SYSTEM MANUALS

of **BASEMENT**

SYSTEM MANUAL

of *BASEMENT*

CONTENTS

Preface

Preface to Versions 1.0 − 1.3

The development of computer programs for solving demanding hydraulic or hydrological problems has an almost thirty-year tradition at VAW. Many projects have been carried out with the application of "home-made" numerical codes and were successfully finished. The according software development and its applications were primarily promoted by the individual initiative of scientific associates of VAW and financed by federal instances or the private sector. Most often, the programs were tailored for a specific application and adapted to fulfil costumer needs. Consequently, the software grew in functionality but with little documentation. Due to limited temporal and personal resources to absolve an according project, a single point of knowledge concerning the details of the software was inevitable in most of the cases.

In 2002, the applied numerics group of VAW was invited by the Swiss federal office for water and geology (BWG, nowadays Swiss Federal Office for the Environment FOEN) to offer for participation in the trans-disciplinary "Rhone-Thur" project. With the idea to build up a new software tool based on the knowledge gained by former numerical codes - while eliminating their shortcomings and expanding their functionality - a proposal was submitted. The bidding being successful a partnership in terms of co-financing was established. By the end of 2002, a newly formed team took up the work to build the so-called "BASic EnvironMENT for simulation of environmental flow and natural hazard simulation – BASEMENT".

From the beginning, the objectives for the new project were ambitious: developing a software system from scratch, containing all the experience of many years as well as state-of-the-art numerics with general applicability and providing the ability to simulate sediment transport. Additionally, professional documentation is a must. As to meet all these demands, a part wise reengineering of existing codes (Floris, 2dmb) has been carried out, while merging it with modern and new numerical approaches. From a software-technical point of view, an object-oriented approach has been chosen, with the aim to provide reusability, reliability, robustness, extensibility and maintainability of the software to be developed.

After four years of designing, implementing and testing, the software system BASEMENT has reached a state to go public. The documentation at hand confirms the invested diligence to create a transparent software system of high quality. The software, in terms of an executable computer program, and its documentation are available free of charge. It can be

used by anyone who wants to run numerical simulations of rivers and sediment transport – either for training or for commercial purposes.

The further development of the software tends to new approaches for sediment transport simulation, carried out within the scope of scientific studies on one hand side. On the other hand, effectiveness and composite modelling are the goals. On either side, a reliable software system BASEMENT will have to meet expectations of the practical engineer and the scientist at the same time.

Minor

em. Prof. Dr.-Ing. H.-E. Minor
Member of the steering committee of Rhone-Thur Project 2002-2007
Director of VAW, 1998-2008

October, 2006

Preface to Versions 1.4

The work since the first release of the software in October 2006 was exciting and challenging. To go public is paired with interests and demands of users – although user support for the software never was intended. But interchange with users is definitely one of the most crucial factors of successful software development. Feedback from academic or professional users conveys a different point of view and enables the development team to achieve costumer proximity as well as to consolidate experience. Accordingly, the project team tried to meet the demands as effectively as possible. In version 1.3 of BASEMENT, which was released in April 2007, there were some errors fixed, a few new features added and the documentation was completed. Since then, many things have changed: on the personnel, on the project as well as on the software technical level.

In summer 2007 one of our main software developers, Dr. Davood Farshi, left VAW and changed to an international hydraulic consultant. Dr. Farshi supported our team from 2002 to 2007 as a profound numeric specialist and was mainly involved in the development of BASEplane. At his own request, he is still engaged in the development of BASEMENT as external advisor and tester. Dr. Farshi's position in the project team was reoccupied by Christian Volz, an environmental engineer from southern Germany. Mr. Volz has broad experience in numerical modelling as well as object-oriented programming.

On the project level the framework slightly changed. The initial scope within BASEMENT was developed, the "Rhone-Thur" project, has been finalized by the end of 2007. The sequel is called "Integrales Flussgebietsmanagement". It has the same co-financer as its predecessor, namely the Swiss federal office for the environment (FOEN), and basically the same participating institutions (EAWAG, WSL, LCH(EPFL) and VAW(ETHZ)). The funding runs until the end of 2011. Due to the retirement of Prof. Dr.-Ing. H.-E. Minor in summer 2008, our laboratory is solely represented in the project committee by Dr. R. Fäh at the moment.

The emphases of the new proposal for the further development of BASEMENT are advanced topics of hydraulics and sediment transport, such as secondary currents and lateral erosion. Furthermore, the efficiency of the software should be increased by the implementation of appropriate parallelisation and coupling approaches.

Since the last minor release a long time passed, which was mainly consumed by a general revision of the software. After five years of development a diligent consolidation was expedient. In addition, the coincidence of a new team member offered an unbiased reflection of the source code. All in all it was very worthwhile.

Last but not least, there are numerous bugs fixed and some new features in the current version. Mainly the efficiency of the software has been improved. The first stage of parallelisation is completed. The current implementation of the code includes the OpenMP interface which allows for parallel execution of the basic computation loops. In other words, the software is now able to exploit the power of current multi-core processors with a convincing speedup. Furthermore, the revision of some data structures and output routines as well as the application of an optimised compiler led to a reduction in execution time.

Concerning sediment transport, the one-dimensional model BASEchain now supports the modelling of fine material, either as suspended or bed load. Also the advanced models for boundary conditions are worth mentioning. On the one hand, it is now possible to model domain boundaries with momentum and on the other hand, special boundary conditions inside the computational region, such as a weir or a gate, are implemented. The fact, that

the version 1.4 of BASEMENT is also available for the Linux operating system the first time, rounds off the new additions and features of the software package at hand.

Summarised one may say that the release 1.4 of BASEMENT is a major release due to all the different kinds of changes, but it's still a minor release concerning the new features – let's call it a "major minor" release. We are looking forward to Version 2.0 of BASEMENT, which is planned for next year.

D. Vetsch
Project Supervisor

October, 2008

Preface to Version 2.0

Four years ago, in spring 2006, the first version of the software system BASEMENT was completed and ready for internal use. In autumn of the same year, the first official version 1.1 of the software was released and made available as free download on the project website www.basement.ethz.ch. Since then, the functionality of the program has been enhanced and the international user community has grown gradually. Over the last years, BASEMENT has become a reliable tool for professional investigations, especially within the scope of flood prevention, and for scientific studies. Furthermore, the software is part and parcel of the lecture "Numerical Models in Hydraulic Engineering" to ensure education of young engineers in the field of hydrodynamic numerical simulation. The lecture is held on a regular basis by VAW staff for master students of civil and environmental engineering at ETH Zurich.

In February 2009, I have become the successor of Prof. em. Dr.-Ing. H.-E. Minor as Director of the Laboratory of Hydraulics, Hydrology and Glaciology (VAW) at ETH Zurich. In the meantime, I have joined the project committee "Integrales Flussgebietsmanagement"as a further representative of VAW besides Dr. R. Faeh.

Furthermore, there are some changes concerning the personnel of the project team of BASEMENT to mention. Lukas Vonwiller joined the team last autumn after having obtained his master's degree at ETH Zurich. Within the scope of his master thesis at the VAW, he studied the hydrodynamics and ecological impact of floods at the river Flaz using BASEMENT. Some of his experiences with the application of BASEMENT and selected results are documented in the new tutorial on 2-D simulations in the user manual UIV. His current duties are the application and testing of the software in terms of project work. We were also very lucky being able to engage Dr. Ratko Veprek as a distinguished software engineer for a limited period of time. His contributions to the software, such as flow control of river systems, computational efficiency and the graphical user interface, just to name a few, are of great value. Unfortunately he will leave us by the day of the release to take on a post doctoral position abroad.

According to the announcement in the preface to version 1.4, the second major version of BASEMENT is released with little delay but with all the more important improvements and substantial new features. First of all, the new version 2.0 of the program comes with a graphical user interface (GUI), which allows running or stopping simulations and tracking the progress. Furthermore, the model setup and configuration, i.e. the assembling of the command file, is completely integrated into the GUI. The user is guided through the setup and any input is validated directly. In addition, the integrated help function, which is based on the command file reference, provides detailed information on the meaning of input parameters. This gives way to a clearer model setup compared to the rather fault-prone manual text editing, which is still available and also accessible through the GUI. Another main feature of the new GUI is the editing of the topography for BASEchain. Besides the GUI based setup, interpolation and thinning out of model cross sections, a graphical viewer helps the user to check the configuration and subdivision. For this reason, the new version of BASEMENT comes with its own topography file format for BASEchain. The new format has a clear structure similar to the style of the command file.

Moreover, the visualisation of actual results during a simulation with BASEviz has been improved and is now more interactive, i.e. the simulation can be paused, continued or the variable shown can be switched. Other improvements concern computational efficiency and

sediment transport, especially gravitational bed load transport. Please refer to the release notes in the section "introduction and installation" of this manual for further details about new features and bug fixes.

The software system BASEMENT in its current version 2.0 has reached the point to be termed as a state of the art numerical modelling tool for flow and sediment transport in rivers. The incorporated well established or new numerical approaches, software technical features like parallelization or the coupling of sub domains, advanced features for sediment transport and flow control are making it a reliable tool for professional as well as scientific applications. With the new GUI another hurdle has been cleared and a new era of the software in terms of usability has begun. We are looking forward to the further development as well as upcoming releases of BASEMENT and we are curious about how the software will establish itself in the future.

Prof. Dr. R. Boes
Committee Member of Project "Integrales Flussgebietsmanagement"
Director of VAW

May, 2010

Preface to Version 2.8

End of an era

More than 11 years ago, the first official version 1.1 of the software system BASEMENT was released and has been made available as free download on the project website www.basement.ethz.ch. Since then, the functionality of the program has been enhanced and the Swiss as well as the international user community has grown gradually. Over the last years, BASEMENT has become a reliable tool for professional studies, especially within the scope of flood prevention and morphodynamics, and for research at universities. Furthermore, the software is part and parcel of different lectures at ETH Zurich to ensure education of young engineers in the field of hydro- and morphodynamic numerical modelling. The lectures are held on a regular basis by VAW staff for master students of civil and environmental engineering at ETH Zurich.

With version 2.8, an era of BASEMENT development comes to an end. During the last 2 years, the software has been rewritten from scratch to make it more efficient and to allow for using new technologies like general purpose graphics possessing units (GPGPUs). After successful testing of the prototype, version 3.0 is almost ready and will be released in the upcoming months. Therefore version 2.8 will be the last of its kind (i.e. no version 2.9 but maintenance updates and bug fixes will be released as versions 2.8.x). However, version 3.0 will not have all the features of version 2.8 right from the start. Thus version 2.8 remains the working horse for many applications and will be long-term supported.

The current version contains important improvements and substantial new features. First of all, the software environment (i.e. third party libraries) was upgraded to most recent versions and to 64bit to avoid compatibility problems with new hardware and operating system versions such as MS Windows 10 and Ubuntu 18. In doing so, the stability of the GUI on high-DPI devices was improved. Furthermore, a vegetation model was added to BASEplane that affects flow resistance and erodibility related to growth. Several improvements related to morphodynamics were made, e.g. updated and new transport formulae and internal sediment boundary conditions.

We hope to maintain good user experience and wish you effective simulations.

In the name of the project team
Dr. D. Vetsch
Project Director

May, 2018

This page has intentionally been left blank.

Preamble

Credits

Project Team

Software Development, Documentation and Test (alphabetical)

M. Bürgler, MSc. ETH Environmental Eng.
F. Caponi, MSc. Environmental Eng.
Dr. D. Conde, MSc. Civil Eng.
E. Gerke, MSc. ETH Civil Eng.
S. Kammerer, MSc. ETH Environmental Eng.
Dr.techn. M. Weberndorfer, MSc.

Scientific Board

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

Former Project Members

See https://www.basement.ethz.ch/people

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

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., Bürgler M., Caponi F., Ehrbar D., Facchini M., Faeh R., Farshi D., Gerber M., Gerke E., Kammerer S., Koch A., Mueller R., Peter S., Rousselot P., Vanzo D., Veprek R., Volz C., Vonwiller L., Weberndorfer M. 2020. System Manuals of BASEMENT, Version 2.8.1 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, 2020. http://www.basement.ethz.ch

For Software:

BASEMENT – Basic Simulation Environment for Computation of Environmental Flow and Natural Hazard Simulation. Version 2.8.1 © ETH Zurich, VAW, 2006-2020.

License

<div align="center">

BASEMENT SOFTWARE LICENSE

between

ETH

Rämistrasse 101

8092 Zürich

Represented by Prof. Dr. Robert Boes

VAW

(Licensor)

and

Licensee

</div>

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

BASEMENT uses third party software. For instance, the BASEMENT executable directly links the following external libraries:

- CGNS
- HDF5
- Qt5 (non-cluster version only)
- Qwt (non-cluster version only)
- Shapelib
- TecIO
- VTK (non-cluster version only)

The libraries (and their dependencies) are included in the BASEMENT distribution if they are not provided by the operating system.

Please refer to `ThirdPartySoftwareLicenses.txt` in the distribution and/or the operating system documentation for the third party software licenses and copyright notices. The external libraries for Windows 10 have been built using `vcpkg` version 2020.07 (`HDF5` was compiled without `szip`).

INTRODUCTION & INSTALLATION

of **BASEMENT**

Contents

1

Summary of Contents

1.1 Introduction

The intention of the BASEMENT system manual is to guide and support the user in carrying out numerical simulations with BASEMENT. The documentation, in particular the introductory and theoretical chapter, shall provide a deeper insight into the implemented numerical models, their application and the according preparation of necessary input data.

As a matter of fact, the application of state-of-the-art numerical models for channel flow, flood plain and natural hazard simulation is a demanding task. For this reason and despite of its incompleteness, this manual should provide helpful information to gain satisfying simulation results in terms of quality and time.

The BASEMENT system manuals are made up of four main parts – the user manual (denoted by the letter "U"), the reference manual (letter "R"), the tutorials (letter "T") and the test cases (letters "TC"). These parts are available as individual download packages. In the following, a short description of each main part is given.

1.2 Content of System Manuals

1.2.1 User Manual

The user manual mainly gives an introduction to the basic simulation environment and its application. All necessary input data and its preparation are discussed. The procedure to setup a running model and the command file, which strictly defines a scenario to be simulated, are described in detail herein.

The Basic Simulation Environment

The first part of the user manual describes the basic concepts and the components of the software system BASEMENT and gives a general introduction to its application. Get into

it and learn more about the meaning of "BASEchain" and "BASEplane".

Pre Processing

The careful preparation of input data, such as topographical data or a discharge hydrograph of a flood event, is crucial for the scope and the quality of the result of numerical simulations. This chapter provides basic information on how to obtain and prepare the necessary input data for the models. If you don't understand the terms "aspect ratio" or "SMS", this chapter is worth reading.

Model Setup

After the preparation of all input data, a simulation has to be setup. By the use of a command file, all necessary simulation parameters have to be set and the input data is linked therein. The command file can be viewed and edited using a graphical user interface (GUI). This chapter explains the usage of the GUI and its features.

1.2.2 Reference Manual

On the one hand, the reference manual covers the theoretical part of the implemented models of the software system BASEMENT. On the other hand, the detail syntax description of the input files, especially the command file, is given in this section of the manual – maybe the most important part for a proficient user of BASEMENT.

Mathematical Models

The implemented models for water flow and sediment transport are either physical conservation laws, such as the shallow water or Saint Venant equations, or empirically based balance equations, all in the form of partial differential equations. This chapter shows the so-called "governing equations" which constitute the available simulation models. To correctly describe the mathematically problem, the necessary boundary conditions and source terms are given also.

Numerics Kernel

Most often governing equations for flow can't be solved analytically for general boundary conditions. For this reason, the mathematical equations have to be discretized accordingly the desired special and temporary scale of the model. This section of the manual discusses the discretization methods, mainly the finite volume method, the explicit Euler scheme and especially the applied Riemann approach to solve the governing hyperbolic equation system.

1.2.3 Tutorials

This chapter guides the user through the necessary steps to setup a numerical model up working by example. With good reason, the chosen case studies are demanding in setup and

application. This part of the manual lives with your experience! The development team of BASEMENT is looking forward to enrich the tutorials section with your suggestions and practical know-how.

1.2.4 Test Cases

To validate the implemented numerical models and to check for their reliability, different well-known and well-documented test cases have been carried out. The models have been tested against analytical solutions and flume data. This chapter is mainly intended for experts, but also interesting for general users, who like to explore the capabilities and limits of the provided models.

2

Setup and First Start

2.1 System Requirements

2.1.1 Microsoft Windows

BASEMENT is available for the following MS Windows systems:

- Windows 7
- Windows 8
- Windows 10

Releases are tested on Windows 7, 8 and 10. According to reports from users of earlier releases, BASEMENT also runs on Windows Vista and Windows 2000.

For the latest news concerning new features and current changes, please visit the webpage http://www.basement.ethz.ch .

2.1.2 Linux

BASEMENT is available for the following Linux systems. The binary was compiled and tested on the following:

LINUX (x86-64):

- Ubuntu 14.04 (LTS), alias "Trusty Tahr":
    - Kernel version 3.13
    - GNU C Library (glibc) version 2.19
    - VTK-version: 5.8
- Ubuntu 16.04 (LTS), alias "Xenial Xerus":

- Kernel version 4.4
- GNU C Library (glibc) version 2.23
- VTK-version: 5.10

If you need a binary for a different version feel free to contact us. Furthermore, there is a binary without GUI and without real-time visualization (VTK). This binary should run on most (debian-based) linux systems and clusters.

2.1.3 Virtualization with Docker

If you want to run *BASEMENT* on any other operating system (OS) than the ones mentioned above (e.g. Mac OS X, non-debian linux distributions, etc.), we suggest to set up a virtual environment using the docker engine. Get a running version of the docker engine for your OS by following online installation instructions. We provide a Dockerfile to create the virtual environment and the corresponding *BASEMENT* binary on http://www.basement.ethz.ch for download.

