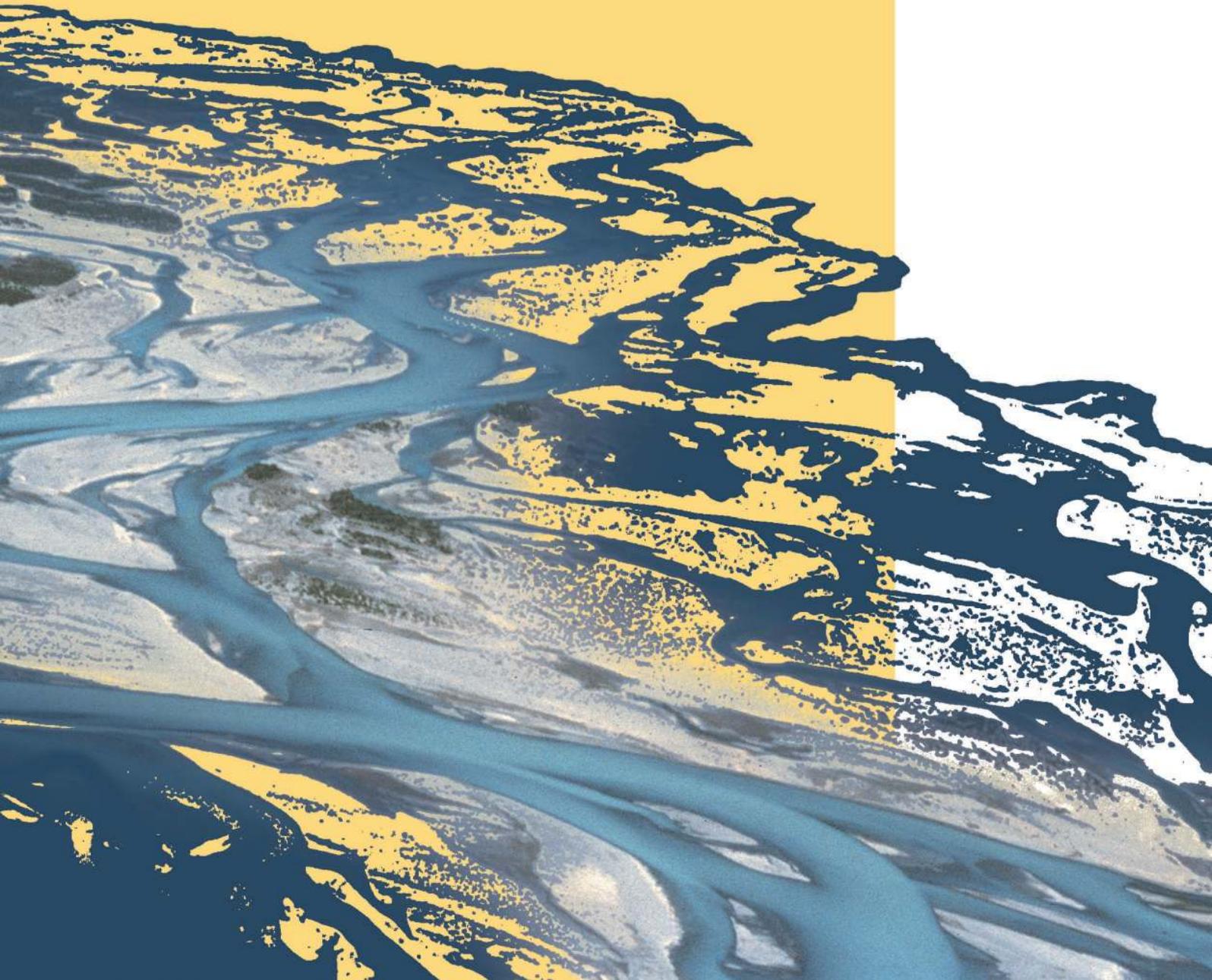




SYSTEM MANUALS

of BASEMENT



T

TUTORIALS

of **BASEMENT**



Contents

Preamble	5
Credits	5
License	7
1 Pre-Processing in QGIS with BASEmesh	13
1.1 Introduction	13
1.2 Tutorial 1: Mesh Generation based on Pointwise Elevation Data	14
1.2.1 Project Settings	15
1.2.2 Coordinate Reference System Configuration	16
1.2.3 Loading Input Data for Elevation Model	16
1.2.4 Saving Layer as Shape File	17
1.2.5 Loading the Model Boundary	17
1.2.6 Editing the Model Boundary	19
1.2.7 Loading Breakline Data	21
1.2.8 Creation of the Elevation Model as TIN	21
1.2.9 Adaption of the Breaklines for Quality Meshing	23
1.2.10 Creation of Region Marker Points	24
1.2.11 Creation of Quality Mesh	26
1.2.12 Interpolating elevation data from elevation mesh (TIN)	27
1.2.13 3D view of the mesh	29
1.2.14 Export of mesh to 2dm	29
1.3 Tutorial 2: Import/Modify an existing Mesh and use Raster Data as Elevation Model	30
1.3.1 Importing a <i>.2dm</i> mesh file	31
1.3.2 Modifying the material indices of elements	31
1.3.3 Manual editing of mesh elements	35
1.3.4 Renumbering the mesh	36
1.3.5 Interpolating elevations from raster data	37
1.4 Tutorial 3: Using dividing constraints along boundary cross sections and setting up a BASEMENT simulation with a mesh from BASEmesh	37
1.4.1 Using dividing constraints for Quality meshing	39
1.4.2 Extraction of Stringdef information from the mesh	40
1.4.3 Set-up BASEMENT command file: Add mesh	42
1.4.4 Set-up BASEMENT command file: Add stringdefs	43
1.4.5 Set-up BASEMENT command file: Add friction values	43
1.4.6 Data Visualization	44
1.5 Tutorial 4: Create 1D BASEchain mesh from a DTM using HEC-RAS/ HEC-GeoRAS	45

1.5.1	Set-up the environment	47
1.5.2	Generate the cross section profiles	48
1.5.3	Set the cross section, main channel and friction properties	49
1.5.4	Export the data from HEC-GeoRAS	49
1.5.5	Import GIS data into HEC-RAS and export a geometry file	49
1.5.6	Convert HEC-RAS format to BASEchain format using BASEmesh	50
1.6	Tips and Tricks	53
1.6.1	Recommended Plugins	53
1.6.2	Tipps and Tricks	53
2	Post-Processing of 2D simulation results	57
2.1	Introduction	57
2.2	2D result visualization with QGIS Crayfish	57
2.2.1	Input data	57
2.2.2	About Crayfish	58
2.2.3	Installation	58
2.2.4	Load and visualize data	60
2.2.5	Creating maps and animations	67
2.3	3D result visualization with Paraview	74
2.3.1	Input data	74
2.3.2	About Paraview	75
2.3.3	Installation	75
2.3.4	User Interface	76
2.3.5	Import Data	77
2.3.6	ParaView Filters	78
2.3.7	Exporting figures and animations	82
3	Hydrodynamics and sediment transport at the river Thur (1D)	87
3.1	Introduction	87
3.1.1	General description	87
3.1.2	Used features	88
3.1.3	Purpose	88
3.2	Setting up the topography file	88
3.2.1	Cross sections	88
3.2.2	Definition of different cross section zones	89
3.2.3	Friction values	90
3.2.4	Computation of water surface elevation	92
3.2.5	Characterisation of the sediments	92
3.2.6	Define flowing zones	93
3.3	Setting up the command file	93
3.3.1	Project	93
3.3.2	Domain	94
3.3.3	Define the physical properties	94
3.3.4	One dimensional simulation	94
3.4	Perform hydraulic simulations	97
3.4.1	Perform steady flow simulation (Thur1)	97
3.4.2	Perform simulation of the floods (Thur2)	98
3.5	Complete the command file for bed load transport	102

3.5.1	Define the bed material	102
3.5.2	Soil assignment	103
3.5.3	Define general parameters for sediment transport	103
3.5.4	Define specific parameters for bed load transport	104
3.5.5	Define bed load transport formula	104
3.5.6	Define boundary conditions for bed load	104
3.5.7	Generate a “geometry” file	105
3.6	Perform bed load simulation (Thur 3)	105
4	Dynamics of longitudinal bed profile due to local river widening (1D)	107
4.1	Introduction	107
4.1.1	General description	107
4.1.2	Purpose	108
4.1.3	Used features	109
4.1.4	Parameter variation	110
4.2	Model setup	110
4.2.1	Definition of 1D topography	110
4.2.2	Determination of the upstream sediment boundary conditions	111
4.3	Results of numerical simulations	111
4.3.1	Temporal evolution of longitudinal bed profile	111
4.3.2	Width and length of the widening	113
4.3.3	Discharge and grain diameter	113
5	Hydrodynamics and sediment transport at the river Flaz (2D)	117
5.1	Introduction	117
5.1.1	Case study description	117
5.1.2	Tutorial structure	118
5.2	Computational grid	118
5.3	Setting up the command file	120
5.3.1	Project	120
5.3.2	Domain	120
5.3.3	Parallel	121
5.3.4	Physical properties	121
5.3.5	Two dimensional simulation	121
5.4	Perform hydraulic simulation	126
5.4.1	Perform steady flow simulation	126
5.4.2	Perform unsteady flow simulation	127
5.4.3	Calibration of the hydraulic model	131
5.5	Morphological simulation with single-grain bed load transport	132
5.5.1	Define the morphological information	133
5.5.2	Define the output	138
5.6	Perform morphological simulation with single-grain bed load transport	139
5.7	Morphological simulation with multi-grain bed load transport	142
5.7.1	Morphological parameters	142
5.7.2	Grain size distribution	142
5.7.3	Grain mixture	142
5.7.4	Define the soil composition	143
5.7.5	Bed load boundary condition	144

5.7.6	Bed load formula	145
5.7.7	Define the output	145
5.8	Perform morphological simulation with multi-grain bed load transport . . .	146
6	Laterally coupled 1D-2D hydrodynamic simulation	149
6.1	Introduction	149
6.2	Set-up of command file	150
6.2.1	General remarks on mesh creation	150
6.2.2	BASEchain (1D) river model	150
6.2.3	BASEplane (2D) floodplain model	152
6.2.4	Lateral coupling set-up	153
6.3	Coupling connections between 1D and 2D subdomains	154
6.3.1	Definition of coupling connections	154
6.3.2	Automatic generation of coupling connections	155
6.4	Perform coupled simulation	157

Preamble

VERSION 2.8

May, 2018

Credits

Project Team

Software Development, Documentation and Test (alphabetical)

F. Caponi, MSc. Environmental Eng.
D. Ehrbar, MSc. ETH Civil Eng.
E. Gerke, MSc. ETH Civil Eng.
S. Kammerer, MSc. ETH Environmental Eng.
A. Koch, MSc. ETH Civil Eng.
Dr. S. Peter, MSc. ETH Civil Eng.
L. Vonwiller, MSc. ETH Environmental Eng.

Scientific Board

Prof. Dr. R. Boes, Director VAW, Member of Project Board
Dr. A. Siviglia, MSc, Scientific Advisor
Dr. D. Vanzo, MSc. Environmental Eng., Scientific Advisor
Dr. D. Vetsch, Dipl. Ing. ETH, Project Director

Former Project Members

em. Prof. Dr.-Ing. H.-E. Minor, Director of VAW 1998-2008
Dr. R. Fähr, Dipl. Ing. ETH, Scientific Supervisor, 2002-2013
Dr.-Ing. D. Farshi, MSc., Software Development, 2002-2007
Dr. R. Veprek, Dipl. Rech. Wiss. ETH, Software Development, 2009-2010
R. Müller, Dipl. Ing. EPFL, Software Development, 2003-2012
P. Rousselot, Dipl. Rech. Wiss. ETH, Software Development, 2006-2014
Dr. C. Volz, Dipl.-Ing. Umwelttechnik, Software Development, 2007-2015
M. Gerber, BSc. Software Eng., 2015-2016
Dr. M. Facchini, MSc. Environmental Eng., 2013-2017

Cover Page Art Design

W. Thürig

Commissioned and co-financed by

Swiss Federal Office for the Environment (FOEN)

Contact

website: <http://www.basement.ethz.ch>

user forum: <http://people.ee.ethz.ch/~basement/forum>

© 2006–2018 ETH Zurich / Laboratory of Hydraulics, Glaciology and Hydrology (VAW)
For list of contributors see www.basement.ethz.ch



Laboratory of Hydraulics,
Hydrology and Glaciology



Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Citation Advice

For System Manuals:

Vetsch D., Siviglia A., Caponi F., Ehrbar D., Gerke E., Kammerer S., Koch A., Peter S., Vanzo D., Vonwiller L., Facchini M., Gerber M., Volz C., Farshi D., Mueller R., Rousselot P., Veprek R., Faeh R. 2018. System Manuals of BASEMENT, Version 2.8. Laboratory of Hydraulics, Glaciology and Hydrology (VAW). ETH Zurich. Available from <http://www.basement.ethz.ch>. [date of access].

For Website:

BASEMENT – Basic Simulation Environment for Computation of Environmental Flow and Natural Hazard Simulation, 2018. <http://www.basement.ethz.ch>

For Software:

BASEMENT – Basic Simulation Environment for Computation of Environmental Flow and Natural Hazard Simulation. Version 2.8. © ETH Zurich, VAW, 2006-2018.

License

BASEMENT SOFTWARE LICENSE

between

ETH

Rämistrasse 101

8092 Zürich

Represented by Prof. Dr. Robert Boes

VAW

(Licensor)

and

Licensee

1. Definition of the Software

The Software system BASEMENT is composed of the executable (binary) file BASEMENT and its documentation files (System Manuals), together herein after referred to as “Software”. Not included is the source code.

Its purpose is the simulation of water flow, sediment and pollutant transport and according interaction in consideration of movable boundaries and morphological changes.

2. License of ETH

ETH hereby grants a single, non-exclusive, world-wide, royalty-free license to use Software to the licensee subject to all the terms and conditions of this Agreement.

3. The scope of the license

a. Use

The licensee may use the Software:

- according to the intended purpose of the Software as defined in provision 1
- by the licensee and his employees
- for commercial and non-commercial purposes

The generation of essential temporary backups is allowed.

b. Reproduction / Modification

Neither reproduction (other than plain backup copies) nor modification is permitted with the following exceptions:

Decoding according to article 21 URG [Bundesgesetz über das Urheberrecht, SR 231.1]

If the licensee intends to access the program with other interoperative programs according to article 21 URG, he is to contact licensor explaining his requirement.

If the licensor neither provides according support for the interoperative programs nor makes

the necessary source code available within 30 days, licensee is entitled, after reminding the licensor once, to obtain the information for the above mentioned intentions by source code generation through decompilation.

c. Adaptation

On his own risk, the licensee has the right to parameterize the Software or to access the Software with interoperable programs within the aforementioned scope of the licence.

d. Distribution of Software to sub licensees

Licensee may transfer this Software in its original form to sub licensees. Sub licensees have to agree to all terms and conditions of this Agreement. It is prohibited to impose any further restrictions on the sub licensees' exercise of the rights granted herein.

No fees may be charged for use, reproduction, modification or distribution of this Software, neither in unmodified nor incorporated forms, with the exception of a fee for the physical act of transferring a copy or for an additional warranty protection.

4. Obligations of licensee

a. Copyright Notice

Software as well as interactively generated output must conspicuously and appropriately quote the following copyright notices:

Copyright by ETH Zurich / Laboratory of Hydraulics, Glaciology and Hydrology (VAW), 2006-2018

5. Intellectual property and other rights

The licensee obtains all rights granted in this Agreement and retains all rights to results from the use of the Software.

Ownership, intellectual property rights and all other rights in and to the Software shall remain with ETH (licensor).

6. Installation, maintenance, support, upgrades or new releases

a. Installation

The licensee may download the Software from the web page <http://www.basement.ethz.ch> or access it from the distributed CD.

b. Maintenance, support, upgrades or new releases

ETH doesn't have any obligation of maintenance, support, upgrades or new releases, and disclaims all costs associated with service, repair or correction.

7. Warranty

ETH does not make any warranty concerning the:

- warranty of merchantability, satisfactory quality and fitness for a particular purpose
- warranty of accuracy of results, of the quality and performance of the Software;
- warranty of noninfringement of intellectual property rights of third parties.

8. Liability

ETH disclaims all liabilities. ETH shall not have any liability for any direct or indirect damage except for the provisions of the applicable law (article 100 OR [Schweizerisches Obligationenrecht]).

9. Termination

This Agreement may be terminated by ETH at any time, in case of a fundamental breach of the provisions of this Agreement by the licensee.

10. No transfer of rights and duties

Rights and duties derived from this Agreement shall not be transferred to third parties without the written acceptance of the licensor. In particular, the Software cannot be sold, licensed or rented out to third parties by the licensee.

11. No implied grant of rights

The parties shall not infer from this Agreement any other rights, including licenses, than those that are explicitly stated herein.

12. Severability

If any provisions of this Agreement will become invalid or unenforceable, such invalidity or enforceability shall not affect the other provisions of Agreement. These shall remain in full force and effect, provided that the basic intent of the parties is preserved. The parties will in good faith negotiate substitute provisions to replace invalid or unenforceable provisions which reflect the original intentions of the parties as closely as possible and maintain the economic balance between the parties.

13. Applicable law

This Agreement as well as any and all matters arising out of it shall exclusively be governed by and interpreted in accordance with the laws of , excluding its principles of conflict of laws.

14. Jurisdiction

If any dispute, controversy or difference arises between the Parties in connection with this Agreement, the parties shall first attempt to settle it amicably. Should settlement not be achieved, the Courts of Zurich-City shall have exclusive jurisdiction. This provision shall only apply to licenses between ETH and foreign licensees

By using this software you indicate your acceptance.

(License version: 2018-05-31)

THIRD PARTY SOFTWARE COPYRIGHT NOTICES

The Visualization Toolkit (VTK)

VTK is an open-source toolkit licensed under the BSD license.

Copyright (c) 1993-2008 Ken Martin, Will Schroeder, Bill Lorensen All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither name of Ken Martin, Will Schroeder, or Bill Lorensen nor the names of any contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHORS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

CVM Class Library

Copyright (c) Sergei Nikolaev, 1992-2016

Boost Software License - Version 1.0 - August 17th, 2003

Permission is hereby granted, free of charge, to any person or organization obtaining a copy of the software and accompanying documentation covered by this license (the "Software") to use, reproduce, display, distribute, execute, and transmit the Software, and to prepare derivative works of the Software, and to permit third-parties to whom the Software is furnished to do so, all subject to the following:

The copyright notices in the Software and this entire statement, including the above license grant, this restriction and the following disclaimer, must be included in all copies of the Software, in whole or in part, and all derivative works of the Software, unless such copies or derivative works are solely in the form of machine-executable object code generated by a source language processor.

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. IN NO EVENT SHALL THE COPYRIGHT HOLDERS OR ANYONE DISTRIBUTING THE SOFTWARE BE LIABLE FOR ANY DAMAGES OR OTHER LIABILITY, WHETHER IN CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Qt Toolkit - Cross-platform application and UI framework

The Qt Toolkit is Copyright (C) 2016 The Qt Company Ltd. and other contributors. Contact: <http://www.qt.io/licensing/>

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; LGPL version 3.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public License (LGPL version 3) along with this library; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Qwt - Qt Widgets for Technical Applications

BASEMENT is based in part on the work of the Qwt project (<http://qwt.sf.net>).

CGNS – CFD General Notation System

This software is provided “as-is”, without any express or implied warranty. In no event will the authors be held liable for any damages arising from the use of this software.

Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.
3. This notice may not be removed or altered from any source distribution.

This license is borrowed from the zlib/libpng License, and supercedes the GNU Lesser General Public License (LGPL) which previously governed the use and distribution of the software.

Shapelib

Copyright (c) 1999, Frank Warmerdam

This software is available under the following “MIT Style” license, or at the option of the licensee under the LGPL (see COPYING). This option is discussed in more detail in shapelib.html.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Included versions

Library	Description	Ubuntu16	Ubuntu18	Windows10
Qt	cross-platform application framework and widget toolkit for creating classic and embedded graphical user interfaces	5.5.1	5.9.5	5.8.0
VTK	open-source, freely available software system for 3D computer graphics, image processing and visualization	6.2.0	6.3.0	6.3.0
Qwt	set of custom Qt widgets, GUI Components and utility classes which are primarily useful for programs with a technical background	6.1.2	6.1.3	6.1.3
CGNS	a general, portable, and extensible standard for the storage and retrieval of CFD analysis data	3.1.4	3.3.0	3.2.1
shapelib	library for reading and writing ESRI Shapefiles - development files	1.3.0	1.4.1	1.5.5
tecIO	library for generating tecplot binary outputs	?	?	?

Pre-Processing in QGIS with BASEmesh

1.1 Introduction

What is the goal of these tutorials? The following tutorials introduce to the creation of computational meshes using BASEmesh and the integrated mesh generator Triangle. Besides mesh generation, features for loading and editing existing meshes are presented. Not all features of BASEmesh can be covered, but using the available tutorials will give you an impression of its workflow and its capabilities. For specific questions, a help function describing the necessary input layers and parameters is included in every tool of BASEmesh and can be accessed under the tab Help. Furthermore, these tutorials introduce to several basic GIS - operations using QGIS. As all features and aspects of QGIS cannot be covered, we recommend the excellent documentation of [QGIS](#) for specific features or tasks.

What is the philosophy of mesh creation with BASEmesh? BASEmesh is a free and open source pre- and postprocessing QGIS plugin. It generates meshes for the numerical simulation software [BASEMENT](#) using Jonathan Richard Shewchuk's mesh generator 'Triangle'. The focus of BASEmesh is on the automatic generation of unstructured meshes based on specific quality criteria. BASEmesh follows the philosophy of separating the tasks of high-quality mesh generation and the generation/use of elevation models.

What is the workflow in BASEmesh? Please take a look at the figure below to see the workflow in BASEmesh. The tutorials cover the indicated items:

- Tutorial 1: Creation and export of a computational mesh for BASEMENT using pointwise elevation data.
- Tutorial 2: Modifying of a computational mesh and using raster elevation data.
- Tutorial 3: Using dividing constraints along boundary cross sections and setting up a BASEMENT simulation.
- Tutorial 4: Create 1D cross sections with HEC-RAS and use BASEmesh to convert them into BASEMENT format.

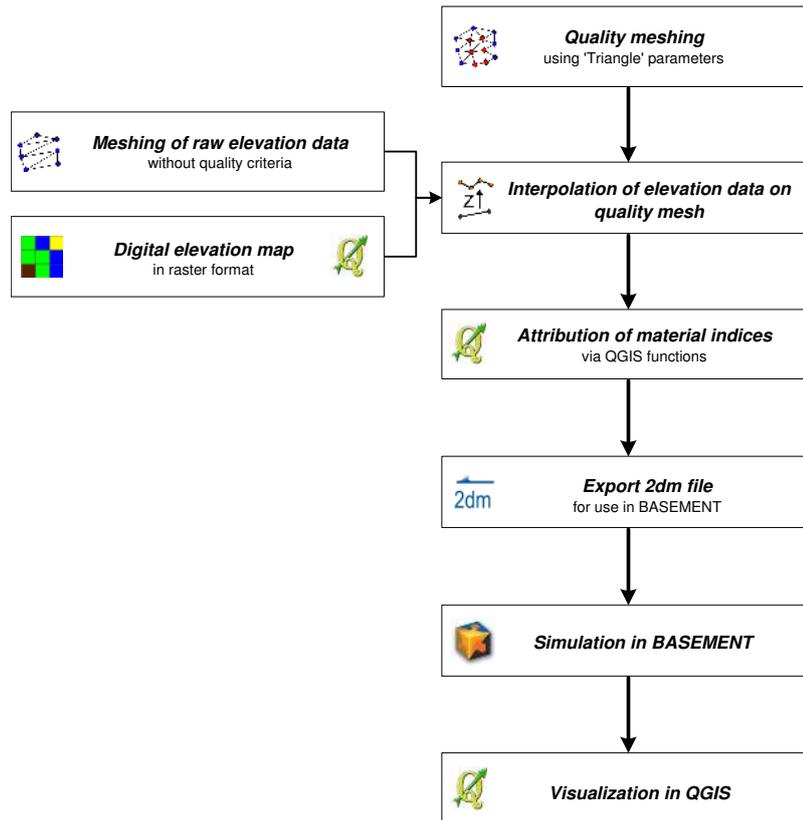


Figure 1.1 General workflow for the creation of a computational mesh with BASEmesh.

For further details, please refer to Section 3.3.5 “Use of QGIS plugin BASEmesh for grid generation” of the BASEMENT User Manual.

1.2 Tutorial 1: Mesh Generation based on Pointwise Elevation Data

The following tutorial illustrates the consecutive steps to create a high - quality computational mesh based on pointwise elevation data stored in a text file. The elevation data in this tutorial is represented by cross section data, gathered in a river restoration project in Switzerland by terrestrial survey. All other files have been edited or created based on this elevation data. Therefore this tutorial exemplifies the mesh generation based on given river cross section data. Another typical task, e.g. for flood simulations with overland flow, is the generation of meshes based on a digital elevation model (DEM) in raster format (see Section 1.3).

Input Data The data needed to complete this tutorial comes as ZIP - file and needs to be extracted to a location of your choice. All screenshots and figures in this document were taken from QGIS version 2.8 ‘Wien’. All data files have to be loaded into the QGIS - project before executing the different tools, as it is not possible to select files directly on the hard drive by browsing. Furthermore, those files must be activated in the QGIS table of contents (TOC this abbreviation will mostly be used in this tutorial) on the left side of the screen. To prevent the selection of wrong shapetypes, the available fields are populated with the corresponding data type.

Rule of thumb Only data that is displayed on the map can be used for meshing.

Normally, only 4 shapefiles are needed as input data for this tutorial. All other input files are created using BASEmesh and several QGIS functions during completion. In case of difficulties, the result files for each step are additionally provided in the subfolder called ‘additional_files’.

1.2.1 Project Settings

- (1) Start QGIS and make sure that the plugin BASEmesh is successfully installed (see Section 3.3.5 of the BASEMENT User Manual).

It is advisable to create a QGIS project with a meaningful name. The project’s name is used as basis for most of the files created with BASEmesh.

- (2) Go to *Project* → *Project Properties*.
- (3) Under *General* → *General settings* you will find the field *Project title*. Enter a name of your choice. Here the name ‘Tutorial’ was chosen.

In this tutorial we need the Swiss projection ‘CH1903 / LV03’. In the following steps, the project’s coordinate reference system (CRS) will be changed.

- (4) Under *CRS* you can see the coordinate reference system settings for this project. Check *Enable ‘on the fly CRS’ transformation* only to change the reference system.
- (5) Enter the EPSG code ‘21781’ in the field *Filter*.
- (6) Select the coordinate reference system ‘CH1903 / LV03’.
- (7) Click *OK*.
- (8) Again go to *Project* → *Project Properties* → *CRS*.
- (9) Now make sure to **uncheck** *Enable ‘on the fly CRS’ transformation* as in the Figure 1.2!
- (10) Close the project properties window. If everything went well, you should see the chosen project name at the title of the QGIS main window and the EPSG code of your coordinate system at the lower right corner of the QGIS desktop.
- (11) Now save the project: Go to *Project* → *Save*. The project name chosen before is automatically proposed.

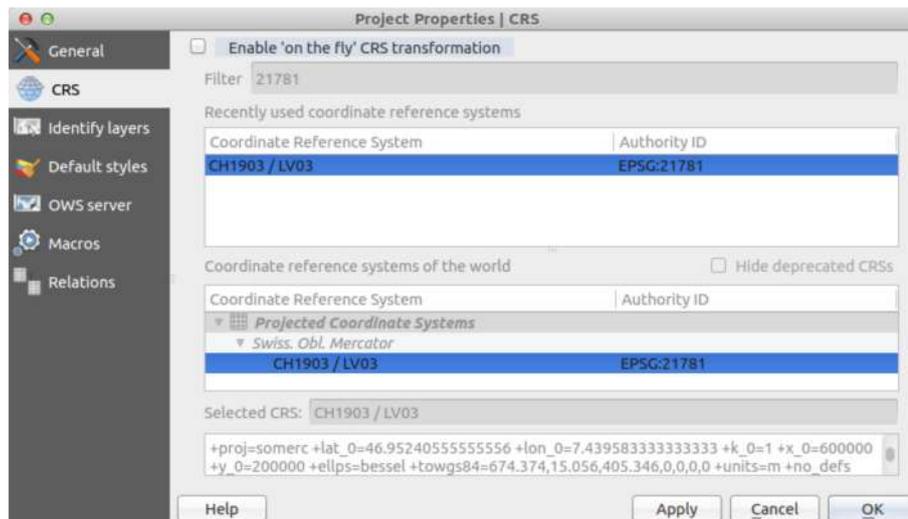


Figure 1.2 Project Properties, CRS

1.2.2 Coordinate Reference System Configuration

Now we need to check how QGIS determines the coordinate system of added layers. This may vary between QGIS versions and operating systems.

- (1) Go to *Settings* → *Options*
- (2) Under *CRS* fill in the dialog as shown in Figure 1.3. Make sure that *Use project CRS* under *CRS for new layers* is checked.
- (3) Close the dialog.

Now that the project and coordinate reference system properties have been set successfully, we can start adding data to the current project.

1.2.3 Loading Input Data for Elevation Model

In this tutorial the elevation model is represented by a triangulated irregular network TIN, in the following referred to as 'elevation mesh'. Thus, our first step is loading the elevation data into our QGIS project. This data originates from cross sectional data and is stored in a delimited text file, where the values are separated by a comma. We will load this text file and convert it to a shapefile, which is more suitable for working in a GIS environment.

- (1) Go to *Layer* → *Add Delimited Text Layer*.
- (2) Fill in the dialog that shows up (see Figure 1.4): Browse for the file containing the elevation data, 'XS_points_straightened.txt' provided with this tutorial. Be sure to select *Comma* as delimiter. Otherwise QGIS will not be able to separate the values found in the file. Under *Geometry definition* select *Point coordinates* and verify if the X and Y coordinates are set correctly.
- (3) Be sure to check *Use spatial Index*.

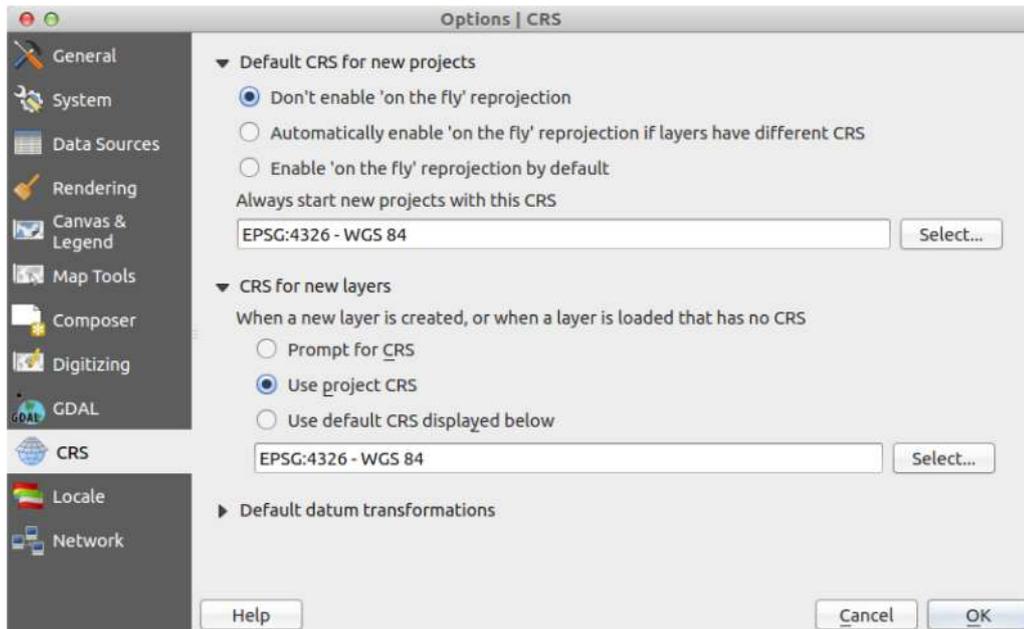


Figure 1.3 Settings, CRS

- (4) At the bottom of the dialog a preview of the file content using the selected delimiter and coordinate fields is given. If everything seems to be correct and according to Figure 1.4 , click *OK*.
- (5) After successful import you will see the river cross sections displayed in the QGIS map canvas. The flow direction is from the bottom left corner to the upper right corner.

1.2.4 Saving Layer as Shape File

Loading the data from a text file into QGIS will not create a shapefile automatically. The data shown as vector layer is still stored in the text file. We will now convert the data into a shapefile.

- (1) Right-click on layer 'XS_points_straightened' in the TOC.
- (2) Go to *Save As...*
- (3) Fill in the dialog according to Figure 1.5 and click *OK*.
- (4) You will now see two layers in the TOC with the same name. The layer that represents the data from the text file can now be deleted by right-clicking on it in the TOC and selecting *Remove*. If both layers have the same symbology, you can check its properties by right-clicking and selecting *Metadata*.

1.2.5 Loading the Model Boundary

- (1) Go to *Layer → Add Vector Layer...* (or Ctrl-Shift-V).

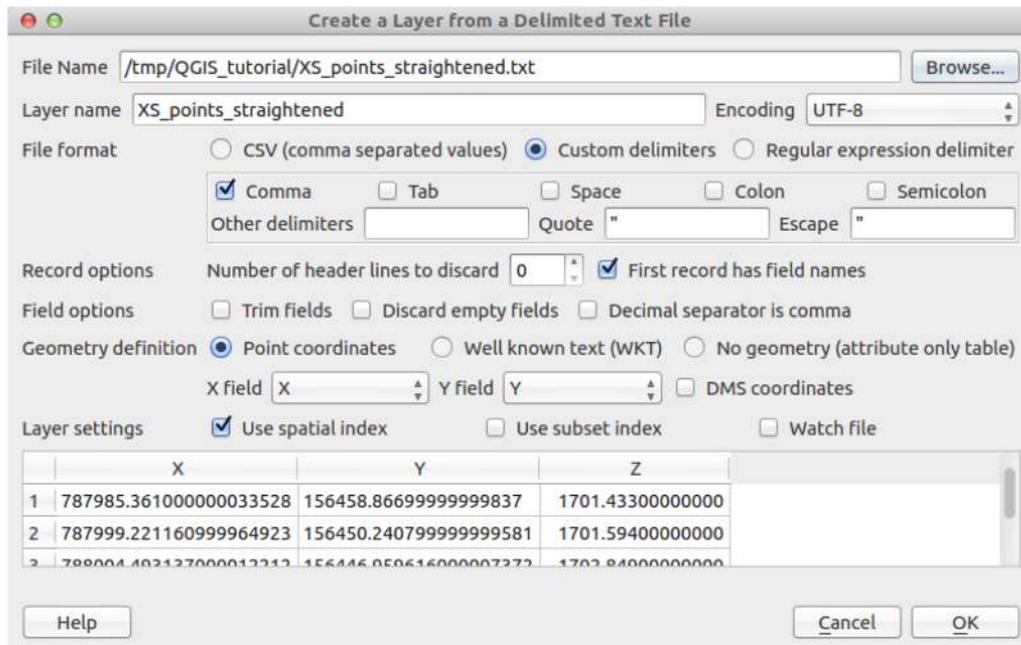


Figure 1.4 Add delimited text layer

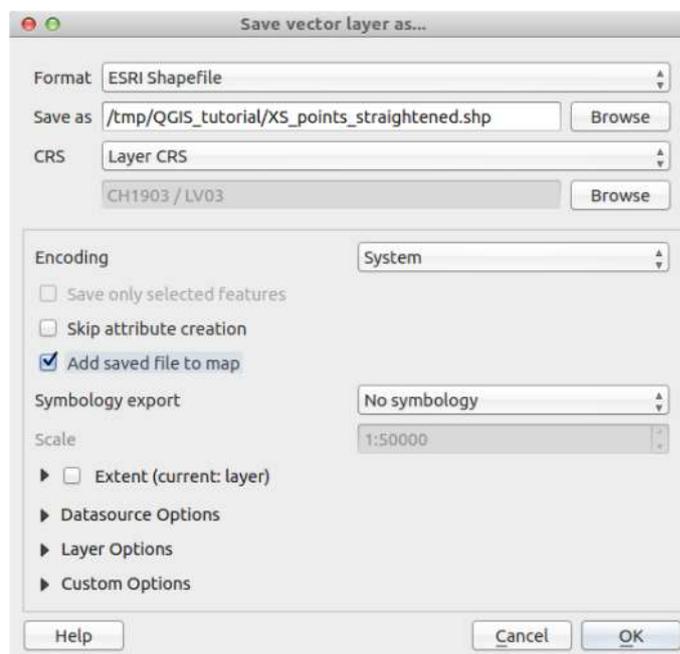


Figure 1.5 Save vector layer

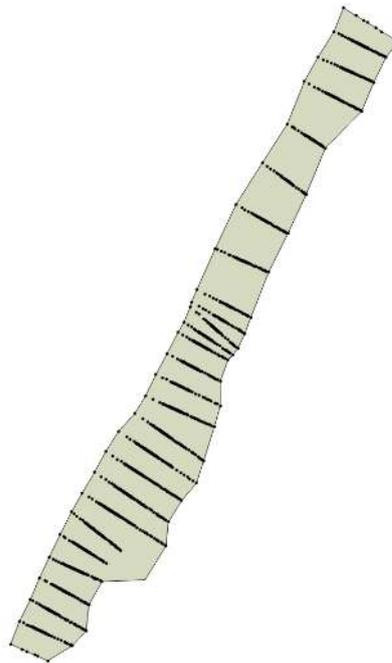


Figure 1.6 Model Boundary

- (2) Browse for the shapefile provided with this tutorial 'model_boundary.shp'.
- (3) Click *Open*.

The color schemes of newly added layers are randomly chosen by QGIS. You can change them by double-clicking on a layer and selecting *Style*. QGIS displays the loaded layers according to their order in the TOC.

- (4) Pull the newly created layer 'model_boundary' below the initial file 'XS_points_straightened'. Your QGIS canvas should now look like in Figure 1.6 .

Please note that for generating an elevation mesh, all corner vertices of the model boundary polygon must lie on elevation points. Otherwise, there might be interpolation errors in the following meshing steps. A suitable model boundary can be created by using the *Convex hull(s)* feature of QGIS. For more information, please refer to the Tips and Tricks (Section 1.6) of this Tutorial.

1.2.6 Editing the Model Boundary

All vertices of the model boundary polygon **must** lie on elevation points. Otherwise there might be interpolation errors in the following steps. Obviously this is not the case in our example (see lower right region of the model data). Therefore we have to move/add nodes of the boundary layer:

- (1) Right-click on layer 'model_boundary' in the TOC and go to *Toggle Editing*. The layer can now be edited.

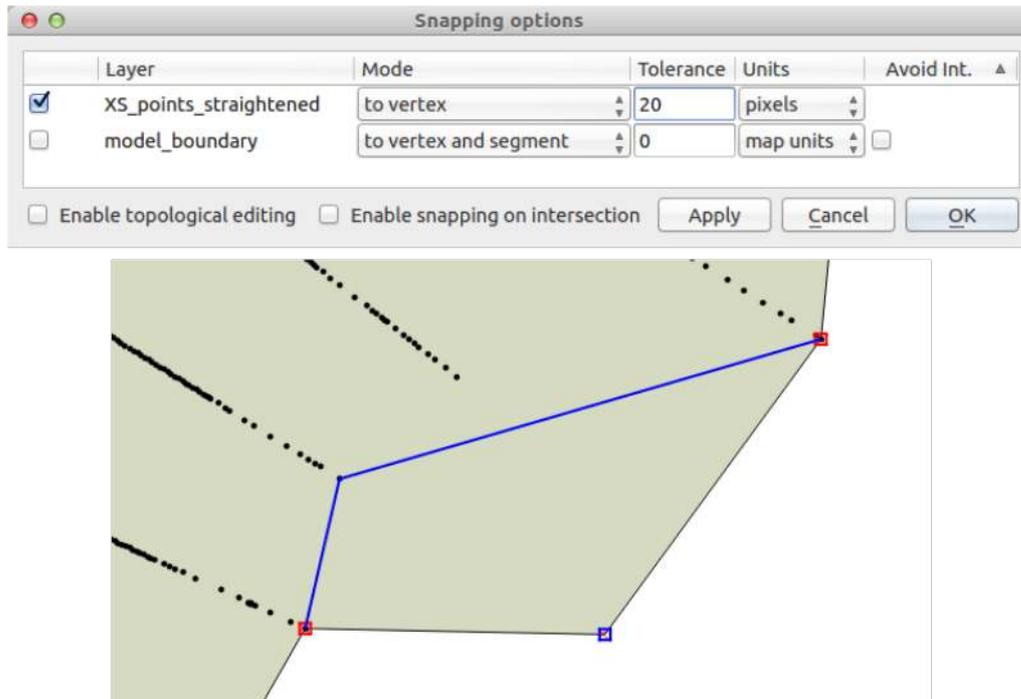


Figure 1.7 Snapping Options

- (2) Zoom into the area of interest. You should see small red 'x' for each polygon vertex.

We must ensure that the new/moved vertex will lie on an elevation point:

- (3) Go to *Settings* → *Snapping Options*. Fill in the form as shown in Figure 1.7 and click *OK*. Depending on your QGIS version and installation, the window with snapping and digitizing options might be docked. In this case, your edits are applied immediately and there are no *Apply* or *OK* - buttons available. (Be sure to only select to vertex under Mode. Otherwise, QGIS will not only snap to breakline vertices, but to all breakline segments. This would lead to incorrect intersection points and erroneous interpolation results during meshing.)
- (4) Go to *Edit* → *Node Tool*.
- (5) Click on the vertex of the boundary which is free. The polygon feature gets selected (all vertices turn to a red square).
- (6) Drag & drop the free vertex to a neighbouring elevation points (see Figure 1.7).

We need to add a vertex to the polygon to include the remaining cross section end-point into the boundary:

- (7) Double-click on the segment somewhere between the two boundary vertices where you want to have the new one. (Be careful with double-clicking: As a new node is created with each double-click, vertices that lie on top or very near to another node are easily created by mistake. Those vertices can lead to meshing errors or very fine triangulations in the following steps.)

- (8) Again, drag and drop the vertex to the wished position.
- (9) We are done with editing, go to *Layer* → *Toggle Editing*. Click *Save*.
- (10) Go to *Settings* → *Snapping Options* and deactivate snapping for 'XS_points_straightened'.

1.2.7 Loading Breakline Data

Here the breaklines represent the left and the right bank lines as well as the river bed boundary. The same rule as for the model boundary applies here: All vertices of a breakline lie on elevation points. Otherwise there might be interpolation errors in later steps. To load the data:

- (1) Go to *Layer* → *Add Vector Layer...* and browse for the shapefile provided with this tutorial 'breaklines_elev_mesh.shp'.
- (2) Click *OK*.

By default, the layer style is defined as *Single Symbol*. Instead, the layer content can also be displayed according to attributes stored in the shapefile. You can color the the breakline data according to the categories 'bank' and 'river' as follows:

- (3) Double-click on the layer 'breaklines_elev_mesh' in the TOC.
- (4) Go to *Style* (Figure 1.8).
- (5) Choose *Categorized* in the drop down menu (by default *Single Symbol*).
- (6) In the field *Column* choose 'type' as the source for the classification.
- (7) Click on *Classify*.
- (8) A table should be generated with an entry for each breakline type. Double-click on the symbol left to the value entry, and you can set a color and other style attributes of your choice.
- (9) Click *OK*.

1.2.8 Creation of the Elevation Model as TIN

Until now no functionalities of BASEmesh have been used, only basic GIS tools provided by QGIS. Based on the data that we have loaded and processed, we can now create the elevation model as a triangulated irregular network using the plugin BASEmesh. We call it *Elevation Meshing*.

- (1) Go to *Plugins* → *BASEmesh* → *Elevation meshing*, or click the respective button  in the toolbar.

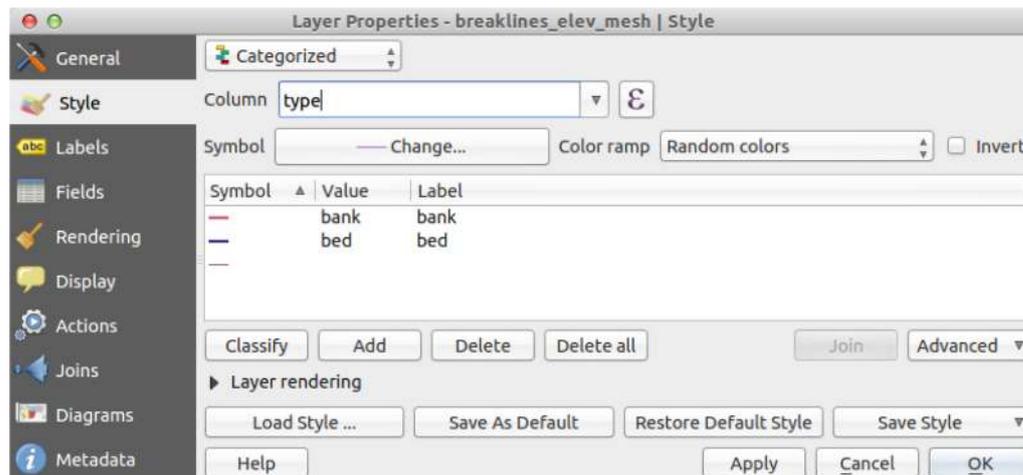


Figure 1.8 Layer properties

On the left side of the dialog (Figure 1.9) you can define the input layers. See Section 3.3.5 “Use of QGIS plugin BASEmesh for grid generation” of the BASEMENT User Manual for further explanations. On the right side status messages as well as Triangle’s output messages are displayed during meshing. Tool-specific help can be found in the *Help*-tab. To prevent the selection of wrong shape types, the available fields are populated with the corresponding data types. E.g. the model boundary has to be represented by a polygon, and therefore only layers with polygon shapes can be selected.

- (2) Select the different fields according to Figure 1.9 .
- (3) Check the optional layer *Breaklines* to include our breakline layer 'breaklines_elev_mesh'.
- (4) Choose an output filename. The default is the name of the project.
- (5) Click *Generate elevation mesh*.

Two shapefiles will be generated and loaded into QGIS canvas. One shapefile contains the nodes of the newly generated mesh, the second shapefile contains the elements which connect the nodes. Both are marked with corresponding suffixes, whereas the project title is set as default output file name. This elevation mesh is an intermediate step and will be used as basis for the further interpolation of the elevation data. Due to its low mesh quality, it should **not** be used as computational mesh for any numerical simulations!

At this point it is advisable to check whether the *Elevation meshing* worked correctly. To do so:

- (6) Right-click on the new layer 'Tutorial_Elevation_nodes.shp' in the TOC and go to *Open Attribute Table*.
- (7) Check if an elevation (column *Z*) has been assigned to every single node of the layer.
- (8) In case of a *Z*-value equals 0, you defined a vertex in your layer 'model_boundary' and/or 'breaklines_elev_mesh' which doesn't lie exactly on an elevation point.
- (9) Check the vertices at the location of the respective points missing an elevation and make sure to use the snapping option.

