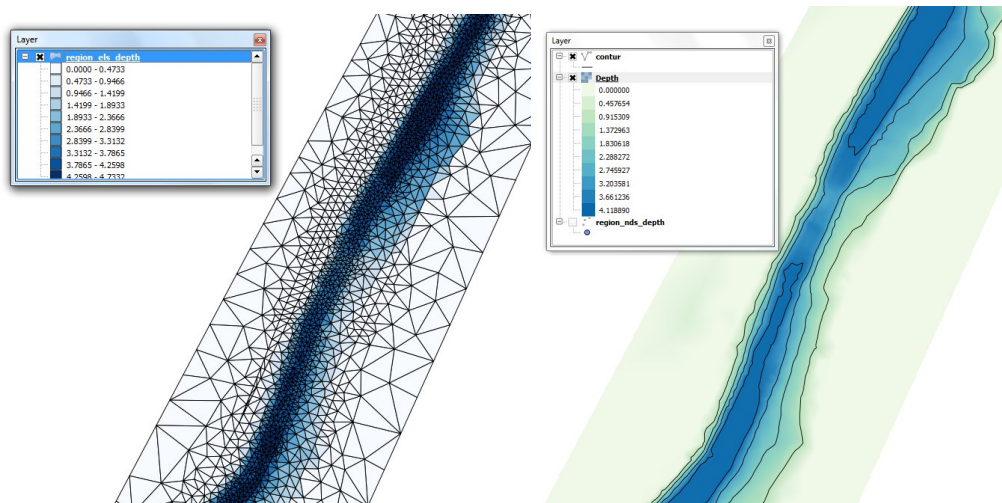


Figure 12: Screenshot of data visualization using shapefiles (left) and raster data (right).

¹⁰It is important to note that output can be generated as 'element - centered' data (in the middle of elements) and as 'node - centered' data (at nodes). Most data values are computed on the elements in BASEMENT. Therefore 'element - centered' data usually equals exactly the computed values, whereas 'node - centered' data is interpolated onto the node coordinates.

¹¹To rasterize the pointwise data values you can use the QGIS option *Raster* → *Interpolation*, which offers multiple interpolation algorithms, like TIN or inverse distance weighting. To colorize the raster data, open the properties dialog and select the option *singleband pseudocolor* and choose your preferred color map.