2.1.4 Hardware Configuration

We recommend the following hardware configuration:

- single- or multi-core processors (x86/x86-64):

    - Intel (Xeon, Core 2, Pentium 4)
    - AMD (K7, K8, K10)

- RAM: 1 GB per core
- shared memory architecture

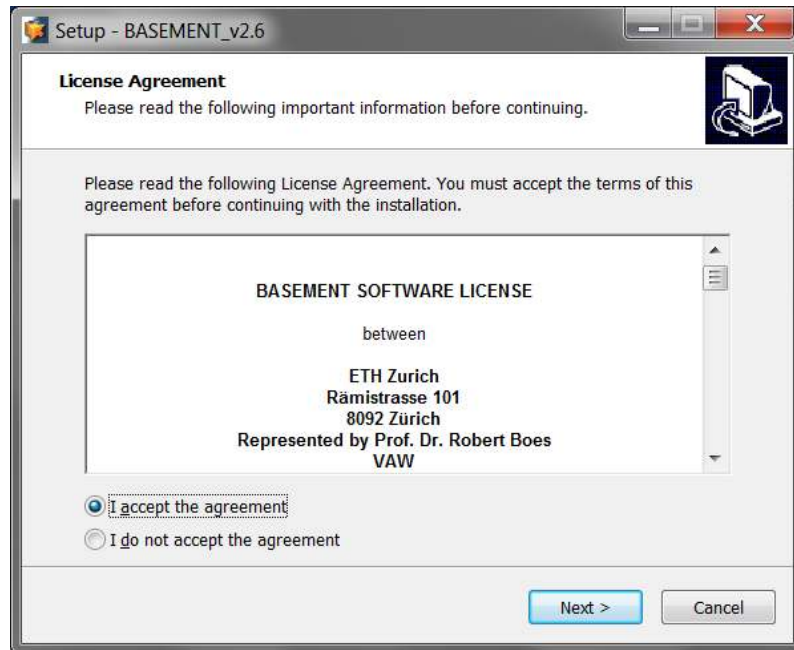Up to this moment, BASEMENT was successfully tested on 1 to 16 core Intel and AMD computers.

Figure 2.1 *License Agreement*

2.2 Installing under Windows

2.2.1 Getting the binaries

First of all, you need to get a copy of the recent software package. You can either download the most recent version from the projects webpage (http://basement.ethz.ch) free of charge or you can order the official manual as print version including a CD and a specially illustrated ring binder using the order form on the project webpage (only production costs will be charged, approx. CHF 100).

Then, you need to get a copy of the latest software package. Therefore go to the project webpage http://basement.ethz.ch and download the latest version free of charge. There are two different installers available. Choose the 'noadmin' to install BASEMENT when having no administrator privileges on your computer. Please note that with the 'noadmin' installer, double clicking on a BASEMENT command file (*.bmc) in the Windows Explorer for example will not open BASEMENT automatically. Therefore, no BASEMENT symbol will be displayed with BASEMENT command files.

The installation procedure remains the same for both regular and 'noadmin' installers.

2.2.2 Installation procedure

Before installing the latest version, please deinstall all previous versions on your computer. After downloading the latest version from the project webpage, start the installation by double-clicking on the BASEMENT installer.

Step 1: Accepting the License Agreement

Please read the License Agreement carefully and activate the 'I accept the agreement' button to accept the terms and conditions and proceed with the installation.
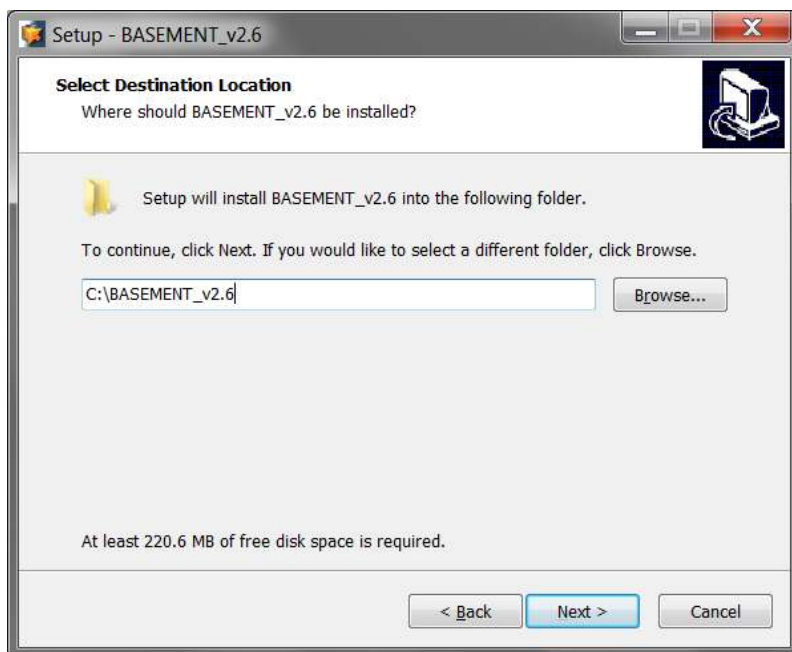
Figure 2.2 *Installation folder*

Step 2: Select the Installation Folder

After accepting the License agreement, you can choose where to install the binaries. Default proposition is "C:\Program Files (x86)\BASEMENT_vX.Y" respectively "C:\BASEMENT_vX.Y" for the noadmin version. You are free to choose any other directory.

The regular version is installed for all users whereas the noadmin version is installed for the current user only.

Step 3: Select program's shortcuts

Select a Start Menu folder to create the program's shortcut. Optionally an additional shortcut can be placed on the desktop and the application directory can be added to your system path (Figure 2.4). Please note that this is necessary in case you plan running BASEMENT in headless mode from command line for instance.

Step 4 and 5: Confirming and Finishing the Installation

Clicking 'Install' will start the installation process. After all files are copied, a final window (Figure 2.6) informs about the success of the installation. Click 'Finish' to close the installer and launch the program.

2.3   Running BASEMENT on Microsoft Windows

2.3.1   Running BASEMENT using the Graphical User Interface

After having successfully installed the program, you may notice the icon for BASEMENT on your desktop:
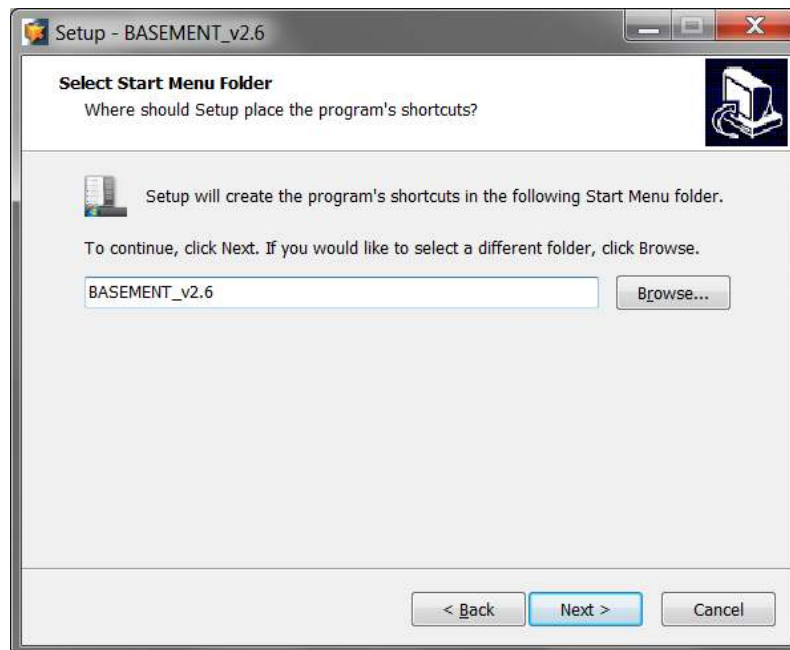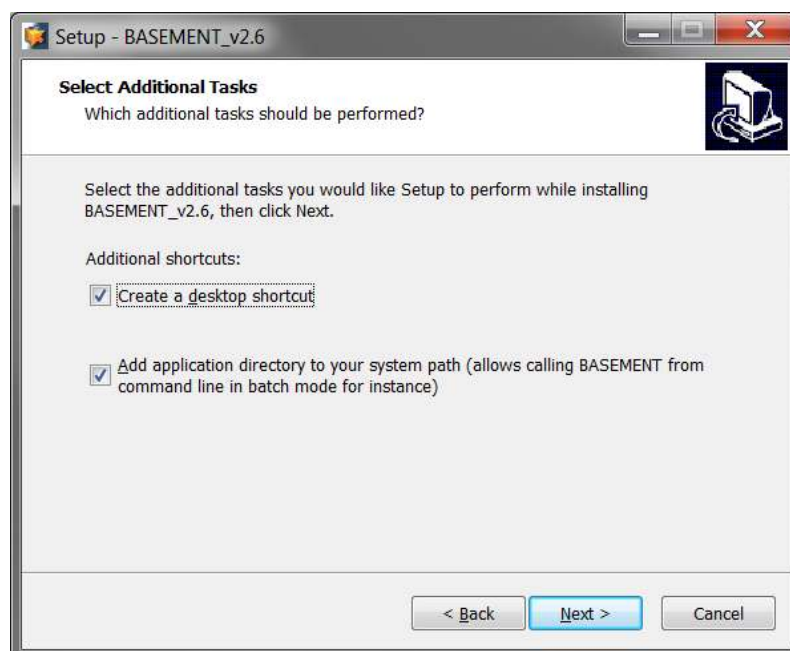
Figure 2.3 *Start Menu shortcut*
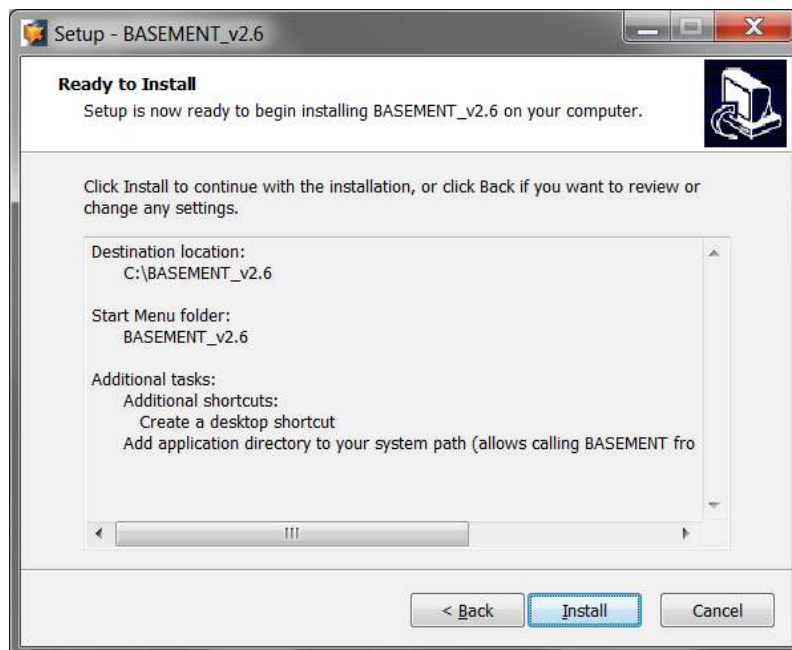

Figure 2.4 *Additional shortcuts*

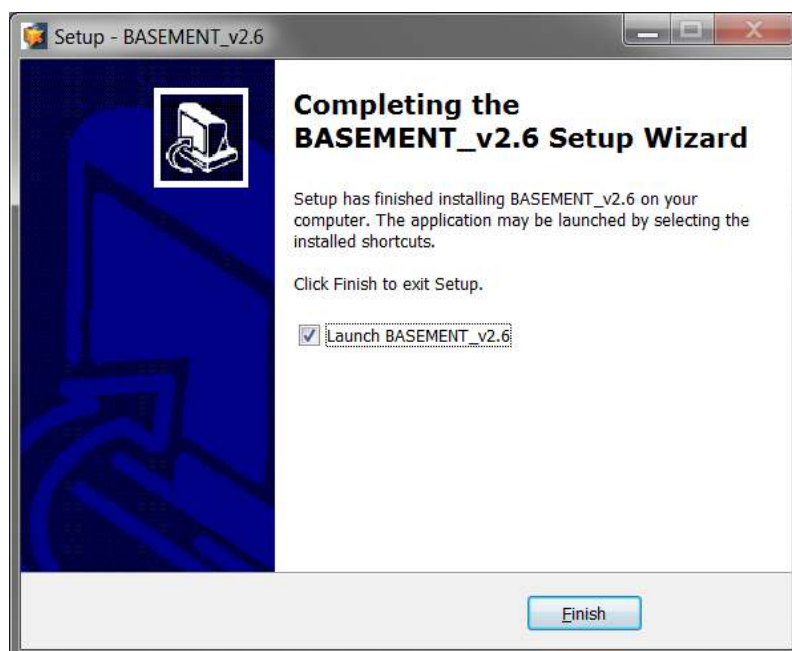**Figure 2.5** *Confirm installation*

Figure 2.6 *Installation complete*
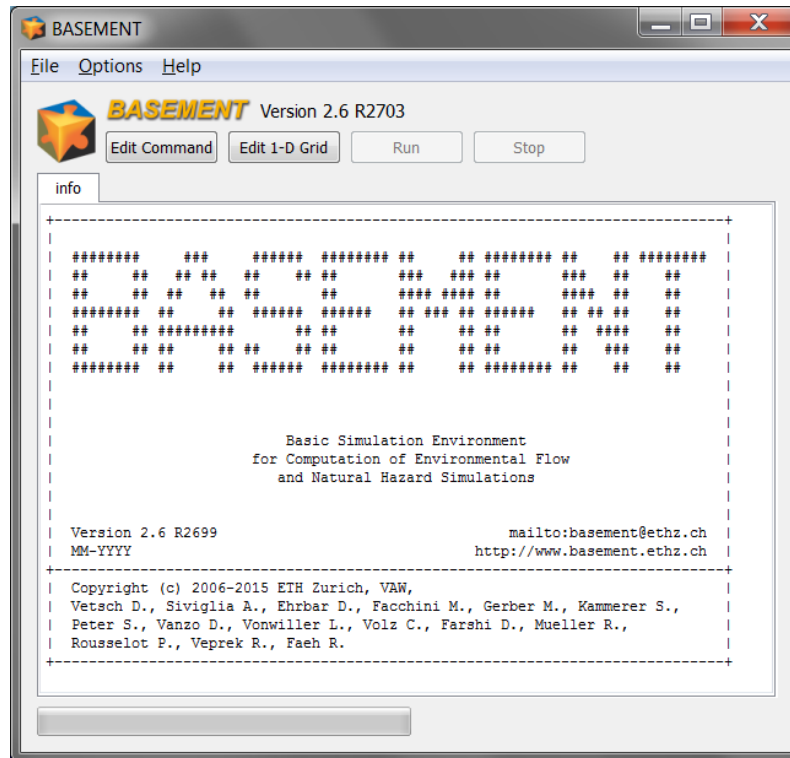

Figure 2.7 *BASEMENT icon*

Figure 2.8 *BASEMENT GUI*

This is the icon used for the program executable. The command files have a similar icon but are additionally identified by "CMD" within the logo. To get the icon for a command file displayed, the command file must end with ".bmc".

There are two ways to start the program. Either by clicking on the program symbol or directly by double clicking a command file (File with ".bmc" ending) for BASEMENT. Since version 2.0, BASEMENT runs as a standalone application including a simple graphical user interface (to run BASEMENT in batch mode see Section 2.3.2):

The difference between the two starting mechanisms however is simple: Starting the program by clicking the input file sets the path for the working folder and the scenarios name automatically according to the position of the input file and its content.

If you just start the program by clicking the executable, you have to select the scenario by pressing the "Edit command" button, which will open the command file editor. Using this editor, you can load ("Open File") your command file and thereby setting the working directory to the directory where the command file is located.

Once you loaded a command file it remains the active until you load the next one or save the current one with a new name.

When a command file is available to BASEMENT, the "Run" button is enabled. Pressing it starts the simulation. The console messages of BASEMENT are printed to the console view (Figure 2.9)

Below the console, the progress of the simulation is shown, together with the elapsed time and a guess of the required overall time. At the lower right, the real time speedup (RTS) of the simulation is shown. The real time speed up gives the number of seconds calculated in the simulation within one real time second.
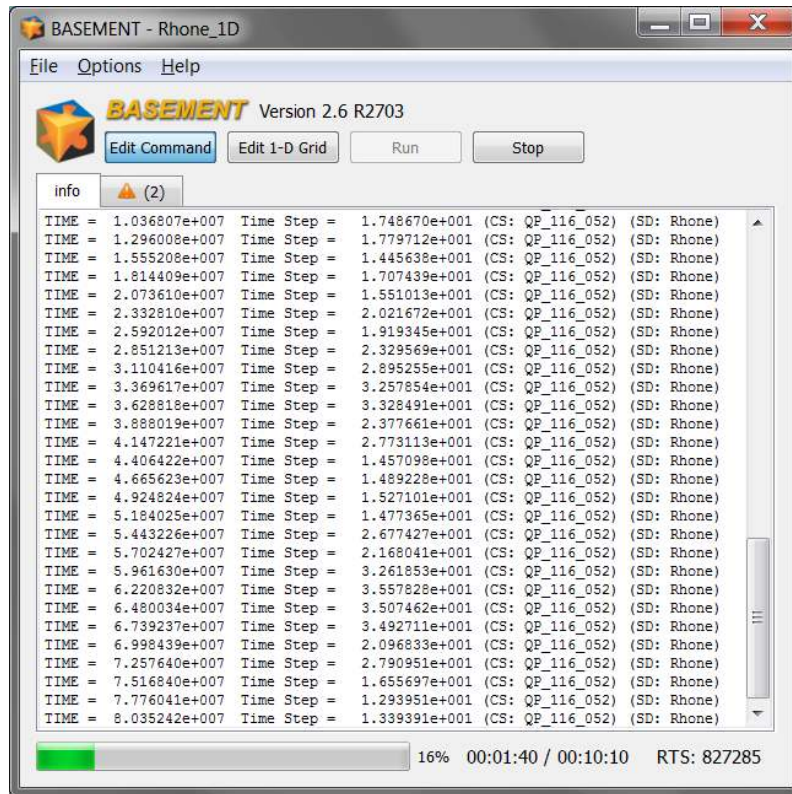
Figure 2.9 *BASEMENT GUI simulation*

All files containing the results will be saved in the working directory containing the command and auxiliary files.

Further details concerning the GUI of BASEMENT are explained in 'BASEMENT Graphical User Interface' of the user manual.

Notice:

The command filename must not contain any spaces or special characters like ä, ö, ü, è, etc.

2.3.2   Running BASEMENT on the Console

The following command line arguments are available, which can be also obtained in the command prompt (console) with the help command '-h' (typing BASEMENT_vX.Y.exe -h, whereas 'X.Y' has to be replaced by the actual version number):

- **version** display BASEMENT version (0 args)

- **h** display help information (0 args)

- **f "filename.bmc"** name of commandfile to be executed (1 arg)

- **t** pares input file and validate for errors (combination with –f)

- **b** no user inputs required (0 args)

- **verbose** detailed information outputs (0 args)

- **no_cls** no clear screen (0 args)

- **log** log level of output (1 arg, [2-6])

- **doc** output of documentation in html-format

Of particular interest is the possibility to run BASEMENT in the batch mode. The batch mode is started by calling the BASEMENT executable using the -b command line switch. The command line switch -b selects the batch mode and prevents the GUI to be started. The desired command file is defined using the -f command line switch.