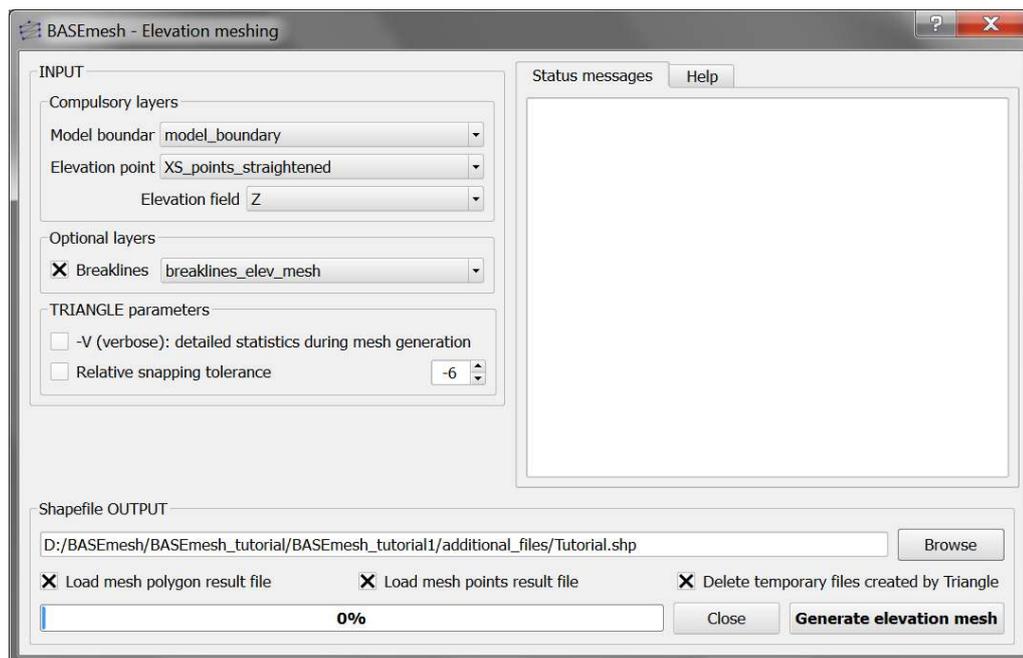


Figure 1.9 BASEmesh Elevation Meshing

(10) Repeat the *Elevation meshing* (steps 1 - 5).

In case the problem cannot be solved by redoing the snapping, try to vary the *Relative snapping Tolerance* in the *Elevation meshing* dialog (Figure 1.9). Increase the tolerance and try a value of -3 for example. Check the Tipps and Tricks (Section 1.6) for further explanations on this important parameter.

1.2.9 Adaption of the Breaklines for Quality Meshing

For most tasks, quality meshing requires the same basic breaklines as elevation meshing does. Nevertheless, some content might be added, e.g. building outlines or lines along which we wish to have special outputs from the future numerical computations. In the following, the breakline layer used before will be duplicated and a building outline will be added:

- (1) Right-click on the layer 'breaklines_elev_mesh' in the TOC and go to *Save as...*
- (2) In the next dialog choose an appropriate name and location for the new layer (e.g. 'breaklines_qual_mesh.shp'). Be sure to check *Add saved file to map* at the bottom of the dialog (compare with Section 1.2.4).
- (3) Load the shapefile 'building.shp' into QGIS (provided with this tutorial).
- (4) Select the layer 'building' in the TOC.
- (5) Go to *View → Select → Select Single Feature*.
- (6) Click on the building outline in the map (it should get colored differently).
- (7) Go to *Edit → Copy Features* (or Ctrl-C).

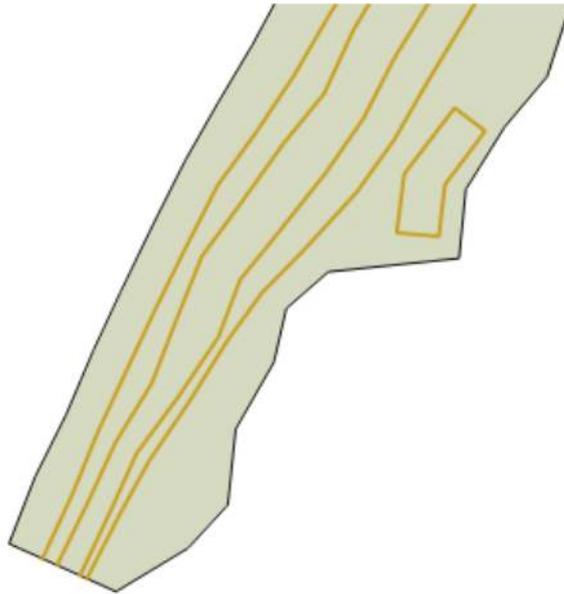


Figure 1.10 Breaklines Quality Meshing

- (8) Right-click on the layer 'breaklines_qual_mesh' in the TOC and go to *Toggle Editing*.
- (9) Go to *Edit* → *Paste Features* (or Ctrl-V).
- (10) As copying the new line feature is completed, select *Edit Toggle Editing*. QGIS asks to save the changes. Click *Save*.
- (11) Having activated only the two layers 'model_boundary' and 'breaklines_qual_mesh' in the TOC, the lower part of your model should now look like Figure 1.10. If the building outline is still highlighted, select *View Select Deselect Features from all Layers*.

1.2.10 Creation of Region Marker Points

In this new Point-Shapefile, three attributes are defined that will be used for the *Quality meshing* (Figure 1.13) afterwards: *maximum area*, *material index* and *holes*. The attributes have to be set individually for each region, which is embraced by breaklines or boundaries and are specified by placing a point into this area. Be careful, a misplaced region marker point can lead to very fine and computationally intensive meshes or wrong definition of Material Indexes respectively.

maximum area: The area attribute of the layer constraints the maximum area of the elements created during *Quality meshing*. If left blank, no maximum area constraints will be taken into account for this particular region (see Figure 1.12)

material index: This attribute determines the material index of a certain region. If left blank, the material index is by default set to 1. The material indices are used in BASEMENT to group elements into zones with similar properties, e.g. to set different friction values or soil properties in certain mesh regions. These material indices are stored in the attribute field *MATID* of the mesh elements layer during mesh generation.

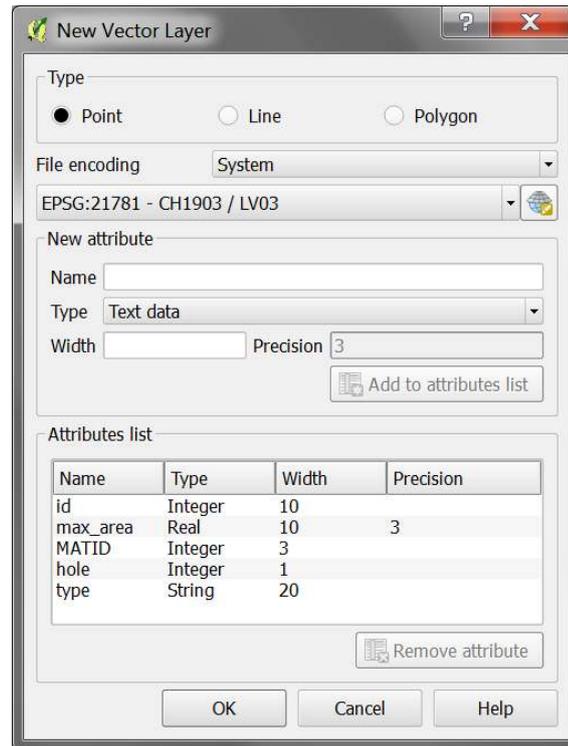


Figure 1.11 New point layer 'region_points'

holes: These points define regions that will be neglected during meshing. These areas will be cut out and therefore not be integrated in the final mesh, preventing water flow through these regions.

- (1) Create a new point layer: Go to Layer → New Shapefile Layer... (or Shift-Ctrl-N).
- (2) Fill in the form as shown in Figure 1.11. Be sure to define the correct CRS (EPSG:21781).
- (3) Add an attribute for the *maximum area* with name e.g. 'max_area' and type 'Decimal Number' (e.g. *Width*=10 and *Precision*=3).
- (4) Add a second attribute for the *material index* with name 'MATID' and type 'Whole Number' (e.g. *Width*=3).
- (5) Add a third attribute for *holes* with name 'hole' and type 'Whole Number' (e.g. *Width*=1*).
- (6) Optionally add another attribute with name 'Type' and type 'Text data' (e.g. *Width*=20) to assign a specific description for each region (e.g. 'River bed', ...)
- (7) Save the layer with name 'region_points'.
- (8) Right-click on the layer 'region_points' in the TOC and go to Toggle Editing.
- (9) Go to *Edit* → *Add Feature*
- (10) Add six features (points) inside the regions displayed in Figure 1.12

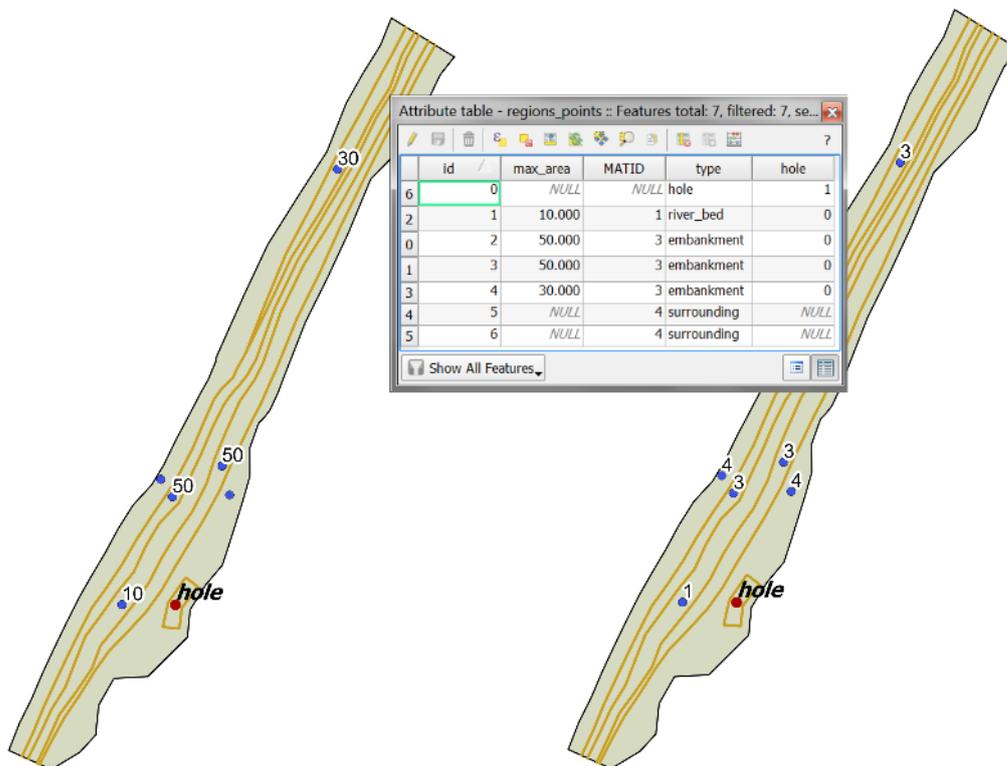


Figure 1.12 Region marker points

- (11) Click somewhere inside the particular region. Enter an arbitrary *id* and fill in the attributes like shown in Figure 1.12
- (12) Go to Layer → Toggle Editing and save the changes.

The value of the attributes 'max_area' (left) and 'MATID' (right) are used as label. In this example three different region 'Types' have been defined: 'river_bed', 'embankment' and 'surrounding'.

1.2.11 Creation of Quality Mesh

We can now use BASEmesh to generate a mesh with high quality properties by controlling cell sizes, using breaklines and holes and other parameters. Please note that the quality mesh created in this section does not incorporate any elevation data. This information will be added in the next step of the tutorial.

- (1) Go to *Plugins* → *BASEmesh* → *Quality meshing*, or click the respective button  in the toolbar.

On the left side of the dialog (Figure 1.13) you can define the input layers. See Section 3.3.5 “Use of QGIS plugin BASEmesh for grid generation” of the BASEMENT User Manual for further explanations. On the right side status messages as well as Triangle’s output messages are displayed during meshing. Tool-specific help can be found in the *Help*-tab.

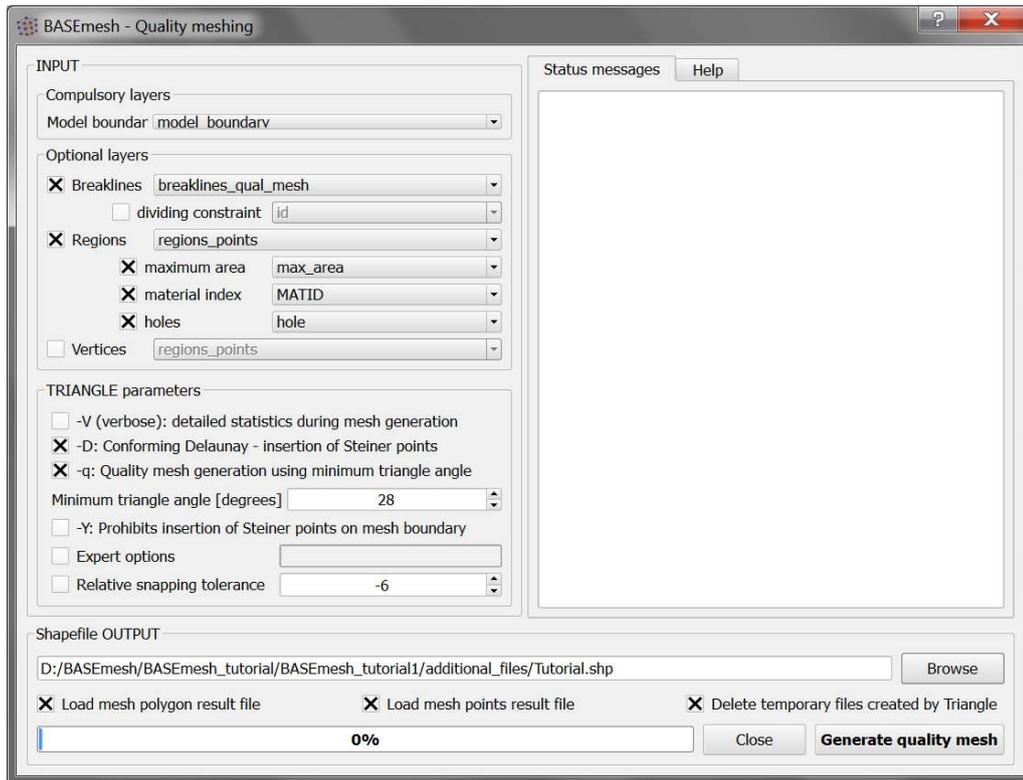


Figure 1.13 BASEmesh Quality Meshing

- (2) Select the different fields according to Figure 1.13
- (3) Check the optional layers *Breaklines*, and *Regions*. Within the Layer 'region_points' check the attributes *maximum area*, *material index* and *holes* like shown in Figure 1.13.

In this tutorial a *minimum triangle angle* of 28 degrees was chosen. This means that no elements with angles smaller than 28 degrees are created. Therefore a smaller value leads to a smaller number of elements in the mesh, while a larger value leads to a higher number of elements but less distorted triangles.

- (4) Finally, click on *Generate quality mesh*.

Similarly to elevation meshing, two shapefiles containing the nodes and the elements of the mesh are generated. Both are marked with corresponding suffixes, whereas the project title is set as default output file name. The style of the layer 'Tutorial_Quality_elements.shp' will be *Categorized* according to the *Column* 'MATID'.

If the mesh contains regions with almost infinitely dense triangulation, again check the snapping of your model boundary and the breaklines used for quality meshing or increase the *Relative snapping tolerance* (see Tipps and Tricks Section 1.6).

1.2.12 Interpolating elevation data from elevation mesh (TIN)

The quality mesh generated in the previous step still lacks any elevation data. Before it can be exported and used for simulations, elevation data has to be interpolated on the

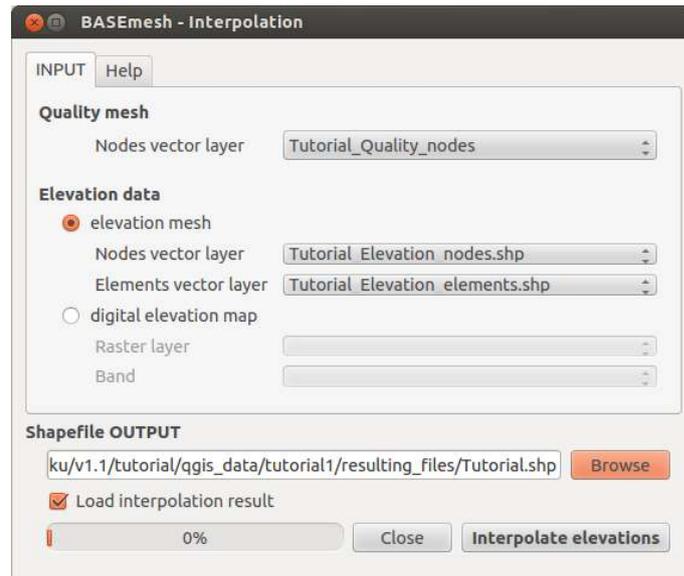


Figure 1.14 BASEmesh Interpolation

nodes of the quality mesh. For this purpose, the elevation mesh created in a previous step will be used. Alternatively, one could also use raster data as source for the elevation model, as it is illustrated in Tutorial 2 (Section 1.3).

- (1) Go to *Plugins* → *BASEmesh* → *Interpolation*, or click the respective button  in the toolbar. A new dialog will open (Figure 1.14).
- (2) As input for *Quality mesh* choose the nodes vector layer of the quality mesh ('Tutorial_Quality_nodes.shp') without elevation data.
- (3) Check the radio button *elevation mesh*.
- (4) As input for the elevation data choose the corresponding layers of the elevation model ('Tutorial_Elevation_nodes.shp', 'Tutorial_Elevation_elements.shp').
- (5) Click on *Browse* and select 'Tutorial' as shapefile output name.
- (6) Click on *Interpolate elevations*. On Windows systems, the interpolation dialog might be marked as Not Responding. On Linux systems, the interpolation dialog might be darkened. This does not necessarily mean that the process stopped or had errors: The system is simply busy.

In case of a large number of elements ($>10^4$), the interpolation might take up several minutes. In general, the interpolation from raster data is faster than the interpolation from the elevation mesh. After successful interpolation, a new nodes shapefile is generated with the suffix '_interpolated_nodes_elevMesh'. For every node of the quality mesh, the elevation has been interpolated based on the elevation mesh given. Therefore, the locations of the newly created interpolated nodes are identical to the nodes of the quality mesh, but contain the interpolated elevation data as attribute. The result can be checked for plausibility by labeling the elevation nodes layer and the interpolation result layer with their elevation attributes and comparing the values.

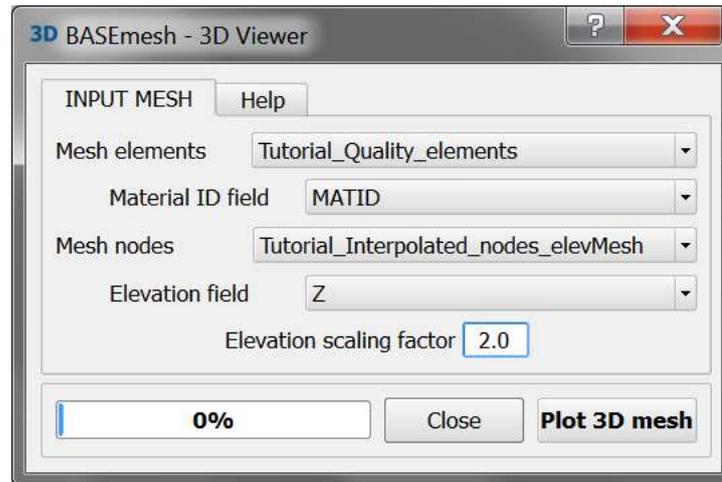


Figure 1.15 BASEmesh 3D view

1.2.13 3D view of the mesh

During the interpolation step, elevation data has been added to the mesh. With this additional information we are now able to generate a 3D view of our mesh. This can be useful to check for plausibility.

- (1) Go to *Plugins* → *BASEmesh* → *3D view*, or click the respective button **3D** in the toolbar. A new dialog will open (Figure 1.15).
- (2) Fill in the dialog as shown in Figure 1.15. The *Elevation scaling factor* is optional and might be useful since the horizontal extend of the computational mesh is normally significantly greater than the amount of vertical change within the domain.
- (3) Click *Plot 3D mesh* and a new window opens up.
- (4) The mesh elements are colored according to the different material indexes.

1.2.14 Export of mesh to 2dm

To this point, all steps for the creation of a computational grid shown in this tutorial resulted in the generation of shapefiles. For the future use in the BASEMENT simulation software, the computational grid has to be provided in *.2dm* format. Before this conversion, it might be necessary to set zones of different material indices in the mesh or to modify parts of the mesh as described in Section 1.3.

- (1) Go to *Plugins* → *BASEmesh* → *Export mesh*, or click the respective button  in the toolbar.
- (2) Select the radio button *2D mesh export*.
- (3) As input for the *Elements* select the element layer of the quality mesh 'Tutorial_Quality_elements'.
- (4) Choose the attribute field with the material indices (*MATID*).

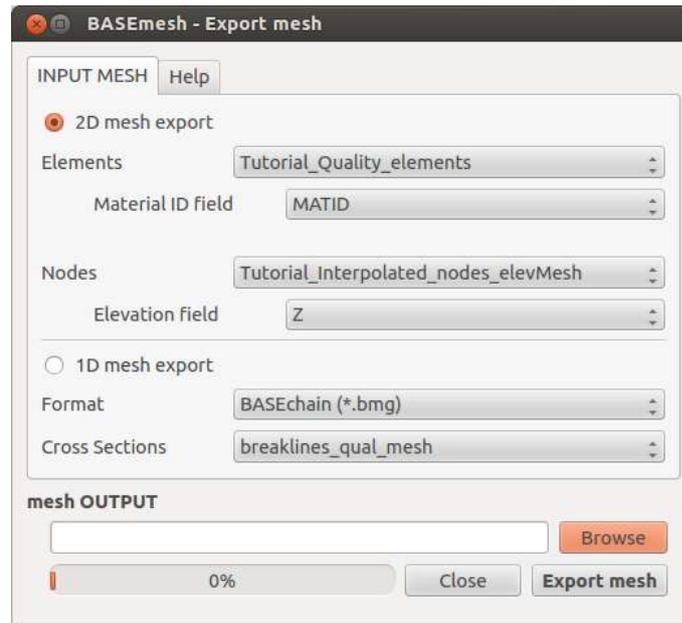


Figure 1.16 BASEmesh Export Mesh

- (5) As input for the *Nodes* select the node layer with the interpolated elevation data 'Tutorial_interpolated_nodes_elevMesh.shp'.
- (6) Choose the attribute field with the elevation data (*Z*).
- (7) Click on Browse and select an output *.2dm* mesh name.
- (8) Click on *Export mesh*.

Finally, the *.2dm* mesh file is written. You can use this mesh file to set-up 2D simulations with BASEMENT.

1.3 Tutorial 2: Import/Modify an existing Mesh and use Raster Data as Elevation Model

This tutorial illustrates the workflow for importing a mesh from an existing *.2dm* file into QGIS and the available features for manual modifications of this mesh. Using BASEmesh and QGIS, mesh elements can be edited (e.g. moved, added, deleted) or the material indices within zones of the mesh can be edited. Furthermore, the use of raster data as elevation model is illustrated. This corresponds to a scenario where a digital elevation model (DEM) is the source for the interpolation of elevation data.

Input Data The data needed to complete this tutorial comes as ZIP - file and needs to be extracted to a location of your choice. The data have to be loaded into the QGIS - project before executing the different tools. It is not possible to select files directly on the hard drive by browsing. Furthermore, those files must be activated in the QGIS table of contents (TOC this abbreviation will mostly be used in this tutorial) on the left side of the screen. To prevent the selection of wrong shapetypes, the available fields are populated with the corresponding data type.

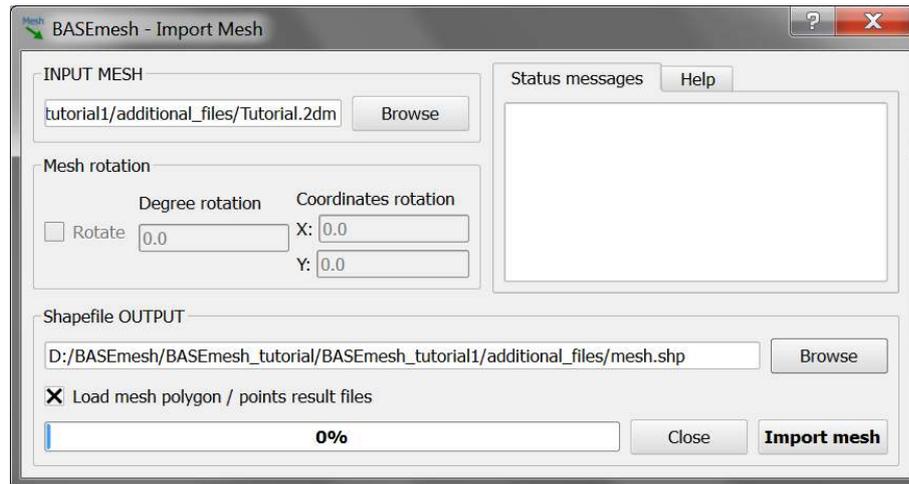


Figure 1.17 BASEmesh Import Mesh

In this tutorial, only a *.2dm* mesh file and a shape file containing the landuse properties are needed as input files. It is recommended to create at first a new project and to set its project settings as exemplified in Section 1.2. In case of difficulties, the results for each tutorial step are additionally provided in the subfolder called 'additional_files'.

1.3.1 Importing a *.2dm* mesh file

- (1) Go to *Plugins* → *BASEmesh* → *Import mesh*, or click the respective button  in the toolbar.
- (2) In the opening dialog *Browse* for the mesh file 'MeshTutorial.2dm' in your file system
- (3) Use the *Browse* Button at the bottom of the dialog to select a name for the mesh data. Select a name of your choice, e.g. 'MeshTutorial.shp'.
- (4) Click on *Import mesh*.

After successful mesh import you will see two new layers in the TOC, representing the mesh:

- A polygon shapefile with the suffix '_elements' containing all mesh elements and the mesh connectivity.
- A point shapefile with the suffix '_nodes', which contains all mesh nodes with x - y - z data.

In the following steps, we will modify these shape layers.

1.3.2 Modifying the material indices of elements

The material indices are used in BASEMENT to group elements into zones with similar properties, e.g. to set different friction values in certain mesh regions. These material indices are stored in the attribute field *MATID* of the elements layer.

Selection of the elements

In QGIS shape features can be selected by different approaches:

- Manual selection using options like *Select single features*, *Select features by rectangle*, *Select features by polygons*, etc. For small meshes, this might be a suitable option, but is time consuming and error prone for large meshes.
- Selection of features according to their location using spatial queries.

The latter and more general approach is illustrated here to select all elements lying within the river bed. For this purpose an additional plugin needs to be installed first:

- (1) Load the QGIS plugin manager by choosing *Manage and Install Plugins...* in the menu *Plugins* of the QGIS main toolbar.
- (2) In the tab *All type* 'Select Within' into the search field and click *Install plugin*

Once the plugin is installed successfully, an additional icon  should appear in your Layers toolbar

- (3) Go to *Layer* → *Add Vector Layer* (or Shift+Ctrl+V) and load the shapefile 'landuse_polygon.shp'.
- (4) Open the attribute table of the layer 'landuse_polygon.shp' by right click on the layer in the TOC. The attribute table dialog will open (see Figure 1.18).
- (5) Select the feature of *type* 'river_bed' by clicking the corresponding number in the first column of the attribute table dialog (numbered automatically starting with '0').

QGIS highlights the selected features in a color depending on your system settings (default: yellow).

- (6) Click on the *Select Within* icon  in the toolbar. A new dialog will open (see Figure 1.18).
- (7) For *Select features from:* chose 'MeshTutorial_elements.shp'.
- (8) Check *Where the: Centroid*
- (9) For *Is within:* chose 'landuse_polygon.shp'.
- (10) Check *Using selected features* and *Creating new selection* below and click *OK*

When the spatial query has finished (this may take some time), all mesh elements within the extent of the river bed are selected and highlighted yellow (see Figure 1.18).

Changing the material indices

As next step, the material indices of the selected elements will be edited.

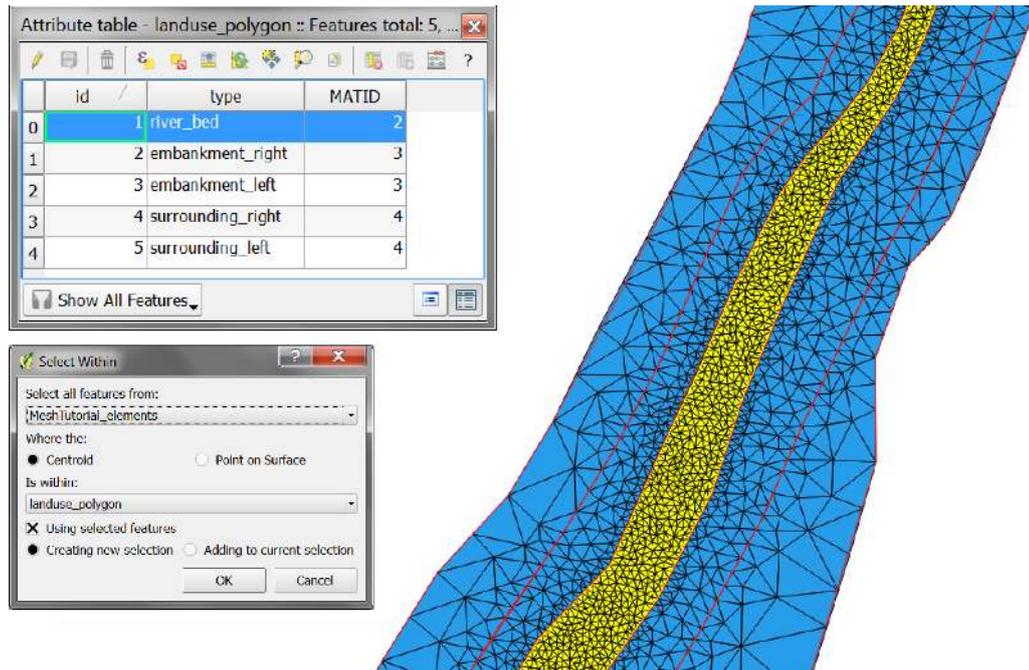


Figure 1.18 Selected mesh elements according to their location

- (11) Open the attribute table of the layer 'MeshTutorial_elements.shp' by right click on the layer in the TOC. The attribute table dialog will open.
- (12) On the bottom left of the dialog, select *Show Selected Features*. Now, only the selected features of the river bed are shown (see Figure 1.19 left).
- (13) Click on the upper left icon *Toggle editing mode* (or Ctrl+E).
- (14) Click on the upper right icon *Open field calculator* (or Ctrl+I). A new dialog will open (see Figure 1.19 right).
- (15) Select the checkbox *Update existing field*.
- (16) Chose the attribute field *MATID* in the combobox below.
- (17) Enter '2' into the *Expression* box at the bottom of the dialog. Click *OK* to change the material indices of all selected elements from 1 (default value) to the value 2.
- (18) Uncheck *Toggle editing mode* (or Ctrl+E) and save the made changes.

If the selected elements are still highlighted, select *View* → *Select* → *Deselect Features from all Layers*. So far only the *MATID* of the 'river_bed' mesh elements has been modified. Repeat the steps demonstrated before in order to modified the remaining mesh elements of the 'embankment' and 'surrounding' according to the proposed values listed in the attribute table of the layer 'landuse_polygon'

To verify that the material indices within the different regions were successfully altered, you may want to color the elements layer *Categorized* according to the attribute value of the *Column MATID* (see Section 1.2.7). If everything went well, your QGIS canvas should now look like in Figure 1.20.

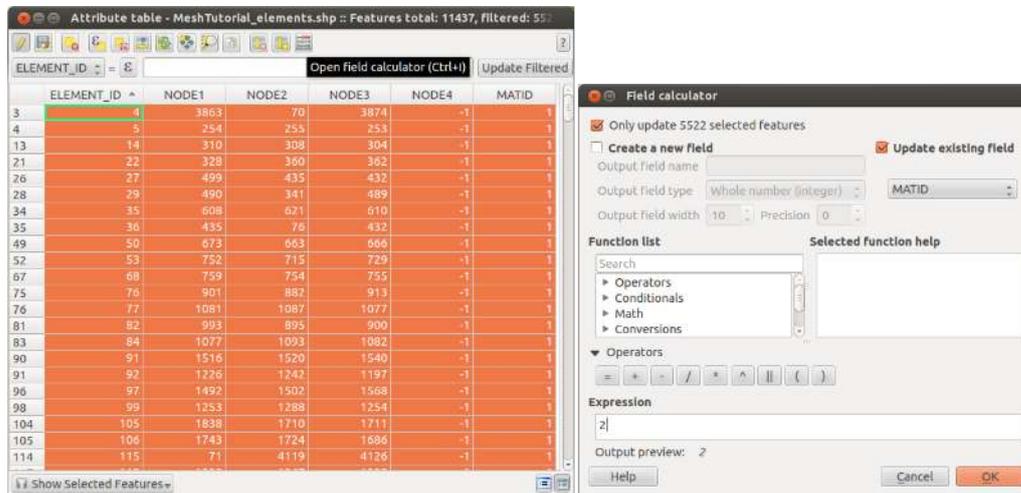


Figure 1.19 Editing of selected mesh elements

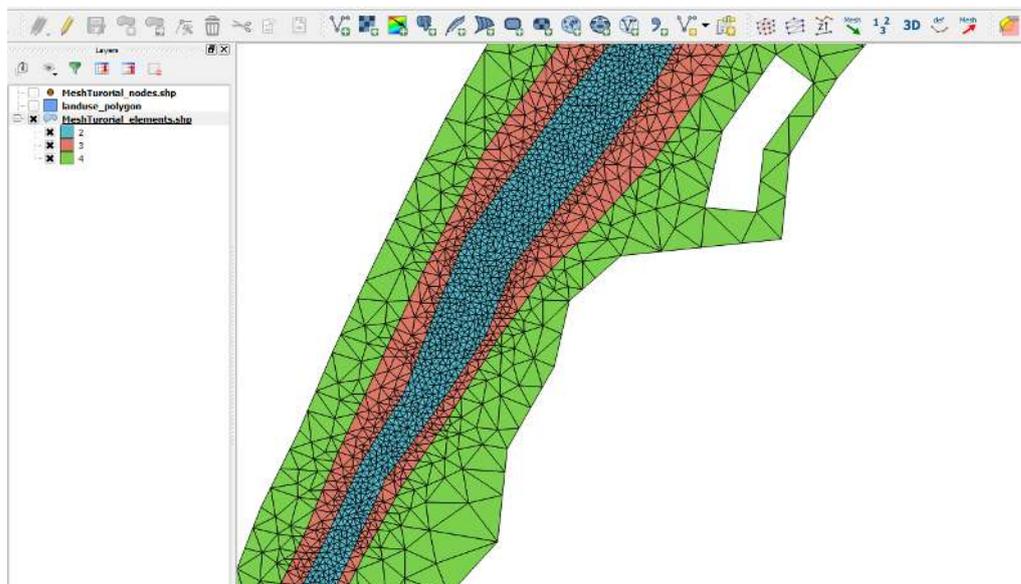


Figure 1.20 Mesh material indices: river bed=2, embankment=3, surrounding=4

1.3.3 Manual editing of mesh elements

In this section, we want to modify the mesh by deleting two triangles and merging them into a new triangle. Before we start editing the mesh, let's make copies of the layers (this is optional, just for safety if something goes wrong). As it will not be needed in this step, you can now delete the 'landuse_polygon' layer by right click in the TOC and selecting *Remove*.

- (1) Right click on each layer and select *Save As...* to save it to file. Add the suffix '_mod' (modified) at the end of the filenames ('_elements_mod.shp', '_nodes_mod.shp'). In the *Save As* - dialog, be sure to check the box *Add saved file to map*.

Next, label all nodes and elements of these layers with their ID:

- (2) Label the '*elements_mod.shp*' layer with the *ELEMENTID* attribute.

Finally, let's start modifying the mesh:

- (3) Zoom to elements #9357, #9350 at the top of the canvas (see Figure 1.21).
- (4) Activate the '_nodes_mod.shp' layer. Go to *Layer Toggle editing*.
- (5) Select the node #4736 (use the *Select single feature* icon), so that it is highlighted. Go to *Edit Delete selected* to remove it from the layer or press 'delete' on your keyboard.
- (6) Go to *Layer Toggle editing* and save the changes.
- (7) Activate the '_elements_mod.shp' layer. Go to *Layer Toggle editing*.
- (8) Select the elements #9357 and #9350, so that they are highlighted. Go to *Edit Delete selected* to remove them from the layer.
- (9) Go to *Settings Snapping options*. Check the checkbox '_nodes_mod.shp' and set the tolerance to 10 pixels. Go to *Settings Snapping options*. Check the checkbox '_nodes_mod.shp' and set the tolerance to 10 pixels.
- (10) If not done yet, activate the '_elements_mod.shp' layer. Go to *Edit Add Feature*. Now click on the nodes #4741, #4737 and #4738, the snapping ensures that the correct coordinates are selected. After a right click, a new dialog opens to fill out the attributes of the new cell. Leave all attributes empty except of the *MATID* attribute, which is set to 3 according to the position of the modified elements within the left embankment.
- (11) Go to *Layer → Toggle editing* and save the changes.

You have now merged two small triangles into a larger triangle. Please note that the mesh connectivity is no longer valid and needs to be updated! In an analogous way it is also possible to manually generate quadrilateral cells instead of triangles. Similar, you may also add new nodes to the mesh. When adding a new node, leave all attributes empty with the exception of the *Z* attribute, which must be specified.

Thumb rules for mesh editing:

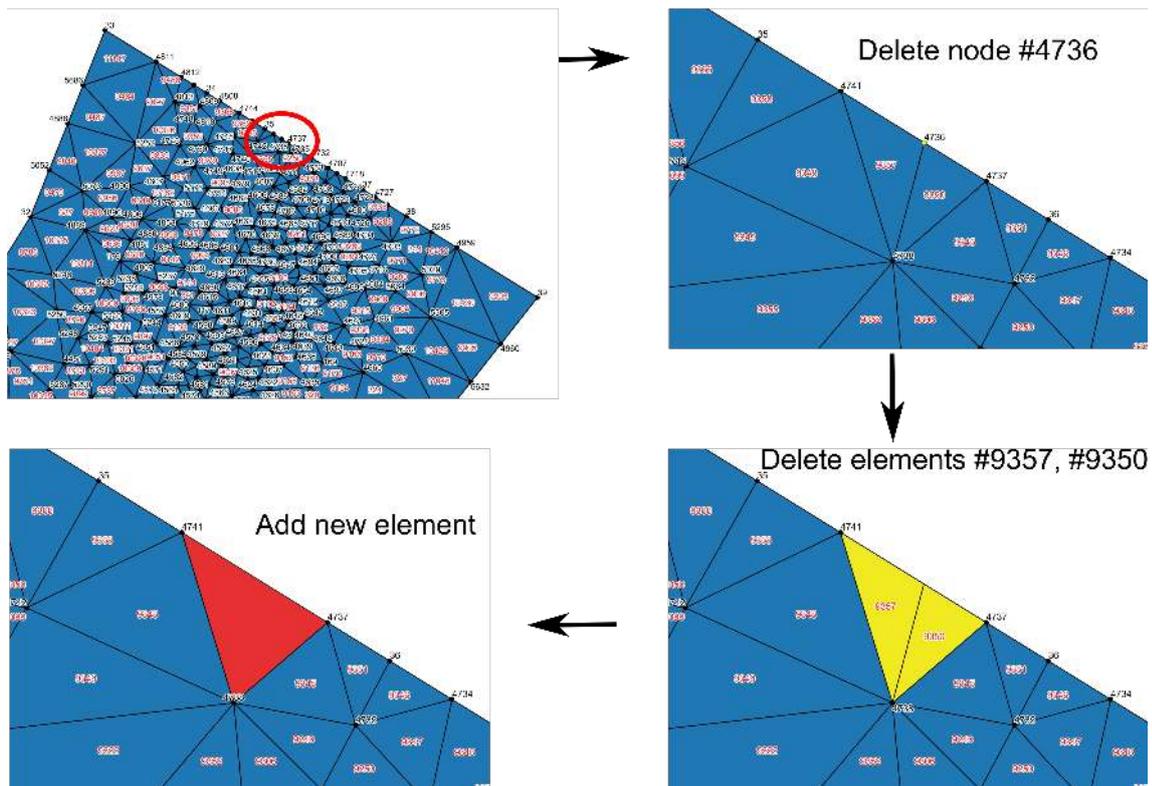


Figure 1.21 Modify mesh elements

- Element vertices and nodes must be at exactly the same locations (snapping!)
- Never generate elements without corresponding nodes and vice versa.
- Never create multiple nodes or elements at the same location.

1.3.4 Renumbering the mesh

After mesh modifications, the connectivity of the elements and nodes becomes invalid. Therefore it is necessary to renumber the mesh and update the mesh connectivity.

- (1) Go to *Plugins* → *BASEmesh* → *Renumber mesh*, or click the respective button ¹/₃ in the toolbar.
- (2) As input for the *Mesh elements layer* select 'MeshTutorial_elements_mod.shp'.
- (3) As *Material ID field* chose *MATID*.
- (4) As input for the *Mesh nodes layer* select 'MeshTutorial_nodes_mod.shp'.
- (5) As *Elevation field* choose *Z*.
- (6) Click on *Browse* and set the output file name to 'MeshTutorial'.
- (7) Click on *Renumber mesh*.

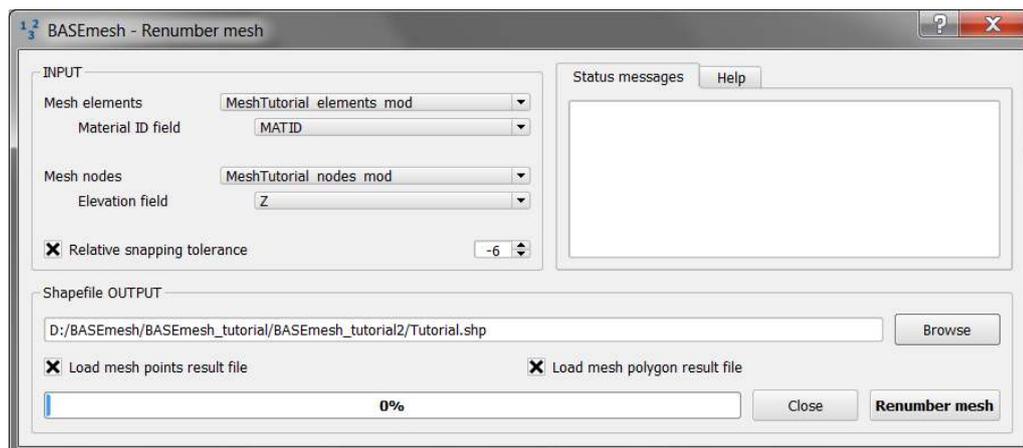


Figure 1.22 Renumbering the mesh

After renumbering, two shapefiles are generated with the suffixes '*_renumbered_nodes*' and '*_renumbered_elements*'. This mesh is in a valid state again! Please note that all node IDs and element IDs were renumbered. Hence, it may become necessary to update all parts of your BASEMENT command file referring to these numbers (see for example Section 1.4).

1.3.5 Interpolating elevations from raster data

As final step of this tutorial, it is illustrated how to interpolate elevations from a raster data set (DEM) on the mesh node layer.

- (1) Go to *Layer Add Raster Layer* (or Shift+Ctrl+R). Choose the raster data file '*raster_elevations.asc*'.
- (2) Go to *Plugins* → *BASEmesh* → *Interpolation*, or click the respective button  in the toolbar.
- (3) As input for the quality mesh chose the nodes layer '*MeshTutorial_Renumbered_nodes.shp*'.
- (4) Check the radio button *digital elevation map*.
- (5) As input for the *Raster layer* chose the raster data '*raster_elevations*' and set the *Band* to 1.
- (6) Click on *Browse* and set the output file name to '*MeshTutorial*'.
- (7) Click on *Interpolate elevations*.

1.4 Tutorial 3: Using dividing constraints along boundary cross sections and setting up a BASEMENT simulation with a mesh from BASEmesh

This tutorial illustrates how to use the generated mesh from the BASEmesh plugin to set-up a numerical model with BASEMENT. In advance it is demonstrated, how to use

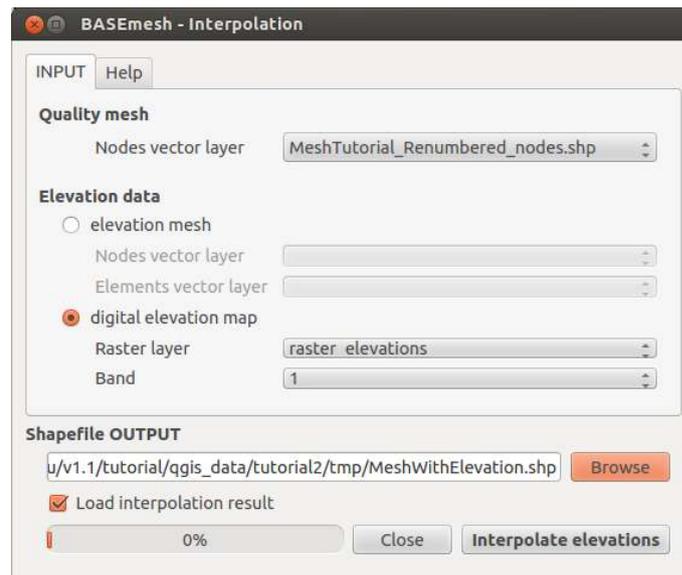


Figure 1.23 Interpolation from raster data

dividing constraints along cross sections in order to determine the number of mesh elements along a certain breakline during quality meshing. This is especially relevant for the use of Inner Boundary Conditions in BASEMENT, where an equal number of elements is required at the upstream and downstream edge of the Inner Boundary. The mesh generation itself is covered in other tutorials and for the general set-up of BASEMENT command files, please take a look at the documents provided on the homepage of [BASEMENT](#). BASEmesh / QGIS can be used to generate or determine the following BASEMENT input data:

- The *.2dm* - mesh file,
- Node numbers at the boundary cross sections (optional),
- Element numbers at external sources (optional), and
- Material indices.

Input Data The data needed to complete this tutorial comes as ZIP - file and needs to be extracted to a location of your choice. You also need to have the software BASEMENT installed on your system. Example files for a BASEMENT simulation are given in the folder 'basement_model/input':

- Input hydrograph: 'hydrograph.txt',
- BASEMENT Command file: 'run.bmc'.