In a batch file several simulations can be runned consecutively (for example over the weekend). To generate a batch file just create an empty text file and replace the ending '.txt' by '.bat'. In this file several command lines can be defined as for example:

```
"C:\Programs\Basement\BASEMENT_v2.2.1.exe" -f F:\Project_1\run.bmc -b
"C:\Programs\Basement\BASEMENT_v2.2.1.exe" -f F:\Project_2\run.bmc -b
"C:\Programs\Basement\BASEMENT_v2.2.1.exe" -f F:\Project_3\run.bmc -b
```

Then run the batch file by double clicking.

2.4 Installing under LINUX

2.4.1 Getting the binaries

You need to get a copy of the actual distribution as described in the Windows installation section. One can either download the most recent version from the projects webpage http://basement.ethz.ch or one has an official manual including a CD which can be ordered on the same webpage.

2.4.2 Installation procedure

The installation of program and documentation on LINUX systems is to be done via the console. Please replace all occurrences of 'X.Y' in the installation introduction by the actual version number.

Step 1: Preparation of the installation

Before unzipping the zipped installation file you probably want to create a temporary directory ('/tmp') and copy the downloaded file into this directory. Change to this directory where the downloaded zip-file is located. Unzip the file by typing:

```
unzip BASEMENT_vX.Y_linux.zip
```

Step 2: Installation of BASEMENT and license agreement

Under the unzipped files you can find an executable named 'setup' and an 'install.txt' which has additional information on the installation process. To start the installation, run setup

```
./setup
```

and follow the instructions. The license text of BASEMENT is displayed and must be agreed on to proceed with the installation. After running setup the password protected zip-file is extracted and the debian package file is available.

Step 3: Installation of debian (* .deb)-package

To ease the installation of BASEMENT a debian-package is created which automatically installs all needed files, the documentation and the tutorials and test cases on the pc. To install this package administrator rights are needed. Therefore on Ubuntu the 'sudo' command is needed. To start installation type

```
sudo dpkg -i basement-X.Y-XXX.deb
```

The debian package automatically detects if your configuration misses one of the needed requirements. In such a case you can use 'apt-get install ...' on Ubuntu systems to install the missing packages. Previous versions of basement are automatically detected by the installer and deleted before the installation starts.

If the installation of the package was successful, than the binary file is copied to

`/usr/bin`

and the program files, the documentation, the test cases and the tutorial are available under

`/usr/share/basement`

2.5   Running BASEMENT on LINUX

If the installation succeeded, start BASEMENT with its graphical user interface in a terminal window with a shell command prompt just by typing:

basement

Selecting an actual simulation, running a simulation on LINUX or using batch mode works the same way as it does on Windows (see Section 2.3.1 and Section 2.3.2).

Notice:

The command filename must not contain any spaces or special characters like ä, ö, ü, è, etc.

2.6   Installing under virtual environment

Follow the instructions to get *BASEMENT* successfully run in a virtual environment using the docker engine:

- open your shell where docker engine is running
- build user:basement image

```
$ docker build -t basement:user <pathToDownloadedDockerfile>/.
```

- run the container and mount a local volume, where you want to have stored the simulation data, into the container

```
$ docker run -it -v <absolutePathToMountVolume>:/home/basement/media
    --name basement-user basement:user
```

- you are now in a Ubuntu "Xenial" bash shell
- the password for the user 'basement' is 'BM#user-16'
- copy the downloaded zip-file *BASEMENT_vX.Y_docker.zip* to the local directory that is mounted into the container
- unzip the archive and run the setup

```
$ cd media
$ unzip BASEMENT_vX.Y_docker.zip
$ ./setup
```

- you should now find the *BASEMENT* binary ready for execution (without GUI and VTK)

```
$ ./BASEMENT_vX.Y
```

- you can exit the virtual environment and re-entering the other day simply by

```
$ docker start -i basement-user
```

3

Release Notes

3.1 Version 1.3

Subsystem BASEdomain:

A logical error check is performed between reading the input file and the start of the simulation. In case of logical errors (contradictions, missing information), the simulation will stop immediately. In any case, a file 'inputerror.err' is created, where all errors and warnings are listed for the current simulation. It is highly recommended to have a look at this file for every simulation being started.

Subsystem BASEchain:

- problem in computation of cross section area fixed
- problem with omission of values in topography file fixed
- sediment transport fixed
- new output: difference of water volume
- new output: difference of sediment volume
- bed load formula of "Rickenmann"
- output of topology of cross section not automatic any more
- output of sediment mixtures not automatic any more

Subsystem BASEplane:

- Computation of the inlet bed load from the bed load hydrograph has been modified.
- The jacobian matrix has been corrected to calculate the slopes for the quad. Elements.
- The coefficient "C" for the weir calculation has been improved!
- The output for the cell centres is modified. For this instead of the xyz output, a mesh file including the centres coordinates with the elements number are written.

- The flux over the weir has been corrected!

- The time step computation has been also extended to the boundary edges. In the previous version only the internal edges were considered for the time step computation.

- A new parameter "node_elevation_method" has been added to the Morphology-Parameter block. The calculation of the nodal elevation based on the "node_elevation_method" can be done with interpolation or slope method.

System Manuals:

- Part UIV has a new tutorial for 2-D simulations.

- Part RIV and corresponding structures in UIV are completely revised.

- Wrong assigned page numbers in some TOCs are corrected.

- Header of part "Index / Appendix" corrected.

- **Update your hardcopy version with the documentation patch.**

3.2   Version 1.4

General Notice:

All users working on a project with v1.3 at the moment should use v1.3 until the project ends and update for version 1.4 afterwards. Changes made to the boundary condition will lead to a slightly different behaviour of v1.4 compared to v1.3. Therefore, a calibrated scenario for v1.3 is not calibrated to work with v1.4!

Subsystem BASEparse:

- New command file structure allowing for distinction of bed load and suspended load in 1-D. The structure also changed in 2-D for similar reason.

- New structure of command file in HYDRAULICS: All friction parameters are now defined in their own block FRICTION instead of the SOURCE block as in v1.3

Attention: The changes of the command file's structure are not backwards compatible! Input files for v1.3 will not work with v1.4 and vice versa. We recommend all users to update to version 1.4.

Subsystem BASEchain:

- problem with HQ-Relation stored in a file fixed

- problem with update of hydraulic tables after geometry change due to bed load fixed

- necessity for a comment for each cross section in HEC-RAS files eliminated

- problem with cross section names ending with one or more 0 in HEC_RAS files fixed

- beta values in Tecplot file are now mean values over the whole bed width instead of value of the first slice

- new friction type: Manning

- new binary Tecplot files for longitudinal profile data

- new binary output file for cross section topology
- new suspended load simulation

    – advection by QUICKEST scheme
    – advection by Holly-Preissmann scheme
    – advection by MDPM scheme
    – sediment exchange between water and bed
    – diffusion by given factor or internal computation

Subsystem BASEplane:

- Inner Boundary Conditions (weir and gate) implemented
- Weir Boundary Condition modified (velocity component in momentum equation)
- UCD-Output for AVS enabled (Universal Cell Data)
- additional friction types implemented (strickler, darcy, chezy, etc)
- new wall friction added to optionally consider wall friction at boundary edges
- hydrograph boundary redesigned with momentum flux at inflow (please use an external source if you want an inflow without momentum flux)
- new h-Q boundary implemented which allows the specification of a h-Q relation as outflow
- ascii and binary Tecplot output
- performance tuning of some subroutines and file outputs
- additional geometry check for ambiguous gradients
- minor bug fixes

Subsystems BASEchain and BASEplane:

- Parallelization for shared memory multi-core systems implemented
- New LINUX version prepared
- BASEviz visualization tool added (currently ONLY for Windows version)

System Manuals:

- Please read the credits and the new preface, which are introducing a new team member and are giving citation advice
- Part UI-5: new chapter about parallel processing
- Part UII-3.3: some hints concerning mesh quality, especially how to deal with ambiguous elements
- Part UIII: almost complete revision. Examples for and comments on newly added features
- Part UIV: overall correction.
- **Part RI: complete revision**

- Part RII-2.3: treatment of inner boundaries
- Part RII-3.2: schemes for suspended sediment transport
- Part RIII: additional test cases for suspended sediment transport and parallel execution
- **Part RIV: complete revision**
- **Update your hardcopy version with the documentation patch.**

3.3 Version 1.5

General:

- *New Features:*
  - Batch Mode (running BASEMENT with program arguments)

Subsystem BASEchain:

- *New Features:*
  - Source and sink of suspended load
  - Conservation of sediment exchange between bed and suspended load improved

- *Bug fixes:*
  - Wrong warning concerning friction definition eliminated

Subsystem BASEplane:

- *New Features:*
  - Gate Boundary
  - HLLC Riemann solver

- *Bug fixes:*
  - Value Check hQ-Relation
  - Correction of Hydrograph boundary at supercritical flow conditions
  - Bug fix of edges memory reservation in case of very large grids with many holes
  - Vector length scaling added for the vtk velocity vector output
  - Prevent output of negative flow depths

3.4   Version 1.6

Subsystem BASEchain:

- *New Features:*

    - Implicit solution of hydrodynamics
    - Smart-Jaeggi bed load formula

- *Bug fixes:*

    - Correction transport diagram
    - Conveyance computation (negative areas)

Subsystem BASEplane:

- *New Features:*

    - Revision of sediment transport (Dual Mesh concept)
    - Kinematic viscosity and algebraic turbulence model
    - Wu bed load formula
    - Improved treatment of bed source terms for quadrilateral elements
    - New approach of slope calculation for quadrilateral elements
    - Improved concept of wall friction treatment
    - Flux correction algorithm implemented to enhance mass conservation
    - New outflow boundary: ZHydrograph

- *Bug fixes:*

    - HQ-Relation and Hydrograph boundaries

System Manuals:

- Part UIII: Introduction to new features
- Part UIV: Corrections
- Part RI: Complete revision and additions, e.g. sediment transport
- Part RII: Complete revision and additions, e.g. partially wetted elements, sediment transport, dual mesh approach and implicit time integration.
- Part RIII: Additions and corrections
- Part RIV: Documentation of new features

3.5 Version 1.7

General:

- *New Features:*

  - Coupling of subsystems BASEchain and BASEplane, i.e. sub-domains
  - Flow control in river systems

Subsystem BASEchain:

- *New Features:*

  - Multiple boundaries at in- and outflow cross sections
  - Speed improvements for explicit simulations
  - Improvements of zhydrograph-boundary condition
  - Changed behaviour of qlateral source (now uses one cross section)

- *Bug fixes:*

  - Batch execution mode corrected

Subsystem BASEplane:

- *New Features:*

  - Suspension transport 2-D
  - Improvements of inner boundaries
  - New procedure for updating velocities at partially dry elements (prevention of high velocities at wet-dry interfaces)
  - ZHydrograph boundary condition enhanced to be used as outflow or inflow

- *Bug fixes:*

  - Correction of hydrograph boundary in case of no inflow
  - Negative area for element prevented (only occurs when .2dm is manually changed so that nodes are not counter clockwise)
  - Bug fixes in Tecplot output routines
  - Default behaviour of HQ relation changed, partially dry edges are treated as zero_gradient instead of wall

System Manuals:

- new chapters on model coupling and flow control
- additions and corrections

3.6   Version 2.0

General:

- *New Features:*

  - Graphical user interface (GUI) for BASEMENT with integrated reference manual. Enables faster and correct setup of command files, setup and interpolation of BASEchain topography files and running a simulation.
  - Visualization (BASEviz) revised and improved to be more interactive.
  - Additional Q-t-Table implemented for flow control.
  - All sediment transport formulas can now be used for BASEchain and BASEplane.
  - Additional bedload transport formula implemented with Meyer-Peter Müller approach for multi-grain with critical shear stress correction according to Ashida and Michiue.

Subsystem BASEparse:

- Various changes and standardizations of input tags in the command file parser.

Attention: The changes of the command file's structure are not backwards compatible! Input files for v1.7 will not work with v2.0 and vice versa. We recommend all users to update to version 2.0.

Guidance on how to convert old command files into the new format:

There are several possibilities to update old command files. Maybe the easiest way is to use the new graphical user interface (GUI). Start the program BASEMENT and open your old command file. The message "parsing failed" will pop up and ask you to open your file in the raw editor to fix it. Confirm with the OK button. In the raw editor the parse errors will appear in the lower section in red colour. Invalid tags are pointed out and possible tags are proposed. Apply the proposed changes and click on the "Validate" button to see if your correction was successful. If some proposed tags should be unclear to you have a look at the Block List in the Reference Manual R IV. In the information-rich Block List all input blocks and possible tags are described.

Another possibility is to build up a new command file with the GUI of BASEMENT. This way you will learn to use the new GUI and will be provided with all possible input blocks, tags and information.

Subsystem BASEchain:

- *New Features:*

  - New topography format implemented for BASEchain (.bmg). Please use this file format as the default format for all new projects.
  - Python-script to convert files from old Floris-format to new input format. The script can be downloaded from the website and requires 'python' to be installed (www.python.org).

– Tool in graphical user interface (GUI) for cross section interpolation.

– Tool in graphical user interface (GUI) to automatically thin out grid points and thus speed up simulations.

– Significant speed improvements especially for hydraulic calculations.

– Coupling via "off-channel" source enabled.

- *Bug fixes:*

  – Bug fix in Roe solver.

Please note: Due to revision works, at the moment only one soil may be specified for a cross section in morphological simulations. Usage of multiple soils will be enabled again in future.

Subsystem BASEplane:

- *New Features:*

  – New framework for data output with additional features and output variables.

  – Geometric approach for gravitational bed load transport (slope failures).

  – Some speed improvements for hydraulic calculations.

  – Additional sediment inflow boundary condition which sets inflow to transport capacity.

  – Changes in calculation of sediment fluxes. Upwind-factor now has more effect.

- *Bug fixes:*

  – H-Q iterations and default precision in Hydrograph boundary conditions fixed.

  – Bug fix in treatment of fixed bed regions based on material index.

  – Bug fix in deletion of layers.

  – Bug fix for sediment restart.

  – Error at simultaneous binary tecplot output of hydraulic and sediment data fixed.

System Manuals:

- Additions and corrections.
- Revision of hydraulic tutorial for BASEchain.
- New hydraulic and sediment tutorial for BASEplane.
- HTML-documentation files can be automatically generated by starting BASEMENT with command line flag "-doc".

3.7 Version 2.1

General:

- *New Features:*

  – Controller extended for external sources
  – Support of jpeg-output of BASEviz visualization (experimental)

Subsystem BASEchain:

- *New Features:*

  – A simple consideration of internal levees
  – Time dependent grain mixtures as new upper boundary for bed load.
  – Multiple soils in one cross section are possible again, a factor is available to define from which degree of submersion a soil is considered to be wetted and will be moved.
  – For bed load transport the definition of the bed bottom is needed again!

- *Bug fixes:*

  – Transport diagram
  – Monitoring point integration for simulations starting after time 0.
  – No more elimination of elevations smaller than h dry in hydraulic balance.
  – Initialisation of initial_time_step.

Subsystem BASEplane:

- *New Features:*

  – New upwind method for suspended transport
  – New binary restart file format (CGNS) contains all restart data and enables restart from different solution times (hydraulics, bed load and suspension)
  – Reduced mesh dependence in many situations due to changed formulation of Riemann problem on edges
  – Initial depth is set to zero over the domain (and no longer to depth_min)
  – Several improvements in data output (depth/wse output are consistent now, sediment balancing improved, added specific discharge, renaming of "z_bed → z_element" and "zcoord → z_node")
  – Some speed improvements for parallel execution
  – Improvements and parallelization of gravitational transport

- *Bug fixes:*

  – Bugfix for inner gate because unphysical flux could be generated in dry cells
  – Bed load transport rate was determined too small in some cases !
  – Bugfix of 2dm mesh input. Internal edges of last nodes of the mesh could be falsely treated as boundary edges.
  – Tecplot output was restricted to 6 digits, now changed to 8 digits

    – Bugfix in layer interpolation on sediment elements

System Manuals:

- New manual for the graphical user interface (GUI) replaced old UIII manual
- New sediment tutorial for fractional transport for BASEplane
- Additions and corrections

3.8   Version 2.2

General:

- *New Features:*

    – External coupling implemented for data exchange with external programs over TCP/IP (experimental stage)

    – 64-bit linux version available for Ubuntu Natty (11.04)

Subsystem BASEchain:

- *New Features:*

    – Account for grain abrasion due to mechanical stress after Sternberg

- *Bug fixes:*

    – Bugfix in layer creation for sediment transport

    – Bugfix in grain sorting in fractional sediment transport

    – Bugfix in BASEviz-output, pause-function did not work

Subsystem BASEplane:

- *New Features:*

    – Cycle step for quasi-steady bed-load & suspension transport to speed up morphological simulations

    – zhydrograph boundary improved. Ghost cell velocities can be set to zero (e.g. lake) or to the upstream cell velocity (e.g. stream).

    – Threshold value can be set in water surface output in case of dry conditions.

    – Upwind-factor for bed-load transport is assigned now for each material index

- *Bug fixes:*

    – Correction in water balance for partially saturated elements

    – Bugfix in dynamic-depth solver, partially wetted cells were not treated correctly

- – Gravitational transport no longer requires an addtitional BEDLOAD or a SUSPENSION block to be defined
- – External sources for bed-load & suspension moved to MORPHOLOGY-block
- – INITIAL-block in MORPHOLOGY-block is now mandatory
- – Bugfix in output of sediment inflow at first time step
- – Bugfix critical shear stress of incipient motion (theta_critic) was not correctly initialized

System Manuals:

- Additions and corrections

3.9   Version 2.3

General:

- *New Features:*
  - – New pre- and post-processer BASEmesh for unstructured 2-D mesh creation, setting of material indices and visualization. Usable as plugin in the free software Quantum GIS (QGIS) on Windows and Linux.
  - – Enhancement of lateral coupling; automatic creation of connections between 1-D cross sections and 2-D edges (experimental)

- *Bugfixes:*
  - – Bugfix of HEC-RAS conversion script for non-georeferenced files

- *Subsystem BASEparse*
  - – New TIMESTEP block introduced in command file
  - – Improved documentation of controller-features
  - – Units of sediment boundaries and sources clarified
  - – Ranges of tag values in command file are now shown correctly
  - – Additional warnings added for non-plausible inputs

Attention: The changes of the command file structure are not backwards compatible! Input files for v2.2 will not work with v2.3 and vice versa. We recommend all users to update to version 2.3.

Subsystem BASEchain:

- *New Features:*
  - – Dredge source improved and enhanced for additional features

- *Bug fixes:*

– Bugfix of cross section geometry update in case of non-rectangular geometries

Subsystem BASEplane:

- *New Features:*

  – Extension and enhancement of hydro-stepping to allow for increased computational performance in case of quasi-steady flow and sediment transport
  – New output format as SHAPE file (for GIS-applications)
  – New output data type of specific discharge
  – New output option for min/max-values; time of occurrence of min/max-values now can be tracked
  – New computational method for the determination of suspended-load concentrations which limits concentrations to max. 1.0
  – Pick-up calibration factor suspended-load transport can now be set for each grain class separately
  – Corey shape-factor added for the determination of settling velocity after Wu and Wang
  – New output for suspended-load transport for multiple grain classes
  – New output data types for debug data (source terms, etc.)
  – New output option for ASCII-output; possibility to write external IDs instead of coordinates
  – New output option for TECPLOT; possibility to split TECPLOT files into several output files, one for each output value, to prevent too large file sizes

- *Bug fixes:*

  – Bugfix and improvement of wall friction computation; original concept was flawed
  – Bugfixes for external couplin
  – Bugfix of sediment restart (hotstart) in case of multiple soil layers
  – Bugfix of determination of removable water volume in water sinks

System Manuals:

- Additions and corrections
- New manual and tutorial for BASEmesh (QGIS Plugin)

3.10   Version 2.4

General:

- *New Features:*

- – New computation of Zhang's settling velocity
- – New 3-D seepage model BASEsub (experimental)
- – Lateral 1-D/2-D Model coupling improved: automatic assignment of 1D and 2D elements, new options and tutorial. Redesign of block-structure.
- – Added 'documentation' option to help menu.

- *Bugfixes:*

  - – Bugfix: restart (hot-start) for coupled simulations corrected.
  - – Bugfix: coupling type 'confluence_wse' could not be properly activacted.
  - – Improved parsing of series file: comments are possible at the end of a line (everything after '#' or '//' is discarded).

Subsystem BASEparse

- New tag 'start_time' in TIME_STEP block introduced (replaces 'restart_start_time' in BASEplane).
- STRINGDEF output improved, considering the flow direction. Each STRINGDEF definition now needs the tag 'upstream_direction' to be defined!
- New block LATERAL_COUPLING introduced for lateral coupling.

Attention: The changes of the command file structure are not backwards compatible! Input files for v2.3 will not work with v2.4 and vice versa. We recommend all users to update to version 2.4.

Subsystem BASEchain:

- *New Features:*

  - – Smart & Jaeggi sediment transport formula extended to multi grain with hiding & exposure according to Ashida & Michiue
  - – VTK visualization (BASEviz) now takes geo-referencing into ac.count.
  - – HecRas conversion script redesigned and enhanced. Script integrated into BASEmesh.

- *Bug fixes:*

  - – Bugfix: zhydrograph now applicable for upper and lower boundary.
  - – Suspension calculations slightly modified (especially exchange rates sources).

Subsystem BASEplane:

- *New Features:*

  - – SMS outputs (*.sol) enhanced with reference time in Julian date format.
  - – Enhanced output for external water source, separately for each element.

- Added output in VTK-Format (use free visualization program Paraview).
- New boundary type: movement of bed evelation, enforced via file input. Defined in GEOMETRY->BOUNDARY block.
- New output for external (hydraulic) discharge source: "external_source_discharge"" shows the element-history of the external source.

- *Bug fixes:*

  - MDPM scheme works now for pollutant transport only.
  - Bugfix: output of tracked variables ('min', 'max', 'sum',...) did crash in some cases.
  - Bugfix: polygons in shape output need to be closed.
  - Bugfix/feature for cycle-step-algorithm: very small correction time steps are now neglected, if they are smaller than the minimum time step size.
  - Bugfix: bad memory allocation for restart simulations with suspended-load

System Manuals:

- Additions and corrections
- Documentation of transport formula of van Rijn
- New tutorial added for lateral coupling (BASEMENT)
- New tutorial added for 1D cross-section generation using HEC-GeoRAS (BASEmesh)

3.11   Version 2.5

General:

- *New Features:*

  - see subsystems for new features.

- *Bugfixes:*

  - Minor fixes to HECRAS-BASEMENT conversion script

Subsystem BASEparse

- New INITIAL block for 1D morphological simulations introduced
- SOURCE block for 1D morphological simulations moved from BEDLOAD to MORPHOLOGY block
- Renamed tag 'theta_critic' to 'theta_critical' in PARAMETER block in BEDLOAD block for 2D morphological simulations

Attention: The changes of the command file structure are not backwards compatible! Input files for v2.4 will not work with v2.5 and vice versa. We recommend all users to update to version 2.5.

Subsystem BASEchain:

- *New Features:*

  – Additional monitoring output added: mean cross section bottom elevation

  – Monitoring output for multiple cross-sections can now be written to a single file

  – Modification of sediment discharge source (additional checks in case of sediment sink, print volumes to file)

  – Added tag 'use_energy_slope' (on/off) for sediment transport formula of 'smartjaeggi' and 'smartjaeggi_multi' for the calculation of the slope

- *Bug fixes:*

  – Fixed and improved 'start_time' parameter, which allows for starting a simulation at times different from 0.0

  – Removed debug-message indicating too high sediment fluxes

Subsystem BASEplane:

- *New Features:*

  – New inner hQ-boundary added (e.g. to simulate culverts, pipes, bridges).

  – New dredge source added. Sediment can be dredged to prevent sediment deposits from exceeding a specified bed-level. The sediment also can be redistributed at a downstream location (allowing for sediment continuity at inner boundary conditions).

  – Added second bed-load flux computation method (it is recommended to use the old version with averaged bed-load fluxes, rather internal parameter)

  – Added special output of 'sediment grid' as shape file (enables visualizations of hydraulic + sediment grids in GIS)

  – Added tag 'scale_lateral_transport' for lateral_transport in order to define only a rotation or a rotation with increased transport

  – Additional 'stringdef_history' outputs (mean values on a stringdef, approximations): mean[velocity], mean[wse], mean[depth], mean[energy_head], mean[zbed]

- *Bug fixes:*

  – Fixed header description for sms output files

  – Fixed behaviour of 'dynamical_depth_solver'. Previous implementation led to violation of well-balancing

  – Fixed VTK output: vtk does not allow for same variable names in scalar and vector data; resolved by new naming scheme

  – Bux fix for sediment restart with cgns (Windows)

  – Improved and more user friendly concept for considering bedforms

System Manuals:

- Additions and corrections

- Description of hQ-inner-boundary for 2D simulations
- Description of dredge source for 2D simulations

3.11.1 Version 2.5.1

General:

- *Bugfixes:*

    – Improved correction of the overdraft cells (including WARNING in case of failed correction)

    – Output files changed for combined time histories: nd, el, sd in filename

    – Headers in output files: matching the value names of the input parser, adding the units

    – File .sol readable again

3.11.2 Version 2.5.2

General:

- *Bugfixes:*

    – Crash of Basement prevented in case of output values that cannot be generated due to missing simulation blocks; now an error message will appear in case of missing simulation blocks

    – Gate formula improvement:

        * Correction of discharge calculation

        * Added possibility of dynamic calculation of weir coefficient 'mu'

        * Added possibility of taking into account backwater effects (only inner boundary conditions and dynamic 'mu')

    – Tag 'tau_erosion_start' available for multi grain simulations

    – Improved ascii and vtk output files for vector data, vector components are not written separately anymore

    – Removed output format "qgis" (deprecated). QGIS plugin crayfish can now read the file format standard of sms (*.sol)

3.11.3 Version 2.5.3

General:

- *Bugfixes:*

    – QGIS SMS output linux: new time step TS was not properly written on a new line. Therefore SMS and QGIS were not able to detect all time steps

    – Bugfix 'tau_erosion_start' for multi grain simulations

3.12 Version 2.6

General:

- *New Features:*

  – New Windows installers for users with and without admin permission
  – New handling of errors and warnings:

    * Writing two files: run.log with log-info and run.err with all warnings and errors
    * The same philosophy is applied in the GUI: two tabs, one for info, the other for warnings/errors
    * Consoles in GUI are cleaned in case of re-run
    * New log levels: (1) Error, (2) Warning, (3) Info, (4) Debug

  – Open the whole block-tree with command in GUI menu bar (expand all, ctrl+A)

- *Bugfixes:*

  – No segmentation fault if user requests to stop simulation AND simulation has already finished

Subsystem BASEparse

- New FORMULA block in BEDLOAD block (1D and 2D)
- Renamed tag 'bedload_transport' to 'bedload_formula' inside FORMULA block
- New DIRECTION block in BEDLOAD block (2D), with new feature curvature_effect_static (option lateral_bed_slope replaces former tag lateral_transport in PARAMETER block)
- Deleted obsolete and misleading tag 'output_time_step' in OUTPUT block (2D)

Attention: The changes of the command file structure are not backwards compatible! Input files for v2.5 will not work with v2.6 and vice versa. We recommend all users to update to version 2.6.

Subsystem BASEplane:

- *New Features:*

  – Index based definition of river curvature (radius) to simulate curvature effect on sediment transport (change of bedload direction) in new DIRECTION block

- *Bugfixes:*

  – Review of HLL and HLLC Riemann solvers
  – Fixed coupling sediment boundary 'coupling_IODown'
  – Tag 'use_cell_averaged_bedload_flux' in PARAMETER block in BEDLOAD block is set to 'off' per default

 – factor_reference_height set to 0.1 by default (not 1.0) for suspended load
 – Flux orientation in wall flux boundary

System Manuals:

- New Tutorial Post-Processing of 2D Simulation Results
- Updated Tutorial Pre-Processing in QGIS with BASEmesh
- Additions and corrections in System Manual
- Review of HLL and HLLC testcases

3.13 Version 2.7

General:

- *New Features:*

 – New bedload transport formulae: Wilcock and Crowe, Engelund and Hansen (for single-grain)
 – Definition of external sediment and hydraulic source over material index
 – Improved and extended *mpm_multi* and *smartjaeggi_multi* using hiding function according to Ashida and Michiue (1972) with new options to choose reference critical Shields parameter: based on mean, median, geometric diameter, or diameter of fractions
 – New logarithmic friction laws: Yalin, Bezzola

- *Bugfixes:*

 – User-defined settling velocities
 – New tag *bedload_exponent* used for *mpm* and *mpm_multi*, correction of Wong and Parker formulation
 – *dynamic_depth_solver_precision*: eliminated dependency on *min_water_depth*
 – Minor others

Subsystem BASEchain:

- *New Features:*

 – HLL Riemann solver

- *Bugfixes:*

 – Fix Matlab output

Subsystem BASEplane:

- *New Features:*

  - New *local_slope_correction* concept: *local_slope_longitudinal* and *local_slope_lateral* suppressed, introduced *local_slope_vanrijn* and *local_slope_chen*
  - New tag *theta_critical_default* with *theta_critical_vanrijn* and *theta_critical_yalin*
  - Dynamic determination of flow curvature based on water surface inclination
  - Dynamic determination of flow curvature based on local streamline reconstruction
  - Hydraulic boundary *hydrograph* with dynamic slope in case of erosion
  - New element output *momentum* that returns flow intensity (water depth times velocity)

- *Bugfixes:*

  - Local slope: updating local slope correction due to changed bed slope
  - Correction of *zero_gradient* downstream boundary condition
  - Revision of upwinding in bed load transport (bed load flux averaging)
  - Write sediment_grid.shp file to the right location (where .bmc file and result files are written)
  - Revision and new concept for creation and initialization of layers (*create_new_layers*)
  - Improved *node_history* output for *grain_size*
  - *control_volume_type*: replaced *dmean* by *d90*
  - Fix *grain_size_friction* on fixed bed
  - Fix high velocities in upstream BCs for inner weir, inner gate and inner h-q relation
  - Minor others

System Manuals:

- New BASEMENT manuals in pdf and html format for offline and online consultation

3.14 Version 2.8

General:

- Software environment updated for Windows 10 - 64 bit
- Update to latest 3rd party libraries
- *New Features:*

  - Vegetation model
  - User defined hiding_exponent also available for Parker formula

– Colorblind friendly GUI

– Ashida and Michiue transport fomula (non-uniform sediment)

– Output timestep matches exactly given intervall

– OpenMP parallelization revised

- *Bugfixes:*

  – Wilcock and Crowe bug fixes

  – output_time_step and restart_time_step

Subsystem BASEchain:

- *NOTE* : Implicit scheme for time integration not available in v2.8. Please refer to v2.7 for implicit time integration calculation.

- *New Features:*

  – Initial state for suspended load transport (mixture and concentration) can be defined with global values instead of providing a file with initial state for every cross section

  – Water depth and Froude number are available as output (including the tracking of minimum and maximum values)

  – Simple upwind approach for suspended load transport available

  – Inner boundaries improved and extended (weir/gate)

- *Bugfixes:*

  – Matlab output fixed and optimised

  – Sediment exchange between suspended load and bed load with "shear stress" approch

  – Mass conservation in dual-mode sediment transport (combination of bed load and suspended load) improved

  – Mass conservation with QUICKEST scheme improved

Subsystem BASEplane:

- *New Features:*

  – Vegetation model: new BLOCK and all related features added

  – Dynamic slope available for 'hqrelation' (uniform flow) at downstream boundary with the same slope adjustment as at the upstream 'hydrograph' boundary (due to erosion/deposition)

  – Definition of control_volume_thickness over the material index with new tag control_volume_thickness_index

  – INNER Boundary Conditions for MORPHOLOGY (BEDLOAD): "open" and "weir" type

– Suspended load exchange term: exchange_type= shear_stress and additional parameters: tk, factor_m and factor_n

– Generation of the sediment grid parallelized

- *Bugfixes:*

    – Possible to get the output value 'saturation' without BASESUB-block

    – string_def output of flow velocity and improvement of the averaged velocity weighted by wetted area of relevant edges

    – INNER GATE and WEIR in BASEplane reviewed

System Manuals:

- Revision of test cases
- Reference Manual:

    – Ashida and Michiue transport formula (non-uniform sediment)
    – Suspended load
    – Bed perturbation
    – Sediment boundary conditions
    – Description of vegetation model
    – Discretisation of Boundary Conditions (inner boundaries)

3.15   Version 2.8.1

General:

- The third party libraries and the installers were updated.

- The vegetation feature was revised.

- The time-integrated transport capacity is now included in the output "transport_diagram".

- The documenation was revised.

- *Kknown Issues:*

    – Fixed bed starts eroding if both suspended load and bed load transport are activated if the suspended load is below transport capacity.

    – Van Rijn formula does not result in any sediment being transported (1D-model).

    – The upstream and downstream stringdefs of the inner boundary weir must be of the same length, otherwise mass might not be conserved. Summary of Features ====================

3.16 Subsystem BASEchain

3.16.1 Numerics

Spatial Discretization	Finite Volumes
Time Integration	explicit Euler
Riemann Solvers	Roe, HLL, upwind

3.16.2 Geometry

Type	Cross sections : simple, composite (flood plains, main channel)
Slices	Friction and soil description, no flow consideration
Mobile bed	Fixed bed consideration, uniform distribution of deposition and erosion over wetted part of cross section, partially wetted soils (suspended load)
Input Formats	BASEMENT (*.bmg)

3.16.3 Hydraulics

Model	1D Saint-Venant equations
Conservative Variables	A (continuity), Q (momentum)
Source Terms	Friction (semi-implicite), bed slope, lateral source/sink
Friction	Strickler (kstr), Manning (n), grain size friction
Determination of z(A)	Interpolation from table or iteration or consideration of levees
Boundary Conditions	Wall, hydrographs, hQ-relation, in/out, zhydrograph, weir, gate, multiple boundary, zero_gradient
Initial Conditions	Sub-critical backwater calculation, dry, from imput file
Inner boundary	Weir/Gate

3.16.4 Sediment Transport: Bed load

Model	Hirano-Exner equation, sorting equation for sediment mixtures
Transport Formula	MPM, MPM-multi, power law, MPM-H, Parker, Rickenmann, Smart-Jaeggi, Smart-Jaeggi-multi, Wu, Van Rijn, parametrized shear stress, abrasion-law of Sternberg, Wilcock and Crowe, Engelund and Hansen, Ashida and Michiue
Boundary Conditions	Hydrograph, IOUp and IODown, transport capacity, fractionated hydrograph
Active Layer	Constant, variable thickness
Source Terms	Local source/sink, dredge source, bed-load function

3.16.5   Sediment transport : Suspended load

Transport Equation	Advection-Diffusion Equation
Discretization	QUICK, QUICKEST, Holly-Preissmann, MDPM, Upwind
Exchange Terms	Concentration: Van Rijn, Zyserman et al., constant factor
	Shear stresses: Xu
Source Terms	Local source/sink

3.16.6   Output Variables/Formats

- Variables

General Information	Geometry and characteristics of cross sections
Bed load	Total and for each grain class
Grain mixtures	For each cross section, slice and layer

- Format

Standard output	Talweg: z_{dam}(left/right), z_{bottom}(mean), friction, $Q_b{}^{cap}$, C_{tot}
	Cross section: Water surface elevation WSE, energy level EL, velocity v, wetted area A, discharge Q, Froude number Fr, friction
	Edge: continuity flux, momentum flux, $Q_{b,i}$, $Q_s{}^{adv}$, $Q_s{}^{diff}$
Special output	Tecplot Format: Bed level, water surface elevation, energy level, discharge, wetted area, dike levels, bed load, grain class concentrations and suspended sediment concentration in active layer
	Monitoring Output: Q, Qb, A, WSE, v, geometry, C, b (=mean bottom level), energy_line (EL), Fr, h (=mean flow depth)
	Restart files: Hydrodynamics, bedload, suspended load
	Volume files: Variation of water volume, variation of sediment volume, suspended sediment volume, transport capacity volume

3.17   Subsystem BASEplane

3.17.1   Numerics

Spatial Discretization	Finite Volumes
Time Integration	explicit Euler
Riemann Solvers	exact, HLL, HLLC

3.17.2   Geometry

Cell/Grid Type	Triangular, quadrilateral, hybrid
Mobile bed	Fixed bed consideration, bed armor consideration, uniform distribution of deposition and erosion over wet elements
Input Formats	Grid-file *.2dm (SMS, QGIS)
Moving bed	User enforced vertical movement of selected nodes