Additionally a number of shapefile is provided to generate a new mesh:

- Shapefiles: 'breaklines_qual_mesh.shp', 'model_boundary.shp', 'region_points.shp', 'Elevation_elements.shp', 'Elevation_nodes.shp'

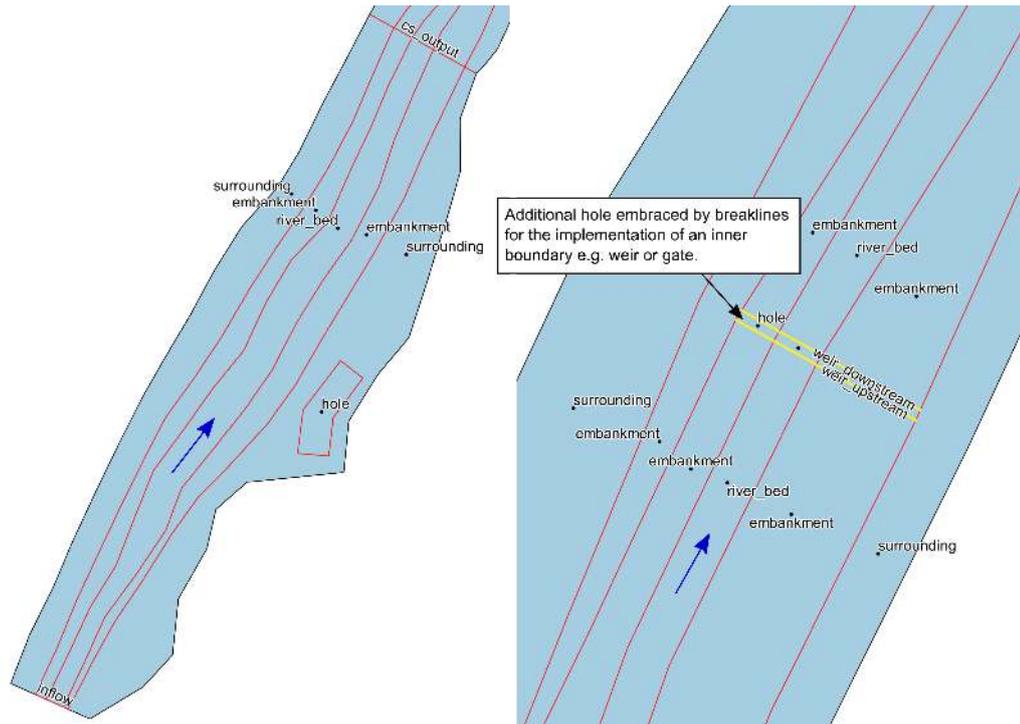


Figure 1.24 Additional breaklines inflow boundary, cross section output and inner boundary

1.4.1 Using dividing constraints for Quality meshing

The general procedure of mesh generation is demonstrated in Tutorial 1 (Section 1.2). Here an additional feature shall be introduced, which can be used especially in the context of Inner Boundary conditions. Therefore we want to use the provided input data to generate an additional mesh:

- (1) Make sure to use the same Project Settings as defined in Tutorial 1 (see Section 2.2.4.1)
- (2) Import the provided shape files 'breaklines_qual_mesh.shp', 'model_boundary.shp' and 'region_points.shp'.

Your QGIS canvas should now look like in Figure 1.24 .

The breaklines with 'stringdef' inflow, outflow, weir_upstream and weir_downstream contain values for the attribute 'nCells'. With these values one can determine the number of elements along a certain breakline that will be generated during *Quality Meshing*.

- (3) Go to *Plugins* → *BASEmesh* → *Quality meshing*, or click the respective button  in the toolbar.
- (4) Check the *Optional layers Breaklines* and *Regions*.
- (5) Select the different fields according to Figure 1.25
- (6) In addition to the Quality Meshing demonstrated in Tutorial 1 (Section 1.2.11), check the *dividing constraint* and select the attribute 'nCells' to determine the number of elements.

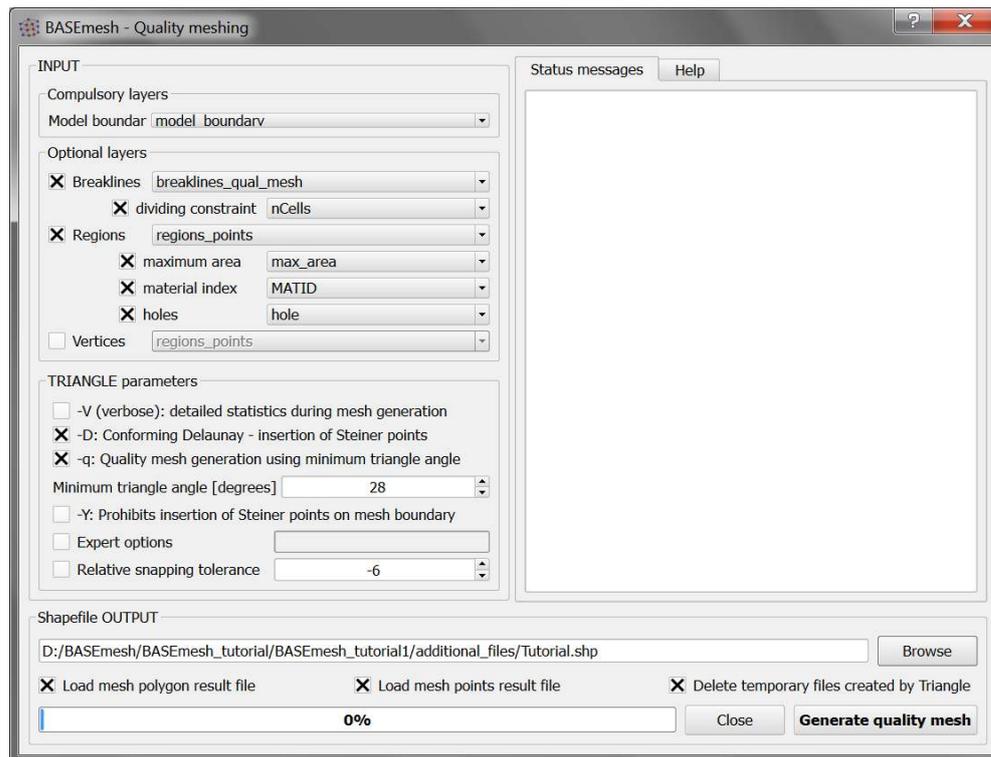


Figure 1.25 BASEmesh Quality Meshing using dividing constraints

(7) Finally click on *Generate quality mesh*

If everything went well, your QGIS canvas should now look like shown in Figure 1.26. The coloring of the mesh elements represents the different material indices (MATID) specified for the different regions. The hole in the mesh is used to implement an inner weir boundary later. As you can see from the figure, the number of Cells (20) along the upstream (weir_upstream) and downstream (weir_downstream) breakline are equal.

So far the generated Quality Mesh does not contain any elevation data. To interpolate the elevation data to the nodes of the quality mesh, import the two provided shapefiles 'Tutorial_Elevation_nodes.shp' and 'Tutorial_Elevation_elements.shp'. Follow the previously described steps in Section 1.2.12 and Section 1.2.14 for the interpolation and mesh export.

1.4.2 Extraction of Stringdef information from the mesh

In the BASEMENT command file, the location of the boundary conditions are specified by a list of node numbers. Therefore, we have to extract this information from the mesh using QGIS and to insert these node numbers into the command file. To this end, BASEmesh provides the tool *Stringdef*. You can either use the *Stringdef* tool during the mesh generation and extract the information from the quality mesh nodes or import an existing computational mesh (*.2dm) to extract the information from the nodes layer, which is generated when importing the mesh. The exemplary usage of this tool is shown in the following steps:

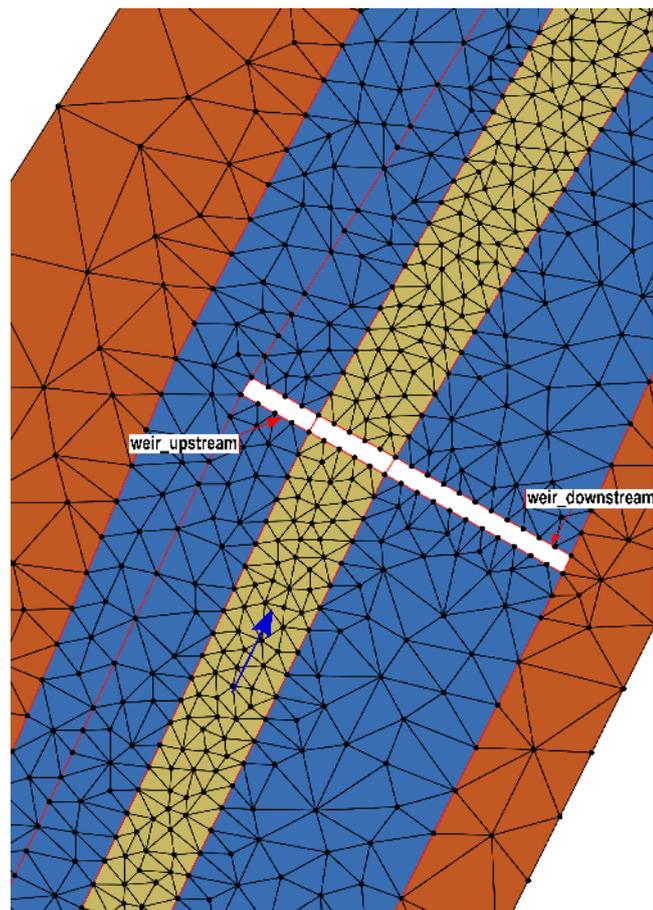


Figure 1.26 Quality Mesh with dividing constraints along breakline

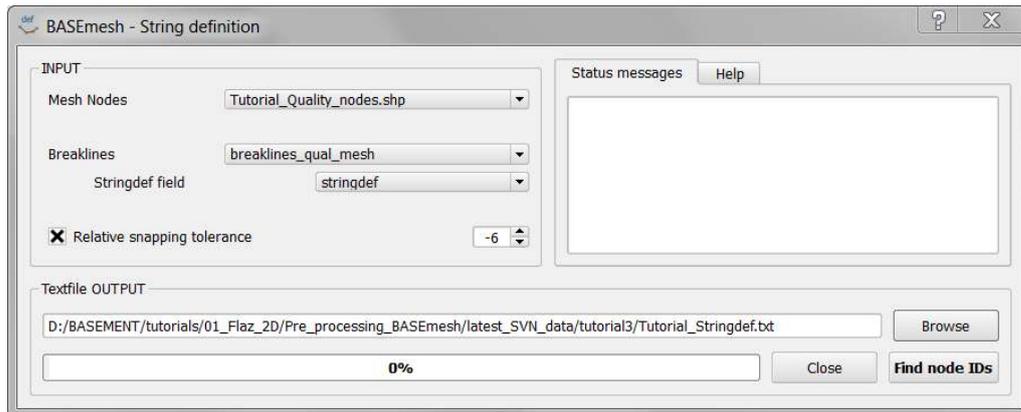


Figure 1.27 BASEmesh extract Stringdef

- (1) Go to *Plugins* → *BASEmesh* → *Stringdef*, or click the respective button  in the toolbar.
- (2) Fill in the dialog as shown in Figure 1.27
- (3) Within the layer 'breaklines_qual_mesh' only features that have a string defined in the field 'stringdef' will be handled as such, here we have 'inflow', 'cs_output', 'outflow' and the Inner Boundaries 'weir_upstream' and 'weir_downstream'.
- (4) Click on *Find node IDs*. **Note:** only nodes that are located exactly on a line will be captured!
- (5) You will find in the text file 'stringdef.txt' five blocks with the stringdef definition. These will be copied into the BASEMENT command file later.

1.4.3 Set-up BASEMENT command file: Add mesh

- (1) If you have BASEMENT installed, then start the program.
- (2) Go to *File Open Command* (or Ctrl+O) and open the 'run.bmc' command file. The *Input Structure* in the left window shows red colored entries, since there are still some missing input data.
- (3) Go to *BASEMENT* → *DOMAIN* → *BASEPLANE_2D GEOMETRY* in the *Input Structure*.
- (4) Use *Add Tag* to add the *file* tag.
- (5) Select 'MeshTutorial.2dm' as *file* input. Alternatively use your own mesh created in the previous steps.
- (6) Go to *File Save* (or Ctrl+S).

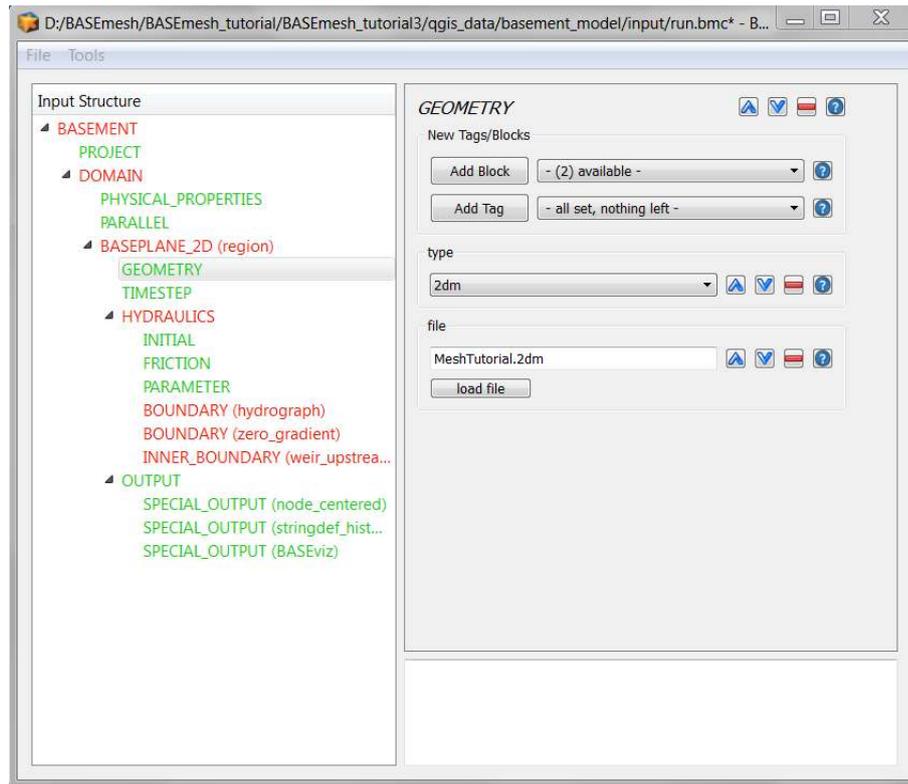


Figure 1.28 BASEMENT command, GEOMETRY

1.4.4 Set-up BASEMENT command file: Add stringdefs

Now we will now put the definitions extracted in Section 1.4.2 into the BASEMENT command file:

- (1) Start BASEMENT and open the 'run.bmc' command file.
- (2) Press *Ctrl+E* (Edit raw). The raw input file will appear.
- (3) Now paste the copied stringdef definitions into the *GEOMETRY* block.
- (4) Click *Apply*. All errors should be gone now.
- (5) Go to *File Save* (or *Ctrl+S*).
- (6) Alternatively you can directly edit the 'run.bmc' command file with a text editor of your choice e.g. Notepad++

1.4.5 Set-up BASEMENT command file: Add friction values

Friction values often differ within regions of the computational domain. For example, the friction is often smaller in the river bed than in the floodplains, due to vegetation influences. In BASEMENT, different friction values can be set for different zones, specified by the material indices of the mesh elements. Here, these indices are:

- 1=river bed

- 3=embankments
- 4=surrounding/floodplain

In the following steps, you can set the friction values for these zones:

- (1) Start BASEMENT and open the 'run.bmc' command file.
- (2) Go to *BASEMENT* → *DOMAIN* → *BASEPLANE_2D HYDRAULICS FRICTION* in the *Input Structure*.
- (3) Use *Add Tag* to add the *input_type* tag and set it to *index_table*.
- (4) Use *Add Tag* to add the *index* tag. Enter '1 3 4', corresponding to the material indices in this tutorial.
- (5) Use *Add Tag* to add the *friction* tag. Enter '33 30 28' as Strickler friction coefficients. Using these settings, the river bed is computed with a friction coefficient of 33, the embankments with a value of 30 and the rest of the domain with 28.
- (6) Go to *File Save* (or Ctrl+S).

Ready to go! Now, you are finished and you can finally start the BASEMENT simulation. To separate the original input files and the generated result files, it is recommended to first copy all input files in the run folder and to execute the model there. Enjoy!

1.4.6 Data Visualization

For a brief information about the available visualization methods of BASEMENT results please take a look in the BASEmesh tutorial (Section 1.2). The Visualization of 2D results of a BASEMENT simulation is covered in detail in the Post-Processing (Section 2.1). In general, data visualization is possible using:

- The QGIS plugin Crayfish (see Post-Processing tutorial)
- The software ParaView (see Post-Processing tutorial)
- Loading shape output from BASEMENT or
- Rasterizing shape/ASCII output from BASEMENT.

Each of these methods has its advantages and disadvantages, whereas we recommend the Crayfish plugin for quick visualizations of time-dependent data.

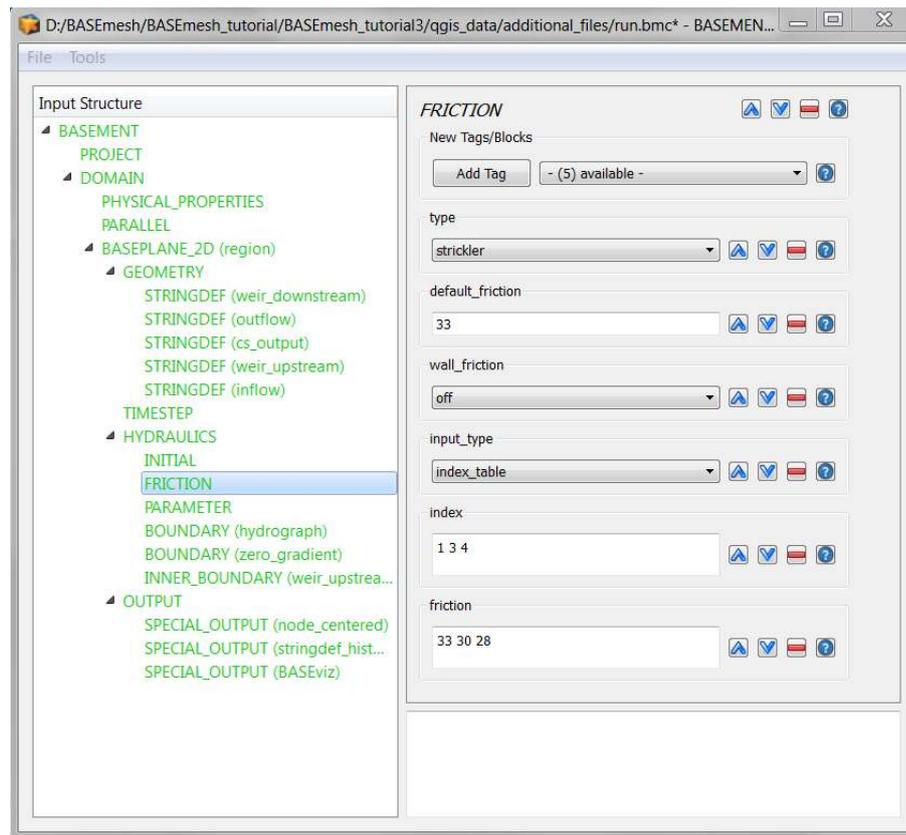


Figure 1.29 BASEMENT command, FRICITION

1.5 Tutorial 4: Create 1D BASEchain mesh from a DTM using HEC-RAS/ HEC-GeoRAS

This tutorial briefly sketches the workflow how to create 1D mesh files using the HEC-RAS software as pre-processor. The tasks are not outlined in detail and more information and tutorials about HEC-RAS and HEC-GeoRAS for pre-processing can be found in the web. Please note, that most tasks are performed within the ArcMap-software, whereas BASEmesh is used only for the conversion of the final 1D HEC-RAS geometry file (.*g01*) to the 1D BASEmesh geometry file format (.*bmg*).

This workflow, and alternativ workflows using HEC-RAS as pre-processor, are illustrated in Figure 1.30.

To follow all steps of this tutorial you need various software installed on your system (Windows only):

- QGIS software,
- BASEmesh plugin,
- [HEC-RAS](#) software, and
- [HEC-GeoRAS](#) plugin for ArcMap, and
- ArcMAP software by ESRI (commercial!).

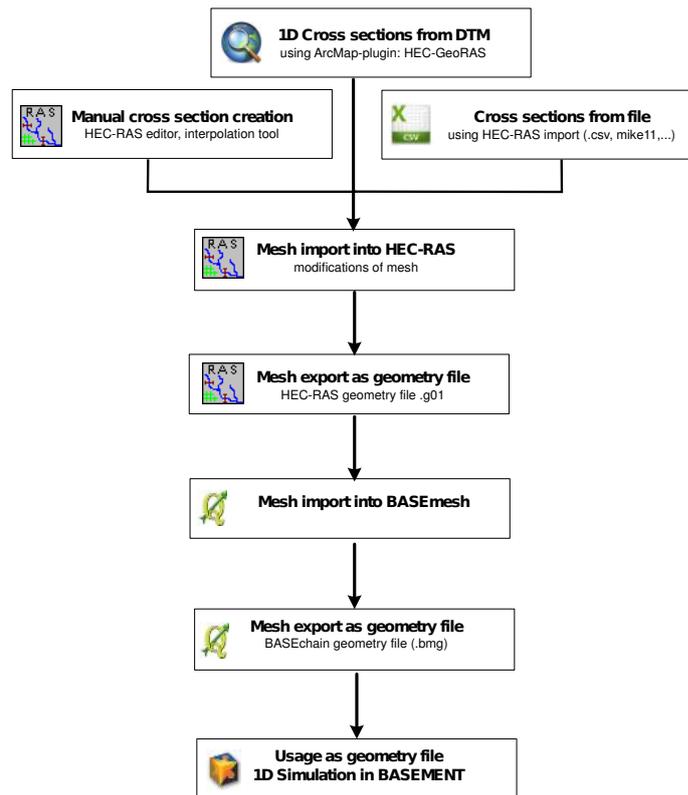


Figure 1.30 Workflow HEC RAS

Input Data The data needed to complete this tutorial comes as ZIP - file and needs to be extracted to a location of your choice. The data comprises:

- ArcMap project file (containing the raster data and river shape data).

The result files of the HEC-GeoRAS plugin and the conversion steps are additionally provided in the folder 'additional_results'.

Conversion HEC-RAS to BASEchain: Using HEC-RAS software as pre-processor requires to convert the resulting geometry files (.g01) to the BASEchain geometry format (.bmg). This conversion can be done with BASEmesh, but not all geometry features of HEC-RAS are supported (e.g. bridges, inner structures). At the moment, the following features can be converted to BASEchain:

- Cross section profiles ('#Sta/Elev'),
- Cross section profiles ('#Sta/Elev'),
- Friction values ('#Mann', optional),
- Levees ('Levee', whereas always BOTH levees must be given, optional), and
- Georeferencing of the cross sections ('XS GIS Cut Line2', optional).

Feel free to enhance the converter script to consider additional features or to remove other limitations or bugs. If you do so, please send us the improved versions of the script-files.

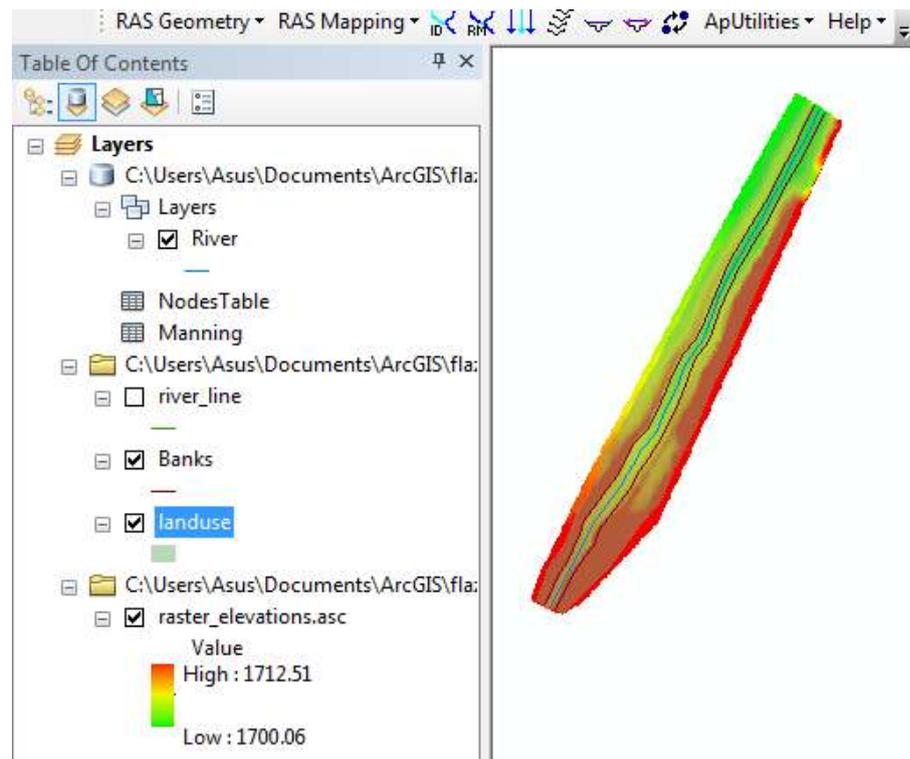


Figure 1.31 ArcMap

1.5.1 Set-up the environment

- (1) Install ArcMap, HEC-RAS and HEC-GeoRAS

To ease this tutorial, the input data was already prepared for the HEC-GeoRAS plugin:

- (2) Open the ArcMap input file named 'Tutorial4.mxd'. This adds the needed raster data and shape layers to the table of contents (TOC).
- (3) Ensure that the HEC-GeoRAS toolbar is activated by checking the option *Customize Toolbars HEC-GeoRAS*.

You can see the raster DTM file and multiple shape files needed for the 1D mesh generation listed in the TOC:

- 'River' is the river streamline,
- 'Banks' are lines indicating the main channel region, and
- 'landuse' are polygons containing the Manning friction data.

Please note that instead of raster data, also TIN data can be used as input data for the topography information.

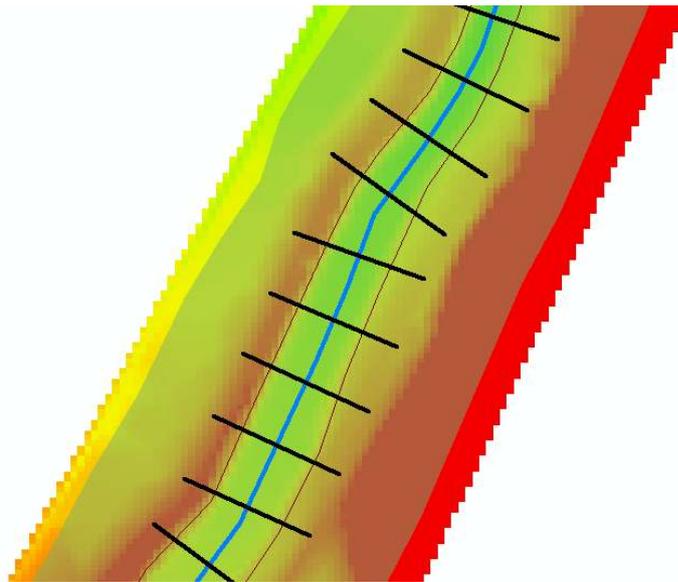


Figure 1.32 XS Cutlines

1.5.2 Generate the cross section profiles

The main task of the HEC-GeoRAS plugin is the generation of 1D cross sectional profiles with elevation data. To create these cross sections, two steps are necessary:

- Define the locations of the cross section lines, and
- Map the elevations of the DTM upon these lines.

To simplify the procedure we use an automatic generation of the cross section lines, which generates cross section lines in a given distance and perpendicular to the direction of the river streamline. Please note that this automatic cross section generation is not the recommended workflow by the HEC-RAS team and usually requires subsequent manual adaptations.

- (1) We need a new layer containing the cross section lines. Click on *RAS Geometry* → *Create RAS Layers* → *XS Cut Lines* and name the layer “XSCutLines”.
- (2) Click on the Icon *Construct XS Cut Lines*. A new dialog opens named *Create Cross Sections*.
- (3) Choose *XS Cutlines* = ‘XSCutLines’ and *Stream Centerline* = ‘River’. Set *Interval* to 40 (distance between the cross section lines along the stream centerline) and *Width* to 80 (width of the the cross section lines perpendicular to the stream centerline). Using these settings, cross sections will be automatically generated.

In the data view, you can see the the generated cross section lines and their placement along the river streamline (see figure, river = blue color, cross section lines = black color). These lines can now be manually modified and adapted to special local conditions.

1.5.3 Set the cross section, main channel and friction properties

Up to now, the cross section lines were only placed along the river streamline. In addition, we must map the data (elevation, friction, ..) onto these cross sections. First, we map the river/reach names on the cross section lines:

- (1) Select *RAS Geometry* → *XSCut Line Attributes* → *River/Reach Names*.

Then we map the geometry/topography data on the cross section lines:

- (2) Select *RAS Geometry* → *Stationing*.
- (3) Select *RAS Geometry* → *Bank Stations*. Use the 'Banks' layer in the opening dialog.
- (4) Select *RAS Geometry* → *Elevations*. Select the DTM 'raster_elevations.asc' as *Terrain* data. An additional 3D shape layer is generated named 'XSCutLines3D'.

Finally, we map the friction data on the cross section lines:

- (5) Select *RAS Geometry* → *Manning's n Values* → *Extract N Values*. Select 'landuse' as *Land Use* layer and select the 'N-Value' attribute field as *Manning Field*. As *XS Cutlines* layer select 'XSCutLines'.

Now, we have mapped the data on the cross section lines. As next step we will export this cross section data to file. Be aware, that additional settings could be made, as for example the levees or the ineffective flow areas. Other features, like bridges or inner structures, however, are not supported within this workflow.

1.5.4 Export the data from HEC-GeoRAS

The data export into GIS-format (.sdf) is quite simple:

- (1) Click *RAS Geometry Export RAS Data*. Choose a file name in the opening dialog and press *OK*.

1.5.5 Import GIS data into HEC-RAS and export a geometry file

- (1) Open the HEC-RAS software.
- (2) Click on *Edit Geometric Data ...*. The empty *Geometric Data* window opens.
- (3) In the *Geometric Data* window select *File Import Geometry Data GIS Format ...*. Select the .sdf-file, which was exported using HEC-GeoRAS, and press *OK*. The *Import Geometry Data*-dialog opens and guides you through the data import.
- (4) Select *SI (metric) units* and click *Next*.
- (5) Click *Next*.

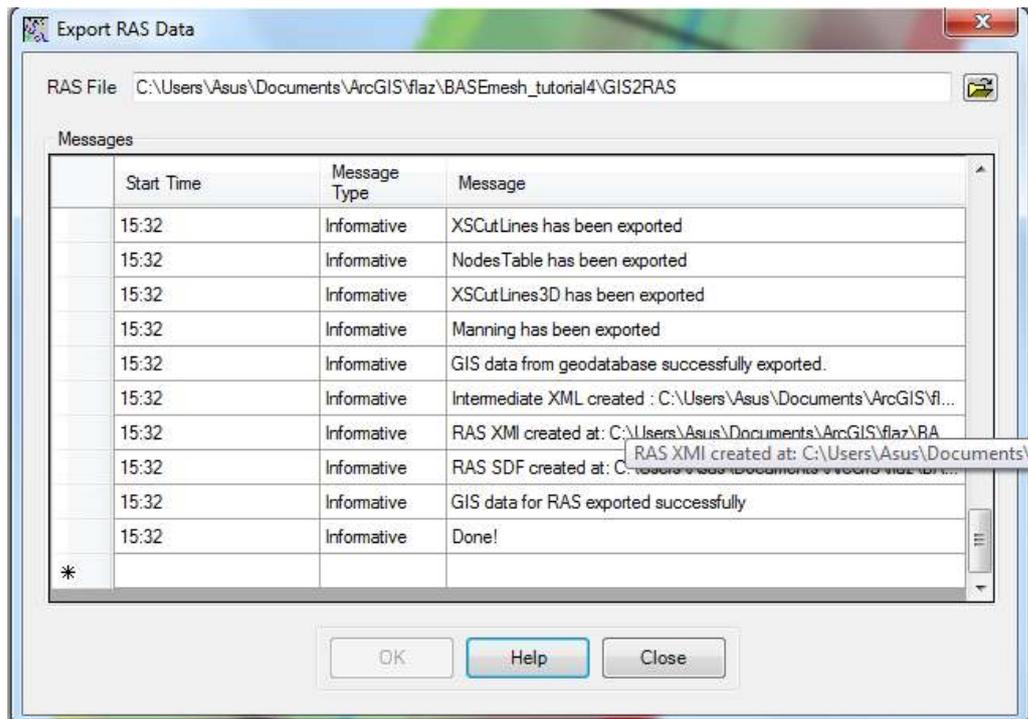


Figure 1.33 Export RAS data

- (6) At the input field *Import River* choose 'Flaz'. At the input field *Import Reach* select 'Reach1'. Then, click on *Finished - Import Data* to finish the data import.

Now the GIS data is imported and the cross sections are drawn in the *Geometry Data* window. Cross sections can now be edited if needed using various HEC-RAS tools. It may be useful, in particular, to reduce the number of cross section points. The automatic generation from a DTM often generates more points than needed to represent the cross section profile accurately, thereby reducing the simulation speed. The total number of points can be reduced using the option *Tools Cross Section Point filter*. If the modifications are finished, the cross sections are ready to be exported as HEC-RAS geometry file (.g01):

- (7) Select *File Export Geometry Data* and choose a filename.

1.5.6 Convert HEC-RAS format to BASEchain format using BASEmesh

Finally, we will use BASEmesh to convert the HEC-RAS geometry file (.g01) to the BASEchain format (.bmg).

- (1) Open QGIS.
- (2) Click on *Plugin* → *BASEmesh* → *Import mesh*. The input dialog opens.
- (3) Browse for the generated HEC-RAS geometry file (.g01).
- (4) Select a name and click *Import mesh*. In the text box some information are displayed about the conversion process. It is strongly recommended to read this information to see the problems encountered at the conversion process.

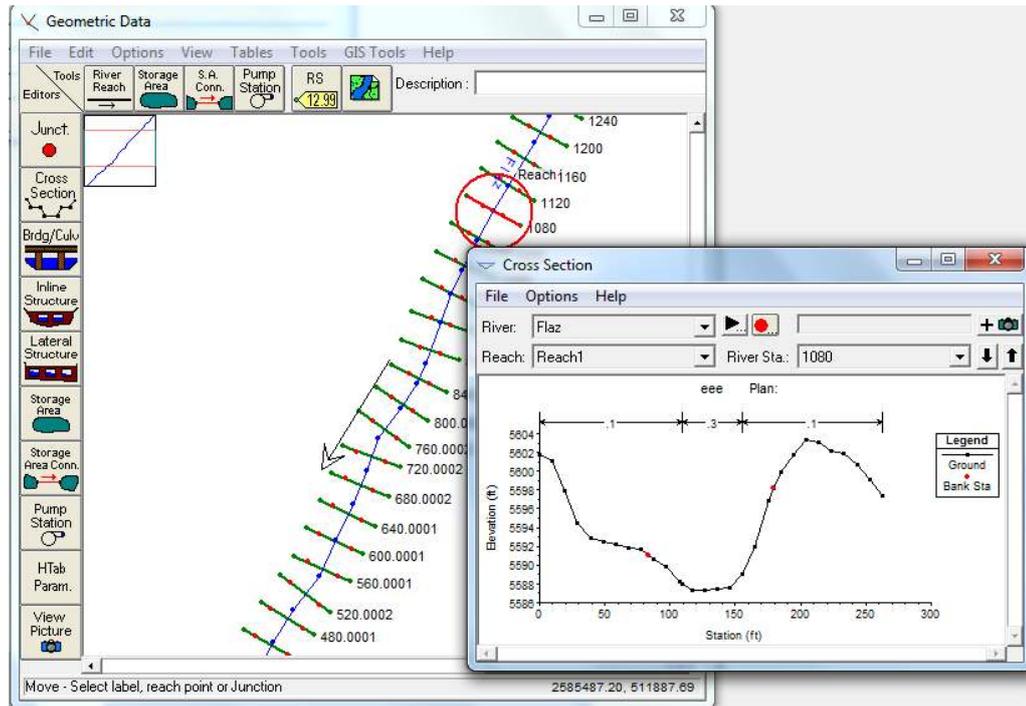


Figure 1.34 HEC-RAS geometry data

If the import succeeds, then two new shapefiles are generated and added to the TOC:

- A point shapefile with the suffix '_points.shp', containing the cross section nodes and their elevations.
- A line shapefile with the suffix '_crossSections.shp', containing the cross section data in the attribute table.

The line shapefile can now be exported to file in the BASEchain geometry format (.bmg). It is also possible to import BASEchain geometry files (.bmg) and export them to HEC-RAS format (.g01). Be aware that the converter does not support all features and is tested only in a limited number of cases.

- (5) Click on *Plugin* → *BASEmesh* → *Export mesh*. The output dialog opens.
- (6) Select the radio button *1D mesh export* and set the output format to *BASEchain (*.bmg)*.
- (7) Select the cross section shape file, containing the cross section data ('_crossSection.shp') and click on *Export mesh*.

Now we have completed the tutorial and created a 1D BASEchain geometry file. This file can be used for 1D simulations in BASEment. Figure 1.35 shows the BASEment cross section editor, where you can check if the workflow was successful and all data was preserved.

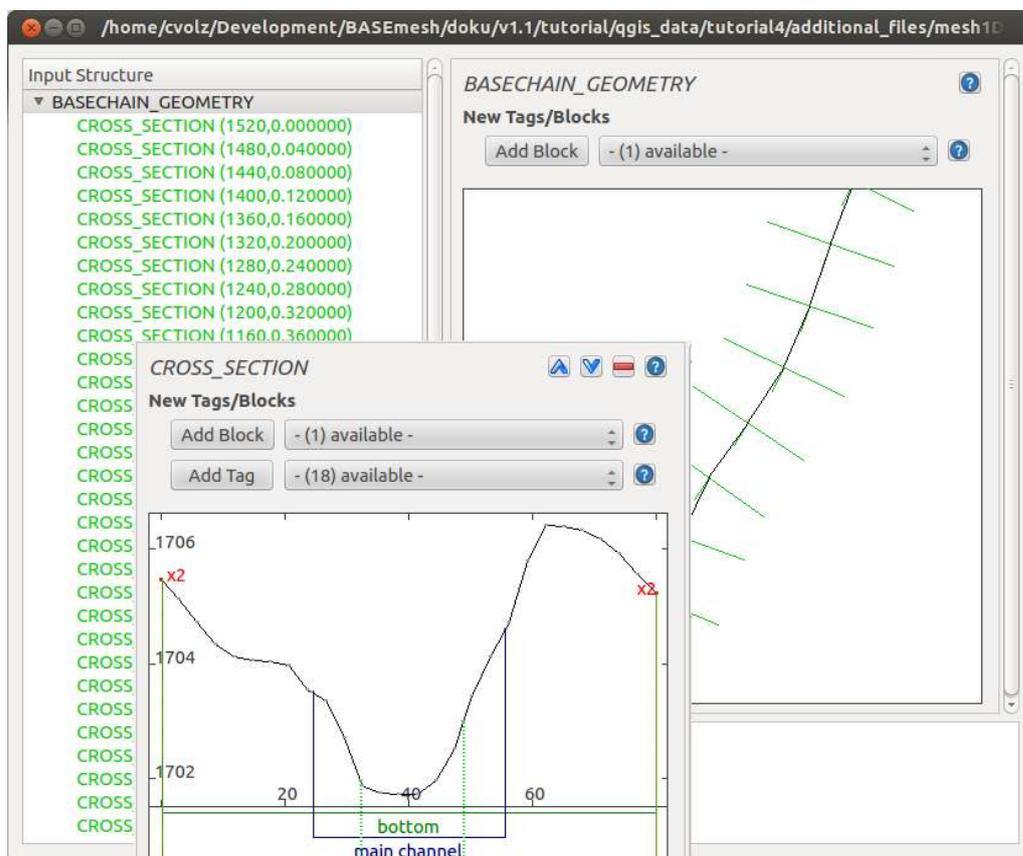


Figure 1.35 BASEchain geometry data

1.6 Tips and Tricks

1.6.1 Recommended Plugins

QGIS offers the possibility to extend its features using plugins from various sources. We recommend in particular the following plugins.

Core plugins:

fTools This is the main plugin for many common vector-based GIS tasks. Core plugin, which can be activated in the plugin manager.

Spatial Query Plugin Core plugin of QGIS, which can directly be activated in the QGIS plugin manager without downloading. *Spatial Query* is comparable to ArcGIS' *Select by Location* feature and enables the user to select features in a target layer with reference to another layer. Possible operators are: contains / equals / overlap / crosses / intersect / touches / within.

Select Within Another spatial query plugin that runs through each geometry you want to select from and tests if the centroid falls within the selected geometry. This plugin is especially useful to select mesh elements for the assignment of material indices (see Section 1.3.2).

Processing Spatial data processing framework, which gives access to a large number of analysis algorithms. It can also connect external algorithms from other GIS packages, such as GRASS, SAGA or Orfeo Toolbox. One of the main features is the graphic modeler, where frequent workflows can be graphically represented by the user and executed automatically.

Python plugins:

Python plugins are mostly contributed by the worldwide QGIS user community. Most of them are stored in the official repository and available as stable or experimental versions.

Crayfish plugin Visualizes result data on structured or unstructured meshes using color maps and vectors. Crayfish is able to directly visualize result data from the numerical simulation software BASEMENT (see Post-Processing tutorial Section 2.1).

Point to One Converts a series of points to lines or polygons based on a common attribute or a sequence field.

Point Sampling Tool Samples polygon attributes and raster values from multiple layers at specified sampling points.

Dxf2ShpConverter Conversion and import of dxf files.

Interpolation plugin Very useful for interpolating raster maps of given elevation data, e.g. point clouds.

1.6.2 Tipps and Tricks

The following list contains some basic tips and tricks for problem handling and the daily work with BASEmesh.

Creation of polygons out of intersection polylines: In many cases, polygons must be created out of intersecting polylines, e.g. for the definition of model boundary polygons based on breaklines. For completing this task, two features are available in QGIS:

- via fTools: Select *Vector* → *Geometry Tools* → *Lines to polygons*. Quite simple menu, no parameters or options can be chosen (standard option). Prone to errors with complex geometries.
- via the Processing toolbox: Select *Processing* → *Toolbox* → *QGIS geoalgorithms* → *Vector geometry tools* → *Polygonize*. Complex geometries are handled well by this tool, multiple options and a log are available.

Creation of a model boundary based on elevation data: When creating an elevation mesh, all vertices of a model boundary polygon must lie on elevation points. Otherwise, there might be interpolation errors in the following meshing steps. Therefore, it is advisable to create a model boundary using the *Convex hull(s)* - feature of QGIS. Below is a short example using the input data provided with Section 1.2.

- Select *Vector* → *Geoprocessing Tools* → *Convex hull(s)*
- As *Input vector layer* select 'XS_points_straightened'.
- Check the radio button *Create single minimum convex hull*.
- Select an appropriate name for the output shapefile.
- Check *Add result to canvas*.
- Select *OK*, after completion close the dialog. Your result should now look like Figure 1.36 .

What to do if Triangle creates (almost) infinitely dense triangulations:

- Polylines that are used for the segmentation of boundary polygons must end in vertices of the polygons. Otherwise, Triangle creates an almost infinitely dense triangulation to ensure its angle and area criteria.
- Be aware that using the *vertex editing tool* generates a new vertex point at each double-click. This can easily lead to the generation of two or more points at the same location or very near to each other. In such a case, Triangle creates a very dense triangulation due to its angle and area criteria.
- Prevent situations where breaklines (and corresponding vertices) have very short distances to each other. The resulting mesh will be very fine in these regions. Meshing sometimes requires to manually adapt the input data and to make compromises between mesh quality and accuracy in certain regions.
- Use the *Relative snapping tolerance* provided in the BASEmesh dialogues *Elevation meshing* , *Quality meshing*  and *String definition* . The mesh shown in Figure 1.37 on the right was generated with the default *Relative Snapping Tolerance* of -6. This led to a very dense triangulation at the intersection of breakline and model boundary since the breakline is not exactly snapped to the vertex of the model boundary. Furthermore MATID 3 was not considered for the embankment as desired. To prevent such behaviour, increase the tolerance to -3 for example. This allows you to generate a proper mesh like shown in Figure 1.37 on the left.



Figure 1.36 QGIS Convex hull

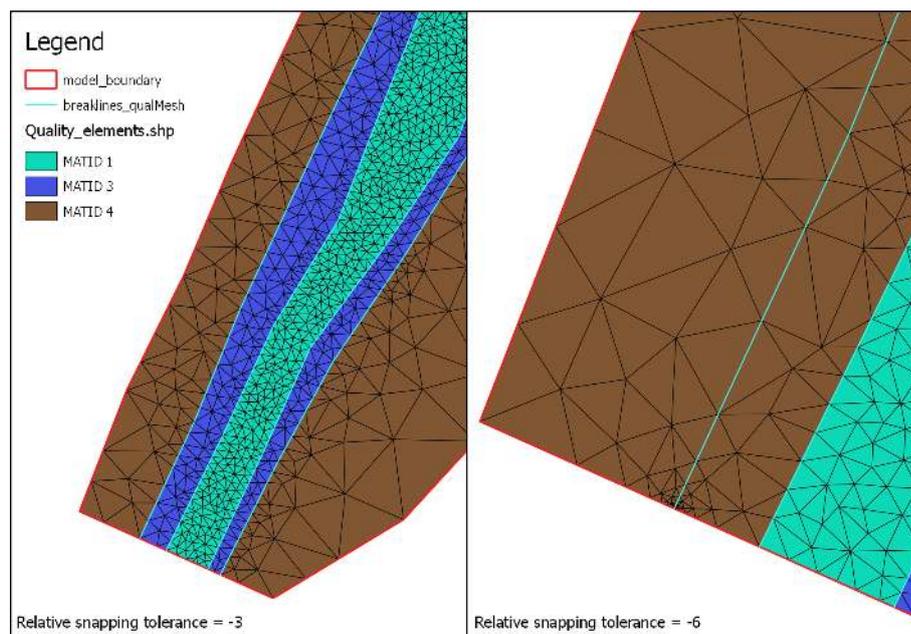


Figure 1.37 Quality meshing using the Parameter 'Relative snapping tolerance'

Do not group layer in the table of contents (TOC): Grouping layers in the TOC may lead to errors because the resulting shapefiles cannot be generated after triangulation.

Conversion of 3D shapefiles to 2.5D shapefiles with elevation attributes This special kind of shapefile is sometimes used in ArcGIS. The 3rd dimension value is not stored in the attribute table (as in 2.5D shapefiles), but in the geometry definition itself. The x and y coordinates of such shapefiles can be displayed in QGIS, but the elevation information is inaccessible and lost. In the following, three workflows are illustrated how to convert 3D shapefiles to 2.5D shapefiles:

- *GDAL - ogr2ogr*: On Linux systems, this GDAL command line tool is directly accessible on the shell. On Windows systems, the tool can be used in the OSGEO4W-shell. For further information, please visit the [GDAL homepage](#)
- *Spatialite*:
 - Open QGIS, add your shapefile and save it as Spatialite (or alternatively use the QGIS DB Manager to drag and drop the shapefile to an already existing Spatialite Database).
 - Add the newly created Spatialite vector and through the QGIS vector properties ('fields' tab) add a new column, that will store the z-values.
 - In the QGIS DB Manager SQL Window use this command: *update tablename set columnname = st_z(st_pointn(geom,1))*.
 - If at the end of the process you really need to have a shapefile (shame on you) then just use the 'save as...' function of QGIS.
- *PostGIS*:
 - Import the shapefile to your PostGIS database
 - `ogr2ogr -f "PostgreSQL" PG:"host=yourhost user=user dbname=dbname password=*****" shapename.shp`
 - Connect to your database
 - `psql -h yourhost -U user dbname`
 - Add a column
 - `ALTER TABLE tablename ADD COLUMN columnname numeric(19,11);`
 - Fill the column with the z-values
 - `update tablename set columnname=st_z(ST_PointN(wkb_geometry,1));`
 - Save result as shapefile

2

Post-Processing of 2D simulation results

2.1 Introduction

In the following tutorials it is demonstrated how to do the post-processing of 2D results generated by a BASEMENT simulation. For this purpose the free and open source GIS software [QGIS](#) is applied together with the QGIS plugin Crayfish developed by [Lutra consulting](#).