3.17.3   Hydraulics

Model	Shallow Water Equations
Conservative Variables	h, u·h, v·h
Source Terms	Friction, bed slope, local discharge source/sink, diffusive fluxes (kinematic and turbulant viscosity)
Friction	Strickler (roughness kstr), Manning (roughness n), Chézy, Darcy-Weissbach, Log. friction law, Yalin, Bezzola
Boundary Conditions	Wall, inlet (hydrograph), outlet (weir, gate, zero gradient, h-Q relation, zHydrograph), inner boundaries (inner weir, inner gate)
Initial Conditions	Dry, user defined condition (based on material index), continuing old simulation

3.17.4   Sediment Transport: Bed load

Model	Hirano-Exner equation, sorting equation for sediment mixtures
Transport Formula	MPM, MPM-multi, power law, MPM-H, Parker, Rickenmann, Smart-Jaeggi, Wu, Van Rijn, Wilcock and Crowe, Engelund and Hansen, Ashida and Michiue
Direction	lateral bed slope effect (Ikeda), curvature effect (Engelund)
Boundary Conditions	Wall, inlet (hydrograph, transport capacity), outlet (outflow = inflow), inner boundaries (open, weir)
Active Layer	Constant, variable thickness
External source Terms	Local source/sink, dredge source

3.17.5   Sediment transport : Suspended load

Transport Equation	Advection-Diffusion Equation
Discretization	Upwind-Scheme
Exchange Terms	Van Rijn of Zyserman et al./Lin
Source Terms	Local source/sink

3.17.6 Sediment transport : Gravitational induced transport

Approach	Geometrical modelling based on critical failure angles
Features	Unstructured mesh, single and multiple grain classes

3.17.7 Output Variables/Formats

- Variables

General Information	Input file, element data (water depth, velocities, bed shear stress, etc), node data (bed elevation, etc), balances
Bed load	Element data (theta_critical, etc.), node data (bedload, delta_z, etc.)
Grain mixtures	Grain size distribution for selected elements

- Format

Standard output	SMS Format and ASCII tables
Monitoring Output	String discharge, Boundaries, Balances, Debug-output
Time History	Time history of selected elements, nodes or edges
Additional Formats	Tecplot, Shape (GIS), AVS-UCD, VTK (Paraview)
Restart file	Binary format (CGNS), hydraulic computation, bed load computation, suspended load computation

3.18 Subsystem BASEsub

3.18.1 Numerics

Spatial Discretization	LBM (Lattice-Boltzmann Method)
Time Integration	explicit Euler

3.18.2 Geometry

Cell/Grid Types	Structured mesh, cubic cells
Input Formats	Self-generated block mesh, parts can be de-activated

3.18.3 Hydraulics

Model	3-D Richards equations, Saturation-formulation, mixed Saturation-Pressure formulation

Primary Variables	Effective water saturation, Pore-water pressure
Source Terms	Water source (infiltration)
Constitutive Models, Material properties	Van Genuchten-Mualem, Brooks-Corey, kf-conductivity value of soil
Boundary Conditions	Bounce-back boundary, Pressure boundary, Effective water saturation boundary, Seepage boundary
Initial Conditions	Constant values: Pores-water pressure, effective water saturation Hydrostatic pressure distribution Hot-start from old simulation file

3.18.4 Output Variables/Formats

- Variables:

  - Primary variables: Effective water saturation, pore-water pressure
  - Secondary variables: Darcy velocities

General Information	Cut-plane or cul-lines through 3-D mesh, seepage line

- Format

Standard output	ASCII (Gnuplot)
Additional Formats	Tecplot, SMS/QGIS
Restart file	Binary file containing values of probability distribution functions

USER MANUAL

of *BASEMENT*

Contents

1

Basic Simulation Environment

1.1 Introduction

The software system „BASEMENT"(basic-simulation-environment) shall provide a flexible and functional environment for numerical simulation of alpine rivers and sediment transport involved. The numerical models for the computation of one- and two dimensional flows with moving boundaries and appropriate models for bed load as well as suspended load are forming the core of the software system. Two of the main project tasks were the renewal and further development of the existing 1-D and 2-D models ("Floris","2dMB"). The one-dimensional model complies with the upper bound of the considered spatial scale (maximum idealization and slightest resolution of spatial processes) and is meant to provide appropriate boundary conditions for the 2-D and 3-D models.

The main focus of conception and development was the stability of the numerical models, the flexibility of the computational grid and the combination and efficiency of the method of calculation (problem dependent equations, coupling of models, parallelization).

The development process was orientated at the concepts of object orientation, to assure transparency, documentation and flexibility of the software system as far as possible. Future developments, applications related to practice and scientific projects shall build upon the environment of BASEMENT to ensure sustainability.

1.2 General Use

1.2.1 Problem Description

In connection with watercourses and river areas, increasingly complex problems have to be addressed. The estimation of floods, the more frequent occurrence of restoration projects or the study of naturally shaped watercourses implicate the examination of larger regions - also outside of the actual waterway - and a more manifold shape of the channels. The simple formulas for the calculation of flow behaviour used in the past showed in several

cases to be insufficient to obtain the desired information. The extent of the considered areas makes the application of hydraulic models in a laboratory - usually employed for difficult cases - impossible or too expensive. So, the numerical simulation of flow behaviour is in many cases the most obvious solution. However, existing programs have still some weak points. Some are limited in their capabilities (e.g. only steady flow, no sediment transport and one dimension only) or may lack in user support caused in incompleteness of documentation or training of users. Furthermore, inherent numerical problems request certain expertise to be overcome. In addition, the preparation of the input data and the processing of the results to a shape, which facilitates the interpretation, are often very laborious.

The aim of the software system BASEMENT, in terms of its free availability and its accompanying scholar programs, is to enable a broader range of people to skilfully process river modelling projects in a justifiable amount of time.

1.2.2 Product Delineation and Employment Domains

1.2.2.1 Product Delineation

BASEMENT is a river engineering tool, which supports the engineer in the solution of tasks in the domain of river area modelling. The program permits reliable computations based on state of the art numerical tools, constant onward development and successive realisation of case studies.

Unlike currently used programs for the simulation of a specific flow behaviour, BASEMENT intends the arrangement of many different problem types with one single tool to gain an integrated understanding for the initial position, the solution process and its results.

1.2.2.2 Employment Domains

The aim of BASEMENT is to permit the solution of as many problems as possible in the domain of river engineering, especially in cases for which the traditional dimensioning tools are insufficient and studies including physical hydraulic models are not possible or too expensive. Typical employment domains are:

- Several problems in relation with the sediment transport of water courses, for instance the future development of deltas and alluvial fans, the long term evolution of the bottom of channels, or the aggradations of storage spaces and the consequences of their scavenging;

- River engineering enterprises, which imply the modification of the channel geometry, as this can be the case for example for revitalisations or protection measures, where the consequences of the interventions have to be evaluated;

- Identification and quantification of dangers for the development of danger maps or of protection and emergency measures, considering the flow behaviour and sediment deposition both inside and outside of the main channel, as well as erosion danger, and consequences of debris flows and dam breaks.

1.2.3 Processed Data Types and Capabilities

1.2.3.1 Processed Data Types

The raw data can be divided into three groups:

- **topographic data**: particularly elevation models and cross sections
- **hydrologic data**: time series of flow discharge, water levels or concentration of suspended sediments, velocity profiles;
- **granulometric data**: grain size distributions from water-, sediment- or line samples.

1.2.3.2 Capabilities

BASEMENT has the following fundamental capabilities:

- Simulation of flow behaviour under steady and unsteady conditions in a channel as well as its transition;
- Simulation of sediment transport (both bed load and suspended load) under steady and unsteady conditions in a channel with arbitrary geometry;
- Simulation of erosion and deposition;
- Choose between different approaches (e.g. choice of problem matched solver-algorithms);

1.3 System Overview

At the current stage of development, the software system consists of the numerical subsystems and the different interfaces to the infrastructural software, such as pre- and post processors. The core of BASEMENT consists of the numerical solution algorithms comprised in the appropriate modules. Pre- and post-processing can be performed with independent products using a well defined common interface. The flexible software design enables a future adoption to a common database for input- and output data (Figure 1.1).

Numerical Subsystems

The core of the software system consists of the numerical subsystems, which actually are:

The one dimensional numerical tool named BASEchain enables the simulation of river reaches (based on cross sections) with respect to sediment transport. Arbitrary coupling with the 2-D tool is possible.

The two dimensional numerical tool named BASEplane enables the simulation

Figure 1.1 *System Overview*
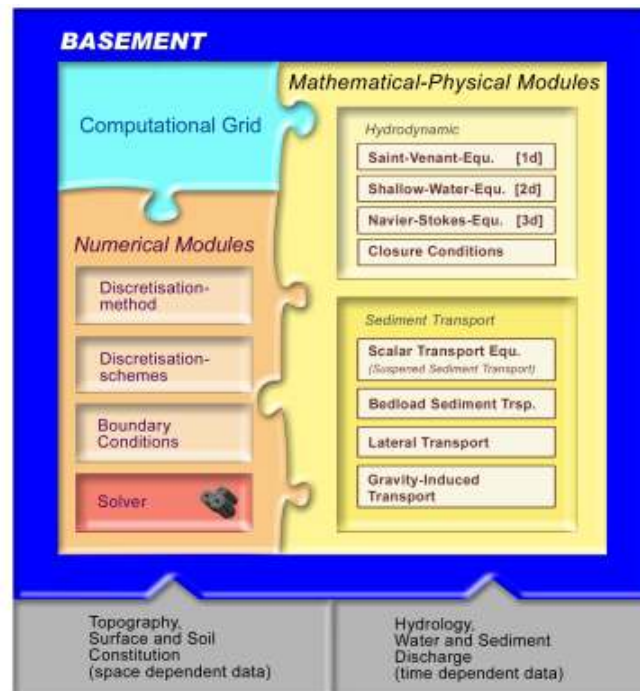
Figure 1.2 *Modules and their Components*

of river reaches as well as flood plains (bases on a digital terrain model) with respect to sediment transport. Arbitrary coupling with the 1-D tool is possible.

The three dimensional numerical tool named BASEsub is meant for the simulation of local subsurface flow fields (based on spatial geometry). The model is coupled with the BASEplane surface flow module.

For a list of implemented features of each model, please refer to the actual release notes.

1.4   Components

To reveal the "black box" of the numerical models, Figure 1.2 gives a graphical insight. The simulation tools of BASEMENT can be subdivided into three different parts:

- the mathematical-physical modules consisting of the governing flow equations
- the computational grid representing the discrete form of the topography
- and the numerical modules with their methods for solving the equations

In the next few chapters an overview of these modules is given. A more detailed description can be found in the reference manual.

1.4.1 Mathematical – Physical – Modules

The behaviour of the fluid hydraulics can be explained with physical models, namely the conservation of mass and momentum. Theoretically, it is possible to resolve the mathematical problem up to small scale phenomena like turbulence structures. In a natural problem however, it is mostly impossible to determine all boundary- and the exact initial conditions. Furthermore, the computational time needed to solve the full equation system is increasing very fast with higher spatial and temporal resolution. Therefore, dependent on the problem, simplified mathematical models are used.

In three dimensions, the flow and pressure distribution are completely described by the Navier-Stokes equations. These equations can only be solved numerically, as analytical solutions exist only for some strongly simplified problems. The 3-dimensional approach is only suitable for local problems, where turbulence phenomena and flow in all directions are essential for the results, e.g. the flow around bridge piers.

Assuming a static pressure distribution and neglecting the vertical flow components, the Navier-Stokes equations simplify to the 2-dimensional shallow water equations. This set of equations provides accurate results for the behaviour of water level and velocities in a plane. Turbulence effects cannot be resolved anymore but are accounted for by an artificial friction factor in the closure condition, which establishes a relation between flow velocity and shear stress. The shallow water equations are used for 2-dimensional flows like dam breaks, curved flow etc.

Reducing the spatial dimension once more results in the 1-D Saint-Venant equations. The main outputs of these equations are the water level and mean velocity in flow direction. This method is still in use for computing large river systems.

The computation of sediment transport is mathematically not as well developed as the hydrodynamic part. Theoretically, the movement of every single stone within the sediment could be computed by solving its equation of motion. However, this approach is yet numerically too expensive. Therefore, sediment transport and behaviour of the riverbed are computed using empirical formulas developed by river engineers. The computation of the sediment flux is physically not really correct, but proved to be accurate enough for a broad range of sediment transport problems. Usually, sediment transport occurs in the main flow direction. More sophisticated models consider also lateral phenomena within a curved flow.

Very small grain sizes are treated as suspended sediment load. Their behaviour can be computed by a physically scalar transport equation.

1.4.2 Computational Grid

1.4.2.1 The Meta Model

An important aspect of every computational task is the grid generation where the real world topography data is transformed into an internal computational grid on which the governing equations are solved. Independent of the discretization method, the construction of the computational grid has great impact on the accuracy of the results and on the computational time needed for the simulation or the numerical time step, respectively. Generally, a suitable mesh is dense at regions, where strong changes in the flow occur and
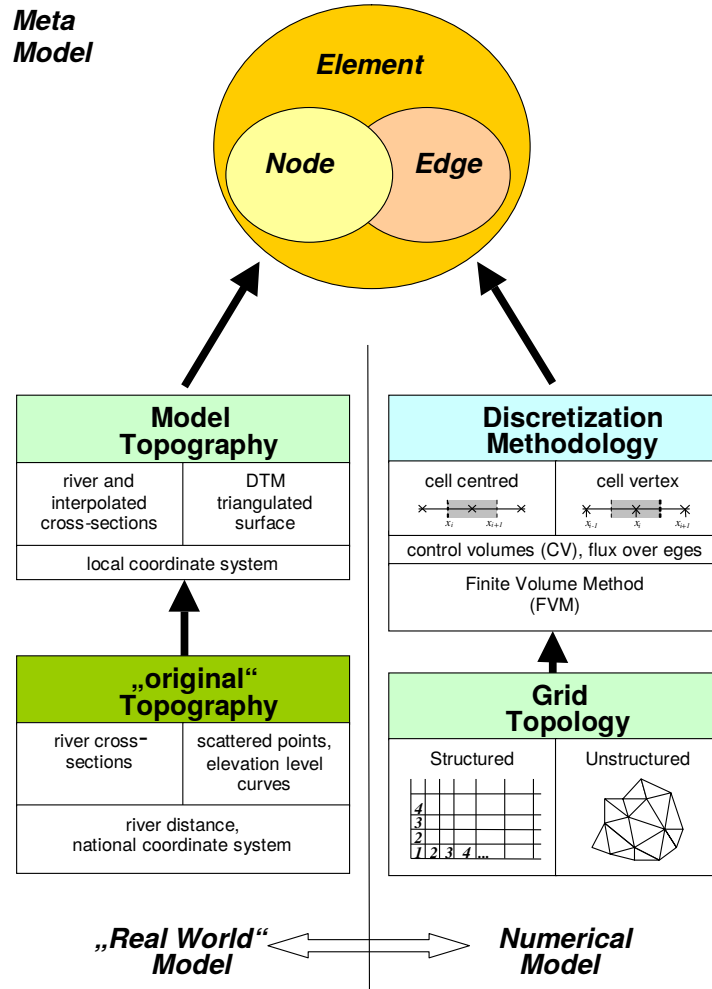
Figure 1.3 *The Meta Model: fusion of "real world" data with abstract numerical considerations*

coarse in regions of lower interest. Additionally, the grid cells should not underlie strong deformations.

Usually, the raw real world data comes in form of river cross sections or geographic terrain models e.g. from a GIS. This elevation information has to be mapped onto a suitable mesh.

There are two types of computational grids: structured and unstructured ones. Structured grids consist of quadrilaterals and can be mapped onto a Cartesian domain. They allow for simple data structures and efficient algorithms. The mesh generation is relatively simple and can even be done manually. However, structured meshes are somehow unhandy for the representation of arbitrary topography data. Unstructured grids are mostly composed of triangles and cannot be mapped onto Cartesian meshes. They usually need more complicated data structures but are highly flexible for automatic mesh generation in complex geometries. An unstructured grid is the most general case of a grid based discretization and is perfectly suitable for object oriented modelling. BASEMENT is built on unstructured grids.

The computational grid consists in general of cells (control volumes). In the software model, the mesh is based on three different objects:

- [node] the nodes – mass free points in relation to a coordinate system;
- [edge] the edges, which are defined by two nodes and define the place of information flux between two elements in Finite Volume Methods;
- [element] the elements, which are defined by several nodes and define the place of the physical variables, e.g. cell centered methods.

This data structure allows for similar treatment of 1-D and 2-D methods and schemes.

As the difficulty of mesh generation occurs in many different computational tasks, a broad range of different triangulation techniques or mesh refinement methods can be found in the literature. Some of them are specially designed for a certain discretization scheme but can also be used elsewhere. As there is nothing such as an ideal or perfect mesh, the user is recommended to produce different grids and compare their behaviour to find the best solution. There are commercial tools available which can be used for the grid generation, e.g. SMS. However, the BASEMENT standard for grid representation (see Reference manual) also allows for self created meshes.

1.4.2.2 BASEchain : one dimensional model

In one dimension, an element consists of two nodes with known cross-section. With a cell-centred discretization, all variables – velocity, flow depth and cross-section geometry - are defined at the location of the nodes. The midpoint of the connecting line between two nodes defines the common edge of the two elements. The more nodes are known, the better the representation of the real world data, especially at regions with strongly curved watercourse.