In addition the free and open source application [ParaView](#) is used to generate 3D views of the BASEMENT simulation results.

2.2 2D result visualization with QGIS Crayfish

2.2.1 Input data

The data needed to complete this tutorial comes as ZIP-file and needs to be extracted to a location of your choice. All screenshots and figures in this document were generated with QGIS version 2.8 ‘Wien’. Besides the computational grid (Flaz_mesh.2dm), the results of the unsteady flow simulation of the tutorial ‘Hydrodynamics and sediment transport at the river Flaz’ (Section 5.1) are used for visualization. In the *SPECIAL_OUTPUT* Block of the BASEMENT command file select *format* ‘sms’ to generate result files that can be visualized with Crayfish (see Figure 2.1). Crayfish can handle both types *node_centered* and *element_centered* results. Be aware that most of the data (e.g. depth, velocity, wse,...) is calculated on the elements during simulation. When choosing type *node_centered*, these results are interpolated to the nodes by BASEMENT and written to the solution files (*.sol).

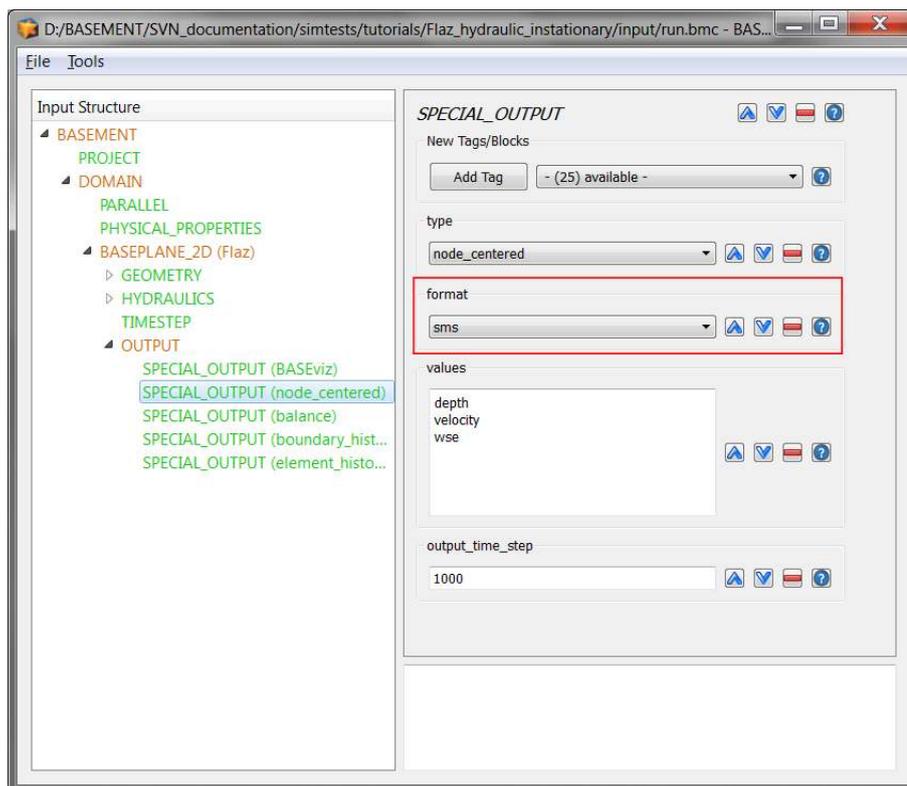


Figure 2.1 Settings defined in the SPECIAL_OUTPUT Block of BASEMENT to generate time dependend result-files (*.sol) of a 2D simulation

2.2.2 About Crayfish

Crayfish is a plugin (extension) developed by Lutra Consulting for the free and open source GIS platform QGIS. The Crayfish plugin aspires to be a complete set of post-processing tools to support numerical modelling within QGIS. With Crayfish, users can load time varying mesh into QGIS. Currently, Crayfish supports a number of hydraulic modelling software packages including BASEMENT.

Crayfish loads and renders results directly rather than converting them to GIS formats before viewing. This allows users to flick quickly through the various output steps in the result files and to create animations. When loading data via Crayfish into your QGIS project, the Crayfish Viewer control panel is docked into your QGIS window in order to modify the properties of the displayed Crayfish Quantities (see Figure 2.2).

2.2.3 Installation

Crayfish can be installed from the official QGIS plugin repository. It requires binary libraries specific to your platform. Lutra consulting distributes Windows (32-bit and 64-bit) DLL files and Linux libraries (64-bit) for Debian-based distributions (Ubuntu, Debian, Linux Mint, etc). If your operating system is one of these, the plugin should automatically download the required libraries. To install Crayfish from the QGIS plugin repository, follow these steps:

- (1) Start QGIS

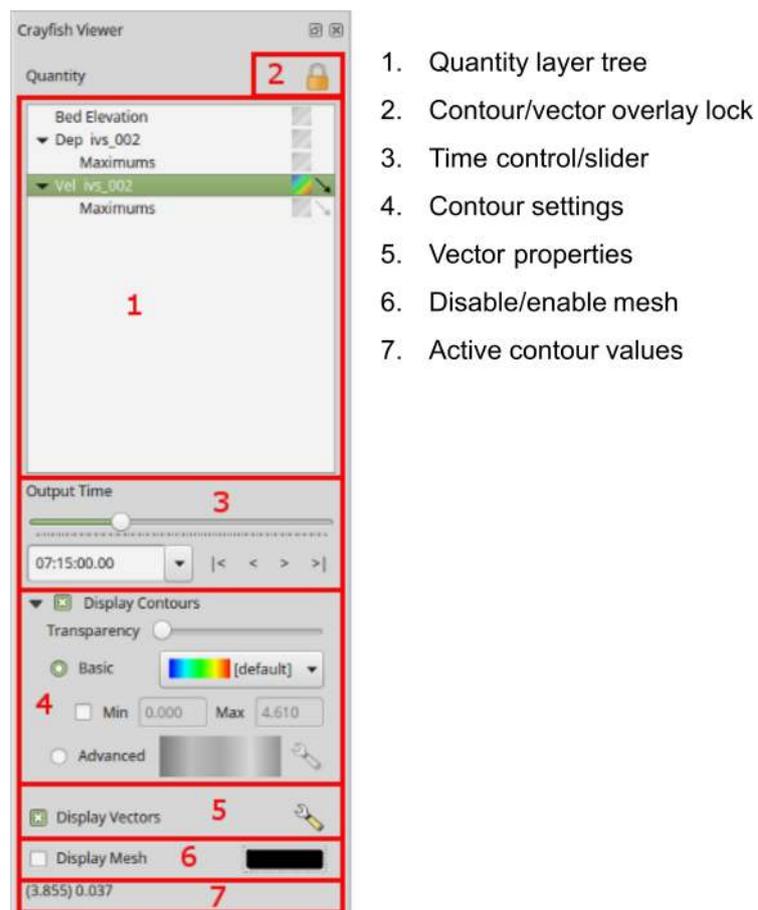


Figure 2.2 Crayfish Viewer control panel, Crayfish version 2.1.3

- (2) Load the QGIS plugin manager by choosing *Manage and Install Plugins...* in the menu *Plugins* in the QGIS main toolbar
- (3) In the tab *All* type ‘Crayfish’ into the search field and click *Install plugin*

Once Crayfish is installed successfully, a new icon  should be added to your Layers toolbar.

2.2.4 Load and visualize data

2.2.4.1 Project Settings

In order to display the provided geodata correctly, some settings regarding the coordinate reference system (CRS) have to be defined. In this tutorial we use the same CRS (CH1903/LV03) like in the tutorial of BASEmesh (Section 1.1). With the following steps, the project’s CRS can be changed:

- (1) Go to *Project* → *Project Properties*.
- (2) Under the tab *CRS* you can see the coordinate reference system settings for the current project. Check *Enable ‘on the fly CRS’ transformations*, just to change the reference system.
- (3) Enter EPSG code ‘21781’ into the field *Filter*.
- (4) Select the coordinate reference system ‘CH1903 / LV03’ and click *OK*.
- (5) Again go to *Project* → *Project Properties* → *CRS*.
- (6) Now make sure to uncheck *Enable ‘on the fly CRS’ transformations* as shown in Figure 2.3 below.
- (7) Optionally define a *Project title* under the tab *General* → *General Settings**.
- (8) Close the Project Properties window. If everything went well, you should see the chosen project name at the title of the QGIS main window and the EPSG code of your coordinate system at the lower right corner of the QGIS desktop.

2.2.4.2 Loading results of unsteady flow simulation (river Flaz)

The provided data is stored in the ‘QGIS’ subfolders ‘background_data’, ‘Flaz_mesh’ and ‘2D_results’. You can either use these data or visualize your own mesh and/or simulation results of the BASEMENT 2D tutorial ‘Hydrodynamics and sediment transport at the river Flaz’.

Loading geodata

- (9) Go to *Layer* → *Add Layer* → *Add Raster Layer* (Alternatively press Ctrl+Shift+R or use the *Add Raster Layer* icon  in the Manage Layers toolbar).

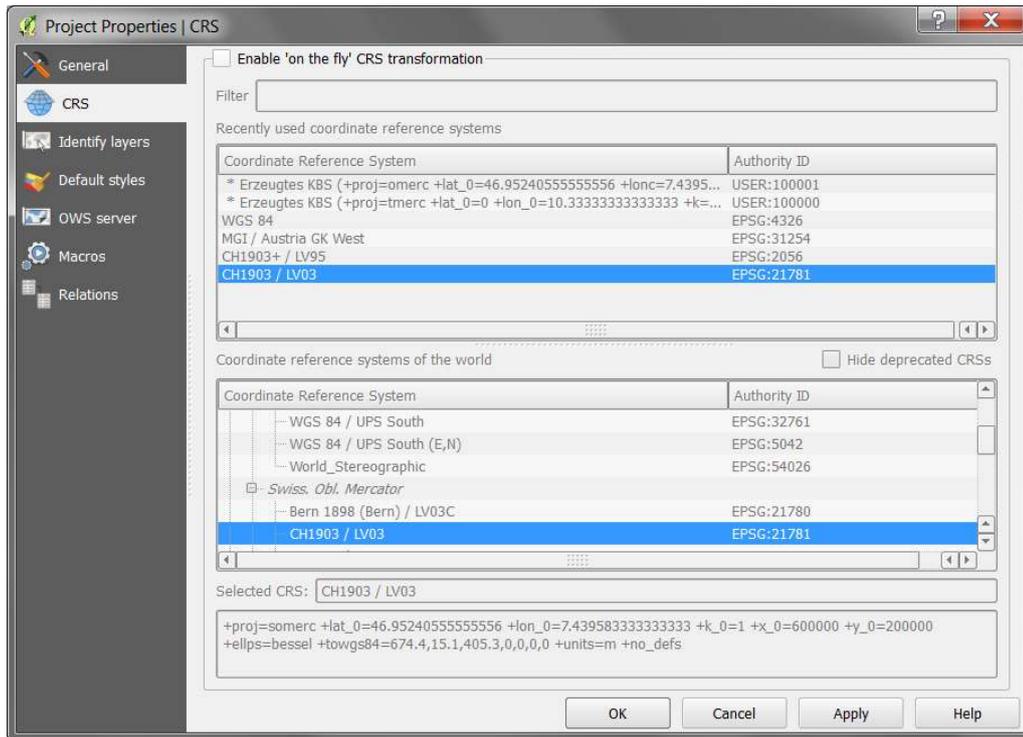


Figure 2.3 Project Properties, CRS

- (10) Browse to the folder 'background_data' and load the areal image 'Flaz_si25.tif'.
- (11) Go to *Layer* → *Add Layer* → *Add Vector Layer* (Alternatively press Ctrl+Shift+V or use the *Add Vector Layer* icon  in the Manage Layers toolbar).
- (12) Browse to the folder 'Flaz_mesh_data' and load the provided shapefiles (*.shp).

The color schemes of newly added layers are randomly chosen by QGIS. You can change them by double-clicking on a layer and selecting *Style*. QGIS displays the loaded layers according to their order in the Layers control panel.

Adding Crayfish layers

To open a set of results, either use the *Add Crayfish Layer button*  on the *Manage Layers toolbar* or select *Crayfish* → *Add Crayfish Layer* from the main *Plugins* menu. You should now be presented with the *Open Crayfish File Dialog*. From here you can either load a mesh file (.2dm) or results files (.dat, .sol or *.xmdf). Crayfish requires projection of the layers to load them into QGIS. If there is a .prj file within the same folder of the mesh file, Crayfish will use the projection information from the .prj file. Otherwise, it will assign a projection based on user's project settings.

First, we load the computational mesh of the 2D simulation. If you use your own data, make sure to load the latest mesh file (*.2dm) you used for your simulations. Otherwise the node ids of the mesh won't match those of the result files and the visualization won't be correct.

- (13) Press the *Add Crayfish Layer button*  and browse to the folder '2D-results'.

(14) Toggle visibility of ‘2DM Mesh Files (*.2dm)’ and load the mesh ‘Flaz_mesh.2dm’.

Now load the result files of the 2D simulation:

(15) Press the *Add Crayfish Layer button*  and browse to the folder ‘2D-results’.

(16) Toggle visibility of ‘Result Files DAT, SOL...’ and load the solution files ‘Flaz_nds_depth.sol’ and ‘Flaz_nds_velocity.sol’

The appropriate mesh file should be loaded automatically when the user opens a .dat or .sol file. If the mesh file is named differently, users will be prompted to locate the mesh file manually. In this case, browse to the location of your mesh and double-click on the file (*.2dm). If a quantity contains time-varying data, you will be able to browse through the output times defined by the ‘output_time_step’ in the BASEMENT command file (see Figure 2.1) and change the timing through the time control. To view the underlying mesh of the result data, you can either activate the Display Mesh on item 6 of the Crayfish viewer panel (see Figure 2.2) or check/enable the shapefile of the mesh (Flaz_mesh_elements) in the Layers control panel.

2.2.4.3 Display properties

The style of the displayed quantities can be modified in the Crayfish Viewer control panel (see Figure 2.2).

Contour styles

Styling of contours can be done either through the Basic settings or the Advanced settings. The basic settings option allows users to quickly change the color profile, minimum and maximum extents and transparency. For advanced options, click on the setting icon and a new window will appear (see Figure 2.4). In this window you can:

- Create your own color ramp
- Set the interpolation in the color ramp
- Save and load color settings
- Fill values below or above min/max

Vector Styles

Vector styling of the velocity vectors can be done through the vector settings dialogue window (Figure 2.5). Under General Options, you can set the shaft length based on min/max values, scaled or fixed. You can set an arbitrary grid, where vectors will be displayed by activating *Display Vectors on User Grid*. This is particularly useful to view a smoother vector interpolation.

Contour and vector overlay

Quantities may contain contour only, or contour and vector data. In the example below, depth contains only contour values, whilst velocity contains both contour and vector. By default, Crayfish renders contours and vectors from the same quantity. If you want to see contours and vectors from different quantities, you need to ‘unlock’ the contour/vector overlay in the *Crayfish Viewer* control panel (see Figure 2.6). In this example depth contours are overlaid with velocity vectors at the **same** output time step.

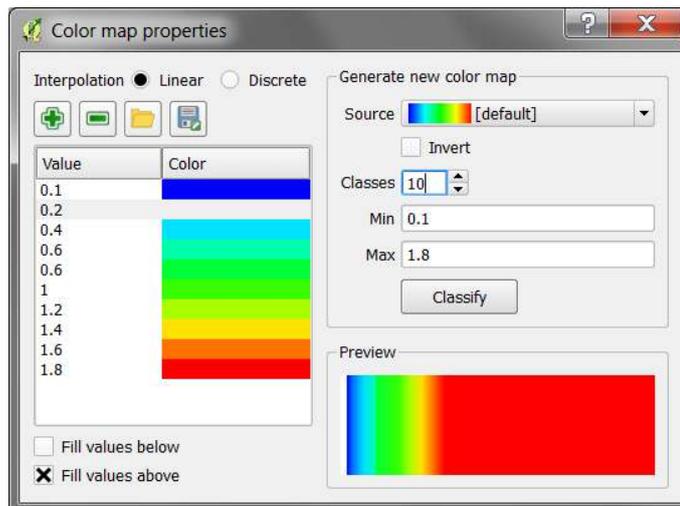


Figure 2.4 Advanced Contour Options window of the Crayfish Viewer

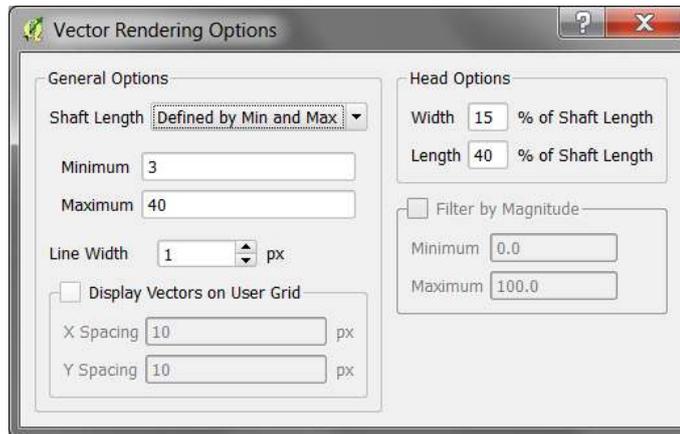


Figure 2.5 Vector Options window of the Crayfish Viewer

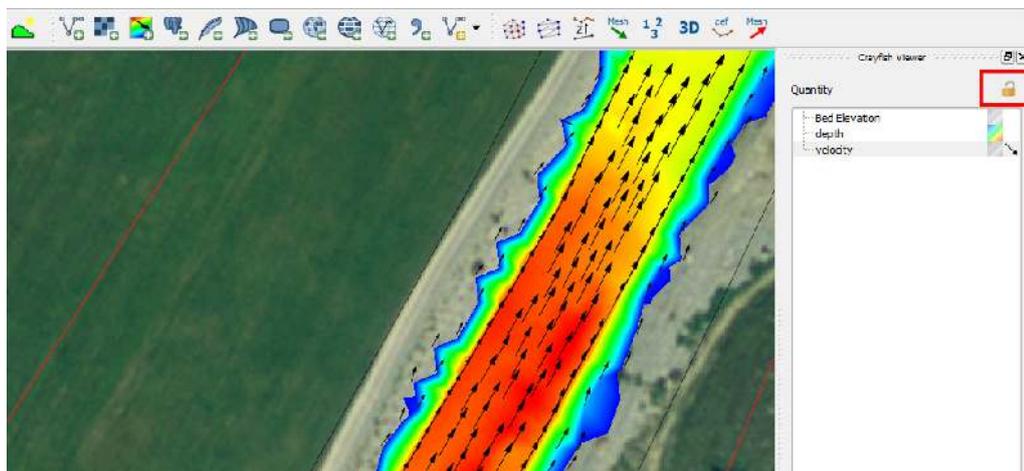


Figure 2.6 Unlock Crayfish Viewer control panel, Orthophoto: © 2016 swisstopo (JD100041)

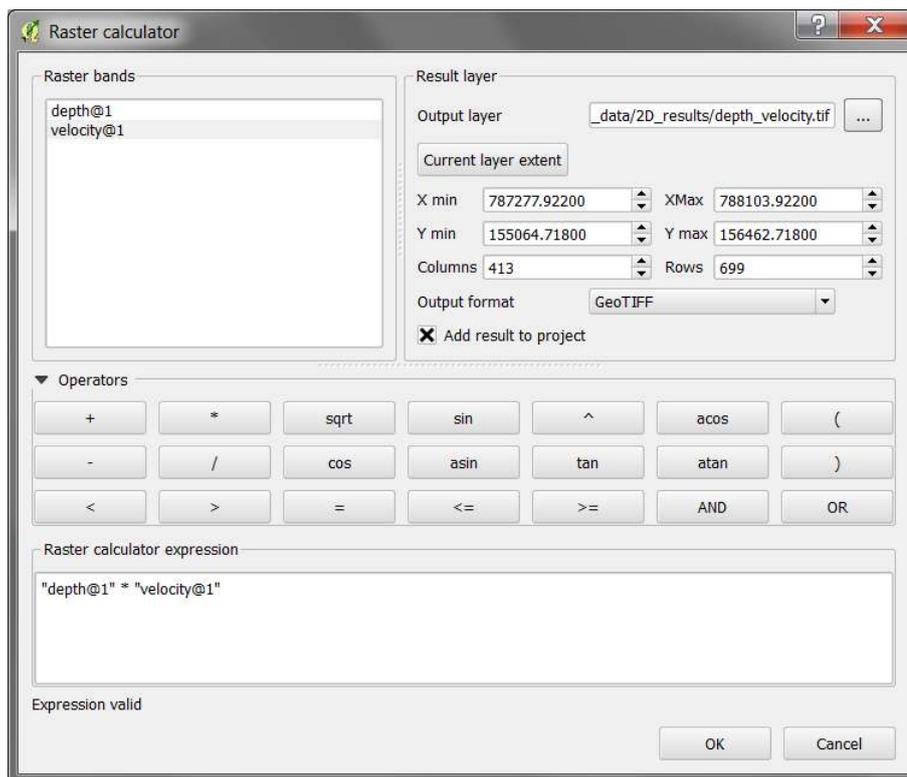


Figure 2.7 Applying QGIS Raster Calculator

2.2.4.4 Export results to raster

For further results analysis, it might be useful to export the simulation results into raster format. Crayfish offers the possibility to directly resample the contour data at the **active output time step** on a desired output resolution.

- (17) Right click on the contour *Quantity* in the *Crayfish Viewer* control panel (in this example Bed elevation, depth and velocity are available).
- (18) Chose *Export to Raster Grid...*, define the desired Grid resolution and press *OK*.

The generated raster is stored in Geo-Tiff format (*.tif) and will cover the extent defined by the model boundary. The flexible raster format allows further post processing steps of the simulation results. After having exported both quantities depth and velocity, you can easily calculated the product depth * velocity using the QGIS *Raster Calculator* for example. This might be of particular interest when generating flood hazard maps for example.

- (19) Go to *Raster* → *Raster Calculator*.
- (20) Specify the *Raster calculator expression* as shown in Figure 2.7 by double-clicking on the *Raster bands* depth and velocity in combination with the required *Operators*.
- (21) Define the *Output layer* and press *OK*.

The new calculated raster dataset will cover the same extent like the two input datasets and will be stored in Geo-TIFF format as well.

2.2.4.5 Profile tool

If you want to analyse the temporal change of variable along a certain cross section of a river reach during a BASEMENT simulation, in principle there are two options:

- Define a *STRINGDEF* along the desired location within your computational grid and chose a *SPECIAL_OUTPUT* of type *stringdef_history* with the required *stringdef_values*.
- Use the QGIS plugin ‘Profile Tool’ and analyse the results of a Crayfish layer.

The advantage of option two is a fast and easy evaluation at any location of the modelled domain, without having defined any *stringdef_history* outputs beforehand.

Installation

The plugin can be installed via the official QGIS repository:

- (22) Load the QGIS plugin manager by choosing *Manage and Install Plugins...* in the menu *Plugins* in the QGIS main toolbar.
- (23) In the tab *All* type ‘Profile tool’ into the search field and click *Install plugin*.

Once the ‘Profile Tool’ is installed successfully, a new icon  should be added to your Layers toolbar.

Usage

In general, there are two options to use the ‘Profile Tool’:

1. Using a temporary polyline:

With this option, you can draw an arbitrary polyline at the location of your choice. Along this line the data of the selected *Quantity* will be extracted.

- (1) Click on the *Terrain profile* icon  and the Profile Tool window pops up.
- (2) In the field *Selection* chose *Temporary polyline* from the dropdown menu.
- (3) Use the black cross to draw a red line along the desired location of your displayed *Quantity* (see Figure 2.8).
- (4) Start drawing the line with a click and finish the line with a double-click

Now the profile of the chosen quantity should be displayed in the *Profile Tool* window (see Figure 2.8 below). From there you can save the graph in different formats or export the raw data via the tab *Table*.

2. Using a selected polyline:

With this option you can chose an existing polyline feature e.g. a certain cross section to extract the desired quantities.

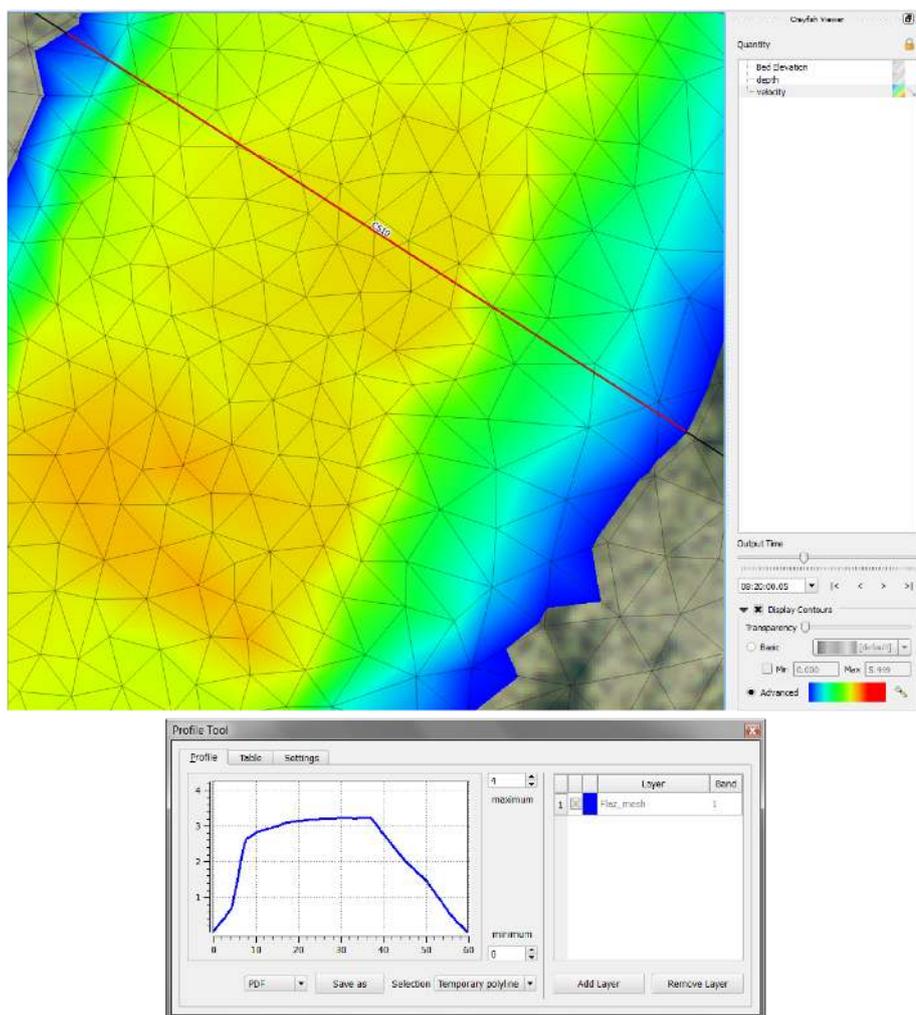


Figure 2.8 Velocity profile along the red line drawn at the location of cross section 'CS10' generated with the 'Profile Tool' plugin

- (1) First make sure to *Deselect Features from All Layers* pressing the respective toolbar button  toolbar button
- (2) Again click on the *Terrain profile* icon  and the Profile Tool window pops up.
- (3) In the field *Selection* now chose *Selected polyline* from the dropdown menu.
- (4) Right-click on the layer 'Flaz_cross_sections' and go to *Open Attribute Table*.
- (5) Select the feature with *name* 'CS10' and left-click into the map displayed in the QGIS main canvas.

Again the profile of the chosen quantity should be displayed in the *Profile Tool* window (see Figure 2.8). Note that the profile also contains entries with value 0 since the extent of the cross section is wider than the wetted main channel.

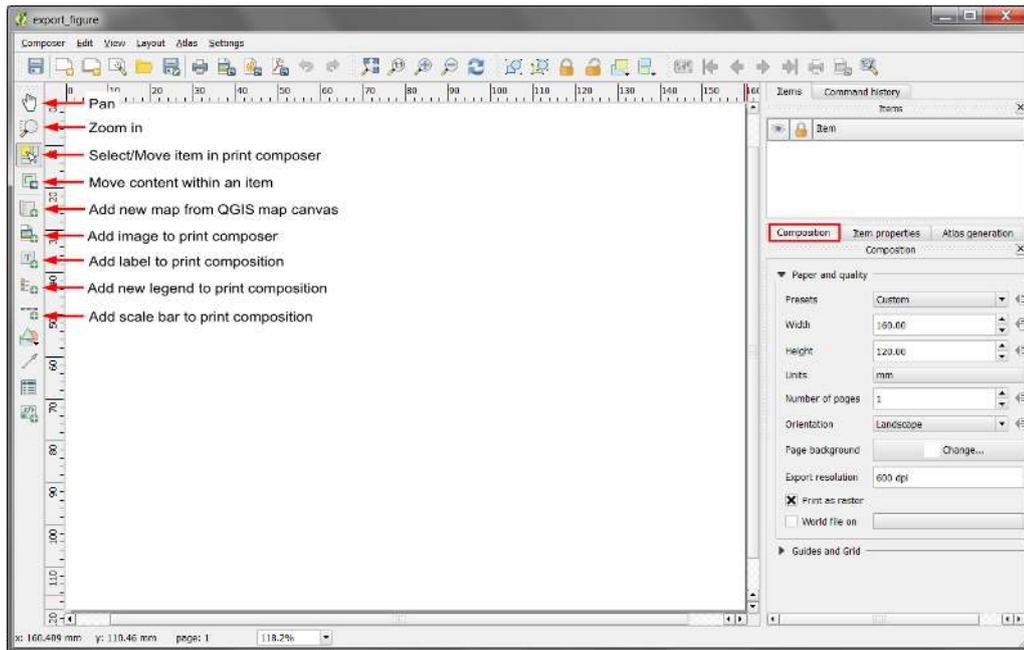


Figure 2.9 QGIS Print Composer with some of the basic control features

2.2.5 Creating maps and animations

An important step of the post processing of simulation results is the creation of maps for example of flow depths or velocities to be published in reports or presentations. For this purpose QGIS offers a powerful tool called ‘Print Composer’ which allows you to take your GIS layers and package them to create professional maps including legend, scale bar or text boxes. With the latest version of the plugin Crayfish it is also possible to display and export Crayfish layers using the QGIS ‘Print Composer’. This tutorial can’t cover the whole functionality of the ‘Print Composer’, but some of the main features, needed to create proper figures and animations of 2D simulation results are explained subsequently. Check the online [User Guide](#) of the software QGIS for further documentation.

2.2.5.1 Customized layout (QGIS print composer)

After loading all the layers to be exported in a map into the main QGIS canvas, we can start to assemble our map. Note, that only layers which are checked / enabled in the *Layers control panel* of the main canvas can be displayed in the *Print Composer*.

- (1) Go to *Project* → *New Print Composer* or press the New Composer icon  in the toolbar.
- (2) You will be prompted to enter a title for the composer e.g. ‘export_figure’. If left blank, a default name such as ‘Composer 1’ will be applied.

A new Print Composer opens up (see Figure 2.9). The main control buttons to add and modify maps can be found in the toolbar on the left site. You can also specify the layout and export resolution in the tab *Composition* on the right site.

After defining a proper layout a new map can be added:

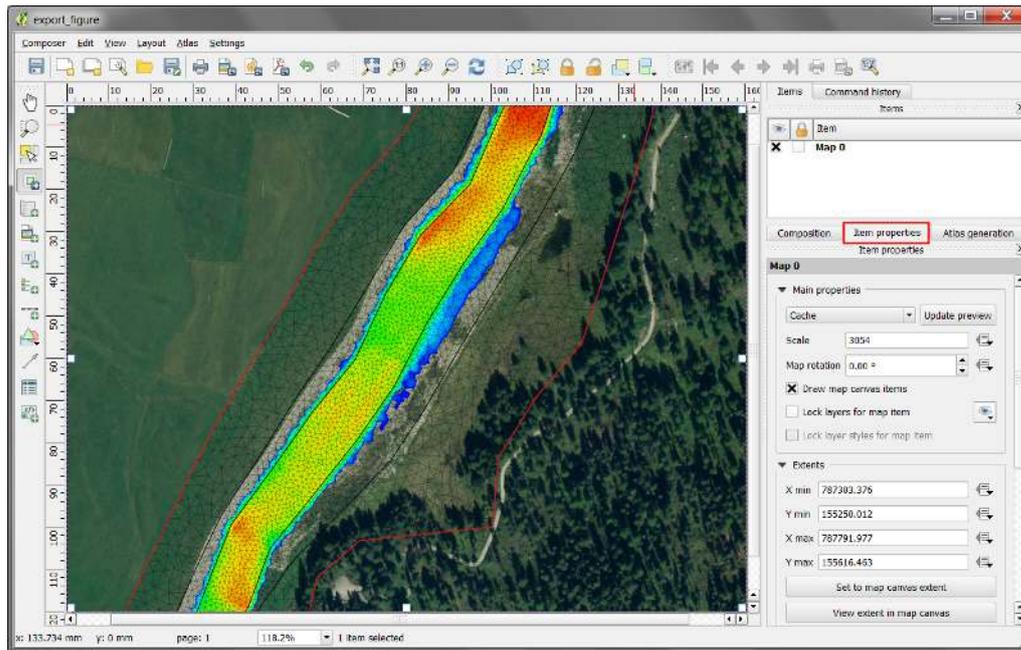


Figure 2.10 QGIS Print Composer with displayed Crayfish layer ‘depth’. Orthophoto: © 2016 swisstopo (JD100041)

- (3) Press the *Add new map button* , hold the left mouse button and drag a rectangle where you want to insert the map.

The rectangle window will be rendered with the currently active map from the main QGIS canvas. In this example the Crayfish *Quantity* ‘depth’ shall be display together with the background areal image, model boundary, breaklines and mesh elements (see Figure 2.10). The rendered map might not cover the desired extent of interest. To alter the displayed map extent and/or zoom:

- (4) Select *Move item content*  to pan the map in the window to desired position.
- (5) Once *Move item content* is active, you can change the map zoom using the mouse wheel.

The map is now listed as ‘Map 0’ in the *Item* dialogue box. Further map properties can be defined via the tab *Item properties*. Additionally, we want to add a legend to our map:

- (6) Press *Add new legend* , hold the left mouse button and drag a rectangle where you want to insert the legend.
- (7) The legend will appear as new *Item* ‘Legend’ in the Items dialogue box (see Figure 2.11). Go to *Item properties* in order to modify the legend entries.
- (8) Under *Legend items* uncheck/disable the *auto update* function. Now you can remove the legend items that shall not be displayed using the *minus button* . In this example everything but the ‘Flaz_mesh’ and the ‘model_boundary’ is removed. The order of the legend items can be changed with the *up*  and *down*  buttons.

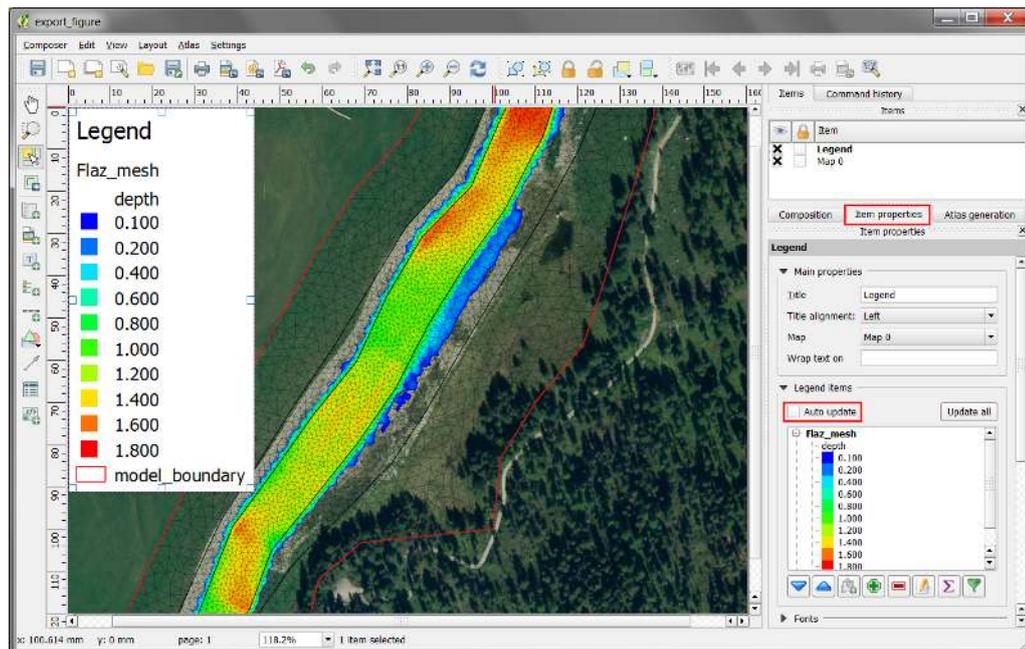


Figure 2.11 QGIS print composer with displayed Crayfish layer ‘depth’ after inserting the legend. Orthophoto: © 2016 swisstopo (JD100041)

So far no units of the Crayfish *Quantity* ‘depth’ are displayed in the legend. Furthermore, we want to add scale bar and a north arrow to our map:

- (9) Click on ‘depth’ in the *Legend items* dialogue box and press .
- (10) Now click on ‘Flaz_mesh’ and press *edit* .
- (11) Type ‘flow depth [m]’ and press *OK*
- (12) With the additional *Item properties* you can modify Fonts size and style of the legend entries as well as the size of the symbols or spacing of the columns and rows.
- (13) To insert a scale bar press the *Add new scalebar* button  and click to the map at the desired position.
- (14) A new *Item* ‘scale bar’ appears in the Items dialogue box and again properties of the scale bar can be modified via the *Item properties* tab.
- (15) To insert a north arrow, press the *Add image* button  and click to the desired location.
- (16) A new *Item* ‘picture’ is listed in the Items dialogue box.
- (17) Under *Search directories* you will find a preview of different arrow styles offered by QGIS, it is also possible to *Add* and *Remove Image search paths* manually.

If everything went well, your Print Composer should now look like shown in Figure 2.12. The map is now ready to be exported or printed. You can add additional features like text

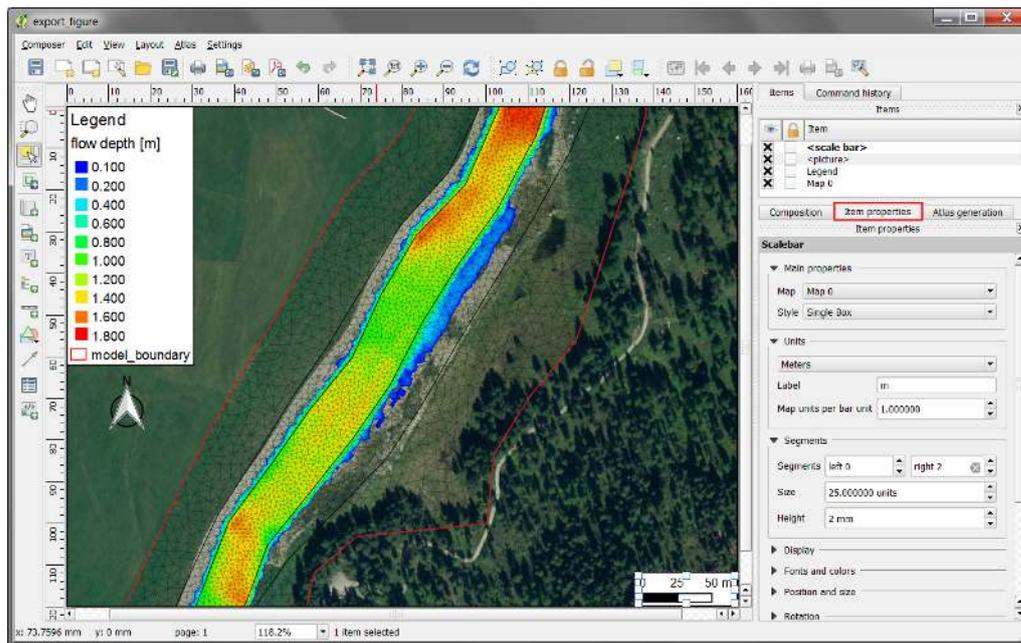


Figure 2.12 QGIS print composer with displayed Crayfish layer ‘depth’ after inserting legend, scalebar and north arrow. Orthophoto: © 2016 swisstopo (JD100041)

or images to the map at your convenience. Just use the *Add image*  or *Add new label*  buttons. You can save your map properties as template for further QGIS projects.

(18) Go to *Composer* → *Save as Template* or use the button  in the main toolbar.

In general the Print Composer is stored together with the main QGIS Project. If you open the project which you defined a Print Composer in, you will always be able to access it via the main QGIS canvas.

(19) Go to *Project* → *Composer manager* where you can find a table of all print composers defined within the main QGIS Project.

For the export of your map several formats are available in the Print Composer.

(20) Go to *Composer* → *export as Image* to export the map as raster image e.g. png, jpeg, tif,...

(21) Alternatively chose *export as PDF* or *export as SVG* to save the map in vector format.

Figure 2.13 shows the final map export. In addition to the steps described above, the Legend Title was replaced by an image of the software BASEMENT.

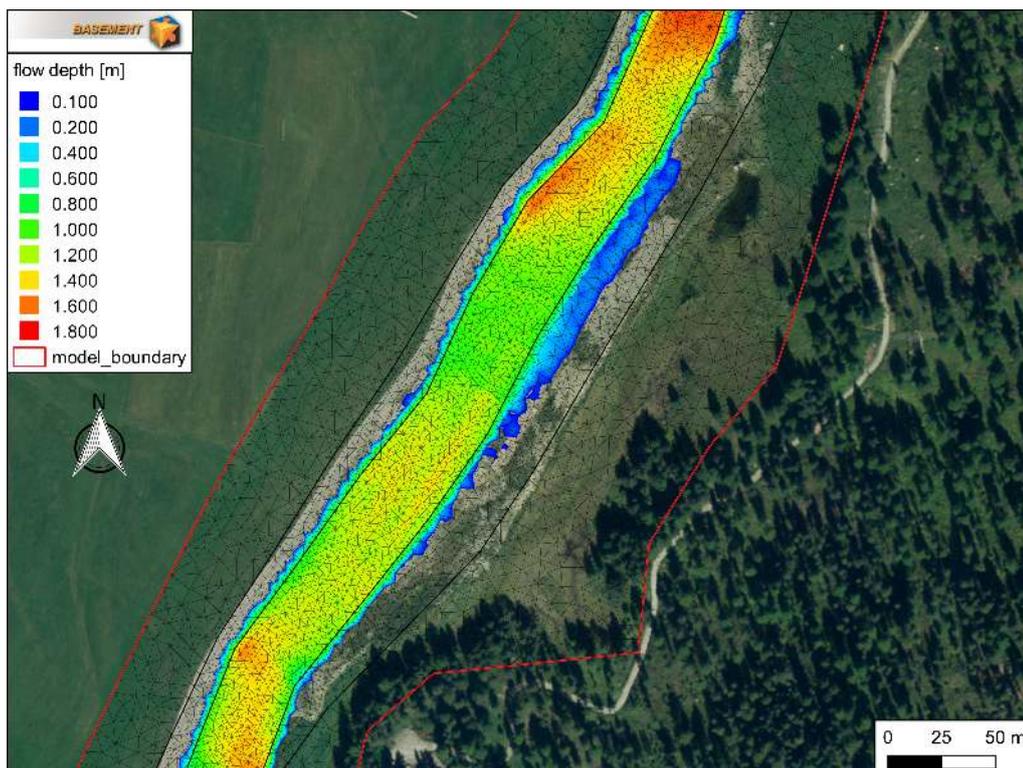


Figure 2.13 Maximum flow depth at the river Flaz during the unsteady flow simulation. Orthophoto: © 2016 swisstopo (JD100041).

2.2.5.2 Export animation

If your simulation results are time dependent like those provided in this tutorial, you might not just want to export selected time steps as maps, but rather create an animation over the whole run time of the simulation. Before exporting your time depending Crayfish Quantity, you have to define the layout of your animation in the QGIS Print Composer. Open a new Print Composer as described in Section 2.2.5.1 and add a new map. In the example shown in Figure 2.14, the same layers like in Section 2.2.5.1 are chosen for visualization. In the tab *Composition* the *Presets* is set to A4 format. The map zoom is changed, so that the whole model domain is visible for the animation. Again legend, scalebar and north arrow are added as described beforehand.

In order to display the runtime of the simulation some additional steps are required:

- (22) Press *Add new label*  and add a text field below the Legend.
- (23) Go to *Item properties* and type 'run time:' instead of the default text 'QGIS' into the *Main properties* field.
- (24) Define *Frame* and *Background* as desired.
- (25) Again press *Add new label* and insert a second text field like shown in Figure 2.15.
- (26) Don't change the default expression 'QGIS' in the *Main properties* field.
- (27) Go down to *Item ID* and type 'time' into the *ID* field

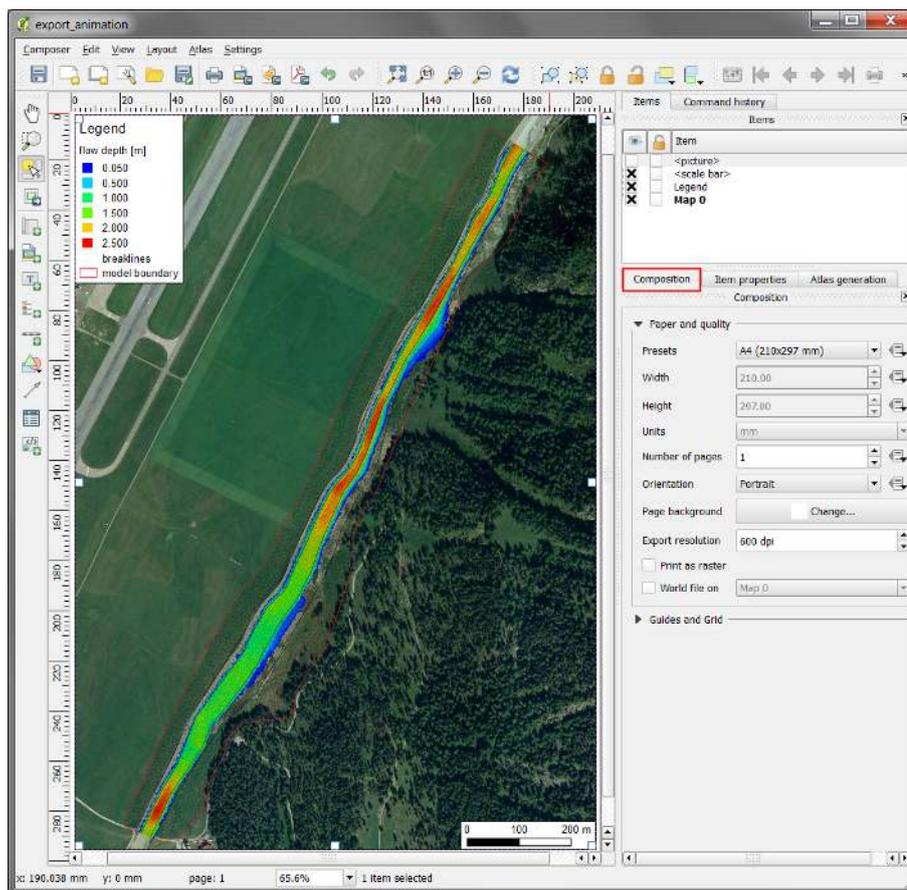


Figure 2.14 QGIS print composer with displayed Crayfish layer 'depth' zoom to the full extent of the model for export as animation. Orthophoto: © 2016 swisstopo (JD100041)

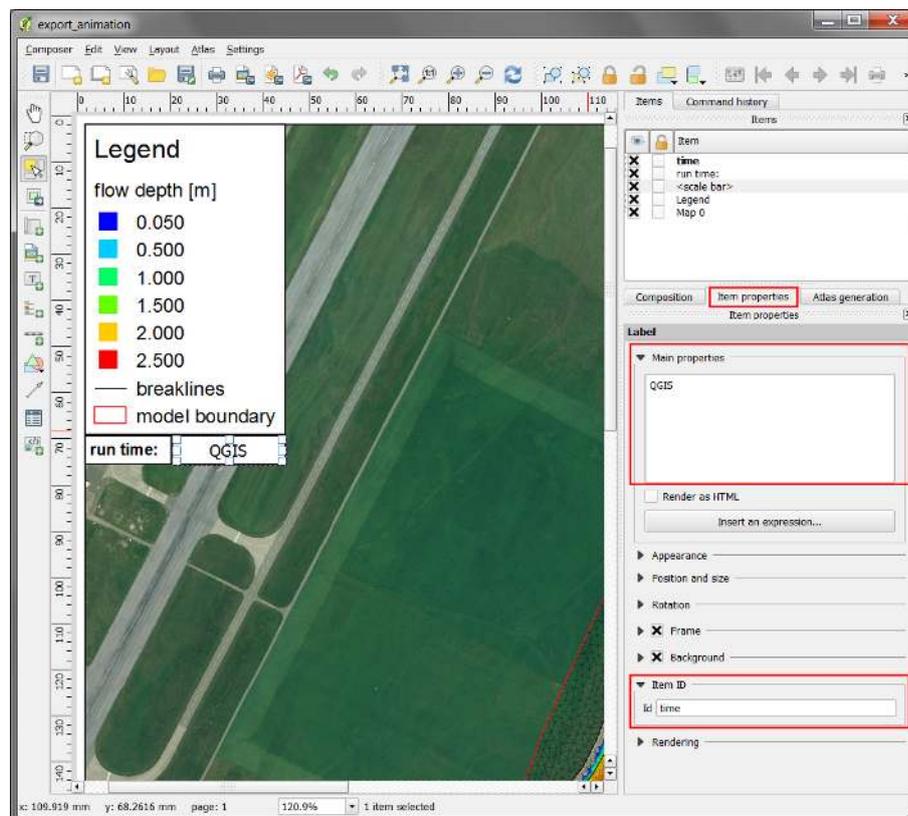


Figure 2.15 Additional label to display the simulation run time during the animation.
Orthophoto: © 2016 swisstopo (JD100041)

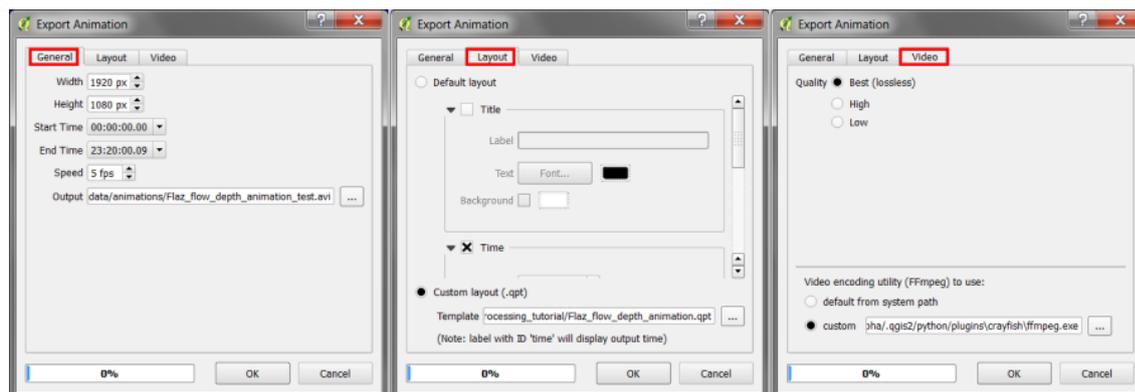


Figure 2.16 QGIS Export Animation window

Now the text field is linked to the simulation runtime of the Crayfish *Quantity* and the runtime will be displayed during the animation.

(28) Go to *Composer* → *Save as Template* or use the button  in the main toolbar.

The customized layout of the animation is saved as QGIS Composer template (*.qpt) and will be used during the animation export.

The final export of the animation is done from the QGIS main canvas:

- (29) Right click to the Crayfish *Quantity* ‘depth’, select *Export animation...* and a new window will appear (see Figure 2.16).
- (30) In the tab *General* you have to specify a output directory and you are able to adjust the Speed of the simulation (fps=frames per second).
- (31) In the tab *Layout* chose *Custom layout (.qpt)* and browse to the location where you saved the Print Composer template (step 56).
- (32) In the tab *Video* you can chose the Quality of the animation. The choice will affect the time to generate the animation as well as the resulting file size.
- (33) Under *Video encoding utility (FFmpeg) to use:* chose custom and press *OK*.

A window will pop up saying that “The tool for video creation (FFmpeg) is missing ...” (Figure 2.17). Follow the instructions depending on your system (Windows or Linux).

- (34) Finally press *OK*. QGIS will inform you that the export of the simulation was successful.

2.3 3D result visualization with Paraview

2.3.1 Input data

The data needed to complete this tutorial comes as ZIP-file and needs to be extracted to a location of your choice. All screenshots and figures in this document were generated

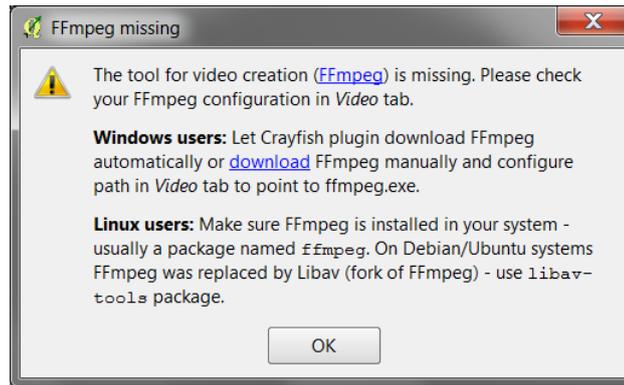


Figure 2.17 QGIS Export Animation FFmpeg

with ParaView version 4.3. ParaView is able to handle result-files of type *node_centered* as well as *element_centered*. Be aware that most of the hydraulic values (e.g. depth, velocity, wse,...) are calculated on the elements during simulation, whereas morphological values (e.g. deltaz) are calculated on the nodes. Therefore, the results are interpolated by BASEMENT either to the nodes or element centres depending on the chosen output *format*.

The result files (*.vtk) used for visualization in this tutorial come from the River Flaz. In the *SPECIAL_OUTPUT* Block of the BASEMENT command file select *format* 'vtk' to generate result files that can be visualized with ParaView (see Figure 2.18).

We want to use the scenario defined in tutorial 3 of the Pre-Processing tutorial with the inner boundary of type weir (see Section 1.4). You can either use your own results or visualize those provided for this tutorial. Note that the provided results were computed using a computational mesh with a higher spatial resolution than the one generated in the Pre-Processing tutorial.

2.3.2 About Paraview

ParaView is an open source application for visualizing 2- and 3-dimensional data sets. The size of the data sets ParaView can handle varies widely depending on the architecture on which the application is running. The platforms supported by ParaView range from single-processor workstations to multiple-processor distributed-memory supercomputers or workstation clusters. Using a parallel machine, ParaView can process very large data sets in parallel and later collect the results.

More Information

In this tutorial, only a very small part of the enormous functionality of ParaView will be covered. The documentation as well as a number of tutorials is available online as [Public Wiki](#). ParaView also offers an online help that can be accessed by clicking the help button



in the application.

2.3.3 Installation

To use ParaView on your own computer simply go to the [Download page](#) of ParaView. The software is available for Windows, Linux and Mac.

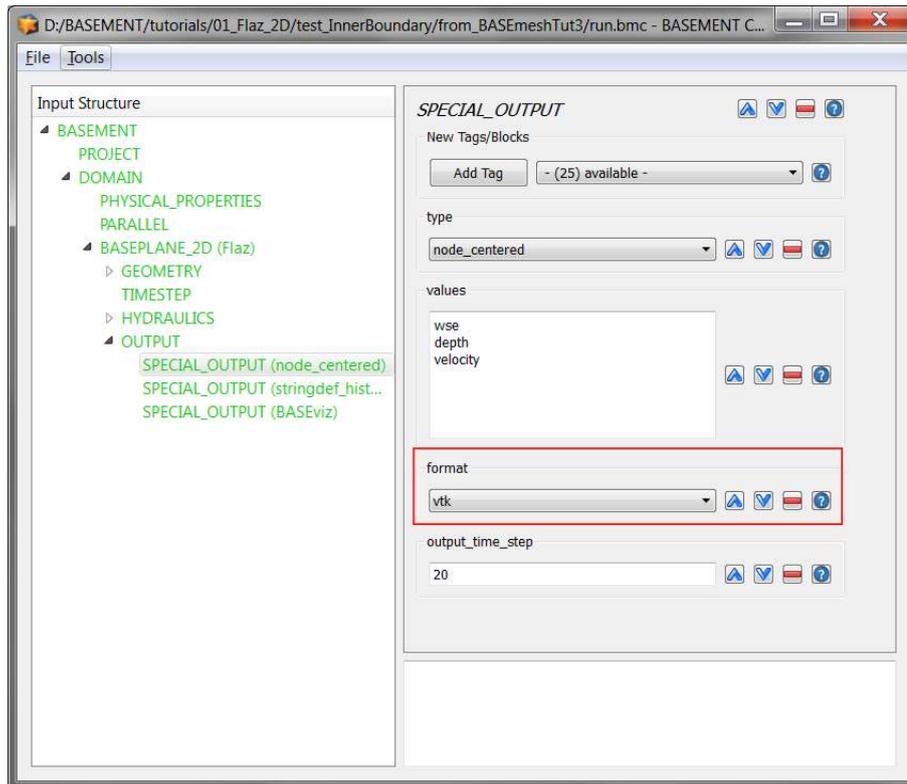


Figure 2.18 Settings defined in the `SPECIAL_OUTPUT` Block of BASEMENT to generate time dependend result-files (*.vtk) of a 2D simulation

2.3.4 User Interface

The layout shown in Figure 2.20 is the default layout given when ParaView is first started. The GUI comprises the following components:

Menu Bar As with any other program, the menu bar allows you to access the majority of the features.

Toolbars The toolbars provide quick access to the most commonly used features within ParaView.

Pipeline Browser ParaView manages the reading and filtering of data with a pipeline. The pipeline browser allows you to view the pipeline structure and select pipeline objects for visualization. To toggle visibility of single objects, click on the ‘eye-symbol’.

Properties Panel The properties panel allows you to view and change the parameters of the current pipeline object. Use the *Toggle advanced properties* button to show and hide advanced options. The properties are by default coupled with an **Information** tab that shows a basic summary of the data produced by the pipeline object.

3D View The remainder of the GUI is used to present the data so that you may view, interact with and explore your data. This area is initially populated with a 3D view that will provide a geometrical representation of the data.

Note that the GUI layout is highly configurable. To toggle the use of a toolbar, go to *View* → *Toolbars* and check/uncheck the available options.

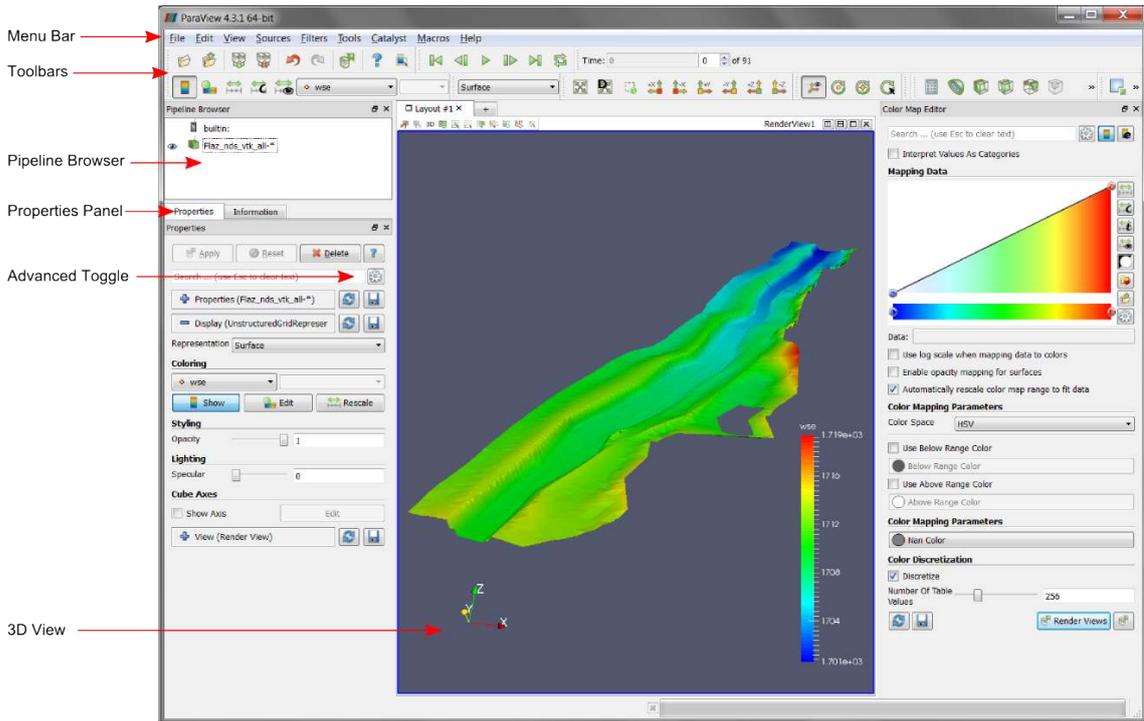


Figure 2.19 ParaView GUI

2.3.5 Import Data

The provided results are stored in the subfolder ‘ParaView’. Although the folder contains 91 vtk-file representing 91 output timesteps, we can load the results as one single data package:

- (1) Go to *File* → *Open* or use the *Open* button  in the toolbar.
- (2) Browse to the ‘ParaView’ folder, select the file ‘Flaz_nds_vtk_all..vtk’ and press *OK*.

Note that opening a file is a two step process in ParaView, so you don’t see any data yet.

- (3) In the *Properties Panel*, click on *Apply* to load the data.

Now a 3D preview of the domain Flaz should be rendered to the 3D view. From the dropdown in the toolbar, we can select, which quantity shall be visualized. Alternatively, the quantities can also be selected under *Coloring* in the *Properties* panel. In this example the quantities ‘depth’, ‘velocity’ and ‘wse’ are available (see Figure 2.20). With the option ‘Solid Color’, just the geometry of the domain without any time depending quantity can be displayed. To visualize the mesh elements, select ‘Surface with edges’ instead of ‘Surface’ from the dropdown menu. The color of the mesh elements can be modified under *Edge Styling*, which is enabled by toggeling the advanced properties (see Figure 2.20).

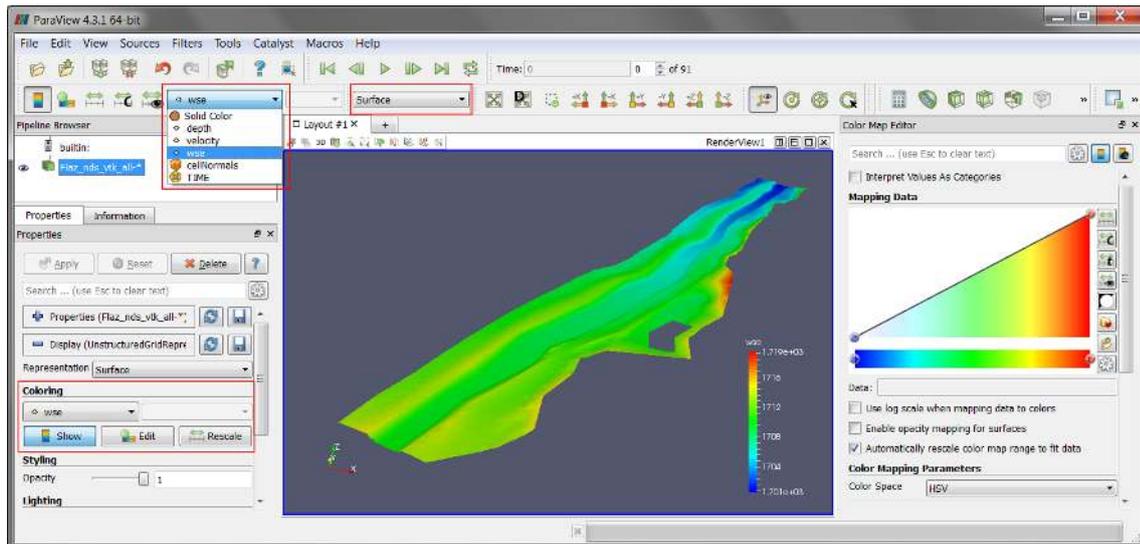


Figure 2.20 ParaView GUI with 3D view of the domain Flaz

2.3.6 ParaView Filters

2.3.6.1 Overview

In ParaView a large number of so called *Filters* can be applied. *Filters* are functional units that process the data in order to generate, extract or derive features from the data. Here are some of the most common, which are available by clicking the respective icon in the filters toolbar.

Calculator  Evaluates a user-defined expression on a per-point or cell basis.

Contour  Extracts the points, curves, or surfaces where a scalar field is equal to a user-defined value. This surface is often also called an isosurface.

Clip  Intersects the geometry with a half space. The effect is to remove all the geometry on one side of a user-defined plane.

Slice  Intersects the geometry with a plane. The effect is similar to clipping except that all that remains is the geometry where the plane is located.

Threshold  Extracts cells that lie within a specified range of a scalar field.

Extract Subset  Extracts a subset of a grid by defining either a volume of interest or a sampling rate.

Glyph  Places a glyph, a simple shape, on each point of the mesh. The glyphs may be oriented by a vektor and scaled by a vector or scalar. With this *Filter* it is possible to visualize velocity vector for example.

For further information about available *Filters*, please refer to the [documentation](#) of ParaView.

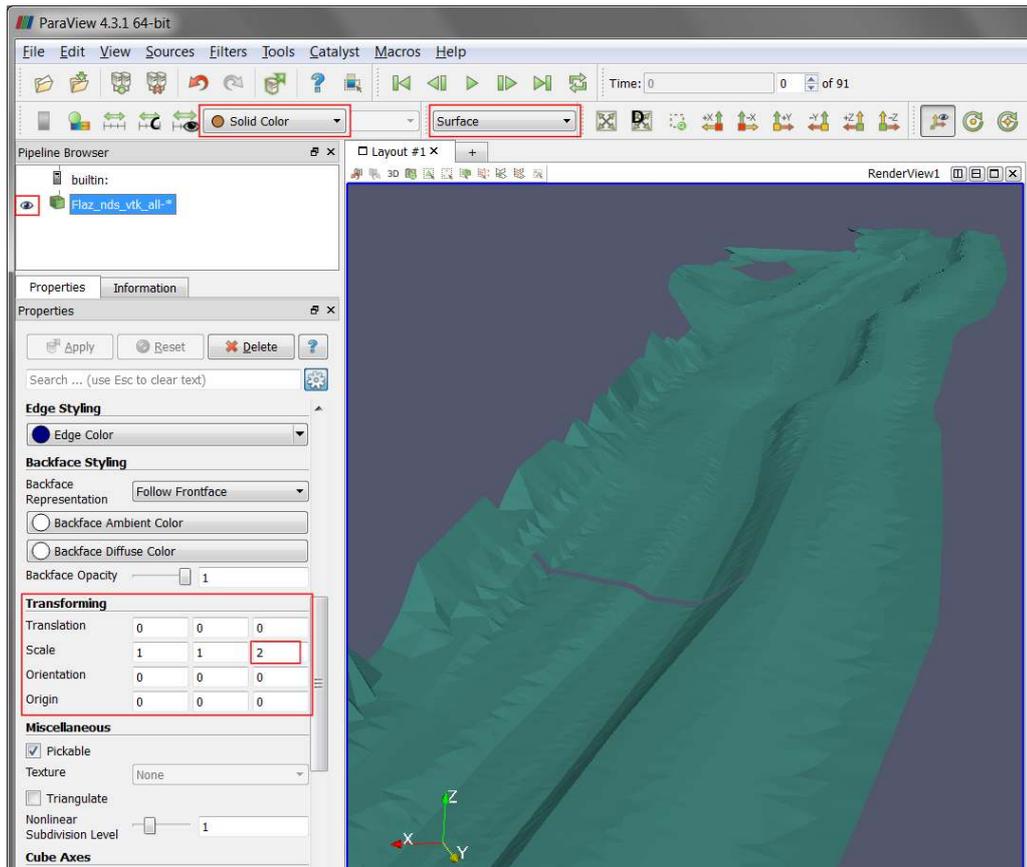


Figure 2.21 Vertical exaggeration of solid mesh

2.3.6.2 Applying Filters to data

After introducing the basic concept of *Filters*, we are now ready to apply three different *Filters* to our dataset. The goal is, to generate a 3D view of the time depending water surface elevation (wse) superimposed on the mesh geometry. First we want to display the geometry with vertical exaggeration for better contrast.

- (1) In the toolbar chose ‘Solid Color’ and ‘Surface’ from the dropdown menu (see Figure 2.21)
- (2) Go to *Transforming* in the *Properties* panel (advanced properties) and alter the *Scale* in the third column (z-direction) from 1 to 2 and press Enter.
- (3) Reload the 3D view by clicking on the eye symbol in the *Pipeline Browser*.

Filter: Warp by Scalar

This *Filter* moves point coordinates along a vector by a distance determined by a user defined scalar for the active data set.

- (4) Select the data set ‘Flaz_nds_vtk_all-*’ from the *Pipeline Browser*.
- (5) In the menu bar go to *Filters* → *Alphabetical* and select *Warp By Scalar*.

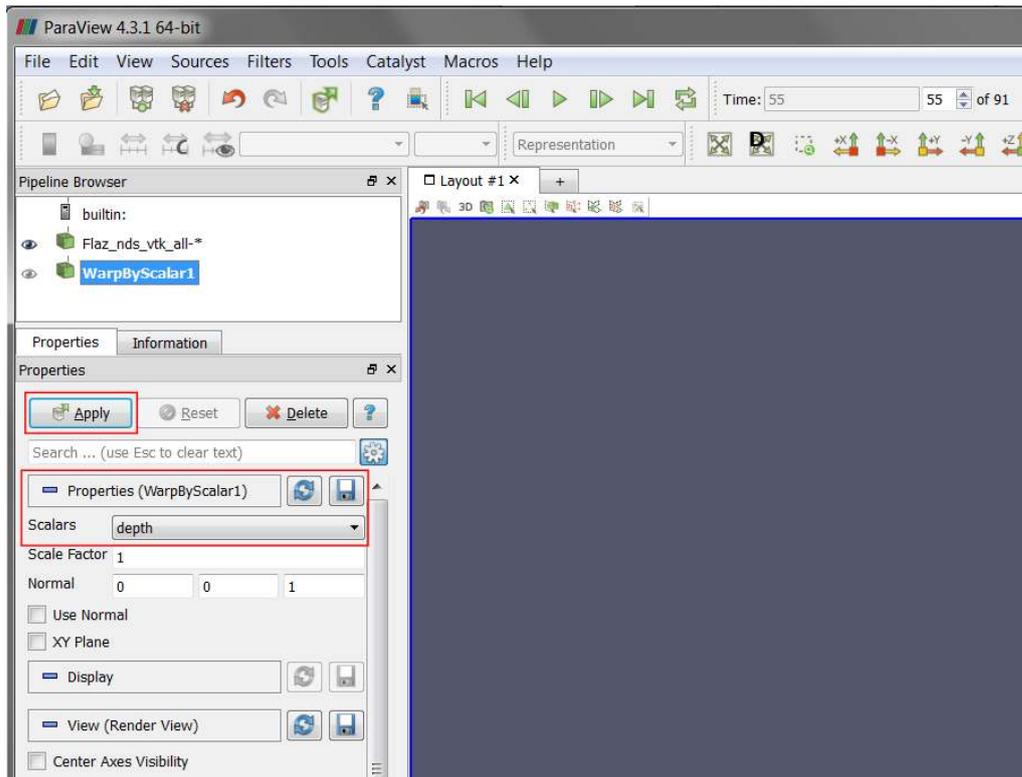


Figure 2.22 Filter, Warp by Scalar

- (6) In the *Properties* panel chose 'depth' as *Scalars* and press *Apply* (see Figure 2.22)
- (7) Select the data set 'WarpByScalar1' from the *Pipeline Browser*.
- (8) In the toolbar chose 'depth' and 'Surface' from the dropdown menu (see Figure 2.23)
- (9) Again alter the *Scale* in z-direction from 1 to 2.

Filter: Threshold

Applying a *Threshold Filter*, we are able to blank out elements with a water depth below a certain threshold:

- (10) Select the data set 'WarpByScalar1' from the *Pipeline Browser*.
- (11) In the menu bar go to *Filters* → *Alphabetical* and select *Threshold* or directly access the *Filter* pressing the respective button  in the toolbar.
- (12) In the *Properties* panel for 'Threshold1' chose 'depth' as *Scalars* and specify the *Minimum* and *Maximum* as shown in Figure 2.24, than press *Apply*.
- (13) Repeat the steps 1-3 for the layer 'Threshold1' in order to adjust the coloring and the vertical exaggeration.

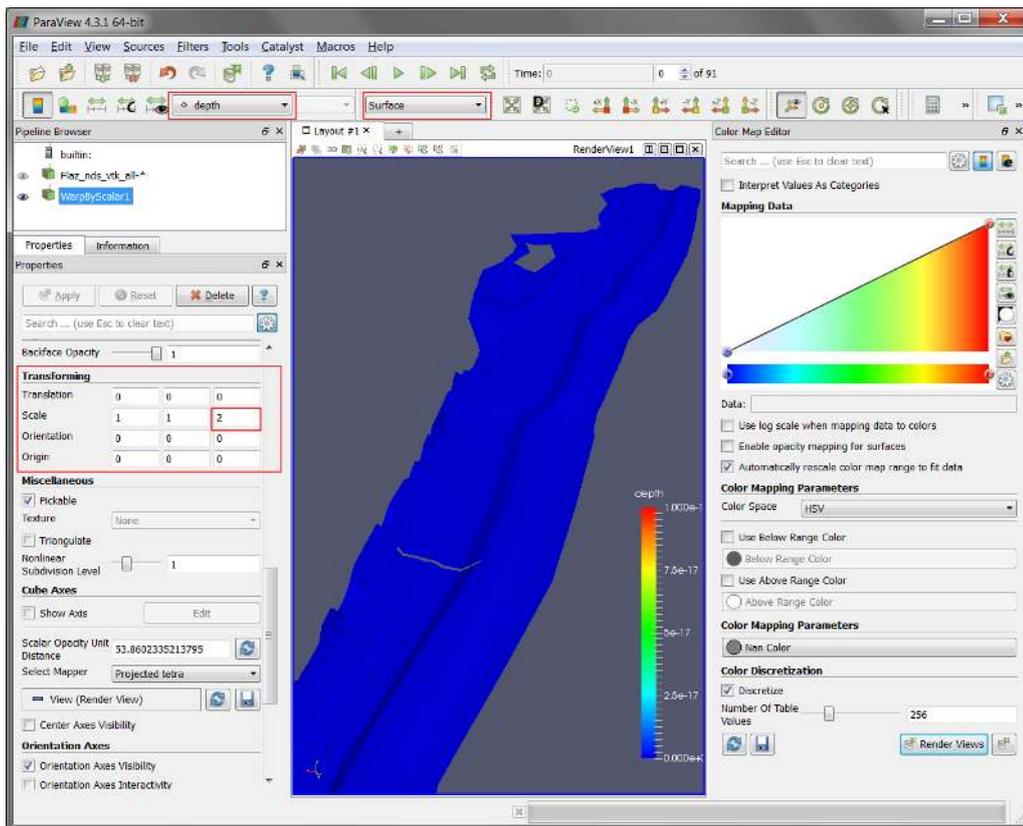


Figure 2.23 Warp by Scalar, properties

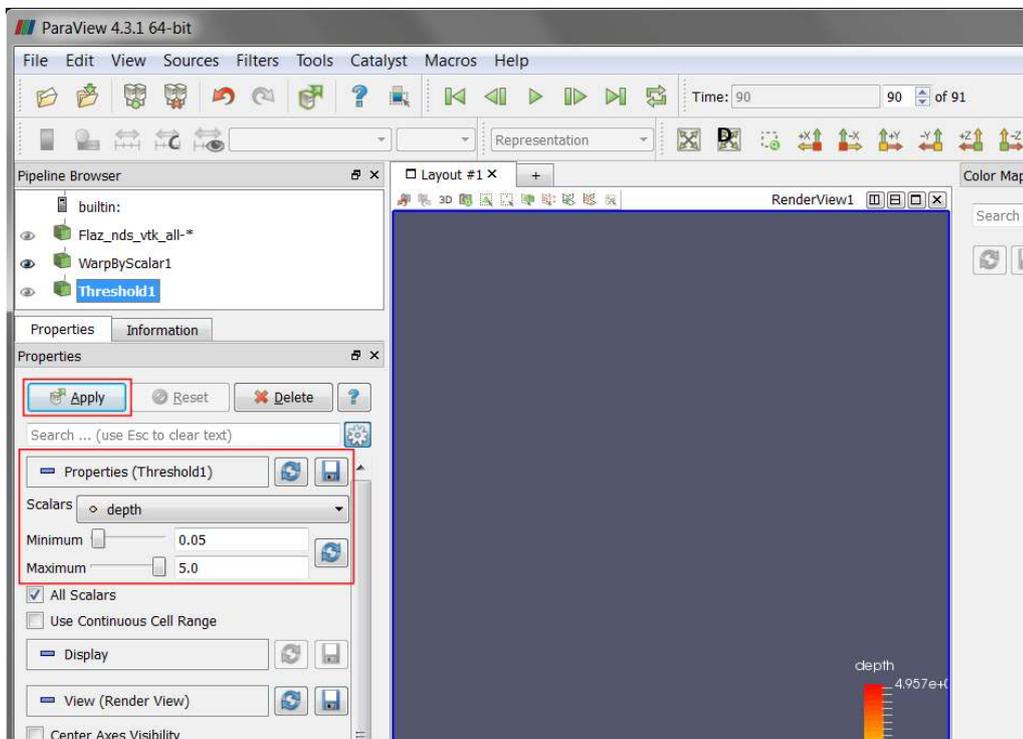


Figure 2.24 Filter, Threshold

- (14) Toggle visibility of the layers ‘Flaz_nds_vtk_all-*’ and ‘Threshold1’ in the *Pipeline Browser*.

Annotate Time Filter

So far our 3D preview of the simulation results is missing a time annotation, which shows the current simulation runtime of the selected output timestep. To add this information:

- (15) Select the data set ‘Threshold1’ from the *Pipeline Browser*.
- (16) In the menu bar go to *Filters* → *Alphabetical* and select *Annotate Time Filter*
- (17) Specify the *Properties* like shown in the *Properties* panel of Figure 2.25 and press *Apply*.

Note that the definition of *Format*, *Shift* and *Scale* is highly dependent on the data you want to visualize. For the provided data a runtime of 1800 seconds and an output timestep of 20 seconds have been defined in the *SPECIAL_OUTPUT* Block of the BASEMENT command file. This leads to 90 time steps you can browse through with the time control buttons in the toolbar (see Figure 2.25).

- Format %4.0f: Double with a width of 4 characters and a precision of 0 decimal places.
- Shift 0: Time annotation starts at 0.
- Scale 20: The timespan between the single output steps in 20 seconds.

If everything went well, your 3D view should no longer look like Figure 2.26. At the beginning (Time: 0) only the geometry is visible due to the initial condition ‘dry’. From time step 1 to 27 water enters the domain at the upstream boundary and a backwater profile appears upstream of the inner weir boundary. After time step 28 the weir is overtopped and water is visible also downstream of the weir boundary.

2.3.7 Exporting figures and animations

One of the most important products of any visualization is screenshots and movies that can be used for reports and presentations.

2.3.7.1 Save Screenshot (picture)

- (1) Go to *File* → *Save Screenshot* and a new window with several controls will appear. (see Figure 2.27)

The *Select resolution for the image to save* entries allows you to create an image that is larger (or smaller) than the current size of the 3D view.

- (2) Press *OK* and specify a location for the file to be exported. From the dropdown you can select several file types. It is recommended to save images as PNG file though.

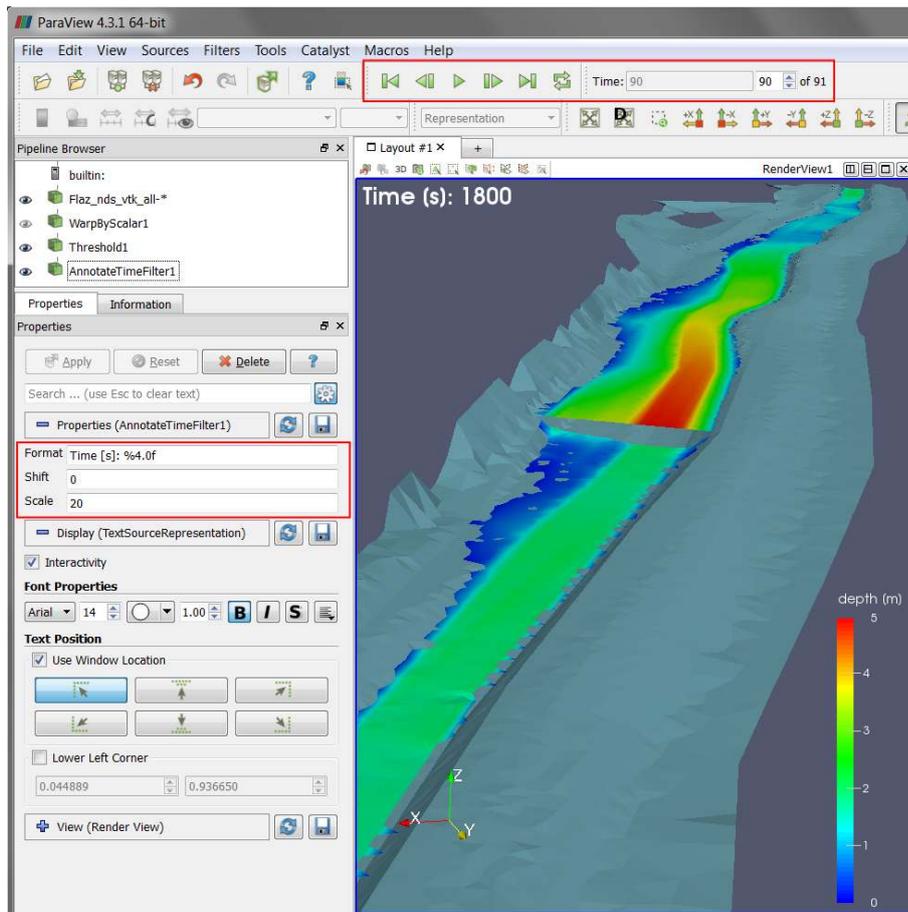


Figure 2.25 Annotation Time Filter

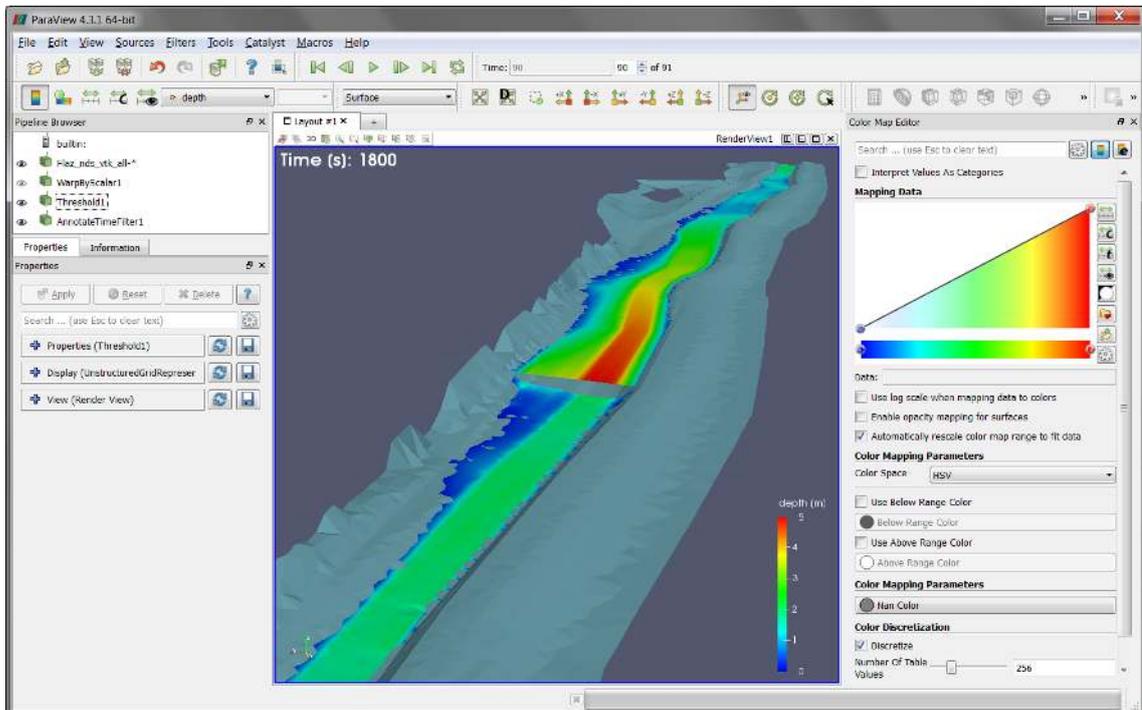


Figure 2.26 Water depth at the steady state of the simulation

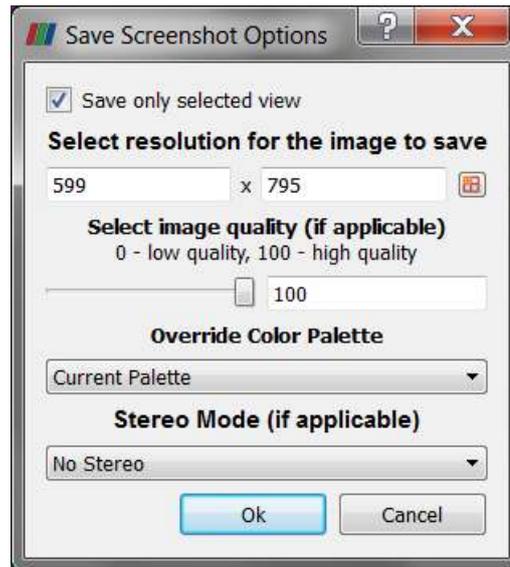


Figure 2.27 Export Screenshot

The save screenshot option saves the image as a raster graphic, which is natural for rendered images and very efficient for large data sets. However, elements like text and other labels often become disfigured. These elements are better represented as vector graphics, which use geometrical primitives to draw the shapes. To create a vector graphic:

- (3) Go to *File* → *Export Scene* and specify an output location. Several file types are available e.g. *.pdf or *.svg.

2.3.7.2 Save Animation (movie)

- (1) Go to *File* → *Save Animation* and a new window with several controls will appear. (see Figure 2.28)

Again, you can create an animation that is larger (or smaller) than the current size of the 3D view modifying the *Resolution (pixels)* entries. It is advisable to reduce the *Frame Rate (fps)* from 15 to 8, otherwise the animation might run to fast.

- (2) Press *Save Animation* and specify an output location.
- (3) Select AVI files (*.avi) as file type and press *OK*.

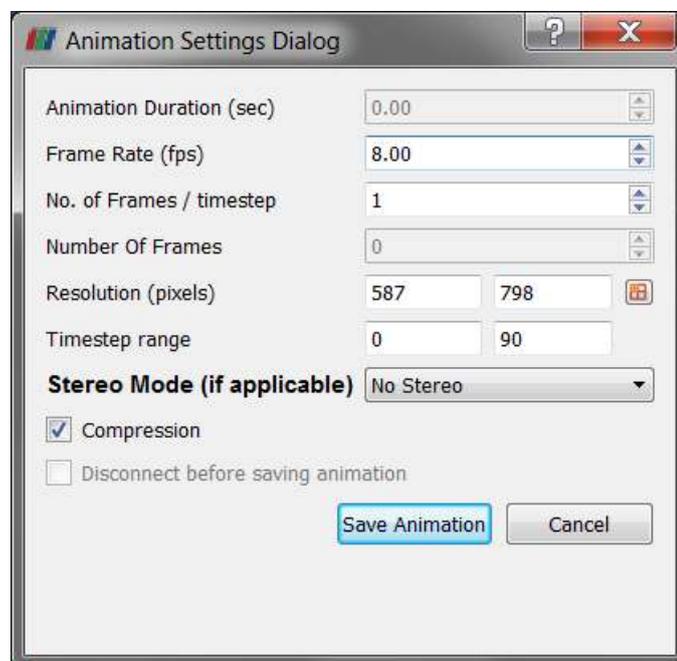


Figure 2.28 Export Animation

3

Hydrodynamics and sediment transport at the river Thur (1D)

3.1 Introduction

This tutorial gives an introduction to the capabilities of the 1D modelling module BASEchain of BASEMENT. It provides a step-by-step guidance on how build up a model for BASEchain.

3.1.1 General description

This Tutorial describes the necessary steps for the simulation of hydrodynamics and bed load in a specific section of the river Thur. In the considered section a river widening has been realized during the last years. It's located in Altikon and illustrated in the figure below. The flow direction is from right to left. The bed modification over a year including an important flood will be simulated.



Figure 3.1 View of the simulated river section

3.1.2 Used features

In this tutorial the following points will be treated:

- Preparation of the needed input files;
- Simulation of a steady flow to use for the following simulations;
- Use of composite cross sections;
- Simulation of bed load with formula of Meyer-Peter Müller;
- Use of dry initial condition;
- Use of a file to define the initial conditions;
- Use of the following boundary conditions:
 - Inflow hydrograph
 - Inflow of sediment in/out
 - Outflow h-q relation
 - Outflow of sediment in/out
- Representation of the results.

3.1.3 Purpose

In the year 2005, intensive rainfall led to a large flood event. The aim of the simulations in this tutorial is to study, which influence this flood had on the channel geometry of the river Thur.

3.2 Setting up the topography file

3.2.1 Cross sections

The data of the topography are available in the form of cross section measurements, where each measured point is given by its x, y and z coordinates. This is an extract from the raw data:

x	y	z
698578.504	272450.223	376.841
698578.494	272446.999	374.991
698578.32	272444.286	373.748
698577.929	272441.889	373.229
698578.081	272439.244	372.207
698578.533	272437.229	371.544
698578.56	272434.366	370.869
698578.612	272431.522	370.766
698578.86	272429.064	370.401
698579.201	272426.526	370.388
698579.323	272424.56	370.617

x	y	z
698578.937	272422.91	370.341
698579.208	272421.645	370.436
...		

These data have to be separated into groups belonging to one cross section and then transformed in a way to have a $z(y)$ relation, where the smallest y is the extreme point on the left river side.

Example:

y	z
0	376.264
1.455081097	376.327
4.349044033	377.804
5.134094857	378.133
6.803278107	378.238
8.241452785	378.227
9.123103693	377.965
10.23346129	377.395
11.41786604	376.664
12.57016281	376.221
14.53240603	376.21
16.34176138	376.215
17.09961488	375.99
18.74552432	375.296
...	

Additionally the distance from the upstream end of the channel (first cross section) has to be determined for each cross section.

The obtained geometry points can be introduced in the topography by copy pasting it directly into the node coordinates field in the grid file editor (Figure 3.2). The editor will translate the column wise data into the proper syntax. Another, more efficient way is to use the python scripts available on <http://www.basement.ethz.ch> to transform topography data in excel format into the BASEMENT format in a first step.

The minimum information we have to provide for each cross section besides the node coordinates are the cross section name and the distance coordinate measured from upstream to downstream in km.

3.2.2 Definition of different cross section zones

To reduce some drawbacks of the 1-D simulation, in the present case it is useful to define a main channel and flood plains, as well as the bed bottom which is limiting the bed load transport.

In Figure 3.3, the flood plains are given by the part of the cross section not defined as the main channel. The soils by their indexes. The keys 2 or 1 refer to the type of soil which is

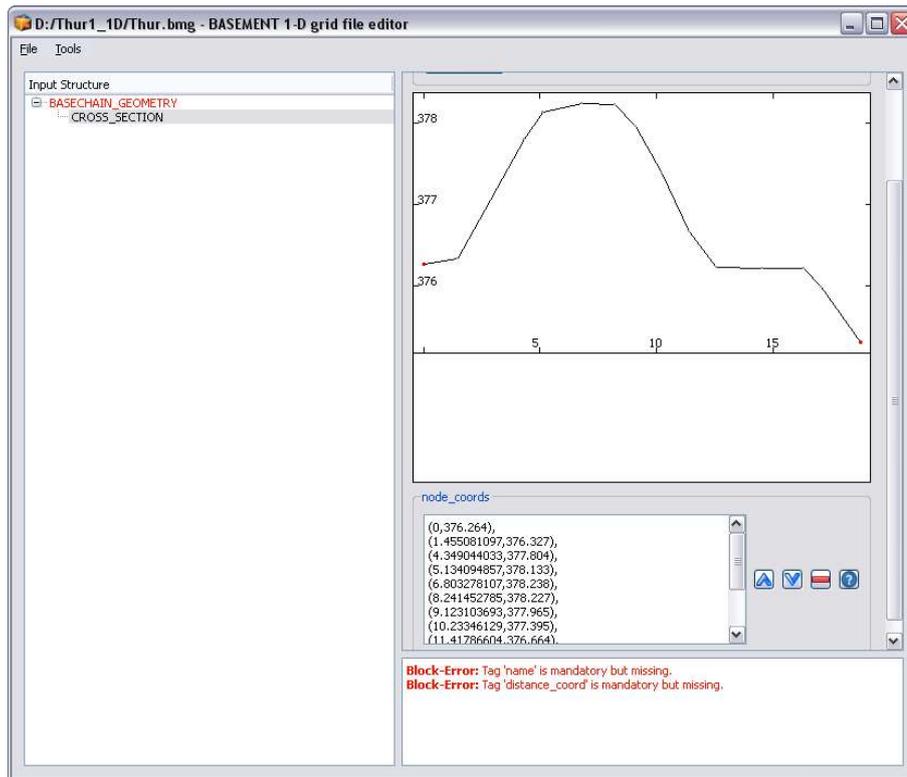


Figure 3.2 Cross section points inserted in the topography editor

defined later in the command file. Here we use only one soil for the whole bottom, but it is also possible to add several soils of the same type or of different types as shown in Figure 3.3. Further, different friction values can be defined for different parts of the cross section. The active range should span from the left to the right dike. Points outside the active range are simply ignored (Figure 3.4).