1.4.2.3 BASEplane : two dimensional model

In two dimensions, an element consists of three nodes with a known ground elevation. Usually, this real world height information is not given exactly at the desired node
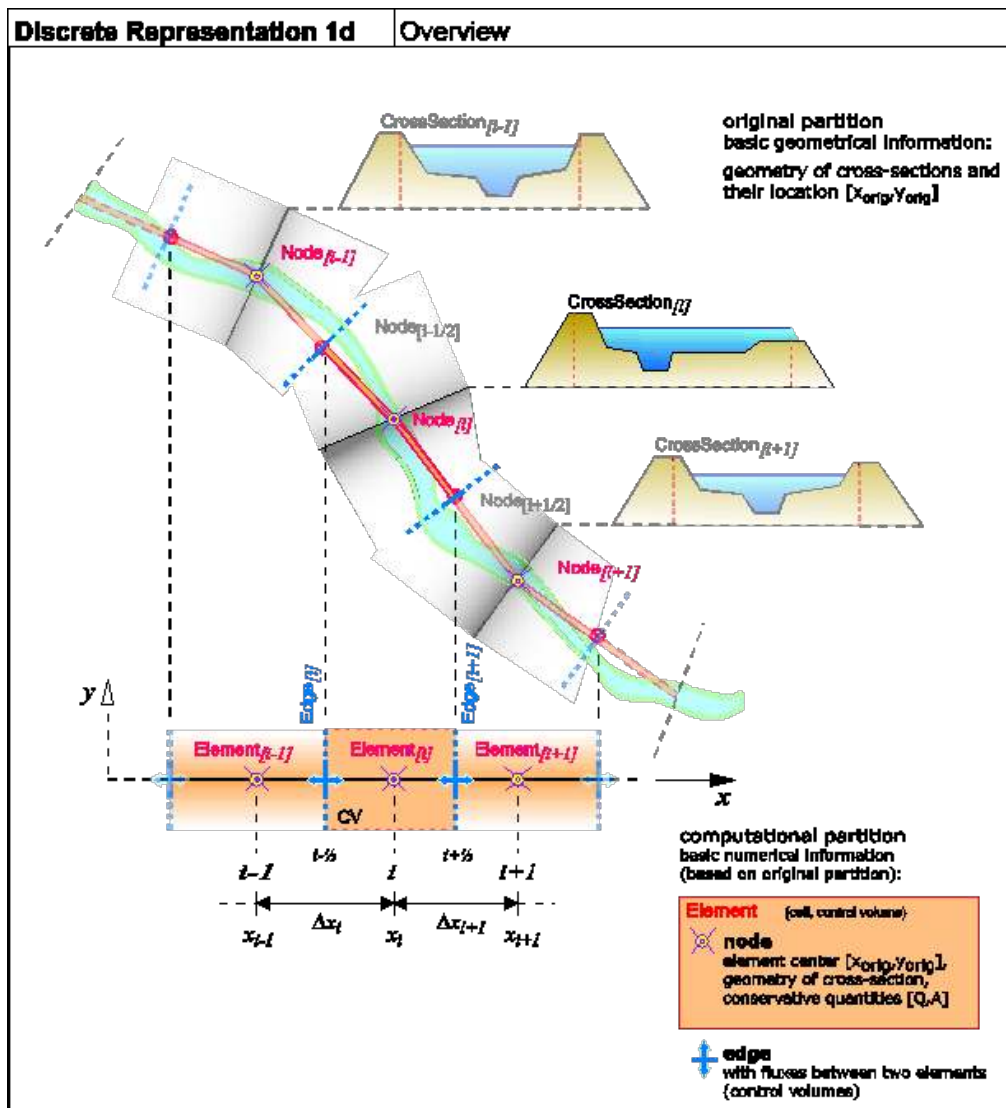
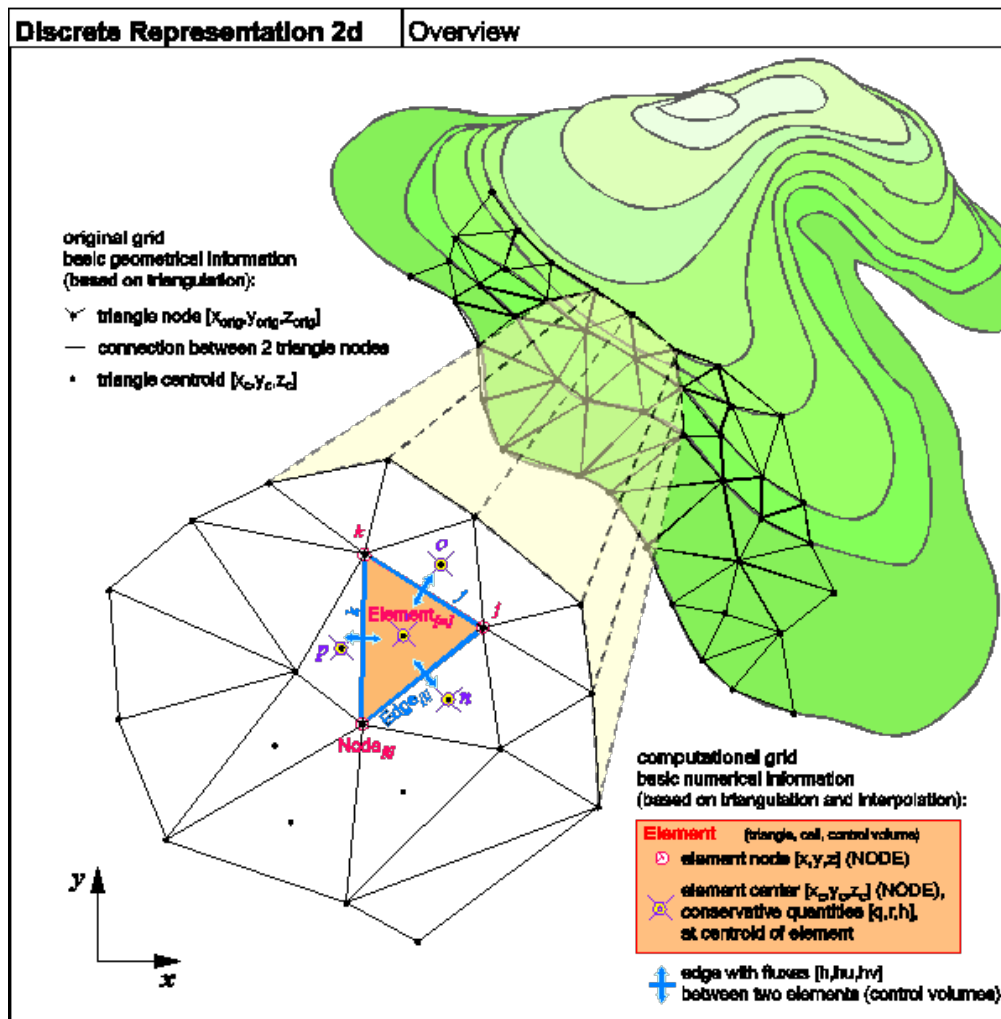**Figure 1.4** *Discrete Representation of the Topography within BASEchain*

Figure 1.5 *Discrete Representation of the Topography within BASEplane*

coordinates and therefore has to be interpolated. The primary variables are defined somewhere inside the element, e.g. the balance point. The fluxes between two elements are defined at their corresponding edges.

1.5 Simulation Procedure

The procedure to simulate a concrete problem setup is not unique. BASEMENT is coded using an object-oriented design which allows for flexibility and interchange ability concerning different application problems. The possible combinations are manifold. On the one hand, the governing equations may change dependent on simplifications or extensions of certain terms, use of sediment transport or pure hydraulics, etc. On the other hand, there are miscellaneous numerical methods, e.g. for time integration (implicit, semi-implicit, explicit) or computation of spatial fluxes. Therefore, the main variables of interest differ from one problem to the other. It is of great importance, to plan carefully each simulation approach to a certain problem. The most difficult and time-consuming part is not the simulation itself but the acquisition of all needed data (topography, boundary- and initial conditions) and a proper setup of this data. This section describes the main activities performed to execute a simulation with BASEMENT in a very general case. In most problems, only a part of them are being used.

1.5.1 Flow of main activities

1 **Build and Name Project**

2 **Preparative Work / Scenario definition**

- Topography / computational mesh
  - Import rough topographic data
  - Import cross sections
  - Add break lines manually
  - Build terrain model (triangulate)
  - Build cross sections from terrain model
  - Determine mean riverbed level
  - Generate computational mesh

- Define properties of mesh elements
  - Roughness
  - Grain sizes / distribution / mean diameters
  - Flow through / no flow through elements
  - Mobile / fixed bed elements
  - Transport relevant / not relevant elements
  - Shear stress

- Define approaches / Simulation Matrix
  - Sediment transport formulas
  - Roughness / Friction formulas
  - Debris flow type and approach

3 **Define initial- / boundary conditions and required results**

- Boundary conditions
  - Time discharge from Inlet Hydrographs
  - Water level time series at outflow
  - Special hydraulic elements (weir, step, bridge, block, wall, ...)
  - Sediment discharge dependent on Hydrographs
  - Concentration of suspended material
  - Grain size distribution
  - Height of Subsurface
  - Viscosity

- Initial conditions
  - Waterlevel
  - Sediment bed level / Slope
  - Flow velocities
  - Grain size distribution

- Required results
  - Monitoring point or sections
  - Values: discharge, velocity, water/bed-level, concentration, shear stress, ...
  - Result types: max (t), min (t), integral, time series with designed timestep, Simulation times, error estimation, ...

4 **Execute Simulation**

Figure 1.6 *Flow of main activities, part I*

Figure 1.7 Flow of main activities, part II
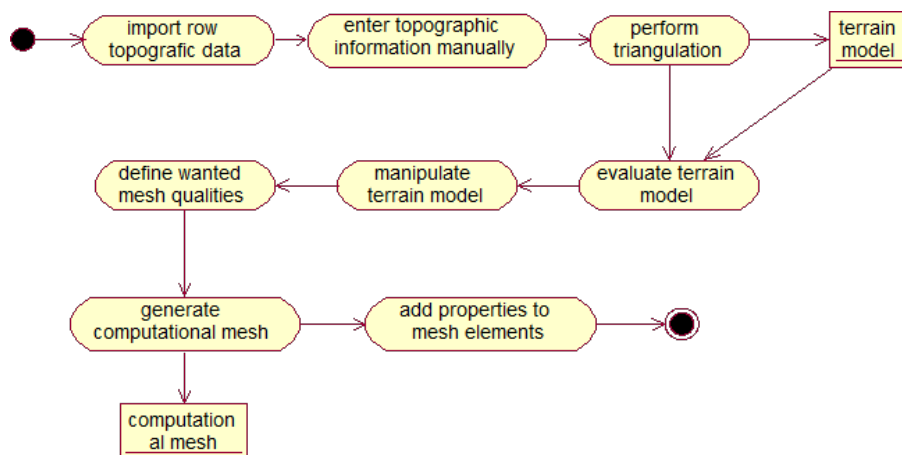
Figure 1.8 Activity diagram generate computational mesh

1.5.2   Scenario Examples

Depending on the chosen scenario and on the available boundary conditions or i.e. topography data, the approach for a successful simulation differs from case to case. As there are different ways to reach a certain target, the following activity diagrams just present possibilities but not a strict guideline.

An important part is always the grid generation. Usually, the raw topography data needs a lot of treatments (manual correction, interpolation, adjustment of single elements, etc.) until a suitable computational mesh can be generated. Although programs for grid generation like SMS provide some powerful tools to manipulate mesh transformations, the user still has to retain an overview over the required steps leading to the final computational mesh.

The following activity diagrams show a possible procedure for different project scenarios.
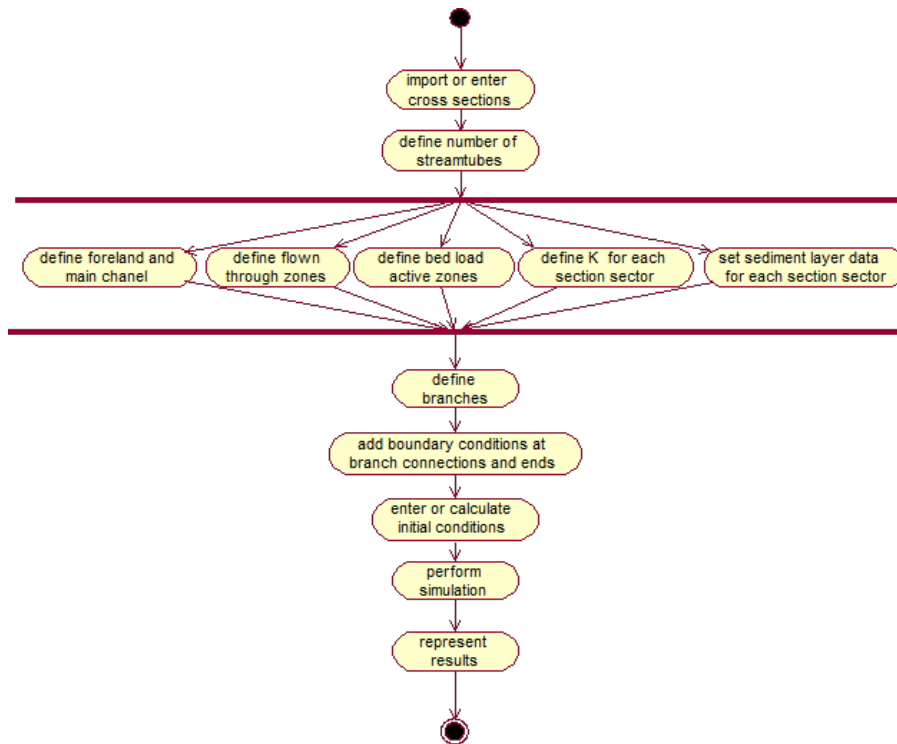
Figure 1.9 *Activity diagram Sediment balance in a river*

1.5.2.1   Sediment balance in a river 1-D

1.5.2.2   Flood 1-D + 2-D
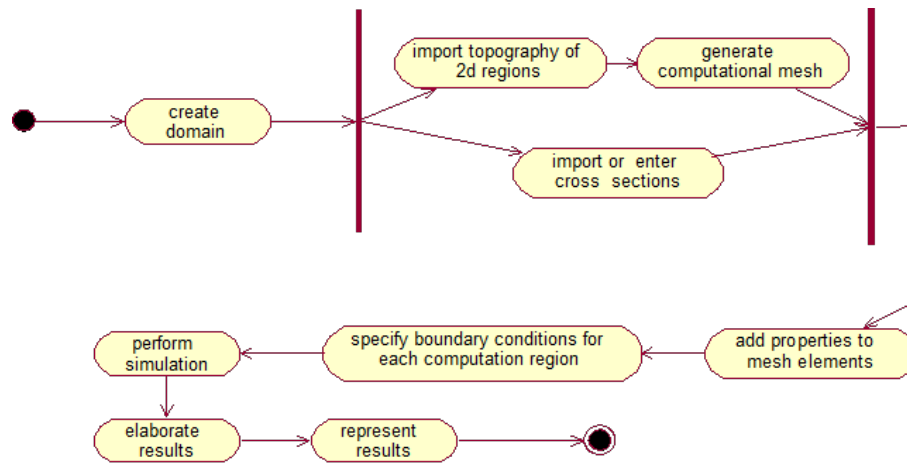
Figure 1.10 *Activity diagram Flood 1-D + 2-D*

2

Simulation procedure

2.1 Run BASEMENT with graphical user interface (GUI)

The start and executing of the BASEMENT software is described in the part "Introduction and Installation" of this manual. Further details concerning the GUI of BASEMENT are explained in Section 4.1.

2.2 Run BASEMENT on Microsoft Windows

When running BASEMENT under Microsoft Windows operating system, the easiest way to start a simulation is by double clicking on the command file ending with ".bmc". Otherwise the program can be executed choosing the "Run. . . " command for the Windows "Start"-menu or by double–clicking the executable file in Windows Explorer. After running, BASEMENT will open the graphical user interface where the command file can be loaded and the simulation can be started.

BASEMENT creates an initialization file in the user's HOME-directory 'bm.ini', which stores the present work directory and scenario name to ease the input procedure for repeated simulations of the same scenario.

According to the existence of a main control block, either BASECHAIN_1D or BASEPLANE_2D, the appropriate simulation will be carried out.

2.3 Run BASEMENT on Linux

BASEMENT runs as a console application without screen graphics output. On LINUX you open a console and type 'BASEMENT_vX.Y' (replace X.Y with current version number) to start the executable (if no environment variables have been set, change into your 'bin' directory of the installation path). After running, BASEMENT will open the graphical user interface where the command file can be loaded and the simulation can be started.

BASEMENT creates an initialization file (as a hidden file) in the user's HOME-directory '.bm.ini', which stores the present work directory and scenario name to ease the input procedure for repeated simulations of the same scenario.

According to the existence of a main control block, either BASECHAIN_1D or BASEPLANE_2D, the appropriate simulation will be carried out.

2.4 Run BASEMENT in batch mode

Executing a simulation with BASEMENT normally opens the graphical user interface (GUI) and requires some input from the user, e.g. to select the model data and to confirm warnings generated by the program at the start and during run-time. But BASEMENT can optionally be started without any graphical interaction and without user input. This feature is especially useful if one or several models shall be run automatically via batch or script file.

But be aware that executing in batch mode requires special attention, since significant warnings may be suppressed without being noticed! It is recommended to study the generated 'log-file' after the simulation to check the program output for warnings which may have been generated during run time.

Executing in batch mode can be specified at the program start of BASEMENT using program arguments. The following list of program arguments is supported at the moment and can be specified in any order.

Table 2.1 *List of program arguments.*

Command line arguments	Description
- b	BASEMENT is run in batch mode without manual user input.
- f filename	The file flag '-f' and the space separated 'filename' argument specify the model filename which shall be executed. The filename must be the full path including the name of the *.bmc-file. (Please note: no empty spaces are allowed to be part of the filename!)
- version	Display the version number of the BASEMENT executable.
- h	This help flag '-h' displays all available command line arguments.
- doc	This generates a reference documentation of all blocks and parameters in .html format.
-log [1-4]	The number [1-4] defines the level of log output generated by BASEMENT.

Example:

Type the following line to execute the model 'Scenario1.bmc' in batch mode without user input.

```
BASEMENT_vX.X.exe -f d:\data\Scenario1.bmc -b
```

2.5   Restart the simulation in BASEplane from a given solution

In 2D simulations with BASEplane an improved and enhanced method for the restart from existing solutions from old simulation runs was implemented. Such a 'restart-file' contains all relevant information and data which is needed for the continuation of an old simulation and therefore often needs a lot of disk storage. These data are now stored in a binary format to reduce the needs for disk storage and to obtain smaller files. For this purpose a standardized CFD format was chosen (CGNS – CFD General Notation System, www.cgns.org). This standardized data format additionally can be edited and visualized by different programs and simplifies data exchange between different programs. (A simple Data Viewer for CGNS files is the adfviewer which can be found on www.cgns.org). The restart from an old solution is possible for the hydraulic computations, bed load transport computations and suspended load computations.

Using this new file format the possibilities to continue a simulation from a given solution were enhanced. It is possible to continue a simulation not only from the last point in time of the old simulation run, but also from different times of the old simulation. This can be very helpful especially for large simulation runs with long durations. For example, if there was an error in the inflow hydrograph at a point in time, than the simulation can be restarted shortly before the time when the error occurred with a corrected hydrograph and without having to repeat the whole simulation from beginning.

<div align="right">

3

</div>

Preprocessing

3.1 General

3.1.1 Requirement

The main purpose of the pre-processing activities is to define the project and the different scenarios, to prepare the available (topographic) input data and to put it in the format needed by the computational module as shown in the overview in Section 1.1. Additionally, the choice of computational approaches and boundary/initial conditions has to be made.