The graphical view of the cross section data helps to identify the correct point and set the ranges to the correct lateral node coordinates. For convenience, one can switch into the text editor mode of the input file by choosing from the *Tools* Menu the option *Edit Raw*.

3.2.3 Friction values

For the friction determination the Strickler approach is used. This is declared in the command file by setting the *type* in the FRICTION block within the HYDRAULICS section to *strickler*. In this case, Strickler k-values have to be defined for the different regions. The banks of the main channel are partially covered with small bushes. The flood plains are covered with grass, stones and sand, but there are also zones with trees.

The following k_{St} values are used:

- Banks of main channel: $k_{St} = 35 \text{ [m}^{1/3}/\text{s]}$
- Flood plains: $k_{St} = 33 \text{ [m}^{1/3}/\text{s]}$.

For the bed bottom the following transformation, based on the grain characteristics of the sediment is used ($d_{90} = 5 \text{ cm}$):



Figure 3.5 Definition of table values

$$\text{Bottom: } k_{Str} = \frac{23}{\sqrt[6]{d_{90}}} = 38 \text{ [m}^{1/3}\text{/s]}.$$

In BASEMENT internally, the cross section is represented by slices, defined by the segment between two nodes. Each slice has its own properties. Therefore we have to provide so called ranges to assign the friction values to the respective slices. The ranges can be defined either referring to node coordinates (note that you have to match the coordinates exactly) or by referring to slice indexes, starting at index 1 from left to right.

3.2.4 Computation of water surface elevation

As it is much quicker, the use of tables for the computation of the water surface elevation and other hydraulic variables is chosen for this example. In the case of tables, all properties are pre-computed for a given set of points and only updated in case of a non-negligible change of the soil. This is accomplished using the block `SECTION_COMPUTATION` in `PARAMETER`. As all variables are calculated for several water surface elevations, the maximum and minimum intervals between the different levels have to be set accordingly. The default spacing is given by `max_interval - min_interval`. Whenever the bed changes, the table is updated accordingly.

3.2.5 Characterisation of the sediments

Two types of ground will be defined:

1. the ground is not erodible;
2. the ground is composed by sediments with a mean diameter of 2.5 cm.

In the topography file the codes of the different types are assigned to different cross sections. A code can be set in the cross sections by creating a new sub block `SOIL_DEF`, where the index is assigned to the respective soil index in the command file and the span of the soil is defined via the range it extends.



Figure 3.6 Declaration of soil types

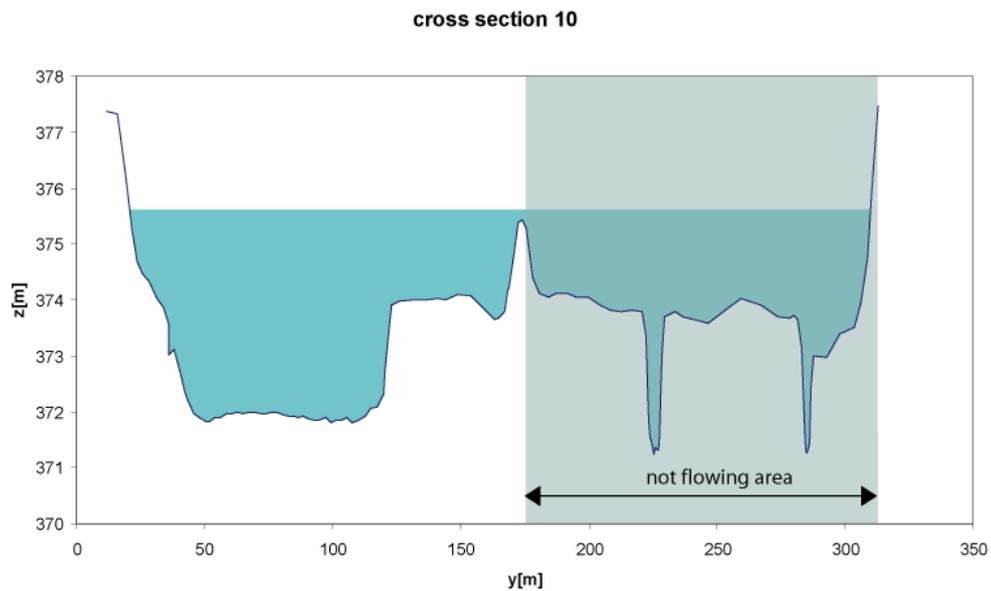


Figure 3.7 Definition of non flowing areas

3.2.6 Define flowing zones

The 1-D model considers the flow velocity to be the same over the whole width of the cross section: This is obviously not true, especially for cross sections where important zones are behind a sort of dike, like it occurs very often at the Thur. This effect has an important influence on the bed load transport. For this reason, regions where the water does not flow are declared using the `water_flow_range` tag. The next figure shows an example of a cross section with the different zones. Of course, we only mention this here. The tutorial topology already contains the required ranges.

3.3 Setting up the command file

3.3.1 Project

The first command file is called “run.bmc”. The first step is to define a project by its name, the author and the date:

```
PROJECT
{
  title = Thur
  author = rm
  date = 10.8.2006
}
```

3.3.2 Domain

A domain is defined by including all relevant parameters for this computation. The first parameter is the name of the computation region.

```
DOMAIN
{
  multiregion = Thur
  [...]
}
```

3.3.3 Define the physical properties

The Physical properties normally do not change from one project to another.

```
PHYSICAL_PROPERTIES
{
  gravity = 9.81
  viscosity = 0.000001004
  rho_fluid = 1000
}
```

3.3.4 One dimensional simulation

The next step is to declare a *BASECHAIN_1D* block. This will make the program execute a 1-D simulation. The name of the computational region is given here.

```
BASECHAIN_1D
{
  region_name = Thur
}
```

3.3.4.1 Define the geometry

The next block defines in which file the topography is stored and the type of geometry file used. The cross section names are listed from upstream to downstream.

```
GEOMETRY
{
```

```

    type = basement
    file = ThurTopo.bmg
    cross_section_order = ( CS1 CS2 CS3...CS54 CS55 )
}

```

3.3.4.2 Define hydraulic information

All information concerning the hydraulic simulation is declared in the block *HYDRAULICS*.

```

HYDRAULICS
{
    [...]
}

```

3.3.4.2.1 Define the upper boundary condition

The upper boundary condition is defined by a hydrograph, which is stored in a separate file. Indications are given about the precision required between the discharge corresponding to the iteratively determined area and the given discharge, as well as the maximum number of iterations allowed to reach this precision. The slope of the first cross section must be given in per mil (the last 3 values are used only in case of supercritical flow).

Create a hydrograph file named *ThurSteadyHydrograph.txt*:

```

// T      Q
0         30
100000   30

```

Then, add the upper boundary block:

```

BOUNDARY
{
    type = hydrograph
    string = upstream
    file = ThurSteadyHydrograph.txt
    precision = 0.001
    number_of_iterations =100
    slope = 0.93
}

```

3.3.4.3 Define the lower boundary condition

The lower boundary is an h-q-relation which is calculated internally. Again, we have to define the boundary condition with a specific slope.

```

BOUNDARY
{
    type = hqrelation

```

```
    string = downstream
    slope = 1.5
}
```

3.3.4.3.1 Define initial condition

The channel is considered to be initially dry. Note that starting with a dry channel is for the depth-average equations a numerically delicate problem. So this option will require some care when we set the numerical parameters.

```
INITIAL
{
    type = dry
}
```

3.3.4.3.2 Define default friction values

The declaration of a default friction type and a default friction value are mandatory. The friction values are overwritten by the values declared in the topography file.

```
FRICTION
{
    type = strickler
    default_friction = 35
}
```

3.3.4.3.3 Declare parameters for hydraulic computation

In the *PARAMETER* block the relevant parameters for the hydraulic simulation are defined. With the flag *minimum_water_depth*, the water depth is defined for the case where the channel is considered to be dry. The simulation is using an explicit Euler scheme using a Roe solver for the Riemann problem at the edges of the control volumes.

```
PARAMETER
{
    minimum_water_depth = 0.01
    simulation_scheme = explicit
    riemann_solver = roe
}
```

3.3.4.4 Define time step information

For the first computation the simulation time is set to 150000 s. For a computation on a dry bed, a small initial time step should be chosen. It is used only at the very beginning, as there is no flow in the channel from which the time step could be deduced. The maximum time step should be bigger than all time steps computed during the simulation.

```

TIMESTEP
{
    total_run_time = 150000
    initial_time_step = 1.0
    maximum_time_step = 60.0
    CFL = 0.95
}

```

3.3.4.5 Define output

If only standard output is needed, only the time step for file printing and for console printing have to be defined. If Tecplot software is available, it is also very useful to generate a tecplot file.

```

OUTPUT
{
    output_time_step = 100
    console_time_step = 100
    SPECIAL_OUTPUT
    {
        type = tecplot_all
        output_time_step = 100
    }
}

```

3.4 Perform hydraulic simulations

3.4.1 Perform steady flow simulation (Thur1)

The first simulation has the aim to create a steady flow as initial condition. Place the three files in the same folder and start the simulation by double-clicking on the command file “run.bmc”, by starting from the console or by loading it as command file in the graphical user interface.

When the simulation has finished, open the main output file named “Thur_out.dat” using a text-editor such as notepad++ or gedit. Scroll to the end of the file, where the output data of the last time step is listed. Check the discharges Q which are given in the column labelled “Mass_Flux_[m³/s]”. If the discharges correspond to the steady inflow discharge of 30 m³/s at all edges between the cross sections, then the steady-state has been reached. You should also have a look at the other output values to see if there are suspicious values, which indicate an error somewhere in your model setup.

Edge_number	Mass_Flux_[m ³ /s]	Momentum_Flux_[m ⁴ /s ²]	...
0	30	21.0328	...
1	30	20.7677	...
2	30	20.8081	...
3	30	18.4419	...
4	30	16.5479	...

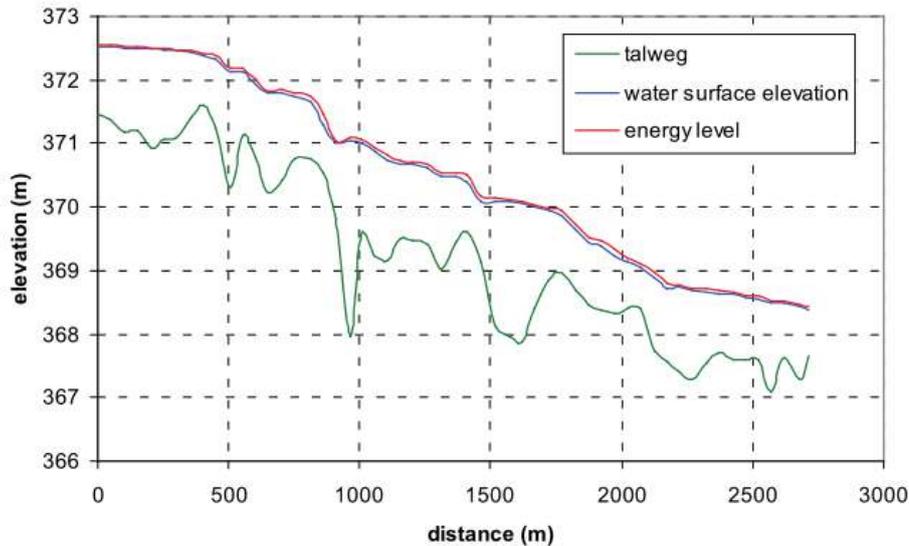


Figure 3.8 Longitudinal profile

Edge_number	Mass_Flux_[m ³ /s]	Momentum_Flux_[m ⁴ /s ²]	...
5	30	18.1862	...
6	30	19.3961	...
7	30	21.1823	...
8	30	21.8026	...
...			

If the steady-state has been reached, then we use this result for subsequent simulations (hot start). For this purpose you can use the file “Thur_restart_150000.dat”, which contains the results of the last time step. It is recommended to rename the restart file (e.g. “ThurInitial.dat”) in order to use it as an initial condition in further simulations.

You should also have a look at the file named “ThurTopo_tables.txt” and verify whether there have been any errors or strange values resulting from your topography file. (Of course, the grid file you have here in the tutorial should not have any errors. It’s just a good exercise to check your geometry if the program does not work as expected).

Now, take from the main output file “Thur_out.dat” the columns named “distance_[m]”, “z_talweg_[m_asl]”, “wse_[m_asl]” and “energy_head_[m_asl]” for the last time step and use them to plot a longitudinal profile (e.g. with Excel).

As it seems that there is no problem, we can proceed to the next step.

3.4.2 Perform simulation of the floods (Thur2)

Copy the following files in a new Folder or take the ones in the second zip file:

```
run.bmc
ThurTopo.bmg
ThurInitial.dat
```

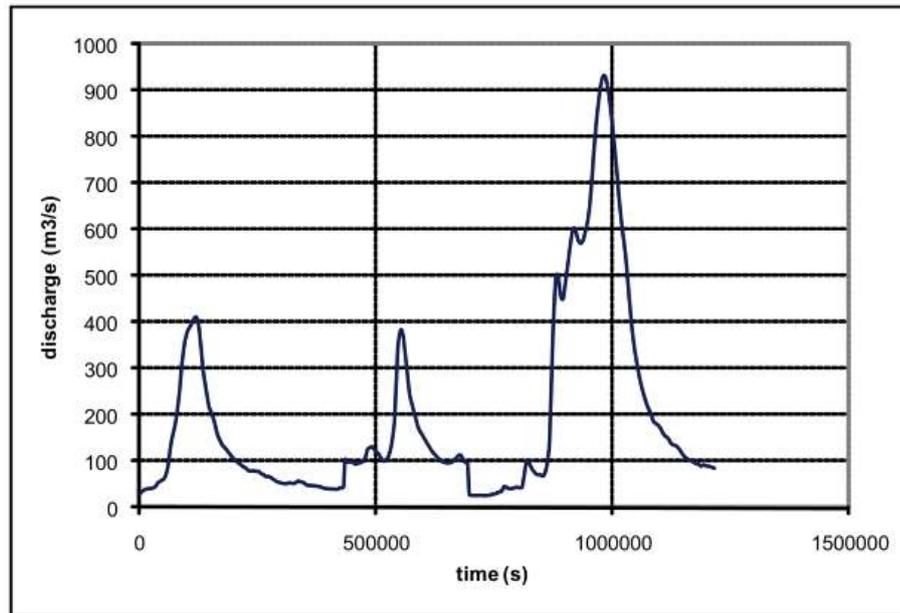


Figure 3.9 Hydrograph of the flood event of 2005

Extract the “ThurHydrograph.txt” from the second tutorial zip file. It contains the hydrograph of the flood event in 2005.

T	Q
0	30.648
3600	34.05
7200	37.305
10800	39.707
14400	41.18
18000	41.916
21600	42.275
25200	42.654
28800	43.646
32400	45.444
...	

For the upstream boundary condition, change the data file into “ThurHydrograph.txt”.

```
BOUNDARY
{
  type = hydrograph
  area = upstream
  file = ThurHydrograph.txt
  precision = 0.001
  number_of_iterations =100
  slope = 0.93
}
```

The renamed restart file “ThurInitial.dat” is used as an initial condition for the new simulation. Therefore we have to consider two things:

1. Choose the restart file
2. Define the start time of the new simulation.

```
INITIAL
{
    type = continue
    file = ThurInitial.dat
}
```

Because we want to restart from time zero we set the flag *start_time* in the *TIMESTEP* block to zero. Be aware that if the *tag start_time* is not defined the time in the restart-file is used (i.e. # time: 150000 sec).

The total running time of the simulation is increased to 338 hours (1216800 sec.). So we change the settings in the *TIMESTEP* block accordingly.

```
TIMESTEP
{
    start_time = 0.0
    total_run_time = 1216800
    CFL = 0.95
    maximum_time_step = 60.0
}
```

The output will be plotted less often. Therefore, we change the console and output time.

```
OUTPUT
{
    output_time_step = 1000
    console_time_step = 1000
}
```

Now we can run the file “run.bmc” by clicking on the Run button. When the simulation is finished, have a look at the main output file “Thur_out.dat”. Take the columns of “z_talweg_[m_asl]”, “wse_[m_asl]” and “energy_head_[m_asl]” for the time of maximal discharge (ca. 933 m³/s) at ca. 982.000 s (= 273 hours) and plot them over the “distance_[m]” (see Figure 3.10).

Then we plot some interesting cross sections with their water surface elevation, for the same time (see Figure 3.11). This can help to see what happens, and which parts of the cross sections are touched by the flood. For this purpose the cross section geometry and can be taken from the topography file and the water surface elevation from the main output file. Alternatively a monitoring point of type “geometry” could be used.

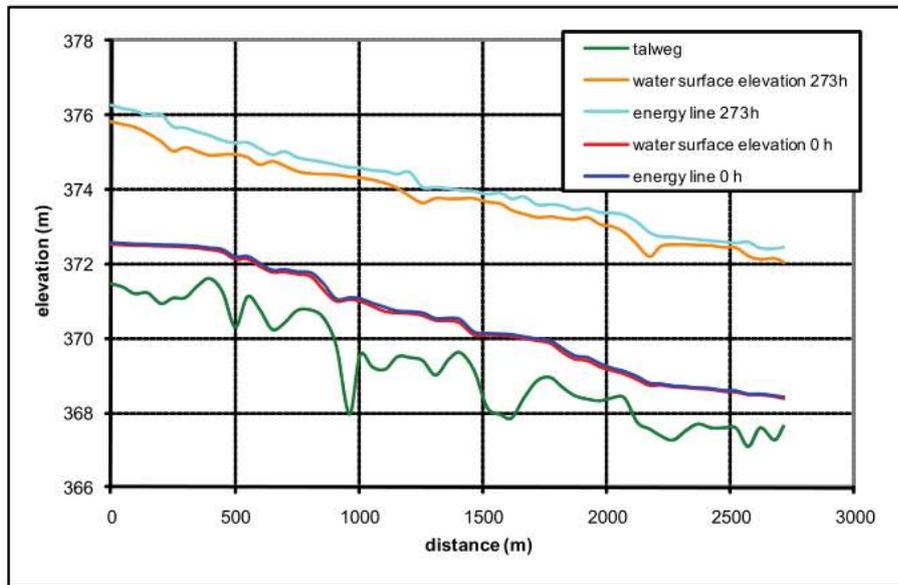


Figure 3.10 Longitudinal profile for maximum discharge

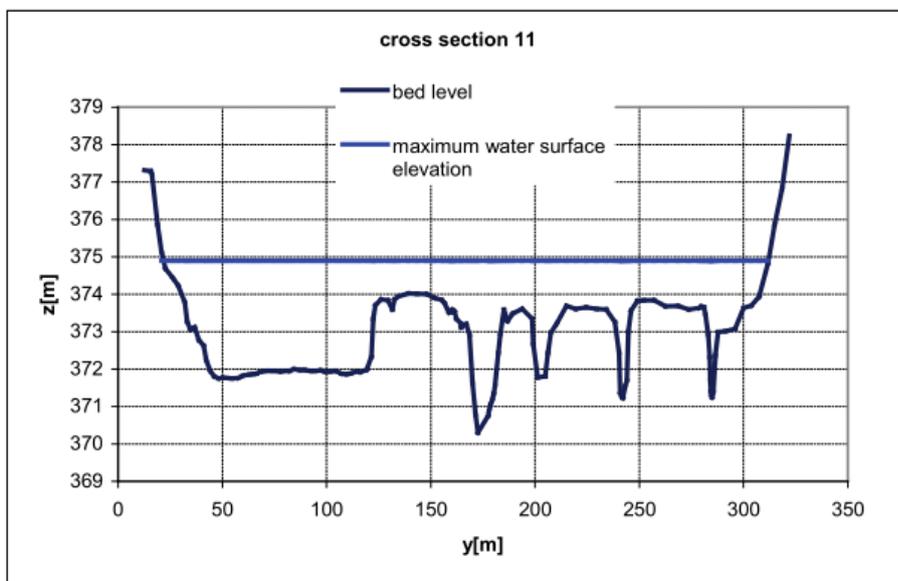


Figure 3.11 Example of resulting water surface elevation

3.5 Complete the command file for bed load transport

After copying the files from simulation 2 in a new folder the command file must be completed with information about bed load, which is grouped in the block MORPHOLOGY. In the block BASECHAIN-1D chose MORPHOLOGY and press Add Block. Go in the block MORPHOLOGY.

3.5.1 Define the bed material

The simulation is executed with a single grain class with mean diameter = 2.5 cm. This means that you have to define one grain class and one mixture. Add a block of type GRAIN_CLASS and one of type MIXTURE. In the GRAIN_CLASS block add the diameter.

In the MIXTURE block add the name and the volume fraction.

```
BEDMATERIAL
{
  GRAIN_CLASS
  {
    diameters = ( 25 )
  }
  MIXTURE
  {
    name = unique
    volume_fraction = ( 100 )
  }
  [...]
}
```

Two types of soils are defined: one which is fixed (code 1) and one with a sub layer of 5 m thickness which is attributed to the bed bottom where bed load takes place (code 2). Add twice a block of type SOIL_DEF. The first one needs only a name as it has no layers of material. In the second one add a LAYER block. Then give the layer a bottom elevation and a mixture.

```
SOIL_DEF
{
  name = fixed
}
SOIL_DEF
{
  name = mobile
  LAYER
  {
    bottom_elevation = -5.0
    mixture = unique
  }
}
```

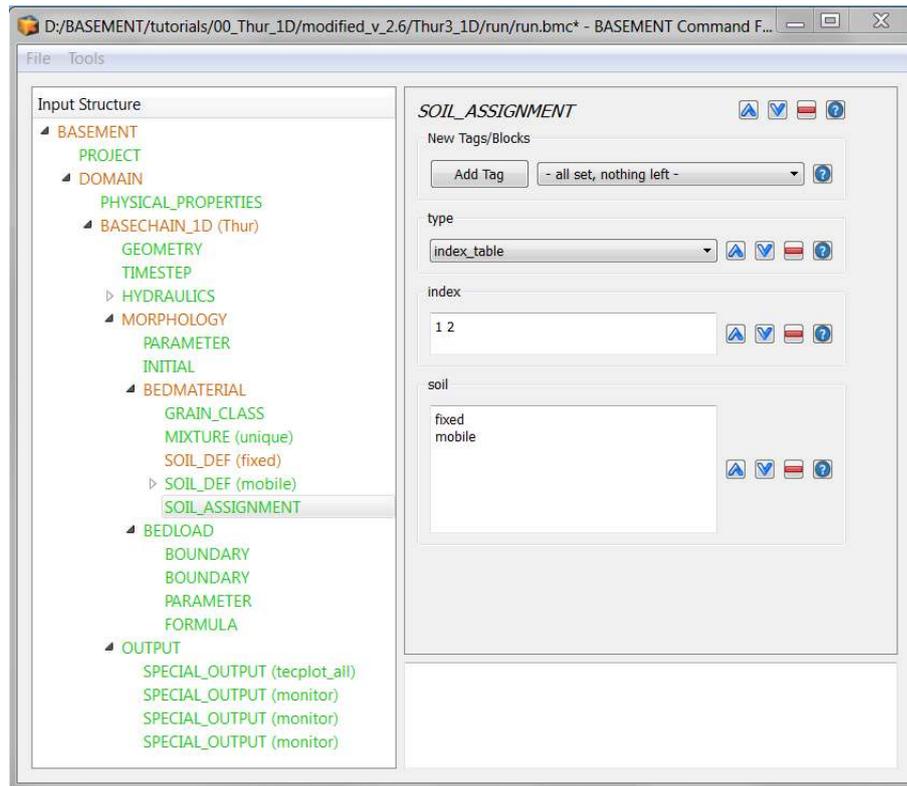


Figure 3.12 Soil assignment in the user interface

3.5.2 Soil assignment

The names of the described soils have now to be assigned to the soil codes used in the topography file. Add a `SOIL_ASSIGNMENT` block and there the attributes `type`, `index` and `soil`. The first value in the index window has to correspond to the first name in the soil window etc.

3.5.3 Define general parameters for sediment transport

The porosity and the density of the material are standard values. The control volume is set to a constant thickness of 20 cm (varying this parameter influences the grain sorting processes in the simulation). The tables for the hydraulic computation will be updated each time when the bed level has changed more than 5 cm.

```
PARAMETER
{
  porosity = 37
  control_volume_type = constant
  control_volume_thickness = 0.2
  density = 2650
  max_dz_table = 0.05
}
```

3.5.4 Define specific parameters for bed load transport

The parameter for upwind scheme is set to 1.0 and the 'velocity_area' which defines which part of the cross section area is taken for the velocity computation for bedload transport, is changed to 'main'.

```
PARAMETER
{
    upwind = 1
    velocity_area = main
}
```

3.5.5 Define bed load transport formula

The Meyer-Peter and Müller bed load approach will be applied without adjusting the calculated transport capacity (bedload_factor = 1.0). For the critical angle a standard value has been chosen.

```
PARAMETER
{
    bedload_formula = mpm
    bedload_factor = 1
    angle_of_repose = 30
}
```

3.5.6 Define boundary conditions for bed load

At the downstream boundary it is considered that the quantity of sediment which enters the last element leaves it by the boundary.

```
BOUNDARY
{
    type = IODown
    string = downstream
}
```

At the upper boundary, the observed modification of the bed level before and after the floods is very small. For this reason it can be assumed that at the upstream boundary there is as much sediment coming in, as is transported out of the first element.

```
BOUNDARY
{
    type = IOUp
    string = upstream
}
```

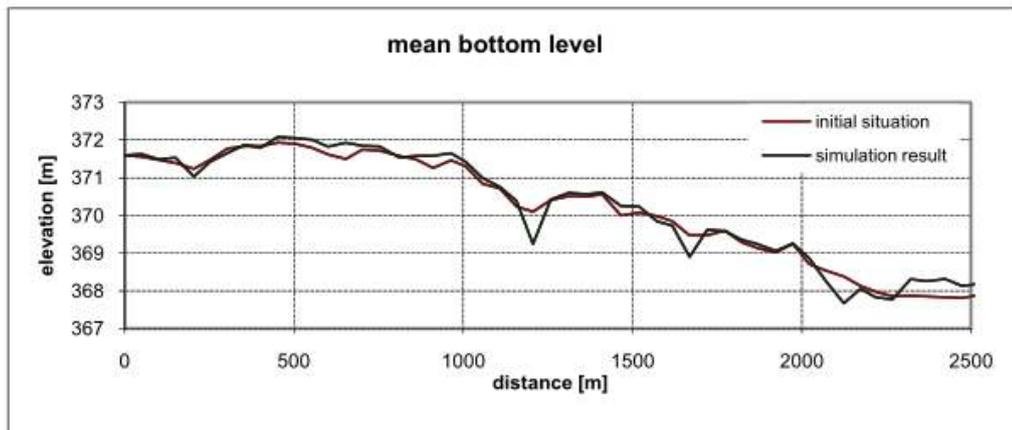


Figure 3.13 Longitudinal profile of mean bottom level

3.5.7 Generate a “geometry” file

To see how the geometry of cross section 14 changes during the flood add a `SPECIAL_OUTPUT` block to the `OUTPUT` block. Set the type of the `SPECIAL_OUTPUT` to monitor.

```
OUTPUT
{
  output_time_step = 1000
  console_time_step = 1000
  SPECIAL_OUTPUT
  {
    type = monitor
    output_time_step = 1000
    cross_sections = ( CS14 )
    geometry = ( time )
  }
}
```

3.6 Perform bed load simulation (Thur 3)

Run the file `Thur3.bmc`. When the simulation has finished, look at the “`Thur_out.dat`” file, take the columns of distance and mean bottom level of the start and end situation and make a longitudinal profile of it (Figure 3.13).

Additionally open the topology file of cross section 14 “`ThurCS14_tec.dat`” and plot the old and new geometry of this cross section to see the morphodynamical changes (Figure 3.14).

Obviously this is only a first run for exercise. This computation now needs calibration and validation before it can be used to make prediction of future bed evolution.

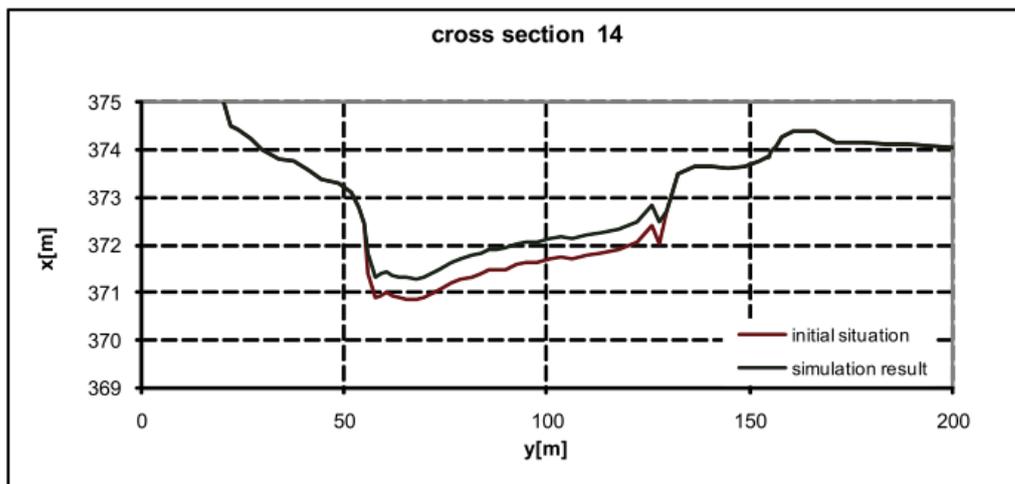


Figure 3.14 Transversal cross section profile

4

Dynamics of longitudinal bed profile due to local river widening (1D)

4.1 Introduction

In this tutorial the capabilities of *BASEchain* in modelling the evolution of the longitudinal bed profile due to a local symmetric widening is shown and discussed. The general concepts of a 1D simulation with *BASEMENT* and how to set up the command file is demonstrated in the previous tutorial ‘Hydrodynamics and sediment transport at the river Thur’.

4.1.1 General description

In the context of river engineering, local river widenings are a measure for river bed stabilisation and ecological restoration. The example river widening can be divided into three main parts (Figure 4.1): Upstream channel (L_{US}) with a certain channel width (W_C), transition zone between channel and widening (L_T), widening with length (L_W) and width (W_W) and the downstream channel (L_{DS}).

The new channel geometry due to the widening affects the hydraulic conditions in the particular river reach. This causes morphological processes that lead to a new equilibrium state of the system (Figure 4.2). Starting from an initially plane bed, erosion occurs upstream of the widening due to a temporary flow acceleration towards the widening (Scour_US). A significantly larger erosion can be observed at the downstream end of the widening (Scour_DS) as a result of the bed load deficit caused by the deposition inside the widening. This deposition (ΔH) increases and migrates in downstream direction until the end of the widening is reached. The material slowly fills up the scour whereby the downstream bed tends towards its initial state again. The longitudinal slope within the widening increases until a new equilibrium slope is reached providing enough transport

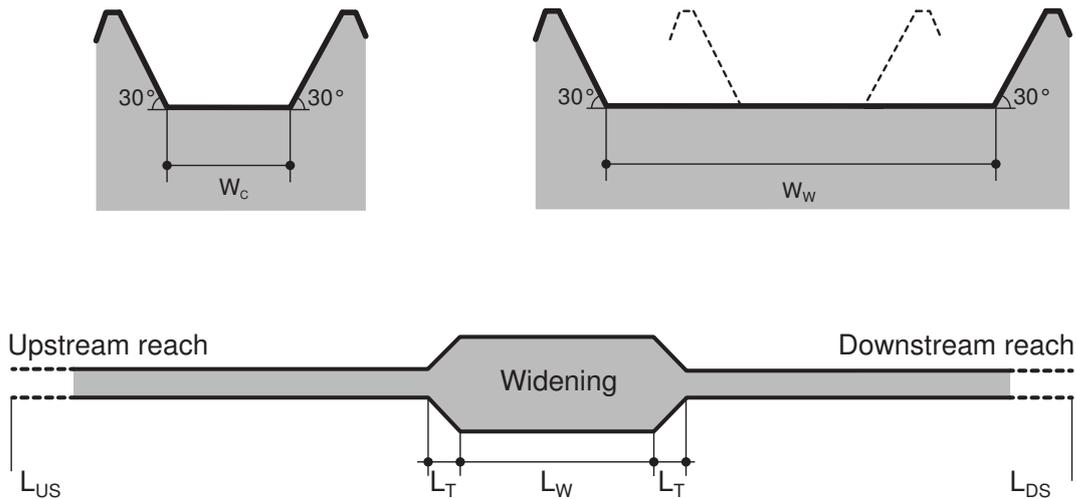


Figure 4.1 Sketch of symmetric widening used for the numerical simulations in this tutorial.

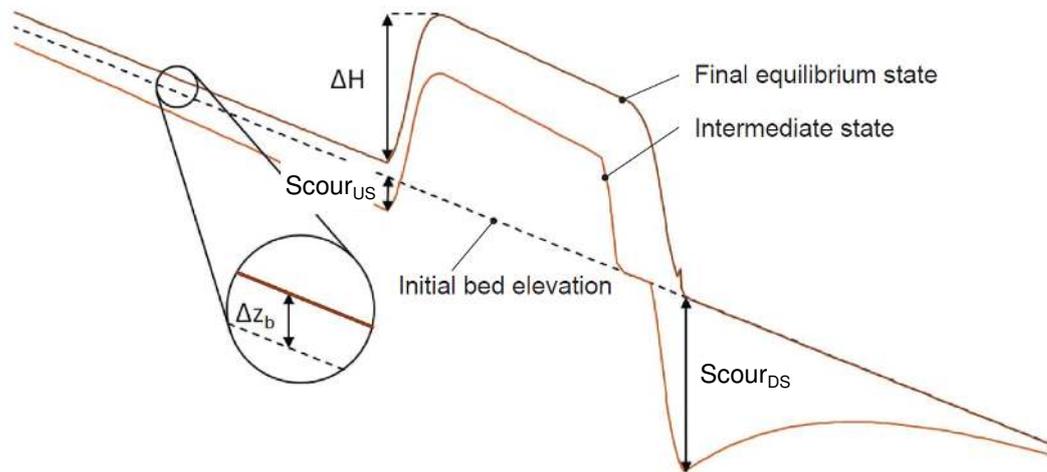


Figure 4.2 Temporal evolution of the longitudinal bed profile due to a local river widening.

capacity to convey the entire bedload discharge downstream. The widening may induce an uplift (Δz_b) of the river bed in the upstream reach. However, before the depositions in the widening, reach equilibrium state, erosion may be observed in the upstream channel.

4.1.2 Purpose

Based on the general conditions of the river widening at the River Thur near Altikon (ZH, Figure 4.3), numerical 1D simulations with simplified geometry are performed in order to analyse the effects of an artificial river widening on the evolution of the longitudinal bed profile (mean bottom elevation). Besides the temporal evolution, the final equilibrium state of the longitudinal bed profile shall be determined by means of a parameter variation for different length of the widening, width of the widening, discharge and mean grain size diameter.



Figure 4.3 River widening at the River Thur near Altikon (ZH) in 2014, source: C. Herrmann, BHAteam Ingenieure AG, Frauenfeld

4.1.3 Used features

BASEMENT version 2.6 is used for the simulations in this tutorial. All relevant files to run the simulations including the topographies for the different scenarios are provided. Bedload transport is calculated using the Meyer-Peter and Müller (MPM) formula for a single grain configuration. In general default values of the program are used. Two important parameters, which have been modified, are described in Table 4.1.

Table 4.1 *BASEMENT* parameters used for this tutorial

Parameter	Description	Default value	Used value
<i>max dz table</i>	Change in bottom elevation to trigger recomputation of tables [tables1D]	0.1 [m]	0.01 [m]
<i>upwind</i>	Weighting for sediment flux calculation on edge		0.7 [-]

4.1.4 Parameter variation

Starting from the default configuration of the reference geometry, the parameters are altered according to Table 4.2 . The input files and *BASEMENT* command files (*.bmc) for each run are provided and can be identified by their run-number.

Table 4.2 Summary of the simulation runs for different parameter combinations

Parameter	run	W_W [m]	L_W [m ²]	Q [m ³ /s]	d [mm]
	default	100	500	800	20
Width variation	1	150	500	800	20
	2	200	500	800	20
Lenght variation	3	100	250	800	20
	4	100	750	800	20
	5	100	1000	800	20
Discharge variation	6	100	500	200	20
	7	100	500	400	20
	8	100	500	1600	20
Diameter variation	9	100	500	800	5
	10	100	500	800	10
	11	100	500	800	40

4.2 Model setup

4.2.1 Definition of 1D topography

All the topography files (.bmg) used in this tutorial are provided to the user. A uniform value for the Strickler roughness is chosen for all cross sections. The `bottom_range*`, which defines the part of the cross section considered for the sediment transport calculations, spans from the left to the right embankment toe (Figure 4.4). This means that deposition and erosion only occur at the bottom, the embankments are fixed. A mobile bed (layer soil 1) with a thickness of 2 m is defined at the bottom.

The parameter variation is performed based on the reference geometry as specified in Table 4.3:

Table 4.3 Default configuration and variation of the investigated parameters.

Type	Symbol	Units	Description	Default value	Variation
Reference geometry	L_{US}	m	Length upstream channel	3000	
	L_T	m	Length transition zones (US/DS)	150	
	L_W	m	Length widening	5000	250-1000
	L_{DS}	m	Length downstream channel	3000	
	W_C	m	Width cannel	50	
	W_W	m	Width widening	100	100-200
	S_C	%	Channel slope	0.15	

Type	Symbol	Units	Description	Default value	Variation
	S_W	%	Initial slope	0.15	
Additional variables	Q	m^3/s	Steady discharge	800	200-1600
	Q_S	m^3/s	Steady sediment discharge	0.178	$f(Q, d)$
	d	mm	Grain diameter (single grain)	20	5-40
	$k_{St,C}$	$\text{m}^{1/3}/\text{s}$	Strickler roughness channel	34	
	$k_{St,W}$	$\text{m}^{1/3}/\text{s}$	Strickler roughness widening	34	

4.2.2 Determination of the upstream sediment boundary conditions

The equilibrium sediment discharge at the inflow model boundary is determined for each discharge Q using a topography with a slope equal to the reference geometry but without river widening. The sediment discharge is calculated for different discharge and grain diameters (see Table 4.4) by *BASEMENT* using the boundary condition of type *transport_capacity*. The resulting sediment discharges are used for the parameter variation simulations afterwards. This approach was chosen to assure a constant sediment inflow independent of local erosions or depositions when running the final simulations including the local widening.

Table 4.4 Upstream sediment boundary inflow Q_S defined for different stationary discharges Q and mean grain size diameters d .

	Q [m^3/s]	d [mm]	Q_S [m^3/s]
Discharge variation	800	20	$Q_{S,Q800} = 0.178$
	200	20	$Q_{S,Q200} = 0.020$
	400	20	$Q_{S,Q400} = 0.043$
	1600	20	$Q_{S,Q1600} = 0.363$
Diameter variation	800	5	$Q_{S,Q800,d5} = 0.282$
	800	10	$Q_{S,Q800,d10} = 0.245$
	800	40	$Q_{S,Q800,d40} = 0.067$

4.3 Results of numerical simulations

4.3.1 Temporal evolution of longitudinal bed profile

The basic morphological processes described in the introduction can be reproduced generally well with *BASEMENT*. The results with default configuration are shown in Figure 4.5. Steady state river bed conditions are reached after a total simulation runtime of about one year.

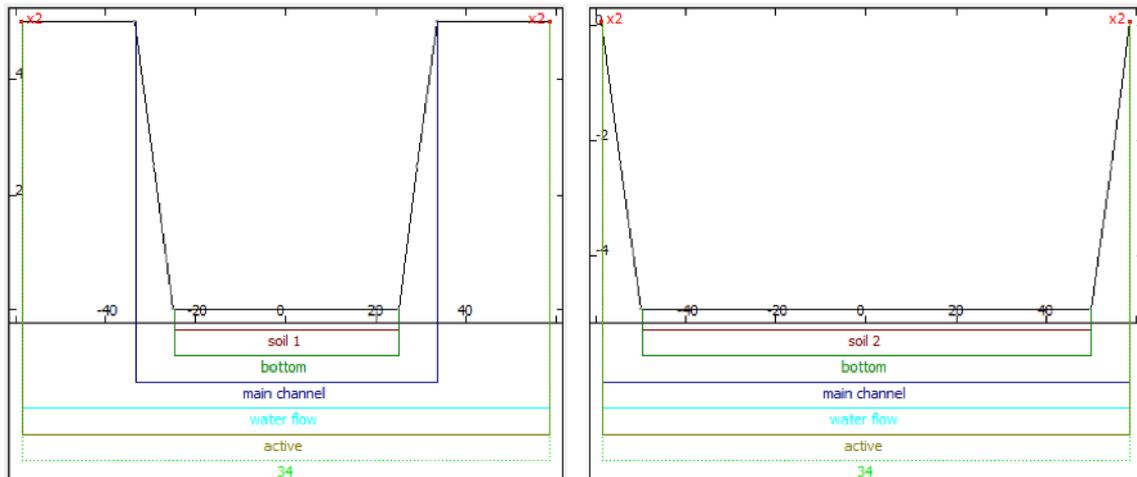


Figure 4.4 Cross sections of the BASEMENT topography representing the channel (left) and the widening (right) to illustrate the geometry and the bottom/soil definition.

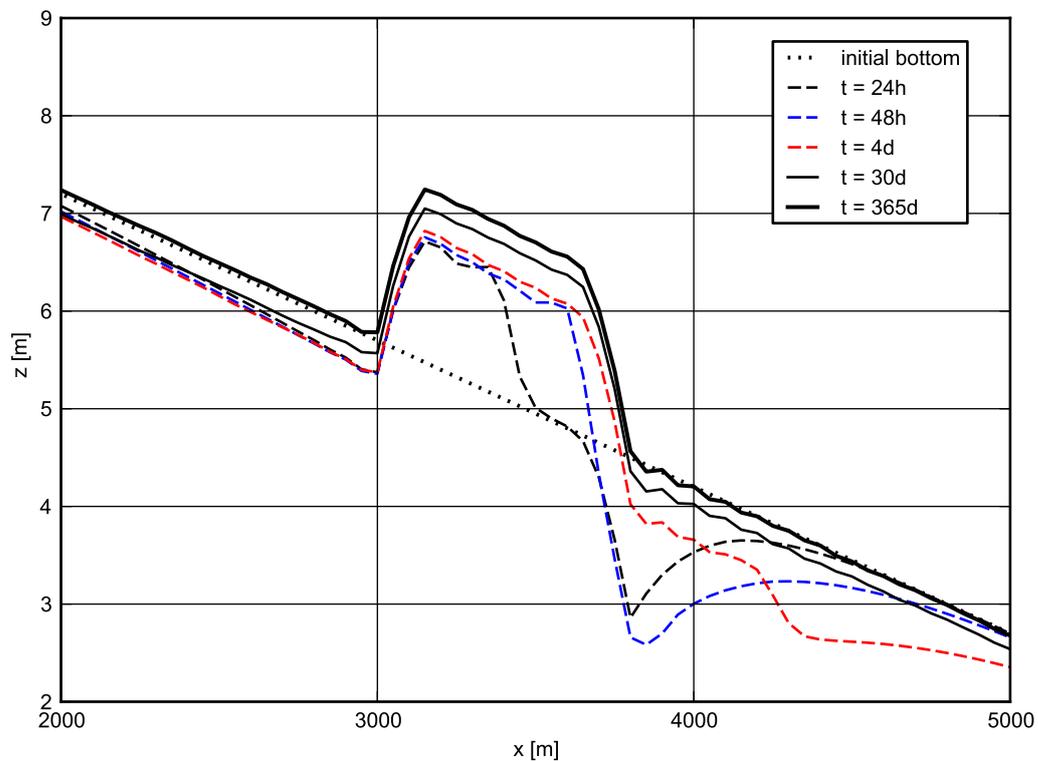


Figure 4.5 Temporal evolution of the mean bottom elevation of the reference geometry with default configuration.

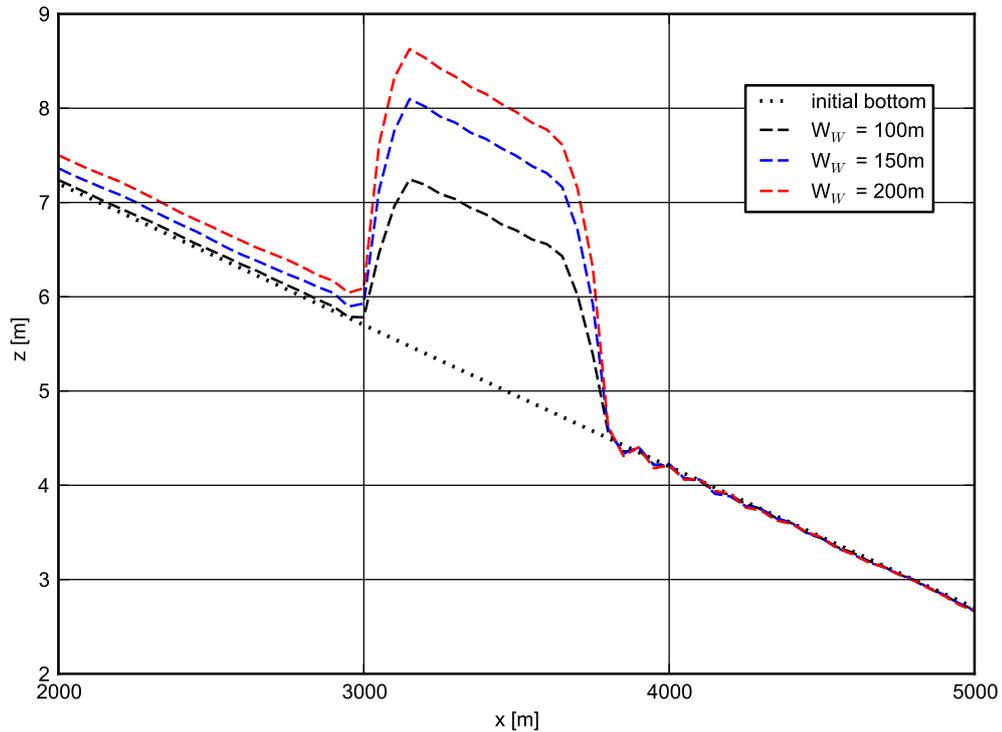


Figure 4.6 Longitudinal bed profile of the initial and resulting steady state for different widening width W_W .