3.1.2 Project / Scenarios

A project is defined for one region which is to be analyzed. Within a project, one or more scenarios can be embedded. To manage a project, it is therefore necessary to define which scenarios, files, and other elements belong to it. This includes the information where all these elements are stored and how they are connected to each other.

For the same project, several scenarios can be created. For each of them a simulation is executed. The different scenarios can vary regarding the computational mesh, other input data, the used approaches and the boundary conditions. In addition, type, location and time of the results to save have to be specified. The simulation can be done in 1, 2 or 3 dimensional computation or as a combination of them. The phenomena to be considered have to be chosen as well, as there are at the moment: dam break, bed load, suspended load, debris flow, mobile bed, erosion, collapse, pollutants etc.

3.1.3 Input data

As mentioned in the introduction, there are three main types of data to be provided for a simulation: topography, hydrology and sediment data. All data has to be transformed in a certain way to satisfy the input specifications of the main computation program. The precise specifications are available in the reference manual.

3.1.3.1   Topography / Computational mesh

The most important, and most laborious to achieve, input is the retrieval and setup of the computational mesh. It is based on the real world topographic data. At the end of the pre-processing task, a grid in a defined shape and format is available for the simulation module.

This mesh can be generated in different ways. The topographical raw data may come from a cluster of points described by three dimensional coordinates, digitized contour lines, break line polygons or cross sections and probably is furnished in different file formats, which have to be interpreted and transformed. For a 2-D simulation, in a first step a TIN (Triangulated Irregular Network) has to be generated. Then this mesh has to be modified and refined in order to satisfy the special mesh qualities needed for stability of computation. For the 1-D model it might be necessary to interpolate cross sections or deduce cross sections from a DEM (digital elevation model).

3.1.3.2   Hydrologic data

At all boundaries of the mesh, hydrologic boundary conditions such as hydrographs or time series of water levels have to be provided. In the case of a simulation with sediment transport, also a sediment concentration in the water might be needed.

The choice of boundary data has to be made with care, considering the type of event being simulated. The data might be created especially for the wanted simulation hypothesis or be adapted from existing measured or statistical data. Some times, special manipulation of the data turns out to be necessary, for instance to omit discharges not relevant for sediment transport. Finally, again, the hydrologic data has to be put in the format wanted by the program.

3.1.3.3   Sediment data

Sediment data primarily comes from water sediment samples or surface samples like the line method. Possibly, a grading curve still has to be built from this raw data. Then the number of desired grain classes for the simulation has to be chosen and the characteristic grain classes need to be identified. Based on the grain distribution, other values like roughness or angle of rest may have to be calculated.

3.1.4   Boundary conditions

The boundary conditions define quantities at the margin of the computational region during the whole simulation time. These are typically hydrographs at the inflow boundary and water levels at the outlet boundary. In general, different special conditions can be applied by defining for instance the water level of a lake, weirs, steps, etc.

3.1.5   Source terms and local properties

For every mesh element, several characteristics can be specified, for example not flown through cells, mobile bed, bed load potential, k-value, roughness, shear stress or others.

Sources and sinks of water and/or sediments might be added within the computational region.

3.2 Model input data

3.2.1 Topographic data sources

The topographic raw data in the form of digital terrain information builds the fundamentals for grid generation and further numerical simulations using the BASEMENT software. In the case of 1-D simulation, the raw data consists of recordings of river cross-sections. In a 2-D or 3-D model, the raw data is built from point clouds, height contour lines or a digital terrain model (DTM).

Dependent on the assignment and its requirements, most of the raw data usually gets collected by experts. However, there are some extensive topographic models with different quality available, see e.g. the cross-sections database from the Federal Office for the Environment (BAFU) and high resolution terrain models from "swisstopo" or "Swissphoto AG".

The next sections provide a short overview of different methods and data sources of data sources for topography and also hydrology and sediment. Most data sources in the next few sections are somehow related to Switzerland. Other regions and countries provide similar data – depending on the actual law.

3.2.1.1 Terrestrial data or surveying with differential GPS

Surveying or special engineering agencies provide terrestrial terrain data or terrain data obtained by differential GPS (DGPS) of the desired quality. A pure terrestrial recording is more expensive because at least two workers are needed in contrast to a DPGS scan, which can be done by one employee only.

The survey with GPS needs a certain visibility concerning the GPS satellites. The accuracy depends on the exactness of positioning for the reference point but usually lies in the range of a few centimetres.

The terrain data is delivered in different formats as e.g. xyz-coordinates or GEOBAU standard. Be sure to specify your claims on quality of the data as resolution, typical terrain deformations but also the desired data format when asking for a tender offer.

3.2.1.2 Official topographic survey

According to a decision of the Swiss federal assembly ("Einführung der amtlichen Vermessung" AV93), the bureau of land charge register provides digital cadastral data almost over the whole country (ca 80%) and data from the country's information system (ca 30%).

The cadastral information is just of secondary use as the parcels are all flat. However, the ground plans of certain constructions and buildings may be helpful.

The data from the country's information system can be used (with some additional work) for the definition of boundary conditions, e.g. data about floor cover types.

The data format used for the AV93 act is called INTERLIS (Data exchange mechanism for "Landes-InformationsSysteme").

3.2.1.3 Laser scanning

Essential factors for an accurate triangulation concerning topographic raw material are accuracy of position and height as well as the density of the measured points. Of special interest are therefore height model data taken by Airborne Laser Scanning (LIDAR) as e.g. provided by "swisstopo". This method allows for a good data base within forests. Only in dense coniferous forest, the determination of the terrain model tends to be impossible. Of course, as with all measurements from the air, neither ground nor surface levels from water bodies can be obtained.

As the surface model generated from the LIDAR method distinguishes between terrain, buildings and vegetation, it provides a good starting point for further processing towards a numerical grid.

The federal office for topography ("swisstopo") has commissioned Swissphoto AG to collect height data all over Switzerland for the identification of agricultural areas. However, this project is restricted to areas below 2000 m asl. For higher located zones (which could be relevant for debris flows), the data could be produced with reasonable costs. Until end of 2007 this project shall be finished.

"swisstopo" delivers their terrain data in two formats:

- DTM-AV raw: a point cloud with averagely 1 point per 2 m^2
- DTM-AV grid2: an interpolated 2x2m mesh

The accuracy of position and height is about +/- 0.5 m. Additionally, "swisstopo" also works with 1x1m meshes which reach an accuracy below +/- 0.3 m.

For large areas and a high density of grid points, the amount of data is beyond the resources even of nowadays computing power. Therefore, there exists a variety of algorithms which eliminate all unnecessary points in a certain area that have no influence on the shape of the triangulation. This may reduce the amount of data by 10 – 60 %.

3.2.1.4 Sources and quality of data

One main factor for the quality of the results is the accuracy of the original topographic data. The accuracy varies depending on the objective of the recording, as well as on the recording methods. The accuracies of the most diffused methods are listed in Table 3.1.

Table 3.1 *Accuracy of current measurement methods*

	Situation accuracy	Height accuracy	Observations
Terrestrial	+/- 2 cm (1)	+/- 5 cm (1)	
Sonic depth finder	+/- 5-10 cm (1)	> 10 cm (1)	

	Situation accuracy	Height accuracy	Observations
GPS	+/- 1-3 cm (1) +/- 1 cm (4)	+/- 5 cm (1)	At least 4 satellites
Orthophoto		+/- 2-3 cm (1) 0.2 – 0.3 ‰ of flying altitude (3)	3 cm with photo scale 1:2000
LiDAR		+/-20-30 cm (2) open lea +/- 15-20 cm forest insecure (3)	
Radar (InSAR)		open lea +/- 20 cm (3)	

(1) *Gewässergeometrie, Landesanstalt für Umweltschutz Baden-Württemberg, Karlsruhe 1999.*

(2) *Swissphoto AG (verbal).*

(3) *Leitfaden Qualitätssicherung Photogrammetrie und DTM-Generierung, Konferenz der Kantonalen Vermessungsämter, 2000.*

(4) *GPS global positioning System, "swisstopo".*

The products which are directly available in Switzerland and their accuracies are listed in the following tables.

__Table 3.2__ Products offered by "swisstopo"

Product	Density	Source	Format	Accuracy
DHM25 Base Model DTM	35-1600 point /km^2	1:25000 map Vectorized contour lines and contour lines in lakes, lake perimeters and spot heights, main alpine break lines (photogrammetric data)	Arc View Shape- Files, DXF,GEN, BMBLT	Mean error 1.5 to 3 meters depending on the regions
DHM25 Matrix Model DTM	1600 point/km^2	Interpolation of the Base Model on a 25x25m grid	MMBLT, MMBL, AIGRID, XYZ, DXF, VRML	

Product	Density	Source	Format	Accuracy
(1)DTM-AV raw (2)DTM-AV grid2	1 point for 2 m^2	Based on official measurement with Lidar Airborne Laser Scanning(1) 2x2m grid interpolated from DTM-AV raw (2)	ASCII , Interlis, others possible if requested	Height accuracy +/- 50 cm
(1)DSM-AV raw (2)DSM-AV grid2	1 point for 2 m^2	With buildings and vegetation (1) 2x2m grid interpolated from DSM-AV raw (2)	ASCII , Interlis, others possible if requested	Height accuracy +/- 50 cm +/- 150 cm for vegeta- tion

Table 3.3 *Products offered by Swissphoto AG*

Product	Density	Source	Format	Accuracy
DSM	10x10m, 20x20m, 50x50m	Photogrammetric interpretation of aerial photos from 1995/96	ASCII xyz or GRID	Height accuracy: 3-5m midland and 7-10 mountains
DEM LIDAR (on commission)	31000/(250x160m), spacing 1-1.3m	Lidar Airborne Laser Scanning	ASCII xyz, ASCII ArcInfo, . . .	Height accuracy 30 cm

3.2.2 River related data sources

The following list provides some more Swiss data sources for real world problem modelling and boundary conditions.

River cross-sections: Primary use for 1-D models; also qualified for 2-D models to improve topography data within a rivers area. Terrestrial mapping, optimal quality and format according to the cross-section database of the Federal Office for Environment FOEN (formerly FOWG / bafu.admin.ch).

Terrain topography: Basic dataset for simulations with spatial (2- or 3-D) models. Optimal quality and format (ASCII x y z) as swisstopo DTM-AV / DOM-AV (New swisswide high resolution DTM/DOM based on laser scanning). (http://www.swisstopo.ch/de/digital/dom.htm)

Surface texture and special buildings: Declaration of roughness coefficients, porosity and erosion resistance of surfaces; Optimal format INTERLIS-1, as e.g. DOM-AV or floor cover plans AV93 (generated by the cadastral register on demand of the cantons).

Sedimentological /geological data: Surface characterization of the riverbed (mostly using line samples), of the erosion capable underground (volume sampling or geological drilling) and of the flow induced transported material (grain size distribution from sieve analysis).

Possible data sources: geology of cantons, FOEN (formerly BWG), Nagra, in situ suspended load and bed load measurements.

The estimation of sedimentologic data and its analysis is surely the most time consuming part. At certain water bodies, a continuous monitoring of suspended load exist. However, the granulometric size distribution is seldom measured. Therefore, the bed load near the surface but also the suspended load often have to be estimated under certain assumptions or they will have to be measured.

Hydrological data: Temporal or stationary boundary conditions for the numerical model, e.g. inflow discharges, water levels and local sources or sinks.

Possible data sources: Hydrological data from the federal measuring facilities, specific rainfall-discharge models (e.g. IFU/ETHZ), retention models.

For larger water bodies within Switzerland (observed by the FOEN), time series and hydrographs are available online. Smaller watercourses are often not covered and need either a hydrologic model or new measuring/monitoring to determine the discharge.

3.2.3   Characteristic quantities of riverbed

The numerical models used for the computation of hydraulic behaviour are always declared to be either 1-D (flow in direction of main flux / x-axis) and/or 2-D (horizontal, depth averaged flow field). However, the fundamental terrain topography is always three-dimensional. The spatial discretization of the transport equations is based in 1-D on cross-sections or in 2-D on an unstructured grid of mostly triangles. The vertical component consists of different layers separated by a planar joint face for both, 1-D and 2-D simulations.

The characteristic values are indicated relative to a joint surface (e.g. riverbed level) or to a layer (e.g. granulometric composition of bed material in layer S1). Their properties are represented in the balance point of the corresponding joint plane. This results in a simplified model of the layering for a computational cell. The characteristic data used in the model to describe the state and conditions of the surface are defined as followed:

Position:

$$
\text{Soil - properties} =
\begin{cases}
z_R : & \text{bottom elevation (bed level)} \\
z_{Sub} : & \text{elevation level of the lower, closing joint plane of the} \\
& \text{corresponding sublayer } Sub_i \ (i \in 1...n, \ with \ \text{n = maximum}} \\
& \text{number of Sublayers)} \\
z_0 : & \text{elevation level of the lower joint face of sublayer Sn, which} \\
& \text{finishes the model downwards, respectively describes the} \\
& \text{position of the non erodable underground.}
\end{cases}
$$

(all variables are measured in [m.a.s.l.] or in [m] relative to a userdefined reference system)

Attributes:
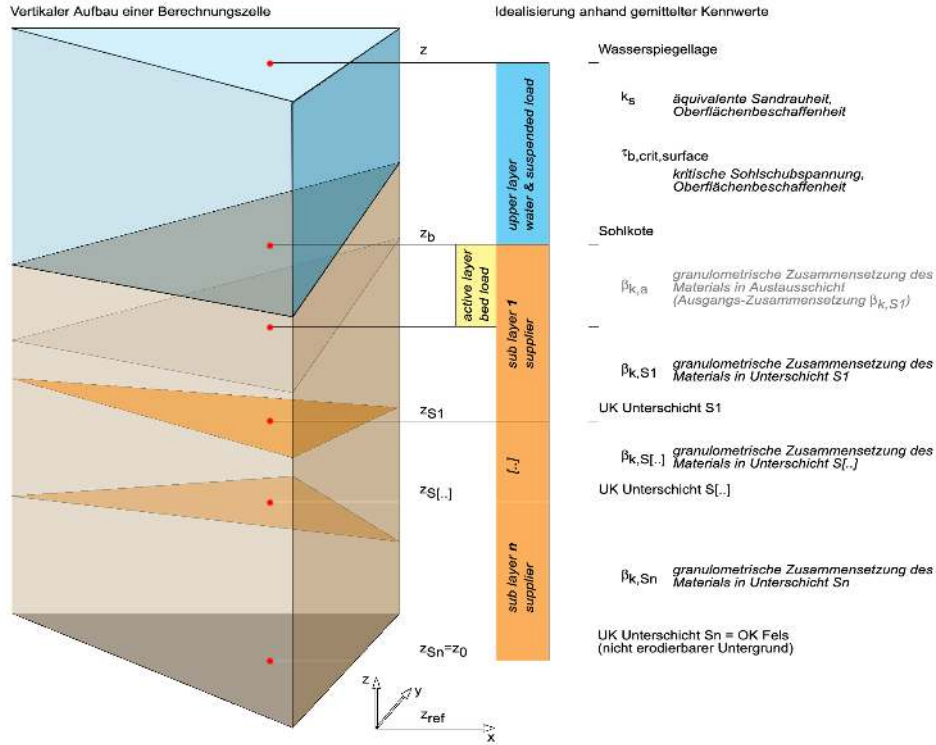
Figure 3.1 *Visualization of the vertical setup for a computational cell within the models.*

$$
\text{Vegetation--} \atop \text{ground covering} \quad = \quad
\begin{cases}
k_s \ [m]: \ \text{equivalent sand roughness based on Nikuradse:} \\
\text{description of the surface roughness (e.g. grass, sealed plane,} \\
\text{etc.). (default value or derived by grain distribution of the} \\
\text{first sublayer)} \\
\tau_{B,crit} \ [N/m^2]: \ \text{critical value for begin of sediment movement at} \\
\text{the surface, in the sense of a rised erosion constancy, e.g. by} \\
\text{natural cover (default values)}
\end{cases}
$$

$$
\text{Soil--properties} =
\begin{cases}
\beta_{g,Sub_i} \ [-]: \ \text{mass fraction of the } g^{th} \text{ grain class relative to the} \\
\text{total mass of sublayer (default values according to the} \\
\text{grain size distribution)} \\
\beta_g \ [-]: \ \text{mass fraction of the } g^{th} \text{ grain class relative to the} \\
\text{total mass of the active layer (possible default values} \\
\text{according to line samples if available)}
\end{cases}
$$

The listed attributes are mostly not completely determinable directly. For example, the equivalent sand roughness k_s based on Nikuradse is an experimentally obtained value for the flow resistance of different kind of surfaces. In the case of a more complex soil structure, k_s may consist of several components. Its actual value has to be determined by a calibration of the numerical model. The same holds true for the critical value $\tau_{B,crit}$ which defines the beginning of movement at the soil surface. This value is affected by the character of the bed surface like natural cover or sealing (at farmlands). Without special conditions, the critical value can be estimated via the grain size distribution.

The mass fraction β_{g,Sub_i} of grain class g in sublayer Sub_i has to be derived from an experimental grain size distribution using a sieve analysis. The material to be sieved may

come from a near surface sample take (e.g. by daggering) or a drill hole. The granulometric composition directly at the surface can be estimated by a line sample. Accordingly, the corresponding mass fractions β_g in the active layer can be determined.

3.2.4   Processing the raw data

It was shown in the previous sections that the numerical model needs some specific characteristic values which usually are not given directly by in situ measurements. Often, some experience in handling computational simulation models is necessary to reproduce the natural conditions in an adequate way and obtain realistic results.

3.2.4.1   Topography

The topographic raw data for 1-D or 2-D simulations comes in the form of cross-sections or by a digital terrain model (DTM). For a 1-D model, cross sections are mostly of accurate quality and need just few corrections and adaptations. However, most DTM's serving as a base for a 2-D model need an elaborate revision because of the surface triangulation:

- the resolution of the DTM and the computational grid are not congruent (e.g. areas of lower interest shall have a coarse grid to reduce computational effort – the DTM usually has a uniform point density)
- the course of water bodies is not continuous or cross sections are not complete (e.g. data only available up to the water level)
- passages are not omitted (e.g. bridges)
- modelling and representation of buildings is not a priori known (e.g. height of buildings, retention volume, flow resistance)
- relevance of waste edges and artefacts are costly to determine (e.g. small walls, temporary dumpsites, hedgerows)
- etc.