4.3.2 Width and length of the widening

The effect of the width and the length of the widening (geometric parameters W_W and L_W) on the longitudinal bed profile and equilibrium slope is shown in Figure 4.6 and Figure 4.7. The corresponding bottom offset ΔH and upstream bed level change Δz_b is shown in Figure 4.8. For increasing width, ΔH rises significantly and reaches about 2.7 m for $W_W = 200$ m. The new equilibrium slope inside the widening is steeper than the initial slope (Figure 4.6). This causes a bed level change (aggradation) of 0.3 m upstream of the widening. The effect of the river widening length on ΔH and Δz_b is rather small as the slope in the widening remains almost constant. Δz_b increases slightly upstream with increasing length of the widening.

4.3.3 Discharge and grain diameter

The effect of the discharge and the grain diameter (parameters Q and d) on the longitudinal bed profile and the equilibrium slope is shown in Figure 4.9 and Figure 4.10. The corresponding bottom offset ΔH and upstream bed level change Δz_b are shown in Figure 4.11. For one geometry but varying discharge, big differences regarding the final topography are observed. For small discharges (200 and 400 m^3/s), ΔH reaches only about 1 m but the slope in the widening is much steeper than the initial one. Therefore the bed level change upstream of the widening is quite large. For very high values of Q however, the slope in the widening is smaller than in the channel and Δz_b becomes negative. The same effect can be observed for small grain diameters ($d = 5$ mm). Note that these extreme values for discharge and grain diameters do not correspond to the real conditions at the

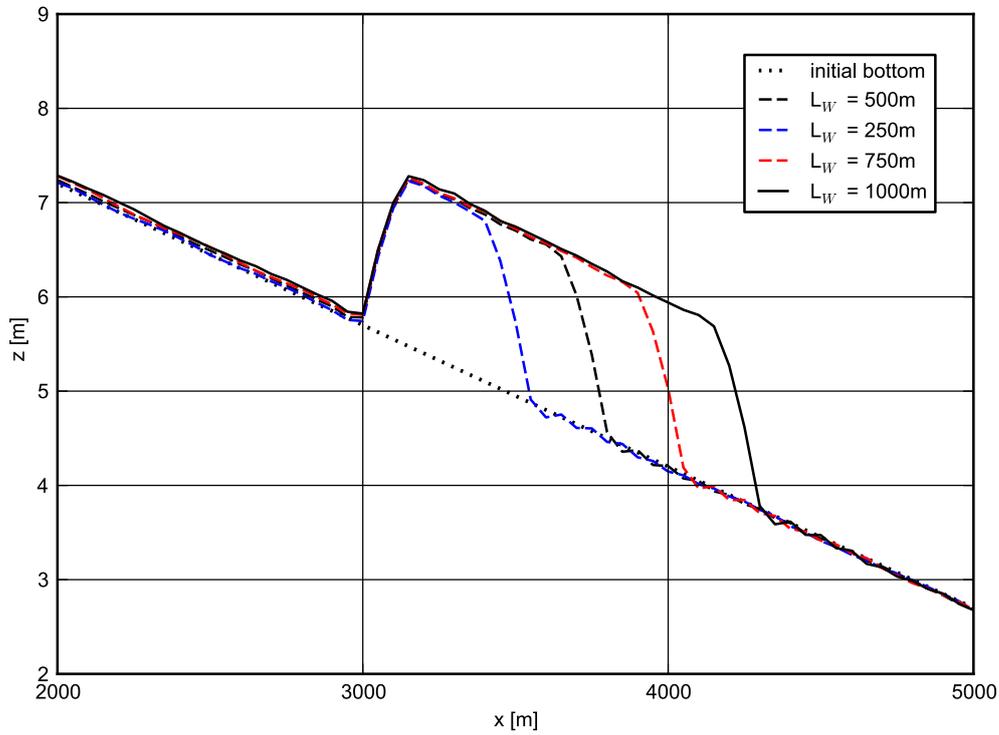


Figure 4.7 Longitudinal bed profile of the initial and resulting steady state for different widening length L_W .

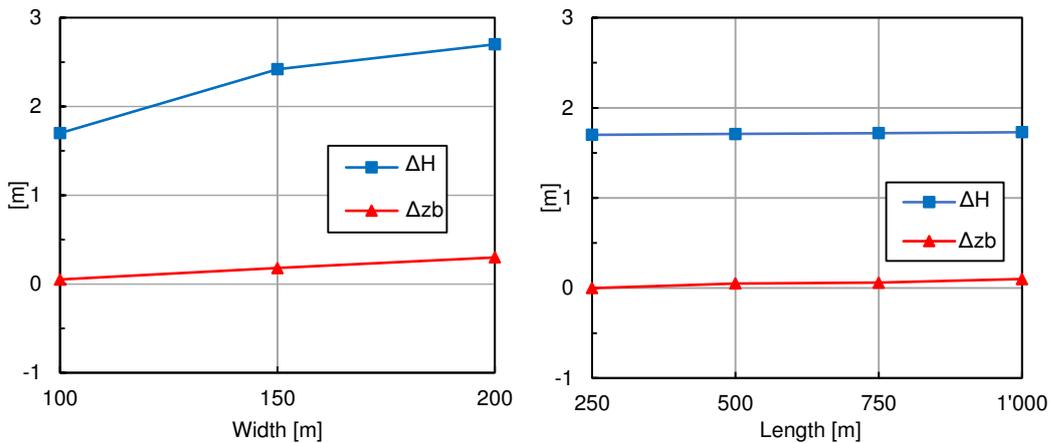


Figure 4.8 Bottom offset ΔH and upstream displacement Δz_b for different W_W (left) and L_W (right), parameter variation according to table 4.2.

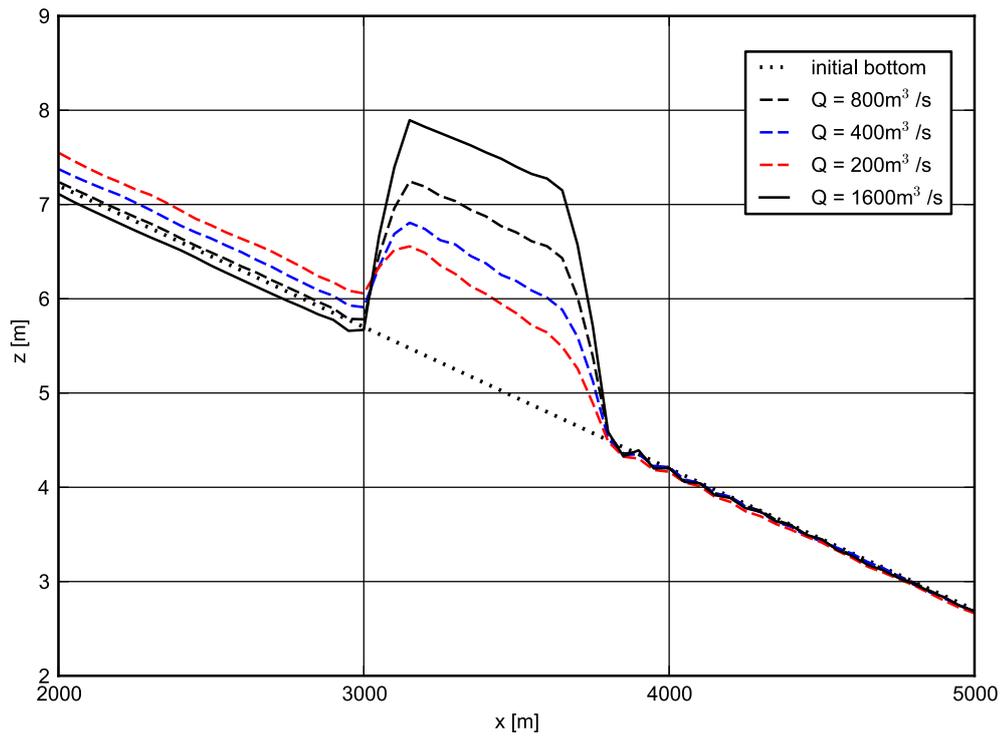


Figure 4.9 Longitudinal bed profile of the initial and resulting steady state for different discharges Q .

river Thur and can therefore be considered as hypothetical configurations. In the case of a coarser grain diameter of 40 mm, the bed slope in the widening get larger than for the default configuration and again depositions in the upstream reach are observed.

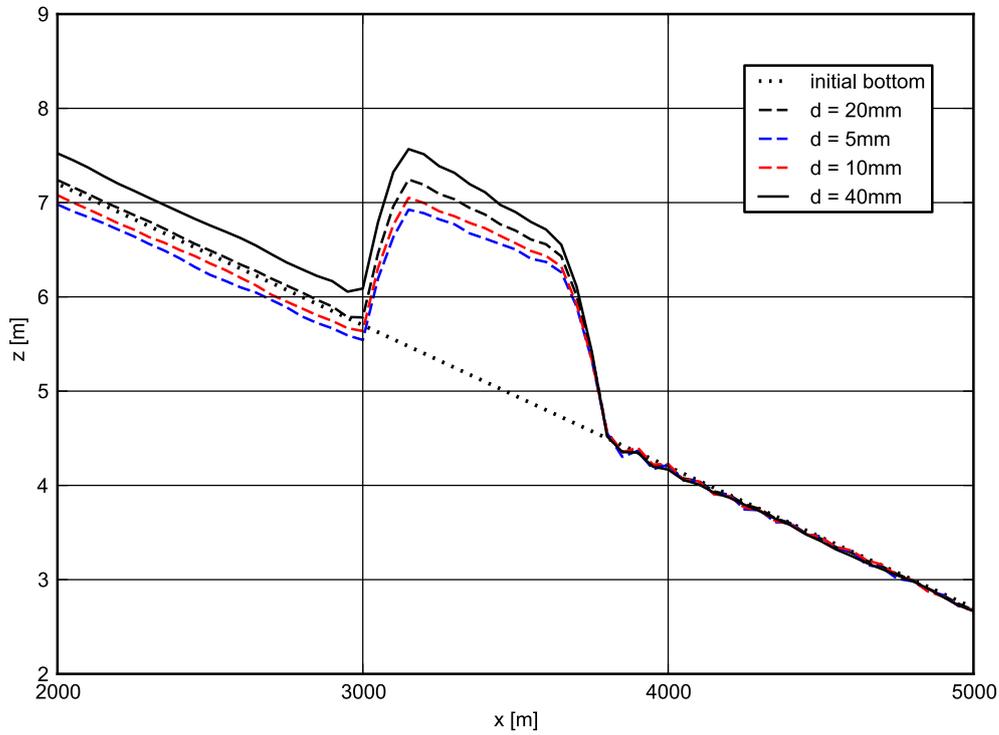


Figure 4.10 Longitudinal bed profile of the initial and resulting steady state for different mean grain size diameters d .

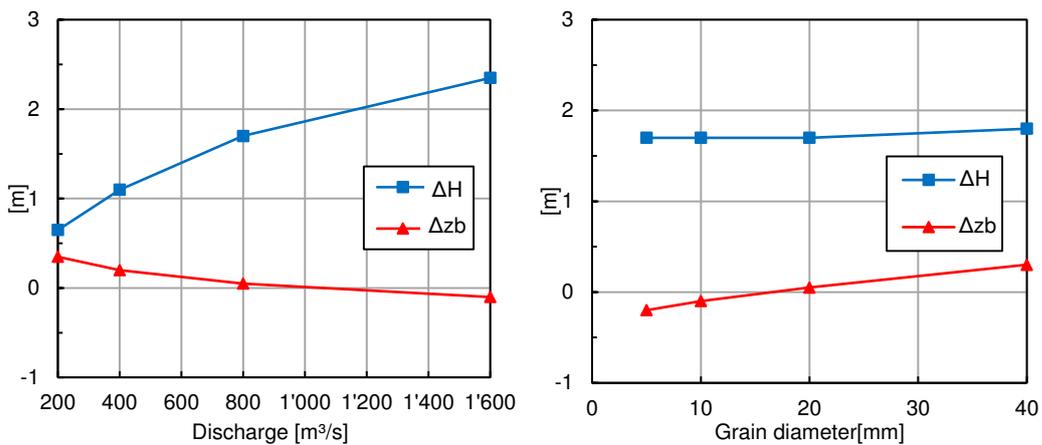


Figure 4.11 Bottom offset ΔH and upstream displacement Δz_b for different discharges Q (left) and mean grain size diameter d (right) parameter variation according to table 4.2.

Hydrodynamics and sediment transport at the river Flaz (2D)

5.1 Introduction

This tutorial gives an introduction to the capabilities of the 2D modelling module BASEplane of BASEMENT. It provides a step-by-step guidance on how build up a model for BASEplane.

5.1.1 Case study description

The tutorial for the 2D modelling module is based on an extract of the case study of the River Flaz in Graubünden. Within the framework of a high water protection project for the village Samedan a completely new section of the River Flaz was built. On a length of 4.1 km morphologically different kind of river subsections can be distinguished (Figure 5.1). The numerical modelling of the whole domain is carried out within the river monitoring project Flaz of the Laboratory of Hydraulics, Hydrology and Glaciology (VAW). In order to reduce the model size (and thus computational running time) only the three most interesting sub-sections are modelled in this tutorial, such as the lower part of the section enriched with roughness elements, the widening part and the part with alternating bars shown in Figure 5.1.

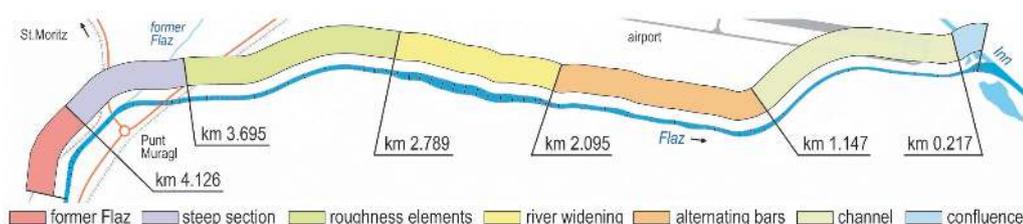


Figure 5.1 Different morphological river subsections of the new section of the river Flaz

5.1.2 Tutorial structure

In a first step the important properties of the mesh file are shown. The tutorial is designed to run BASEplane with the help of the free and open source Geographic Information System [QGIS](#) as pre- and post-processor. The computational grid used in this tutorial (Flaz_mesh.2dm) covers the same perimeter like the one generated in the tutorial of BASEmesh (see Section 1.1). However the position of the breaklines differs slightly from the version of the BASEmesh tutorial. Furthermore the material indices have been changed for the definition of friction factors and soil properties. How to visualize the results of the 2D simulations using the QGIS plugin [Crayfish](#) and how to generate the graphics shown in this tutorial is demonstrated in the post-processing tutorial (Section 2.1). The main focus of this tutorial lies on the setup the command file for the numerical simulation with BASEMENT. The tutorial is structured gradually in the way that first of all a calibrated hydraulic model is set up. Based on this simulation the morphological part can be added to the simulation with a single-grain model. This procedure reflects the proposed way from a calibrated hydraulic simulation to a morphological simulation. An outline of outputs is given and possible visualization is shown for each step.

5.2 Computational grid

The computational mesh is generated with the QGIS plugin BASEmesh. Here just important features and characteristics of the computational mesh for the modelling with BASEMENT are mentioned. The mesh discretizes the topography of the river in such a way that the important topographical information is maintained. Breaklines in the mesh are ensuring that important features of the topography such as the river bed and dike crests are represented correctly (Figure 5.2).

An important feature is the assignment of the material indexes to the different groups of elements. By the material index different properties such as the friction factor and the soil properties can be assigned. The material index is mainly used to assign the friction factor to the different river sections separately. For example it is usual to assign different values for the main channel, embankments and further surrounding land (Figure 5.3).

The mesh file is saved as Flaz_mesh.2dm and has the following structure:

```
MESH2D
.
E3T   eN   n1   n2   n3   eMi   (triangle element)
.
ND    ni  x   y   z
.
```

Where E3T is the flags for the triangular elements; eN denotes the element number; n1, n2 and n3 denote the node numbers of the element and eMi is the material index of the element. The elements are defined in a counter-clockwise direction. The coordinates of the nodes are defined in the second block. ND is the flag for a node; ni denotes node number and x, y and z are the coordinates of the node.

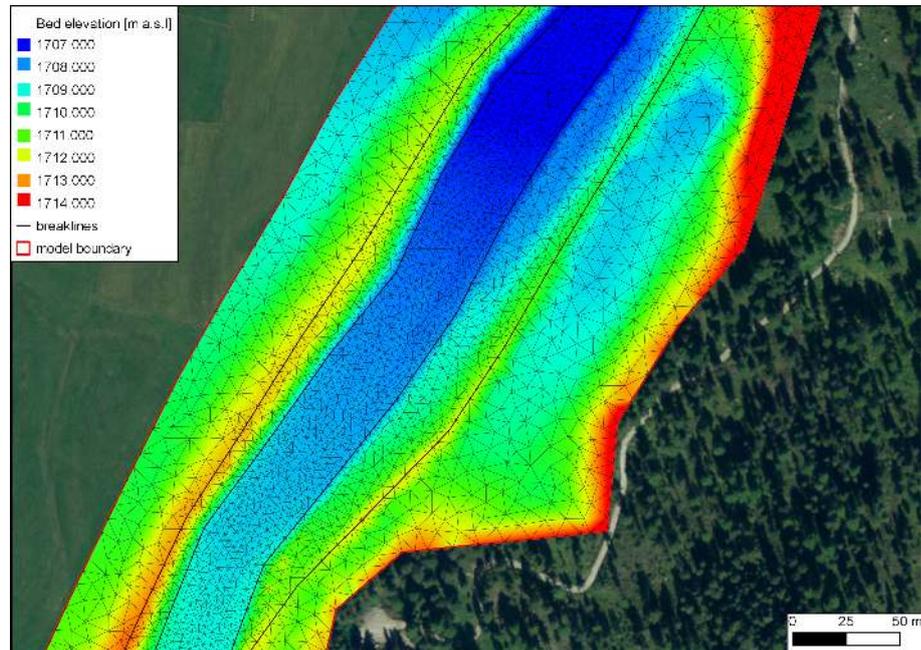


Figure 5.2 Computational mesh generated with BASEmesh for a section in the widening part. The bold black lines are breaklines for the mesh.

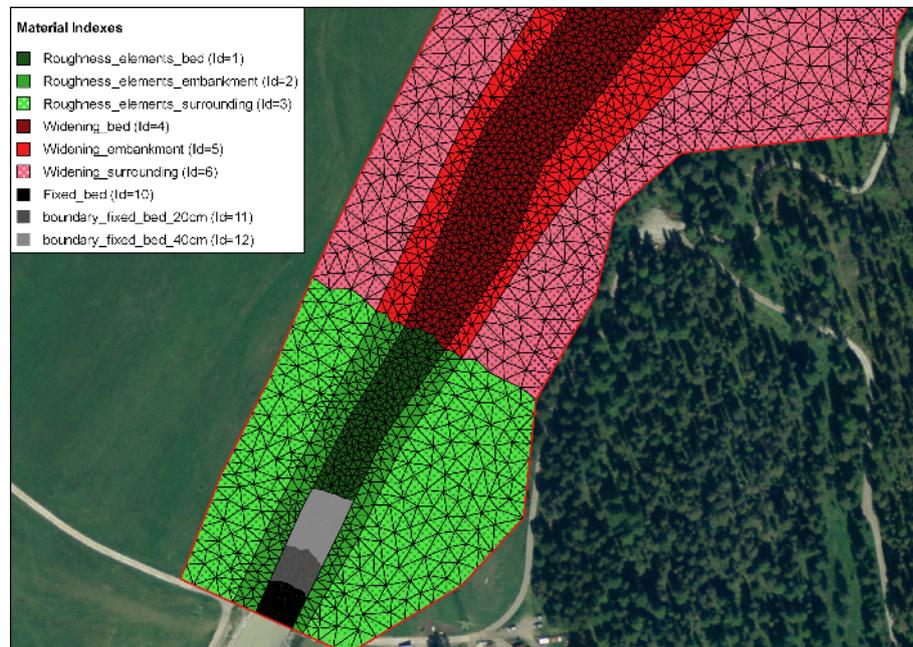


Figure 5.3 Material indexes (Ids) used for assignment the friction factor and the soil properties.

5.3 Setting up the command file

The command file (with the ending .bmc) can be built up and changed within the graphical user interface (GUI) or in any text editor. It has the following general structure:

```
PROJECT
{...}
DOMAIN
{
    multiregion = Flaz
PHYSICAL_PROPERTIES
{...}
BASEPLANE_2D
{
    region_name = Flaz
    GEOMETRY
    {...}
    HYDRAULICS
    {...}
    TIMESTEP
    {...}
    MORPHOLOGY
    {...}
    OUTPUT
    {...}
}
}
```

5.3.1 Project

In this block the project name, the author and the date will be set.

```
PROJECT
{
    title = 2D_Tutorial
    author = KAS
    date = 13.07.2015
}
```

5.3.2 Domain

The DOMAIN-block includes all necessary blocks for a simulation.

```
DOMAIN
{
    multiregion = Flaz
    PARALLEL
```

```

    {...}
    PHYSICAL_PROPERTIES
    {...}
    BASEPLANE_2D
    {...}
}

```

5.3.3 Parallel

In the PARALLEL-block the number of processors can be assigned to the computation with BASEMENT. Depending on the computer the number of threads can be adjusted.

```

PARALLEL
{
    number_threads = 2           // on dual-core system
}

```

5.3.4 Physical properties

The physical properties are global constants in a project.

```

PHYSICAL_PROPERTIES
{
    gravity = 9.81                // [m/s2]
    viscosity = 1.0e-6           // [m2/s]
    rho_fluid = 1000             // [kg/m3]
}

```

5.3.5 Two dimensional simulation

The *BASEPLAIN_2D*-block within the DOMAIN-block contains all information concerning the two dimensional simulation.

```

BASEPLANE_2D
{
    region_name = Flaz
    GEOMETRY
    {...}
    HYDRAULICS
    {...}
    TIMESTEP
    {...}
    MORPHOLOGY
    {...}
    OUTPUT
    {...}
}

```

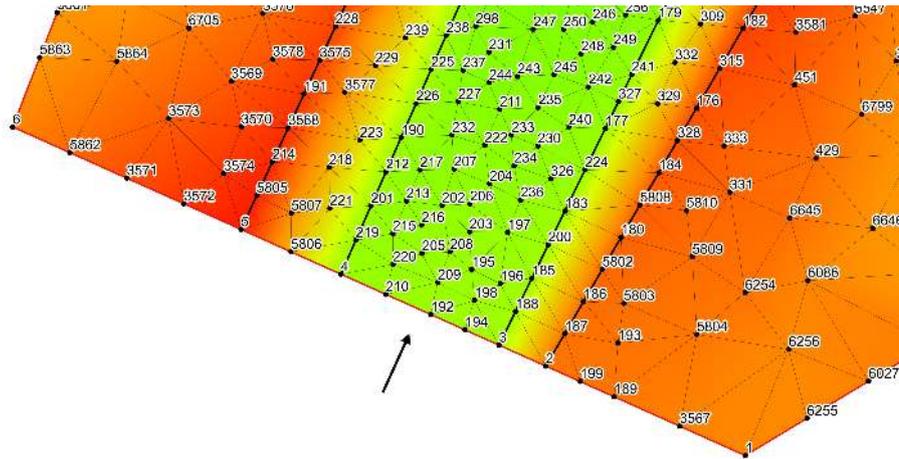


Figure 5.4 Node id numbers for the definition in the `STRING_DEF` block for the inflow boundary.

5.3.5.1 Geometry

The `GEOMETRY`-block defines the mesh file and necessary strings of nodes. Strings are used for inflow and outflow boundaries and can also be used for discharge control. The node ids of the inflow and outflow string can be read out from the mesh in Figure 5.4 and Figure 5.5 respectively.

```

GEOMETRY
{
  type = 2dm
  file = Flaz_mesh.2dm

  STRINGDEF
  {
    name = Inflow
    node_ids = ( 2 3 194 192 210 4 5806 5 )
    upstream_direction = left
  }
  STRINGDEF
  {
    name = Outflow
    node_ids = ( 38 5568 5569 5570 37 5565 5538 5521 5551 36 6004 35 )
    upstream_direction = left
  }
}

```

5.3.5.2 Define the hydraulics

The `HYDRAULIC`-block includes all the information necessary for the hydraulic part of the simulation. This block is divided into the following sub-blocks:

```

HYDRAULICS

```

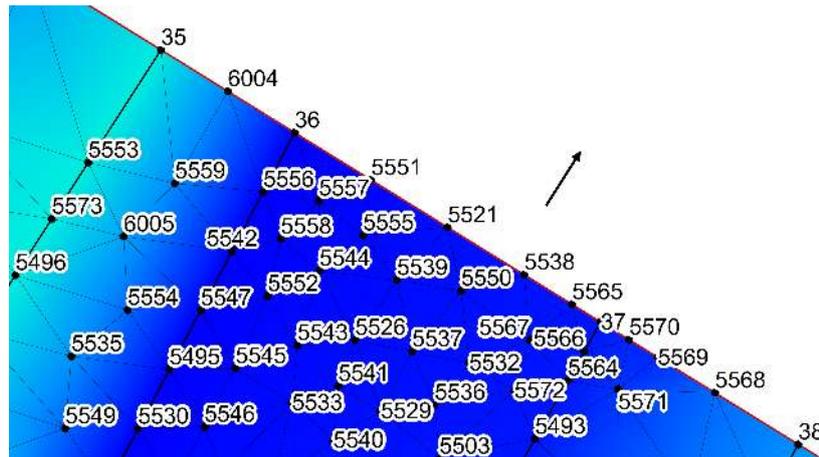


Figure 5.5 Node id numbers for the definition in the `STRING_DEF` block for the outflow boundary.

```
{
  BOUNDARY
  {...}
  INITIAL
  {...}
  FRICTION
  {...}
  PARAMETER
  {...}
}
```

5.3.5.2.1 Hydraulic boundary conditions

For the upper (inflow) and lower (outflow) boundary condition we have to refer to the predefined `STRINGDEFs` (see Section 5.3.5.1). If the boundary condition is not defined explicitly a wall boundary is considered for those edges. Except for the explicitly defined inflow and outflow boundary the model boundary is basically an impermeable wall.

The inlet boundary condition is defined across the predefined string *Inflow*. The hydraulic condition at the boundary is set by the use of a hydrograph and a corresponding slope. The normal slope is used in order to calculate the normal flow depths and the normal flow velocities at the boundary and can be considered as a calibration parameter.

```
BOUNDARY
{
  type = hydrograph
  string_name = Inflow
  file = Inflow_stationary.txt
  slope = 10.0 // [per mill]
}
```

The hydrograph is saved in a text file *Inflow_stationary.txt* in which the first column is the time and the space separated second column is the discharge (Figure 5.6). As a

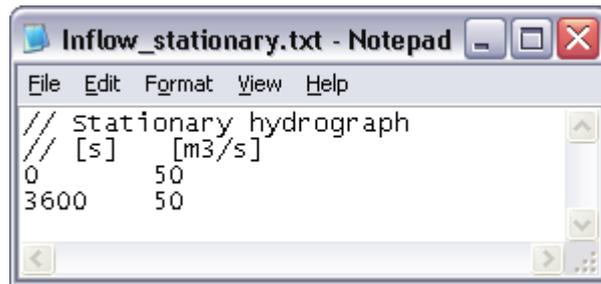


Figure 5.6 Stationary hydrograph file saved as *Inflow_stationary.txt*.

first step, a steady inflow hydrograph is an appropriate choice in order to test the mass conservation of the model. After a certain run time, depending on the size of the model domain, the outflow should counterbalance the inflow. There should be no uncontrolled mass loss within the model domain.

Usually the discharge is taken as the mean annual discharge or the beginning discharge of a flood event. In this case a steady discharge of $50 \text{ m}^3/\text{s}$ is chosen in order to be able to continue later on with a flood hydrograph which starts in this range.

The outlet boundary condition is defined across the predefined string *Outflow*. The normal slope is used in order to calculate the normal flow depths and the normal flow velocities at the boundary and can be considered as a calibration parameter. A sensitivity analysis of this parameter makes always sense. In any case the upper and lower model boundary should be far away enough from the river section of interest, in order to minimise the influence of the boundary conditions.

```
BOUNDARY
{
  type = hqrelation
  string_name = Outflow
  slope = 2.0          // [per mill]
}
```

5.3.5.2.2 Initial condition

The INITIAL-block defines the flow variables at the beginning of the simulation. In a very first step the simulation is started with a dry initial condition.

```
INITIAL
{
  type = dry
}
```

5.3.5.2.3 Friction

The FRICTION-block defines everything related to the friction term in the shallow water equations. Within the computational mesh, a material index is assigned to all elements. By the use of this material index (see Figure 5.3) a friction factor can be assigned. The default friction is used whenever there is no friction assigned to an element.

```

FRICITION
{
    type = strickler
    default_friction = 30
    input_type = index_table
    index = ( 1 2 3 4 5 6 7 8 9 10 11 12 )
    friction = ( 28 30 35 30 30 30 32 32 35 28 28 28 )
    wall_friction = off
}

```

5.3.5.2.4 Computational parameters

The PARAMETER-block defines the control parameters for the numerical simulation of the hydraulic part. The numerical simulation is performed using explicit time integration and the exact Riemann solver for flux computation. The elements with a water depth below the minimum water depth will be considered as dry elements due to stability reasons.

```

PARAMETER
{
    simulation_scheme = exp
    riemann_solver = exact
    minimum_water_depth = 0.05
}

```

5.3.5.3 Define the timestep

The simulation is performed with a total runtime of 3000 seconds. Later on, it has to be tested that after this runtime the flow in the model domain has reached a steady state, meaning that the outflow counterbalances the inflow (see Section 5.4).

```

TIMESTEP
{
    start_time = 0.0
    total_run_time = 3000
    CFL = 0.95
    minimum_time_step = 0.0001
}

```

5.3.5.4 Define the output

In the OUTPUT-block, the desired output has to be defined. During the simulation, output can also be visualized with BASEviz. In order to visualize the 2D results with QGIS Crayfish, chose the output format 'sms'. For ParaView visualization chose type 'vtk'.

```

OUTPUT
{
    console_time_step = 100
    SPECIAL_OUTPUT

```

```

{
    type = BASEviz
    output_time_step = 10
}
SPECIAL_OUTPUT
{
    format = sms
    type = node_centered
    values = ( depth velocity wse )
    output_time_step = 500
}
SPECIAL_OUTPUT
{
    format = vtk
    type = node_centered
    values = ( depth velocity wse )
    output_time_step = 500
}
SPECIAL_OUTPUT
{
    type = balance
    balance_values = ( timestep )
    output_time_step = 100
}
SPECIAL_OUTPUT
{
    type = boundary_history
    boundary_values = ( Q )
    history_one_file = yes
    output_time_step = 100
}
}

```

5.4 Perform hydraulic simulation

5.4.1 Perform steady flow simulation

Open the command file *Flaz_hydraulic_stationary.bmc* either by double-clicking or via the menu of the BASEMENT GUI (File → Open Command). Run the simulation with the *Run* button of the BASEMENT window. If the SPECIAL_OUTPUT of the type *BASEviz* is chosen press the keyboard button *p* to start the simulation. Be aware that the mesh, command file and all other input files have to be in the same folder. The output files are stored in this same folder. In order to check the mass conservation of the model, the files *Flaz_bnd_Inflow_th.dat* and *Flaz_bnd_Outflow_th.dat* are used. After approximately 1600 seconds the outflow counterbalances the steady inflow (Figure 5.7).

In the file *Flaz_balance.dat* the run time of the simulation, the computational time steps and the element which is limiting the computational time step are stored. The identification of the limiting element allows for improvement of the mesh. In order to identify the element

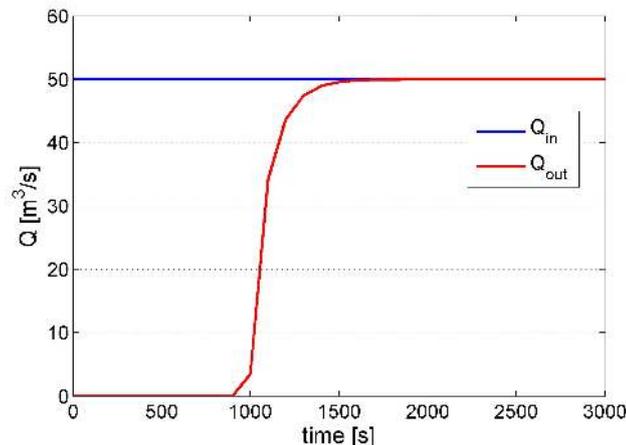


Figure 5.7 Steady inflow hydrograph and outflow hydrograph.

location, load the computational mesh into QGIS as demonstrated in Section 1.3. The solution files with the ending *.sol* can be imported into the program QGIS (see Section 2.1) and the water depth and flow velocities can be visualized with the plugin Crayfish as shown in Figure 5.8. At the end of the simulation the flow variables of the last time step ($t = 3000$ s) are stored in the *Flaz_restart.cgns* file. This file can be used later on to continue the simulation or to use it as an initial condition for a new simulation.

5.4.2 Perform unsteady flow simulation

The unsteady flow simulation is based on the flood event of July 2004 (Figure 5.10). Compared to the steady flow simulation the command file needs some minor changes. First of all the last time step of the steady simulation is taken as initial condition for the unsteady simulation. Therefore the file *Flaz_restart.cgns* from the steady simulation can be renamed and saved for example as *Initial_Condition.cgns*. This file now can be used as an initial condition for the unsteady flow simulation. Therefore we have to consider two things:

1. Choose the solution time to be used as initial condition (note: restart file may contain several solution times).
2. Define the start time for the new simulation.

In this case we set the tag *restart_solution_time* equal to the solution of the last time step of the stationary simulation (*restart_solution_time* = 3000.027 sec). In order to start the simulation from time = 0 again, we have to set the *start_time* in the TIMESTEP block to zero. Be aware that if the *start_time* flag is not defined the *restart_solution_time* defined in the INITIAL block is used (in this case 3000.027 sec).

```
INITIAL
{
  type = continue
  file = Initial_Condition.cgns
  restart_solution_time = 3000.022
```

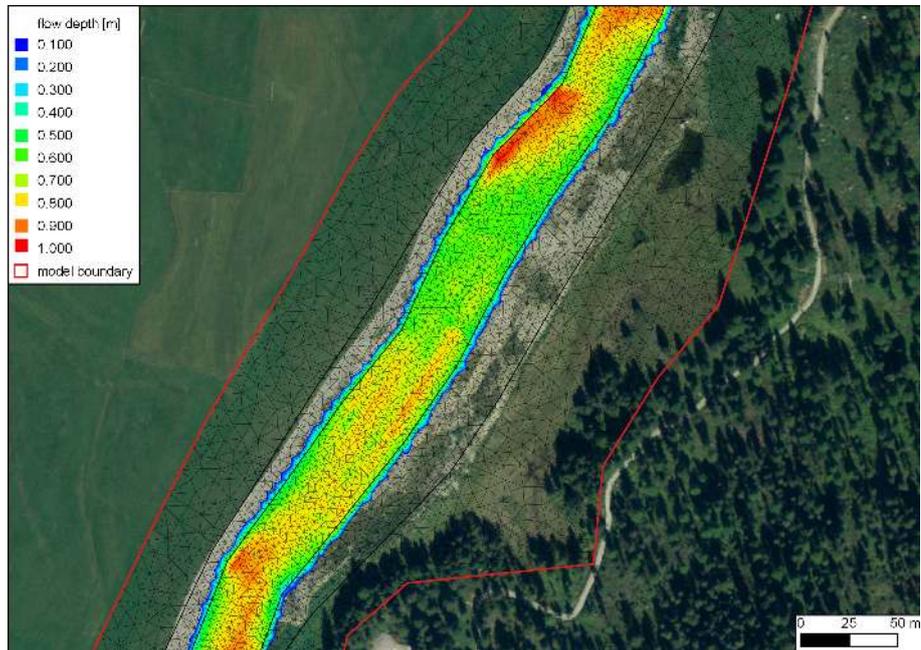


Figure 5.8 Flow depth at the steady state of the model.

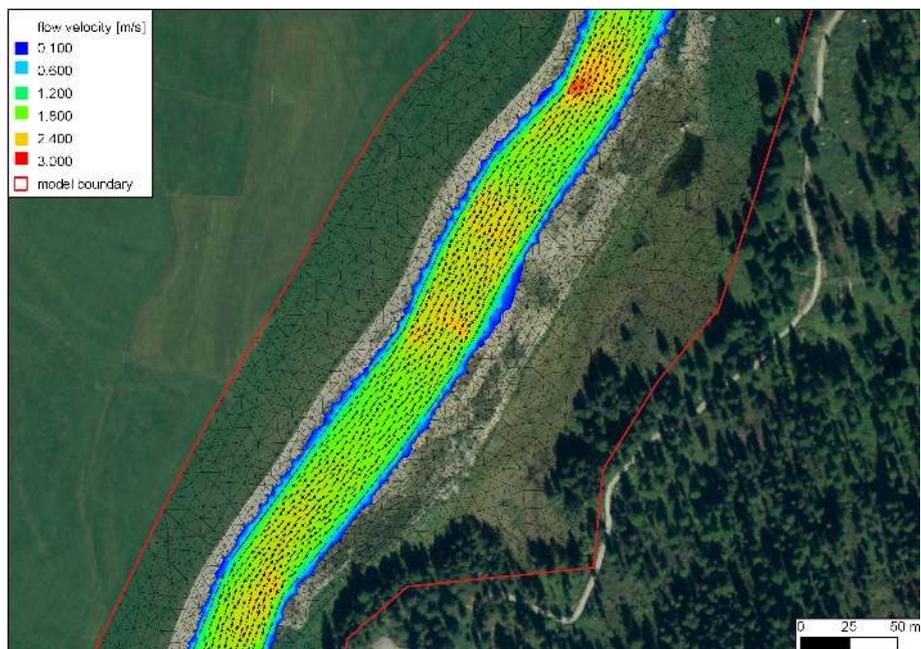


Figure 5.9 Flow velocity and velocity vectors at the steady state of the model.

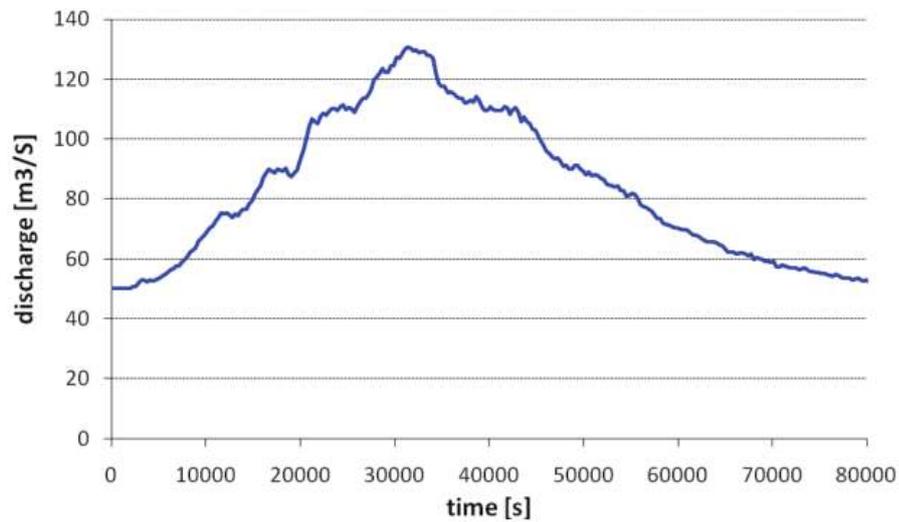


Figure 5.10 Hydrograph of the flood event of July 2004.

```

}

TIMESTEP
{
  start_time = 0.0
  total_run_time = 3000
  CFL = 0.95
  minimum_time_step = 0.0001
}

```

Furthermore the inflow hydrograph of the flood has to be defined and assigned to the upper boundary condition. The hydrograph of the flood shown in Figure 5.10 is saved in the text file *Inflow_transient.txt* with contents as depicted in Figure 5.11. Be aware that the final time defined in this file has to be the same or larger than the computation time. The upper BOUNDARY-block changes to:

```

BOUNDARY
{
  type = hydrograph
  string_name = Inflow
  file = Inflow_transient.txt
  slope = 10.0 // [per mill]
}

```

In the TIMESTEP-block the total run time of the simulation is increased to 84'000 seconds in order to capture the whole flood event:

```

TIMESTEP
{
  ...
  total_run_time = 84000
}

```

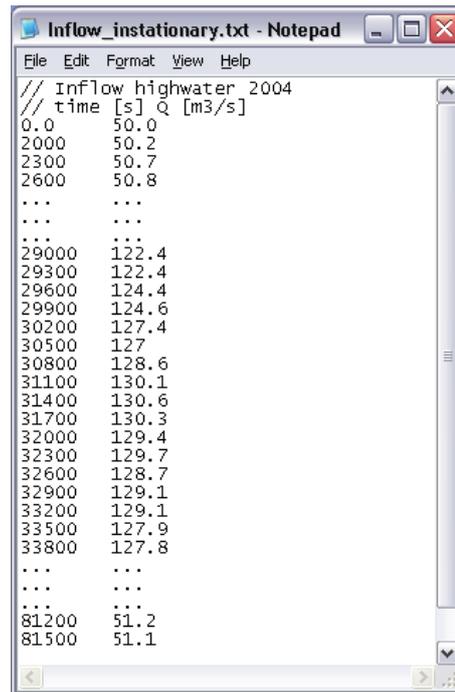


Figure 5.11 Inflow hydrograph stored in the file *Inflow_transient.txt*. Note that the points (...) are just illustrative in order to show the first and last line of the file.

...
}

Last but not least the OUPUT-block has to be adjusted to the needs of the simulation. For the unsteady simulation, the output time step for the SPECIAL_OUTPUT format sms might be reduced in order to capture the maximum values of the output variables .

```

OUTPUT
{
  output_time_step = 2000
  console_time_step = 500
  SPECIAL_OUTPUT
  {
    format = sms
    type = node_centered
    values = ( depth velocity wse)
    output_time_step = 500
  }
  SPECIAL_OUTPUT
  {
    type = balance
    balance_values = ( timestep )
    output_time_step = 1000
  }
  SPECIAL_OUTPUT
  {

```

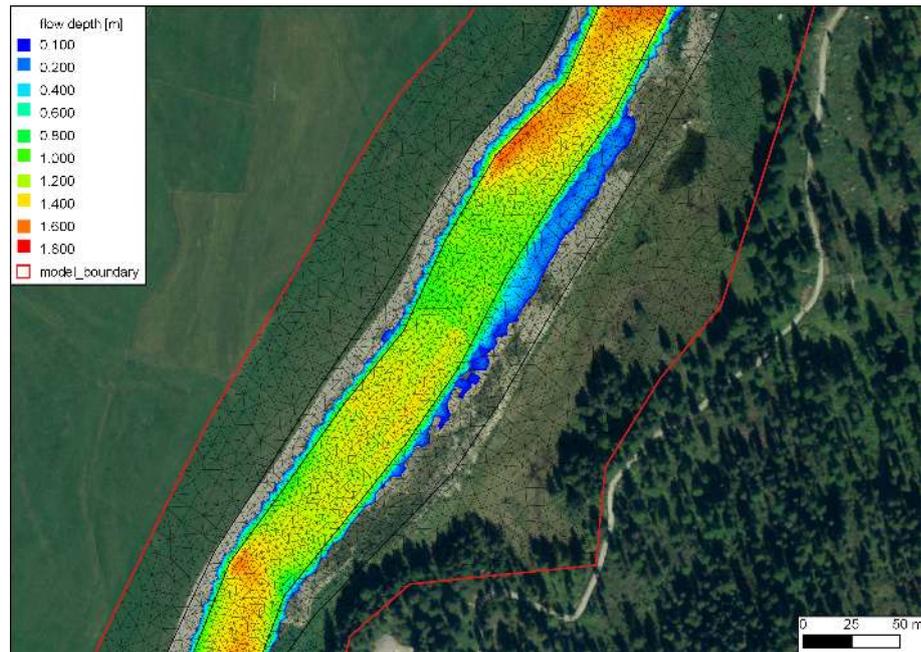


Figure 5.12 Maximal flow depth of the unsteady flow simulation observed at the flood peak after a run time of about 9 hours.

```

    type = boundary_history
    boundary_values = ( Q )
    history_one_file = yes
    output_time_step = 400
  }
}

```

Open the command file *Flaz_hydraulic_transient.bmc* either by double-clicking or via the menu of the BASEMENT GUI (File → Open Command). Run the simulation with the *Run* button of the BASEMENT window. The maxima values of the flow depths and flow velocity vectors can be visualized using QGIS Crayfish as shown in Figure 5.12 and Figure 5.13.

5.4.3 Calibration of the hydraulic model

The hydraulic model can be calibrated for example based on flood level marks by comparing the modelled water surface elevations with the flood level marks. Usually the calibration parameter is the bed roughness introduced with the Strickler value. The calibration procedure may need several adjustments and is an iterative process. The demonstration of the calibration is not part of this tutorial. It should be mentioned that it is important to have a calibrated hydraulic model either for further hydraulic modelling or for morphological modelling in a further step.

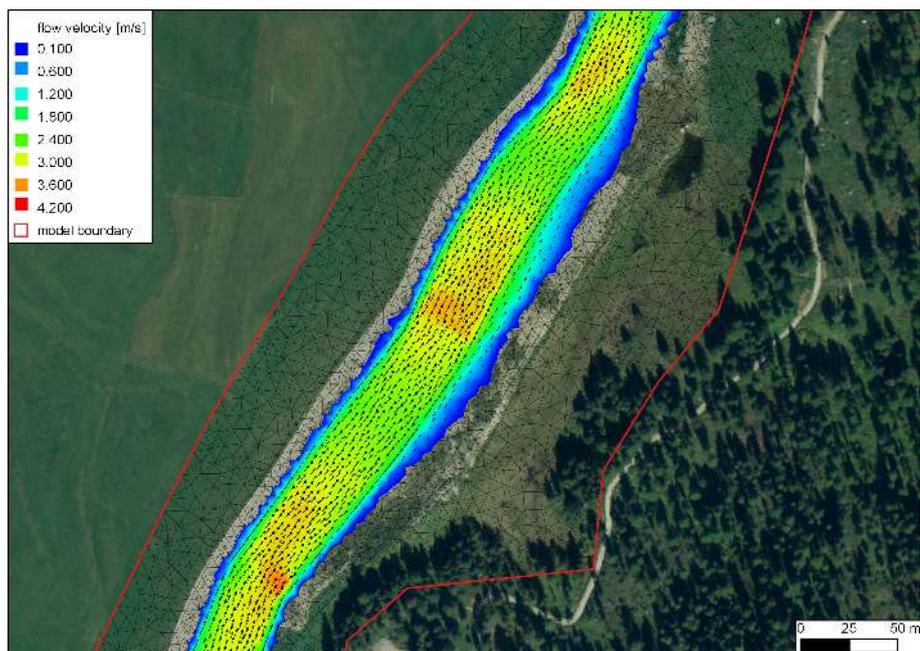


Figure 5.13 Maximal flow velocity and velocity vectors of the unsteady flow simulation observed at the flood peak after a run time of about 9 hours.

5.5 Morphological simulation with single-grain bed load transport

The MORPHOLOGY-block is not compulsory. If this block is not defined the simulation is purely hydraulic. The command file of the unsteady hydraulic simulation has to be completed for the single-grain bed load transport as shown in this section.

The morphological simulation is based on the flood event in July 2004. Therefore a single-grain bed load transport is added to the unsteady hydraulic simulation in Section 5.4.2. In the HYDRAULIC-block a small change has to be done in order to define the boundary string *Inflow_sed* for the bed load inflow. Thus a new STRINGDEF-block is added within the GEOMETRY-block as follows:

```

GEOMETRY
{
  ...
  STRINGDEF
  {
    name = Inflow_sed
    node_ids = (3 194 192 210 4)
    upstream_direction = left
  }
  ...
}

```

5.5.1 Define the morphological information

The necessary information for the morphological part of the simulation is defined in the *MORPHOLOGY*-block.

```
MORPHOLOGY
{
  PARAMETER
  {...}
  INITIAL
  {...}
  BEDMATERIAL
  {...}
  BEDLOAD
  {...}
  GRAVITATIONAL_TRANSPORT
  {...}
}
```

5.5.1.1 Morphological parameters

In the *PARAMETER*-block important parameters for the morphological simulation are defined. The bed load control volume is chosen to be constant with a thickness of 0.1 m.

```
PARAMETER
{
  porosity = 40 // [%]
  density = 2650 // [kg/m3]
  control_volume_type = constant
  control_volume_thickness = 0.1 // [m]
}
```

5.5.1.2 Initial conditions

The initial bed elevation is defined in most cases as the actual topography.

5.5.1.3 Bed material

In the *BEDMATERIAL*-block the grain classes, the composition, the thickness of the soil layers, the level of the fixed bed and the assignment of the soil to the mesh is defined in several sub-blocks.

```
BEDMATERIAL
{
  GRAIN_CLASS
  {...}
  MIXTURE
```

```

    {...}
    SOIL_DEF
    {...}
    FIXED_BED
    {...}
    SOIL_ASSIGNMENT
    {...}
}

```

5.5.1.3.1 Grain size distribution

The single-grain simulation is performed with only one grain class of a given diameter, e.g. the mean grain diameter.

```

GRAIN_CLASS
{
    diameters = ( 50 ) // [mm]
}

```

5.5.1.3.2 Grain mixture

Since we have only one grain size, the volume fraction is equal to 100%.

```

MIXTURE
{
    name = single_grain
    volume_fraction = ( 100 ) // [%]
}

```

5.5.1.3.3 Define the soil composition

The soil layers and the according sediment mixture are defined in the SOIL_DEF-block. For a single-grain simulation it is not important how many layers are defined. The negative bottom elevation defines the thickness of the layer. Below the last layer a fixed bed is assumed. If no LAYER-block is defined, then automatically a fixed bed on the surface is assumed. We use this especially for the river bed near the upper boundary condition to avoid uncontrolled erosion. Furthermore the embankments are kept fixed because the main focus is on the river bed morphology. The two soils *soil_fix_20* and *soil_fix_40* are defined to have a gradual transition from the fixed bed to the movable bed. Anyway, the river section with the roughness elements cannot be modelled accurately, because single roughness elements which are more or less fixed stones cannot be discretized within the computational mesh. They have to be modelled with an increased bed roughness instead.

```

SOIL_DEF
{
    name = soil_element_roughness
    LAYER
    {
        bottom_elevation = -0.8 // fixed bed 0.8 m below the surface
    }
}

```

```

    mixture = single_grain
  }
}
SOIL_DEF
{
  name = soil_widening
  LAYER
  {
    bottom_elevation = -2.0      // fixed bed 2.0 m below the surface
    mixture = single_grain
  }
}

SOIL_DEF
{
  name = soil_alt_bars
  LAYER
  {
    bottom_elevation = -2.0      // fixed bed 2.0 m below the surface
    mixture = single_grain
  }
}

SOIL_DEF
{
  name = soil_fix_20
  LAYER
  {
    bottom_elevation = -0.2      // fixed bed 0.2 m below the surface
    mixture = single_grain
  }
}

SOIL_DEF
{
  name = soil_fix_40
  LAYER
  {
    bottom_elevation = -0.4      // fixed bed 0.4 m below the surface
    mixture = single_grain
  }
}

SOIL_DEF
{
  name = soil_fix              // fixed bed
}

```

5.5.1.3.4 Fixed bed elevation

There are several possibilities to define a fixed bed. In the FIXED_BED-block, the elevations of areas with fixed bed can be defined either with a separate mesh file containing

the fixed bed elevations or with specific fixed bed elevations for some selected nodes. Furthermore a fixed bed can be implemented in the SOIL_DEF-block (Section 5.5.1.3.3). If there is no layer defined, a fixed bed will be assumed. In any case, a fixed bed is assumed below the last layer. In this tutorial the FIXED_BED-block is used as an example to define a fixed bed for a single node. This can be used to consider a big stone for example. A fixed node (node id 8956) is implemented by giving *zb_fix* a value smaller or equal to -100.

```
FIXED_BED
{
  type = nodes
  node_ids = ( 5486 )
  zb_fix = ( -100 )
}
```

5.5.1.3.5 Assignment of the defined soil types

The soil types defined in the SOIL_DEF-blocks (Section 5.5.1.3.3) are assigned to the elements of the mesh by the material index.

```
SOIL_ASSIGNMENT
{
  type = index_table
  index = ( 1 2 3 4 5 6 7 8 9 10 11 12 )
  soil = ( soil_element_roughness soil_fix soil_fix soil_widening
  soil_fix soil_widening soil_alt_bars soil_fix soil_fix soil_fix
  soil_fix_20 soil_fix_40 )
}
```

5.5.1.4 Bedload

In the BEDLOAD-block all needed data for bedload transport as well as boundary conditions are defined in several sub-blocks.

```
BEDLOAD
{
  PARAMETER
  {...}
  FORMULA
  {...}
  BOUNDARY
  {...}
  DIRECTION
  {...}
}
```

5.5.1.4.1 Bed load parameter

The control parameters for the bed load simulation are defined in the PARAMETER-block. Since the *limit_bedload_wetted* tag is turned off, the bed load is computed not only in completely wetted cells but in partially wetted cells as well. Averaging the bedload fluxes over the sediment cells is turned off. This option is computational less expensive and less diffusive.

```
PARAMETER
{
    limit_bedload_wetted = off
    use_cell_averaged_bedload_flux = off
}
```

5.5.1.4.2 Bed load formula

The bed load transport is computed with the Meyer-Peter and Mueller's (mpm) formula.

```
FORMULA
{
    bedload_formula = mpm
    bedload_factor = 0.4
    theta_critic_index = ( 1 2 3 4 5 6 7 8 9 10 11 12 )
    theta_critic = ( 0.04 0.04 0.04 0.04 0.04 0.04 0.04 0.04 0.04 0.04 0.04 0.04
0.04 0.04)
}
```

5.5.1.4.3 Bed load boundary condition

The bed load input is handled with a boundary condition which determines the transport capacity at the inflow cross section. The *IODown* is the only downstream boundary condition available for sediment transport at the moment. All sediment entering the last computational cell will leave the cell over the downstream boundary.

```
BOUNDARY
{
    type = transport_capacity
    string_name = Inflow_sed
    mixture = single_grain
    factor = 0.7
}
```

```
BOUNDARY
{
    type = IODown
    string_name = Outflow
}
```

5.5.1.4.4 Bedload direction

This block contains general settings dealing with the adaption of the bedload transport direction. The lateral transport caused by a *lateral_bed_slope* with respect to the main flow direction is taken into account. The *lateral_index* defines the regions of the computational mesh, where lateral transport should be considered.

DIRECTION

```
{
  lateral_transport_type = lateral_bed_slope
  lateral_transport_factor = 1.5
  lateral_index = ( 1 2 3 4 5 6 7 8 9 10 11 12 )
}
```

5.5.1.5 Gravitational transport

In the GRAVITATIONAL_TRANSPORT-block the parameters for gravitation induced transport are defined. The gravitational transport can be limited to elements which are fully wetted or can be considered for all elements. Over the material index the scope and the applied angles for the gravitational transport can be defined. Note that for soils with a fixed bed the gravitational transport is not active. In this model gravitational transport is applied only for the mesh elements with index 6. This enables river bed widening due to bank collaps on parts of the right embankment of the widening zone.

GRAVITATIONAL_TRANSPORT

```
{
  index = (6)
  angle_failure_dry = (30)
  angle_failure_wetted = (15)
  angle_failure_deposited = (10)
  gravity_transport_on_cells = partially wetted
  angle_wetted_criterion = partially_wetted
}
```

5.5.2 Define the output

The desired output of the simulation has to be defined explicitly in the OUTPUT-block. The *output_time_step* defines the time steps of the results. The *console_time_step* defines the time step to appear in the BASEMENT window during simulation. Specific output modes have to be defined in the repeatable SPECIAL_OUTPUT-blocks. Inside this block the *output_time_step* defines the output time step for this particular output. A detailed overview of all possible output types, values, format types and more can be found in the [Input block structure](#).

OUTPUT

```
{
  console_time_step = 1000
  SPECIAL_OUTPUT
```

```

{
    type = BASEviz
    output_time_step = 5
}
SPECIAL_OUTPUT
{
    format = sms
    type = node_centered
    values = ( depth wse velocity deltaz z_node )
    output_time_step = 1000
}
SPECIAL_OUTPUT
{
    type = balance
    balance_values = (sediment timestep)
    output_time_step = 1000
}
SPECIAL_OUTPUT
{
    type = element_history
    element_ids = (3517 10729)
    output_time_step = 1000
    element_values = (depth velocity z_element)
}
SPECIAL_OUTPUT
{
    type = boundary_history
    boundary_values = ( Q Qsed )
    history_one_file = yes
    output_time_step = 1000
}
}

```

5.6 Perform morphological simulation with single-grain bed load transport

Open the command file *Flaz_single_grain.bmc* either by double-clicking or via the menu of the BASEMENT GUI (File → Open Command). Run the simulation with the Run button of the BASEMENT window. Be aware that the mesh, command file and all other input files have to be in the same folder. The defined outputs are now generated in the same folder as the command file.

The output files with the ending *.sol* can be visualized using QGIS Crayfish. The bed elevation after the flood event is shown in Figure 5.14. Two cross sections are defined and the bed elevation along these cross sections before and after the flood event is compared using the QGIS plugin *Profile tool* (Figure 5.15). The morphological changes (*deltaz*) due to the flood event are shown in Figure 5.16. The position of the simulated depositions matches quite well with the position of the banks observed in the aerial image.

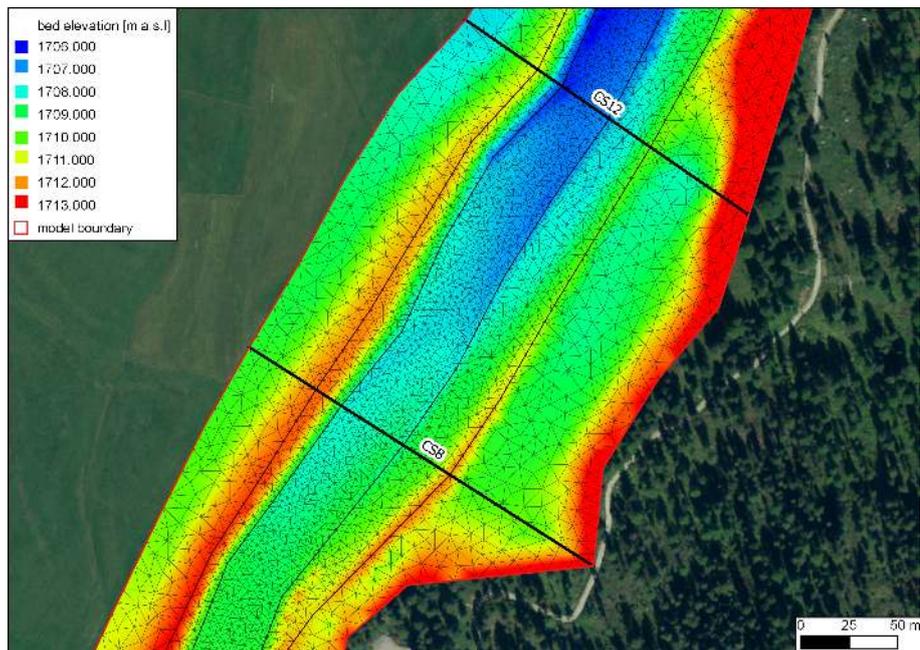


Figure 5.14 Modeled bed elevation (z_{bed}) and two cross sections defined in the widening part.

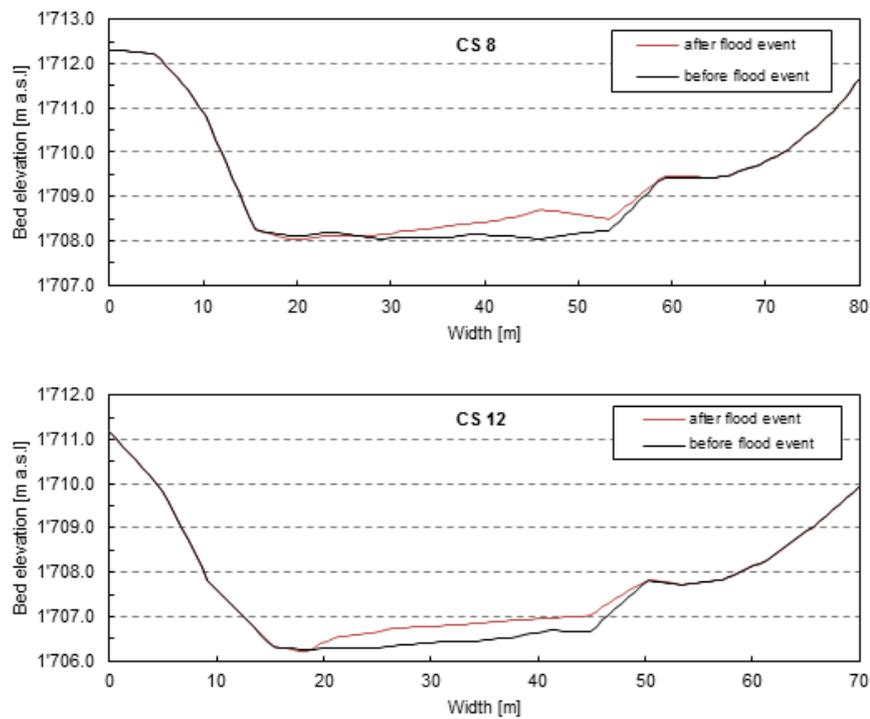


Figure 5.15 Comparison of the river bed before and after the flood event in cross-section CS 8 and CS 12.

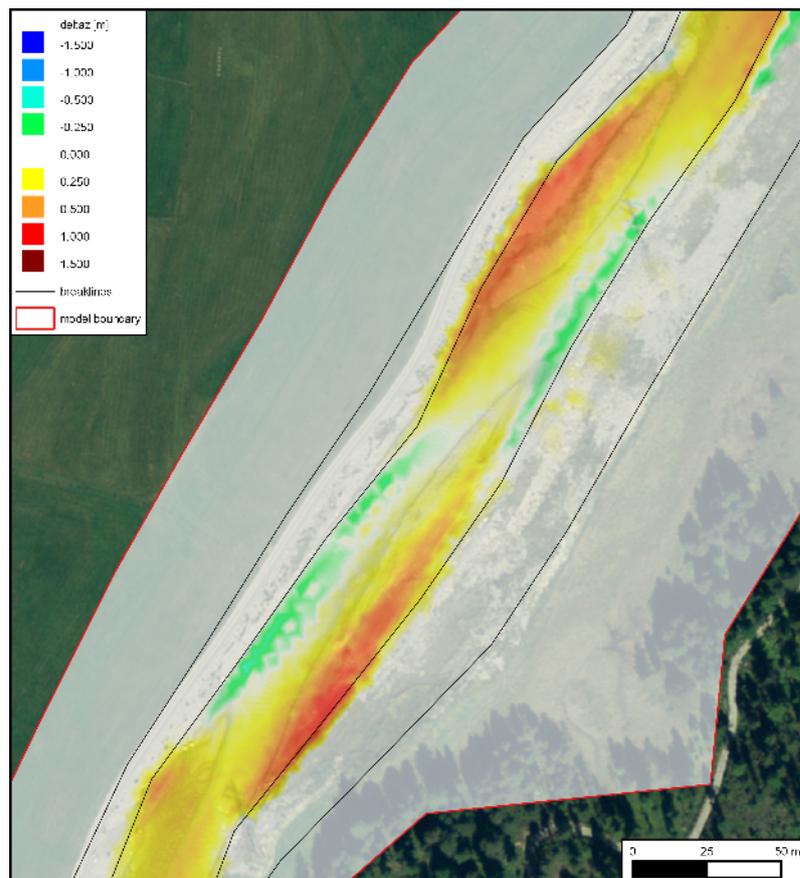


Figure 5.16 Changes of the morphology (deltaz) due to the flood event with the single-grain model. The red colour range represents deposition and the green/blue colour range shows erosion.

5.7 Morphological simulation with multi-grain bed load transport

In order to avoid needless duplication compared to the single-grain simulation just the modifications of the command file are pointed out. Basically there is the possibility to use the grain size distribution to determine the bed friction in the FRICTION-Block. To simplify matters the friction is defined with the Strickler value. Generally it is suggested to try both options and to choose the most suitable for your model purpose.

5.7.1 Morphological parameters

In the PARAMETER-block important parameters for the morphological simulation are defined. In multi-grain simulations the thickness of the bed load control volume is an important calibration parameter. This parameter influences significantly the grain sorting process.

```
PARAMETER
{
    ...
    control_volume_type = constant
    control_volume_thickness = ( 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 ) // [m]
    control_volume_thickness_index = ( 1 2 3 4 5 6 7 8 9 10 11 12 )
}
```

5.7.2 Grain size distribution

The grain size distribution is discretized with six grain classes. They have to be defined in ascending order from the smallest to the largest grain.

```
GRAIN_CLASS
{
    diameters = ( 1 5 15 44 82 150 ) // [mm]
}
```

5.7.3 Grain mixture

In the MIXTURE-block the volume fraction of the different mixtures are defined. The three river sections are considered with different sediment mixtures. Furthermore a mixture for the inflow is defined.

```
MIXTURE
{
    name = mixture_inflow
    volume_fraction = ( 15 15 23 27 10 10 )
}
```

```

MIXTURE
{
    name = mixture_element_roughness
    volume_fraction = ( 17 11 14 27 14 17 )
}
MIXTURE
{
    name = mixture_widening
    volume_fraction = ( 21 13 16 25 11 14 )
}
MIXTURE
{
    name = mixture_alt_bars
    volume_fraction = ( 27 14 14 20 14 11 )
}

```

5.7.4 Define the soil composition

The soil layers with the corresponding sediment mixture are defined in the SOIL_DEF-block. The soil can be defined with several layers of different material, but to keep it simple we assume a single layer. The negative bottom elevation defines the thickness of the layer. Below the last layer a fixed bed is assumed. If no LAYER-block is defined then automatically a fixed bed on the surface is assumed. We use this especially for the river bed near the upper boundary condition to avoid uncontrolled erosion. Furthermore the embankments are kept fix because the main focus is on set on the river bed morphology. The two soils *soil_fix_20* and *soil_fix_40* are defined to have a gradual transition from the fixed bed to the movable bed.

```

SOIL_DEF
{
    name = soil_element_roughness
    LAYER
    {
        bottom_elevation = -0.8    // fixed bed 0.8 m below the surface
        mixture = mixture_element_roughness
    }
}
SOIL_DEF
{
    name = soil_widening
    LAYER
    {
        bottom_elevation = -2.0    // fixed bed 2.0 m below the surface
        mixture = mixture_widening
    }
}
SOIL_DEF
{
    name = soil_alt_bars

```

```

    LAYER
    {
        bottom_elevation = -2.0    // fixed bed 2.0 m below the surface
        mixture = mixture_alt_bars
    }
}
SOIL_DEF
{
    name = soil_fix    // fixed bed
}
SOIL_DEF
{
    name = soil_fix_20
    LAYER
    {
        bottom_elevation = -0.2    // fixed bed 0.2 m below the surface
        mixture = mixture_element_roughness
    }
}

SOIL_DEF
{
    name = soil_fix_40
    LAYER
    {
        bottom_elevation = -0.4    // fixed bed 0.4 m below the surface
        mixture = mixture_element_roughness
    }
}

```

5.7.5 Bed load boundary condition

The bed load input is regulated with a boundary condition which determines the transport capacity at the cross section defined. The factor for the bed load at the boundary is an important calibration parameter and depends on the transport formula. Therefore this factor is different for single-grain and multi-grain simulations. The outflow boundary is handled as in the single-grain simulation.

```

BOUNDARY
{
    type = transport_capacity
    string_name = Inflow_sed
    mixture = mixture_inflow
    factor = 0.7
}
BOUNDARY
{
    type = IODown

```

```

    string_name = Outflow
}

```

5.7.6 Bed load formula

For the sediment transport computation different bed load transport formulas are available. In this tutorial the formula of Meyer-Peter and Mueller for multiple grain classes is chosen. It is suggested to try different sediment transport formulas.

```

FORMULA
{
    bedload_formula = mpm_multi
    bedload_factor = 0.5
}

```

5.7.7 Define the output

The desired output of the simulation has to be defined explicitly in the OUTPUT-block. The specific output is defined in the repeatable SPECIAL_OUTPUT-blocks. For the multi-grain simulation some additional output may be interesting such as for example the grain size distribution in selected nodes. This way grain sorting effects can be observed. A detailed overview of all possible output types, values, format types and more is given in help buttons in the Command File Editor of BASEMENT.

```

OUTPUT
{
    console_time_step = 1000
    SPECIAL_OUTPUT
    {
        format = sms
        type = node_centered
        values = ( depth deltaz z_node )
        output_time_step = 1000
    }
    SPECIAL_OUTPUT
    {
        type = node_history
        node_values = ( grain_size )
        node_ids = (1138 3235)
        history_one_file = yes
        output_time_step = 4000
    }
    SPECIAL_OUTPUT
    {
        type = boundary_history
        boundary_values = ( Q Qsed )
        history_one_file = yes
    }
}

```

```
    output_time_step = 1000
  }
}
```

5.8 Perform morphological simulation with multi-grain bed load transport

Open the command file *Flaz_multi_grain.bmc* either by double-clicking or via the menu in BASEMENT (File → Open Command). Run the simulation with the Run button in the BASEMENT window. Be aware that the mesh, command file and all other input files have to be in the same folder. The defined outputs are now stored in the same folder as the command file.

The morphological changes *deltaz* are shown in Figure 5.17. Here the multi-grain model is not compared quantitatively with the single-grain model. Nevertheless the qualitative comparison is indicating a quite similar behaviour (Figure 5.16 and Figure 5.17). At this state much more details could be investigated such as the grain class fractions, the hiding-and-exposure function (*hiding_exponent*), the amount of grain classes etc. Further important calibration parameters are the critical dimensional shear stress, the bed load factor and the bed load inflow controlled with the bed load factor at the boundary.

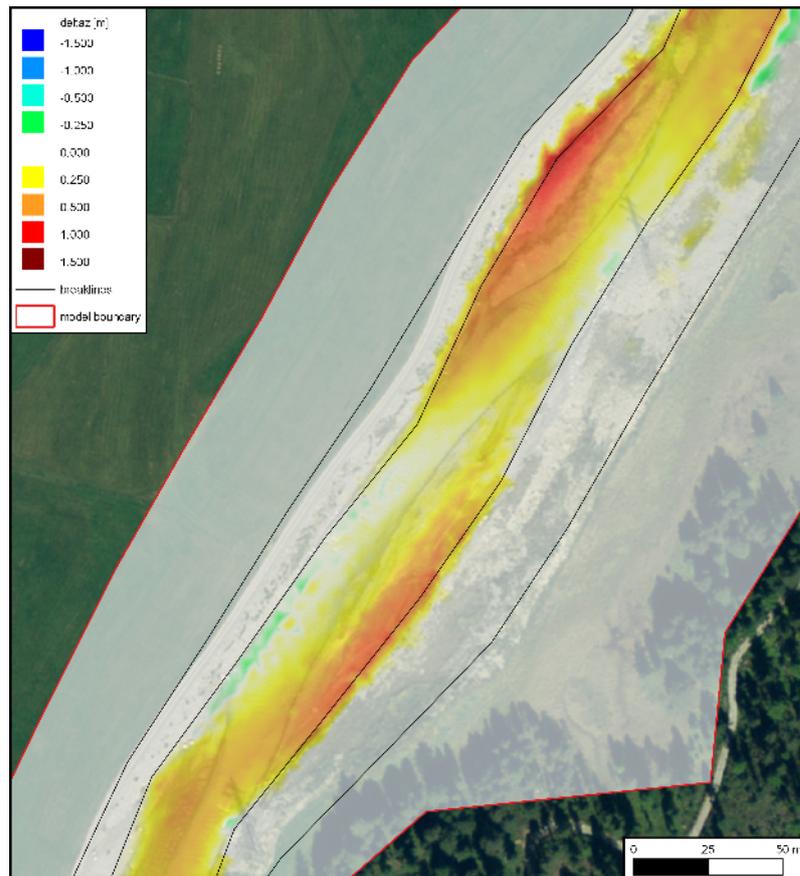


Figure 5.17 Changes of the morphology (deltaz) due to the flood event with the multi-grain model. The red colour range represents deposition and the green/blue colour range shows erosion

Laterally coupled 1D-2D hydrodynamic simulation

6.1 Introduction

BASEMENT offers the possibility to combine BASEchain (1D) and BASEplane (2D) domains in coupled simulations. So-called “laterally coupled” simulations follow the idea to simulate the channelized river flow with a 1D approach. The overland flow in the floodplains, in contrast, is simulated with a 2D approach. Using this coupling concept, the advantages of 1D models (like efficiency, simplicity, representation of 1D flow structures (weirs, gates)) and the advantages of 2D models (like differing flow directions and flow paths, complex topographies) shall be combined.

Coupled 1D / 2D simulations are an interesting alternative modelling approach especially in case of spatially extended scenarios with a clearly defined main channel and overflow into 2D floodplains with complex flow paths or topographies (like urban areas). However, be aware that this concept also suffers from the limitations of both model types (like the 1D flow assumptions, neglect of momentum exchange, etc.) and should be applied with caution.

The model coupling takes place laterally along the 1D river. If the water level exceeds the left or right river dyke crest, the water overtops the dyke and flows into the 2D floodplain. The water exchange is internally modelled by external sources, i.e. the water is removed from the 1D model and added to the 2D model as external sources, thereby neglecting the momentum exchange. In a similar way, water from the 2D floodplain flows back, if its water level exceeds the dyke crest and the water level of the 1D model.

Water exchange takes place at defined connections between 2D boundary edges and 1D cross sections. The overtopping flow is determined using simple 1D weir formulas. To apply these weir formulas, water levels are needed from both models. The 1D water level is hereby interpolated between two cross-sections and the 2D water level is taken from the corresponding 2D cell.

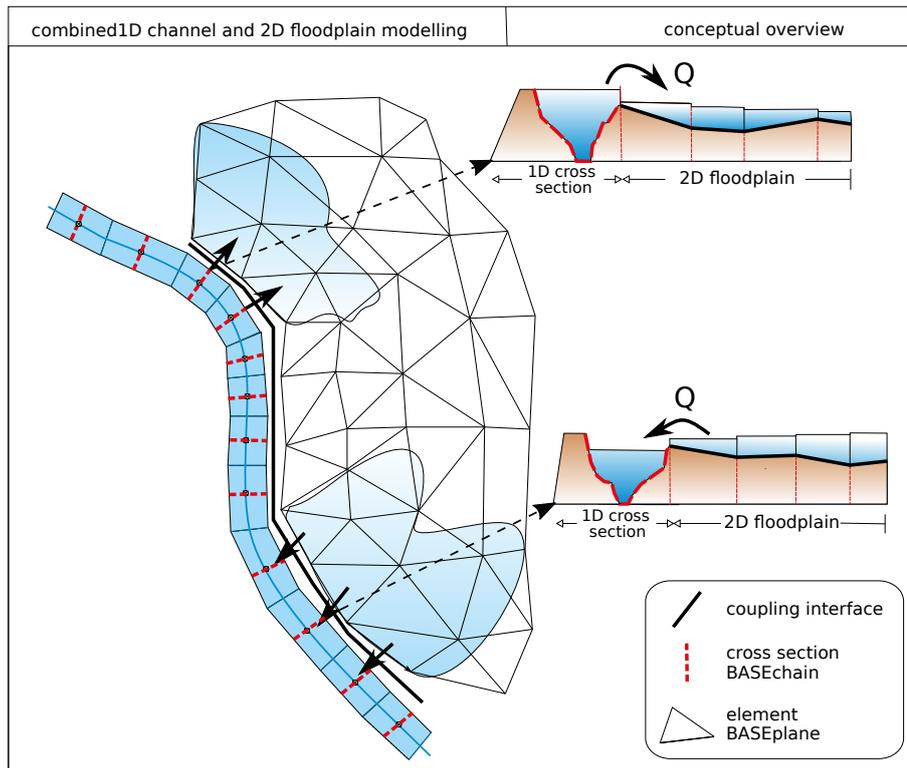


Figure 6.1 Conceptual sketch of lateral coupling of 1D BASEchain (river channel) and 2D BASEplane (overland flow) sub-domains. Water is exchanged as weir overflow over the river dykes.

This tutorial briefly introduces the use of this lateral coupling approach based on a simple scenario. Only hydraulic computations are performed, since the lateral coupling supports no morphological computations up-to-now. The tutorial makes use of the QGIS-plugin BASEmesh (see BASEmesh tutorial Section 1.1) for visual checks, which is recommended for setting-up lateral coupling simulations.

6.2 Set-up of command file

6.2.1 General remarks on mesh creation

Using the 1D / 2D coupling, parts of the domain are represented with the 1D model and other parts with the 2D model. It is important hereby, that both meshes do not overlap, since this would create artificial and unphysical storage volumes and flow paths. In a similar way, there should be no gaps between the model meshes. In this tutorial, both model meshes are adapted to fit to each other along the right river dyke as illustrated in Figure 6.2.

6.2.2 BASEchain (1D) river model

At first, the 1D model is set-up, analogous to the steps of the Thur tutorial (see Section 3.1) outlined above.

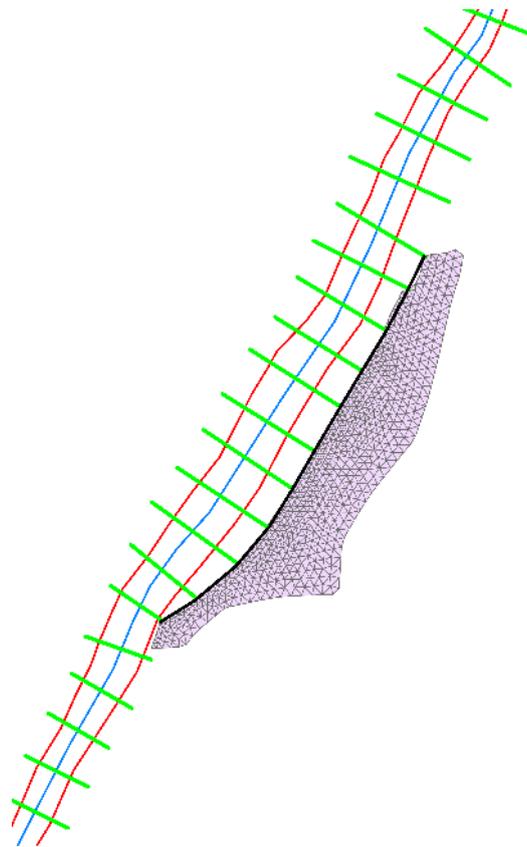


Figure 6.2 1D cross-sections (green lines), 2D mesh (gray) and dyke break-line (black).
Cross-sections and 2D mesh are connected along the river dyke, without gaps or overlapping areas. Flow direction is from bottom left to top right.

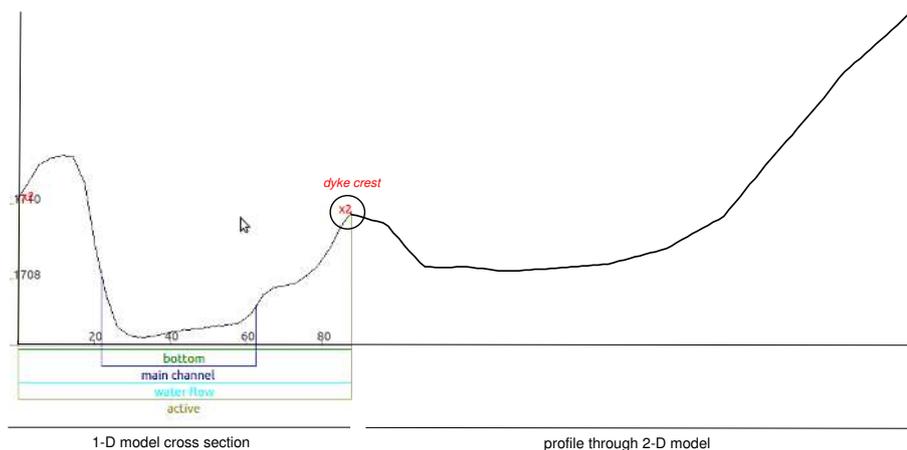


Figure 6.3 Connection of 1D cross section (left) with 2D model (transversal profile, right) at the dyke location. Water overtopping takes place via the coupling connection.

```
BASECHAIN_1D {
  region_name = river_flaz_1D
  GEOMETRY {
    ...
  }
  TIMESTEP {
    ...
  }
  HYDRAULICS {
    ...
  }
}
```

The topography file (*.bmg) contains the cross section points and the main channel and active range definition and was obtained here by cutting profiles through a digital terrain model. Of special importance for the lateral coupling is the accurate definition of the active range (defined by the left and right dykes), because the water exchange into the 2D floodplain takes place along these dykes. Figure 6.3 shows the definition of the right dyke in the 1D cross-section and the connection to the 2D mesh (shown as transversal profile).

The adaptation (“fitting”) of the model meshes along the river dyke was achieved by cutting the 1D cross-sections (*active_range*) at the river dyke location and by using the dyke as break-line for the boundary of the 2D model.

In this scenario, a steady-state inflow of $700 \text{ m}^3/\text{s}$ is defined for the 1D model using a hydrograph inflow boundary. This discharge is large enough to overtop the dyke and to flood the 2D floodplain. The outflow boundary is modelled as zero-gradient boundary. The simulation is run until steady-state conditions are reached, which are used later on as initial conditions for the coupled simulation.

6.2.3 BASEplane (2D) floodplain model

The floodplain outside of the main river channel is modelled with a 2D BASEplane sub-domain. Figure 6.4 illustrates the corresponding 2D mesh.

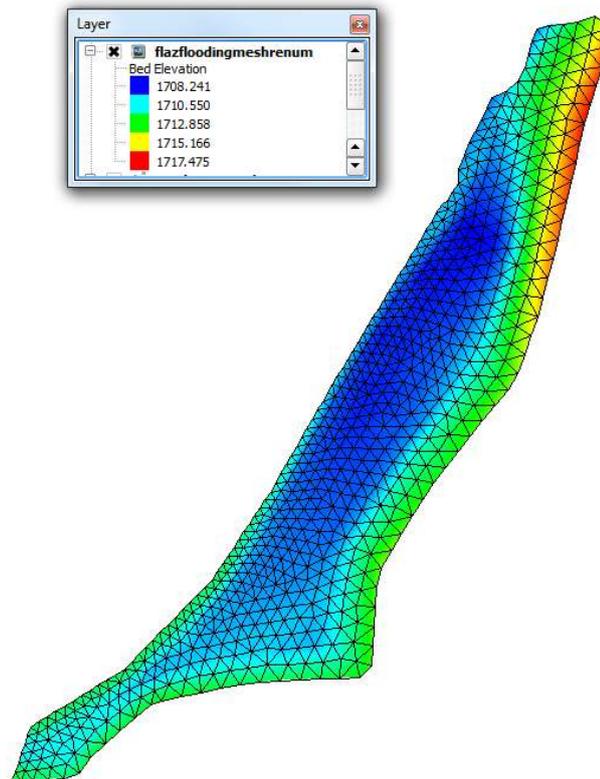


Figure 6.4 Unstructured 2D mesh of the floodplain located to the right of the 1D river channel.

The 2D floodplain model is quite simple to set-up – it has neither inflow boundaries nor outflow boundaries, i.e. water cannot enter or leave the domain, except via the lateral coupling. The model domain is dry at simulation start.

```

BASEPLANE_2D {
  region_name = floodplain_flaz_2D
  GEOMETRY {
    ...
  }
  HYDRAULICS {
    ...
  }
  TIMESTEP {
    ...
  }
}

```

6.2.4 Lateral coupling set-up

The command file of the coupled simulation must contain both, the BASEchain and BASEplane sub-domain definitions. These are coupled with each other by defining an additional COUPLINGS-block:

```

DOMAIN {
  BASECHAIN_1D {
    ...
  }
  BASEPLANE_2D {
    ...
  }
}
COUPLINGS {
  PARAMETER {
    max_time_level = 1
  }
  COUPLING_LATERAL {
    basechain = flaz_river_1D
    baseplane = flaz_floodplain_2D
    connections_filename = connections.dat
    my_weir = 0.6
    side_weir_factor = 0.95
    weir_from_dyke1D = no
  }
}
}

```

The COUPLINGS-block has just two inner blocks in this scenario. The PARAMETER-block is used to define the time level at which the exchange takes place (for details see Section 5.2 in the User manual). Usually, this parameter is set to 1 for lateral coupling, meaning that both sub-domains are executed with the same (minimum) time step size and water exchange takes place after each time step. This setting is recommended for lateral coupling to prevent or mitigate oscillations.

The second inner block is the COUPLING_LATERAL-block. Its main settings are the names of the coupled BASEchain and the BASEplane sub-domains and the filename of the coupling connections file (see Section 6.3). Additional parameters are the empirical coefficients for the weir / side-weir formulas. Furthermore, one must specify whether the weir crest elevation (= the dyke crest) is taken from the 2D mesh elevations (*weir_from_dyke1D* = *no*, default) or is interpolated in between the 1D cross-sections at the connection location. Determining the weir crest elevation from the 2D mesh topography is often advantageous, especially if the 2D mesh topography is based on high-resolution DEM models. Interpolating between 1D cross-sections, in contrast, may neglect local depressions or dyke openings.

6.3 Coupling connections between 1D and 2D subdomains

6.3.1 Definition of coupling connections

The dynamic interactions between the 1D and 2D models take place via a list of defined coupling connections. These connections are defined by the name of a 1D cross-section and the corresponding node numbers of the connected 2D boundary edge. Cross-sections thereby may have multiple 2D boundary edges attached to it (1:n-relation), whereas each 2D boundary edge can only be connected to a single cross-section (1:1-relation).

The geometrical connections must be listed in an ASCII-file, which is read at the start of the simulation. The file contains the following information and has the shown data format:

Cross section name	dyke location	nodeID 1	nodeID 2
CS1	1	44	67
CS1	1	67	43
CS2	1	47	46
...			

1. Column = name of cross-section
2. Column = location of the dyke (0 = left, 1 = right; seen in flow direction)
3. Column = node 1 (of 2-D boundary edge)
4. Column = node 2 (of 2-D boundary edge)

In principle, it is possible to define these connections manually by filling out the connections file. In practice, however, this is usually unfeasible, due to the large number of connections along the river reach. To ease the determination of the coupling connections between both meshes, BASEMENT offers an automatic detection mechanism.

6.3.2 Automatic generation of coupling connections

The automatic detection and generation of coupling connections is based on a search algorithm. It searches from the left and right dyke of each 1D cross section for the nearest 2D boundary edges within a specified distance.

```
COUPLING_LATERAL {
    ...
    connections_automatic = yes
    connections_distance = 30
    connections_index = ( 6 )
    ...
}
```

To use the automatic detection, insert the tag `connections_automatic = yes` in the `COUPLING_LATERAL` block. Furthermore, you need to set a `connection_distance`, which defines the search distance in which it is searched for 2D boundary edges (If you choose the distance too small, some 2D boundary edges may not be connected to a cross-section!). In addition, you can limit the detection of 2D boundary edges to a list of cell material indices. Using this option, you can easily prevent some areas from being automatically connected and from taking part in the lateral coupling by setting corresponding material indices.

The general workflow of the automatic connection detection is as follows:

1. Open the command file in BASEMENT and set the tag: `connections_automatic = yes`.

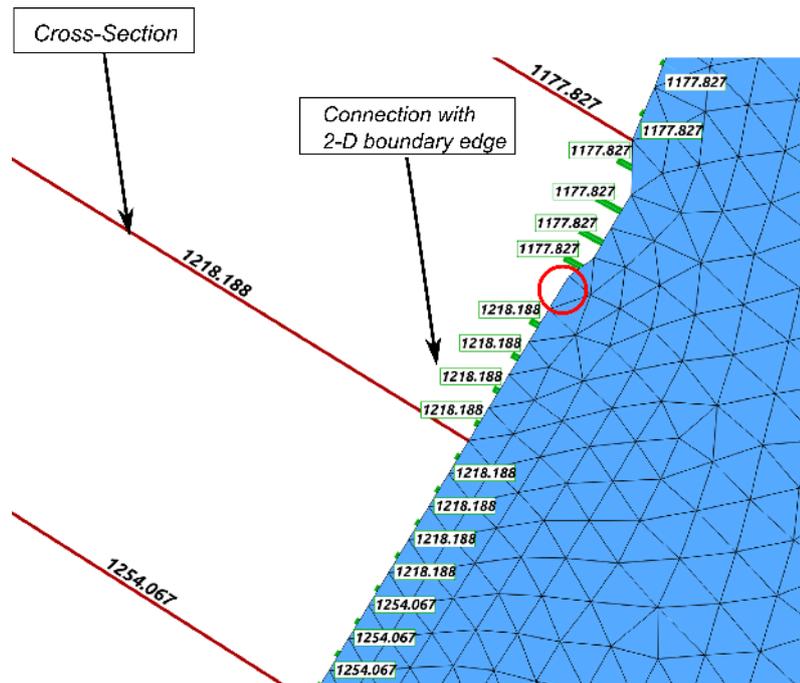


Figure 6.5 1D cross sections (red) connected to 2D boundary edges (blue mesh) via coupling connections (green). The labels show the cross-section names and indicate the corresponding mapping of the 2D edge \rightarrow 1D cross-section. The red circle indicates a 2D boundary edge without a coupling connection, which is not taking part in the lateral coupling.

2. Run BASEMENT. The program will automatically search for the connections and write the detected connections to file and then terminates. Three files are written:
 - lateral_connections_created.dat = the coupling connection file,
 - lateral_connections_stats.shp = shapefile for visualizing the generated connections using GIS-software,
 - lateral_connections_stats.txt = tab-delimited text file for visualizing the generated connections with other software.

Next, you should visually check if the generated connections are correctly set-up, e.g. using the free QGIS-software with the BASEmesh plugin:

1. Open QGIS with installed BASEmesh plugin,
2. Load the 1D river topography file, .bmg (*Plugins \rightarrow BASEmesh \rightarrow Import mesh*),
3. Load the 2D floodplain mesh file, .2dm (*Plugins \rightarrow BASEmesh \rightarrow Import mesh*),
4. Load the generated shapefile with the connection data, *lateral_connection_stats.shp* (*Layer \rightarrow Add Vector Layer*) and label the layers with the cross-section name,
5. Check if all 2D boundary edges are connected or if some should have no connections and modify the connections file accordingly.

The connections are shown as green lines in Figure 6.5 and are drawn from the middle of each 2D boundary edge orthogonal to the imaginary line between the dykes of two

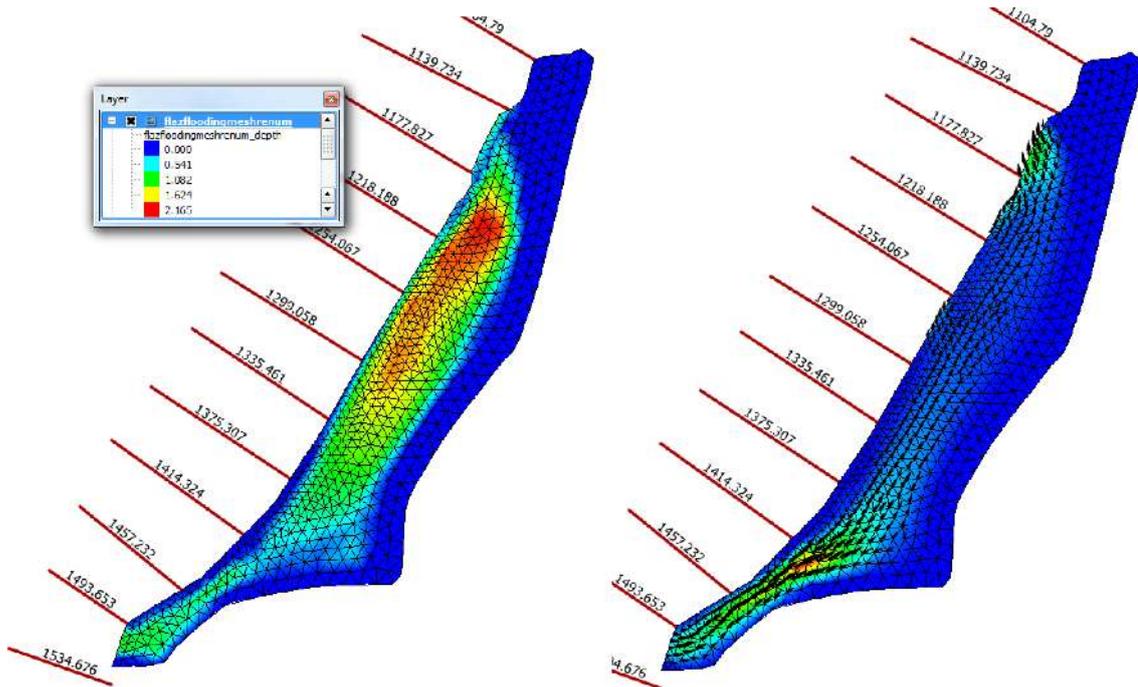


Figure 6.6 Computed water depths (left) and velocity vectors (right) in the floodplain at steady-state conditions. Water enters the floodplain at the lower left and leaves it at the upper right.

neighbourred cross-sections. (These green lines should be as short as possible in order to minimize gaps and overlapping areas of the 1D and 2D meshes).

If the connections are set-up correctly in the connections file, then enter its name in the corresponding tag in the COUPLING_LATERAL-block and de-activate the automatic generation of connections:

```
connections_filename = lateral_connections_created.dat
connections_automatic = no
```

6.4 Perform coupled simulation

Run the simulation with the BASEchain and BASEplane sub-domains connected via the COUPLINGS-block. Be aware that the mesh files, the command file, the coupling connections file and all other input files have to be in the same folder. The defined outputs will also be written into the same folder. The computed water depths (left) and velocities (right) of the steady-state simulation are shown in Figure 6.6. The water enters the floodplain at the lower left, flows through the floodplain and leaves it at the upper right by overtopping the dyke into the river channel.

To check if the water mass balance is fulfilled, we can simply check the inflow and outflow of the 1D model. Both must have the same value, since the water discharges leaving and entering the 1D model over the dykes cancel out to zero during steady-state conditions.