The choice of software resources for processing the topographic raw data is huge, but often, these programs are somehow limited to specific applications and data formats. It is therefore recommended to specify the desired data format and resolution in advance to avoid unnecessary repetitions.

More detailed information about grid generation is given in the next chapter.

3.2.4.2   Hydrology

The hydrologic raw data for discharge and water levels usually has a non uniform resolution in time. Often, the measurements are indexed with date and time. For the simulation module, the time series have to be converted to a system with standard units (e.g. seconds). In an easy case, the converted data can directly be used as a boundary condition. The simulation module will then interpolate the desired values to the actual computational time.

Further modifications may be necessary if e.g. just some time slots have to be simulated or if the main processes for a phenomenon like sediment transport occur at no more than a certain water discharge.

3.2.4.3   Grain size distribution

As mentioned earlier, the granulometric composition of the erodable material gets measured by a line sample or a sieve analysis and results in a grain size distribution (grain diameter versus weight percentages relative to the total sample weight). For the simulation, this distribution has to be discretized and desired grain classes have to be determined. Each grain class is defined by its medium diameter and has a percentage. The choice of the numbers of grain classes and thus the resolution of the material composition depends on the size of the problem and the computing power available. The classification of the grain diameters which represent a grain class strongly affects the transport behaviour of the whole material and the efficiency of the transport model.

It is recommended to run several simulations with one but also with more grain classes and to compare them.

3.2.5   GIS Interface

A geographic information system (GIS) as assistance for a numerical simulation model shall provide several data in best possible spatial and temporal resolution but not necessarily in a model inherent format. The data structures described in the following are just proposals. For all of them, one has to decide a priori whether the raw data shall be saved per se or in a processed form (raw data and pre-GIS-processed data).

3.2.5.1   Administration of geographical data

Generally, one distinguishes two kind of data types: matrix data (based on a uniform matrix, e.g. air taken pictures) and vector data (e.g. property borderlines).

Vector data serves for the visualization of the geometry and uses different data structures. Objects with simple geometry (see "OpenGIS Simple Feature Specifications") are e.g. points, lines and polygons. For complex geometries (see "OpenGIS Geography Markup Language (GML)"), additional information as topology and other attributes (e.g. time) are required.

A further possibility to structure geographical data in a GIS is a splitting on different topic layers. Digital terrain models with height information are represented in a GIS either as TIN (triangulated irregular network) or as a GRID (structured mesh). With regard to numerical simulation models, the topology of a TIN or a GRID corresponds to unstructured resp. uniform computational grids.

3.2.5.2   Plane assignment for point data

In a GIS, real, area-related conditions are given in an idealized way. On the one hand, there is a limited number of attributes and on the other hand the values over the surface are averaged. Partially, the data base is just point wise available and just sparsely against resolution and process scales of numerical simulation models, e.g. at geological
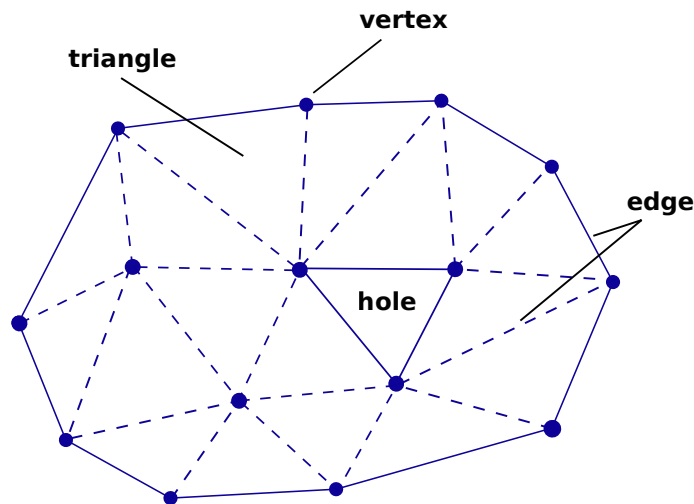
Figure 3.2 *Unstructured grid: Triangulation*

probe holes. Therefore, the coarse values have to be extrapolated on the corresponding surfaces of the numerical grid. The determination of the corresponding surface, or i.e. the surrounding polygon, needs different geometric constructs as e.g. Dirichlet Tesselation (Voronoi-Polygons). Additionally, one should consider related information from superior scales (e.g. geological maps with layering data and downcasts).

3.2.5.3   Interoperability and standard interfaces

To assure interoperability – meaning system independent communication – between different information systems, several norms for a standardized data representation exist. Examples are the "OpenGIS Geography Markup Language (GML)" or the norms used in Switzerland: INTERLIS (SN 612 030 / 612 031) and GEOBAU (SN 612 020).

3.3   Grid generation

3.3.1   Introduction

The numerical methods used for the approximation of differential equations needed in flow simulations are based on a discretization of the domain in simple small shapes. The complex of these elements forms a mesh. In BASEMENT, the finite volume method, being particularly valuable for fluid dynamics, is used.

To describe a topographic surface, the terrain data is often triangulated to allow a perspective representation of the topography. The result is a piece-wise linear interpolation of the surface. This triangulation can be used directly as mesh for simulations, as it permits to have the original data in the vertices of the initial grid and no interpolation is necessary. However, most of the time, the mesh has to be transformed somehow. Regions of high interest need some mesh refinement for higher accuracy and areas of lower interest are often coarsened to save computing power.

In numerical simulations, two types of meshes are used: structured and unstructured grids. Structured grids are simpler to use, but need an interpolation of the data to determine the

values at the desired vertex position if the input is an irregular point cloud. In addition, they are not sufficiently flexible to fit complex geometries.

Among unstructured meshes, the triangulated irregular networks (TIN) are most convenient because they fit the irregular distribution and complex geometry of the topography best. Furthermore, they allow for a rapid transition from small to large elements and the insertion and conservation of break lines. BASEMENT is built on unstructured grids.

The computational mesh consists of vertices, edges which connect the vertices and elements (control volumes) which are bounded by the edges.

The triangulation of terrain data is a special case of mesh generation, as the vertices do not only have a position in the coordinate system but also elevation information. This is a so-called 2,5 dimensional problem.

3.3.2   Topographical input data

The original terrain data typically has a very irregular distribution and a locally variable density. Data can be available in different shapes:

- A cloud of points (x, y, z) e.g. digital elevation model;
- Polygons e.g. digitalized contour lines;
- PSLG (planar straight line graph) e.g. a DEM with addition of break lines. This is the most general case, in which the input is a set of vertices and non-crossing line segments that must be conserved in the triangulation.

Although the raw input data, e.g. from a point cloud, could be directly used as a grid, usually further transformations are performed to gain a suitable computational mesh. A suitable mesh is mainly defined by its quality.

3.3.3   Mesh quality

3.3.3.1   Shape of mesh elements

The shape of the elements of the meshes has an important effect on the applicability of numerical methods. Speed (due to convergence time), accuracy and stability of a simulation depend strongly on the quality of the employed mesh. Therefore, it is important to produce the best possible triangulation for the application.

Size, shape and number of the elements play an important role for the quality of the mesh. Possible characterizations of a triangulation, which can be optimized depending on the application, are:

- minimal angle
- maximal angle
- maximum edge length
- total edge length
- maximum height
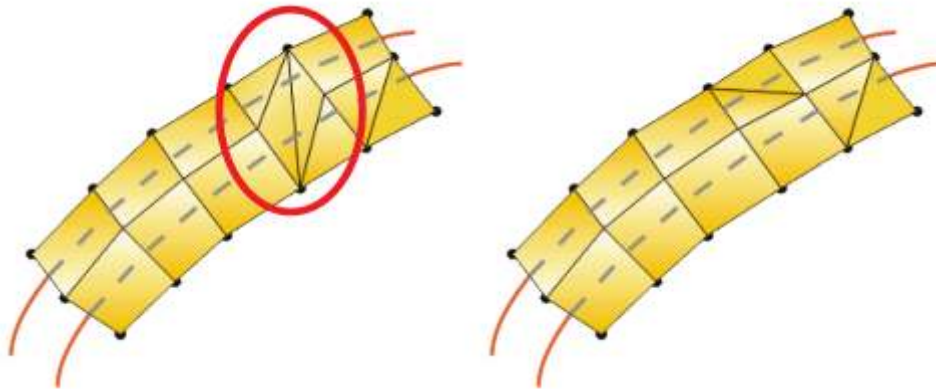
Figure 3.3 *Left: ambiguous quadrilateral element with false break line; right: correct discretisation of the dyke crest.*

- area of the triangle
- aspect ratio: ratio of the length of the longest side to the height (definition after Bern and Eppstein (1995), other similar definitions existent)
- aspect ratio: ratio of the circumference and the in-circle of a triangle
- roughness of the piecewise linear interpolation of a 2,5 D problem (The roughness of an interpolated surface can be measured e.g. by the Sobolev semi-norm)

The criteria which determine the quality of a mesh differ depending on the application (respectively partial differential equation) and the numerical method implemented. Also, the possible optimization depends on the constraints given by the original data. It is recommended to avoid L-shaped overall areas (non convex hull) as such problems often lead to numerical instabilities in the concave corner.

In general, a height aspect ratio, very small angles and especially very large angles are considered as bad, as they lead to a poor numerical condition of matrices and increase the approximation error, which arises with the element size in general, as well. Big differences of size between neighbouring cell elements can have a negative influence on the numeric simulation too.

3.3.3.2 Ambiguous gradients

A further mesh quality criterion specific to meshes consisting of quadrilateral elements, or meshes with mixed element types, is the problem of ambiguous gradients. Quadrilateral elements are defined by four nodal points. These nodes ideally have elevations which guarantee that the four nodes lie within a plane. But in case of strongly varying terrain topographies, a bad placement of quadrilateral elements can lead to situations where this is not the case. Such elements are deformed and have an ambiguous gradient. Figure 3.3 illustrates a quadrilateral element with ambiguous gradient which is situated across a river dyke.

In such situations a splitting of the quadrilateral element into two triangles becomes necessary. The selection of the correct break line within the quadrilateral element (here, along the dyke crest) must usually be done manually according to local topography. To
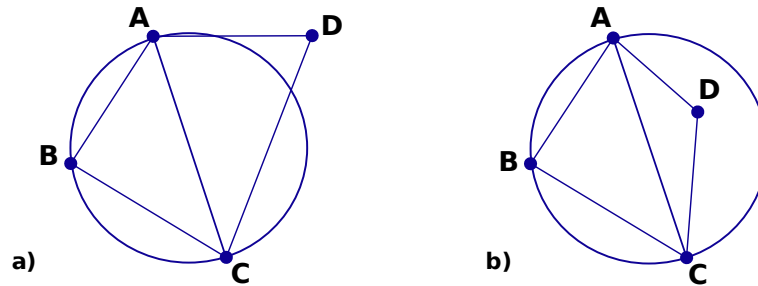
Figure 3.4 *a) Empty circle criterion satisfied. b) Empty circle criterion not satisfied*

facilitate this task, most grid generating tools (e.g. SMS) offer special features for detecting quadrilateral elements with ambiguous gradients in the mesh.

3.3.4 Issues on triangulation

Mesh triangulation and grid refinement play an important role in almost every numerical simulation. Therefore, a lot of different techniques have been developed to achieve suitable computational meshes. The user is basically free to use any tool or method which generates a mesh of accurate quality out of the raw data. This section shall give a short overview on some popular triangulation methods.

BASEMENT does not provide an automated routine for mesh generation. Therefore the plugin BASEmesh for the free and open source geographic information system (GIS) software *Quantum GIS* (QGIS) was developed (see Section 3.3.5).

3.3.4.1 Delaunay triangulation and constrained Delaunay triangulation

The Delaunay triangulation is one of the most employed triangulation methods because it optimizes several quality criteria: It maximizes the minimal angle and minimizes the maximum containment circle radius, the maximum enclosing circle radius and the roughness of a piecewise-linear interpolation. It also provides good results regarding the minimization of the maximum angle, but it does not find a global optimum in this case.

A Delaunay triangulation has the following properties. It

- is the straight line dual of the Voronoi diagram;
- is unique;
- respects the circumcircle criterion;

The circumcircle criterion is respected if the circumcircle of every interior triangle does not contain other points.

This corresponds to say that if ABC + CDA < 180° the empty circle criterion is satisfied.

- respects the edge circle property: for each edge exists some point-free circle which passes through the end points;
- respects the neighbour property: an edge formed by joining a vertex to its nearest neighbour is an edge of the triangulation.
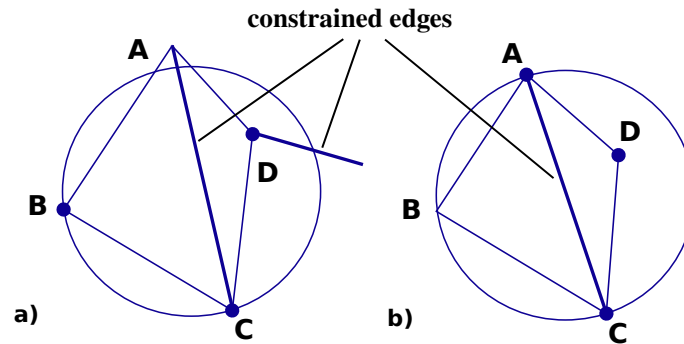
Figure 3.5 *a) edge circle criterion. b) empty circle criterion*

3.3.4.1.1 Constrained Delaunay triangulation (CDT)

The terrain data is sometimes provided in the shape of a PSLG as it contains break lines which must be conserved as edges in the triangulation. In this case, the constrained Delaunay triangulation can be used.

For the definition of a constrained Delaunay triangulation the notion of visibility of a point is needed: In a PSLG domain P a point D is visible to a point C if the open line segment CD lies within the domain and does not intersect any edges or vertices of P. Point D is visible to CB if it is visible to some point on CB.

For the CDT the edge circle and the empty circle criterion are modified as follows:

- Edge circle: for each edge a circle passing through its end-points containing no other point of the domain visible to the edge exists;
- Empty circle: the circumcircle of every triangle contains no points visible to points inside of the triangle.

3.3.4.1.2 MinMax triangulation

The MinMax triangulation minimizes the maximum angle. It has proven to be very useful in CFD (Barth, 1994).

3.3.4.1.3 Data dependent triangulation

A data dependent triangulation can be used for a 2.5 d problem. Its aim is to find the best triangulation under data dependent constraints, by minimizing a local cost function of a piece-wise interpolation. Two examples of local cost functions are described in (Barth, 1994).

3.3.4.1.4 Steiner triangulation

A Steiner triangulation is a triangulation in which extra points are added to the original data to improve the quality of the mesh. The additional points are called Steiner points. The number of Steiner points must be limited, limiting also the quality of the mesh.
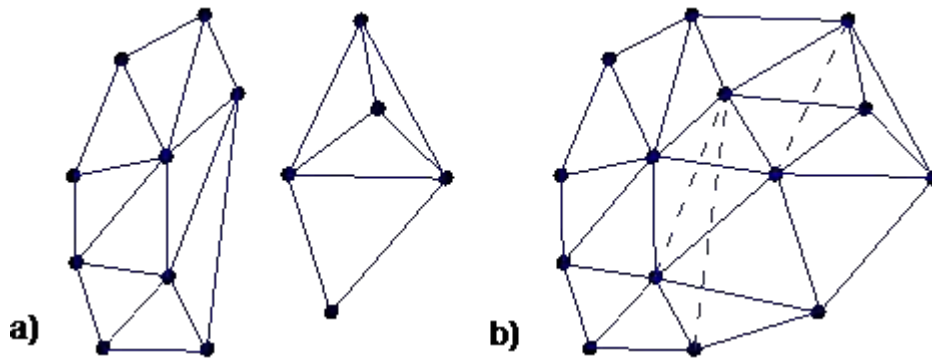
Figure 3.6 *a) triangulated half planes. b) merged triangulation*

3.3.4.1.5   Non obtuse triangulation

One of the most interesting types of Steiner triangulations is the triangulation without obtuse angles. It is a Delaunay triangulation or constrained Delaunay triangulation and also minimizes the maximum angle and maximizes the minimum height (Bern and Eppstein, 1995).

3.3.4.2   Algorithms

3.3.4.2.1   Sweep-line algorithm

The sweep-line algorithm can be used to perform a first triangulation without any quality criteria. It adds the points by x-coordinate order and then connects them to all visible points of the convex hull of the existing triangulation.

3.3.4.2.2   Divide and conquer

For the divide and conquer algorithm the point set is divided in two half planes along the x axis. Then each half plane is triangulated recursively, and finally the two planes are merged.

3.3.4.2.3   Incremental Insertion algorithms

The incremental insertion algorithms successively insert new points to an existing Delaunay triangulation. These algorithms have a worst case running time of (n^2) if the points are badly ordered. But in practice it is near to 0(nlogn) for Green-Sibson.

Watson (or incremental delete and build)

In this algorithm, after the insertion of the point, all the triangles containing the edge q are searched. Then the edges of these triangles visible to q are deleted and the new edge is connected with the vertices of the originated polygon creating new edges.

Green – Sibson

After its insertion, the point q is connected to the vertices of the triangle that contains q.

Then all suspect edges have to be swapped to satisfy the empty circle criterion if we want to obtain a Delaunay triangulation. Other edge swapping criteria can be used, for instance the minimization of the maximum angle.
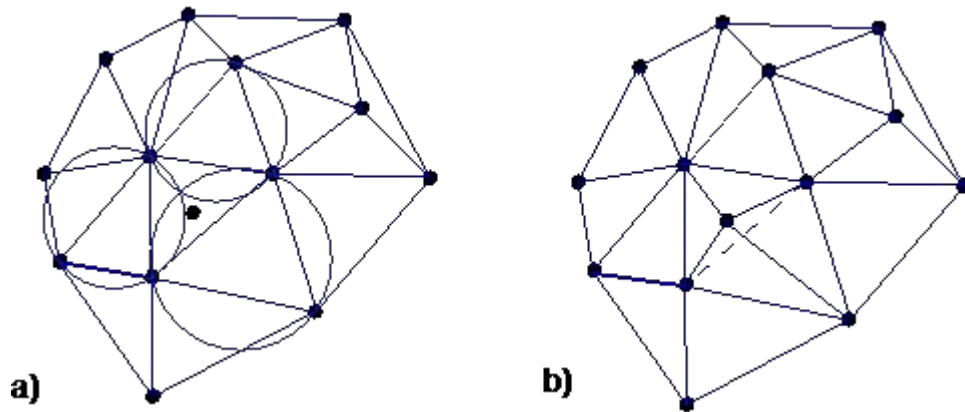
Figure 3.7 *a) point insertion and determination of affected triangles. b) new triangulation*
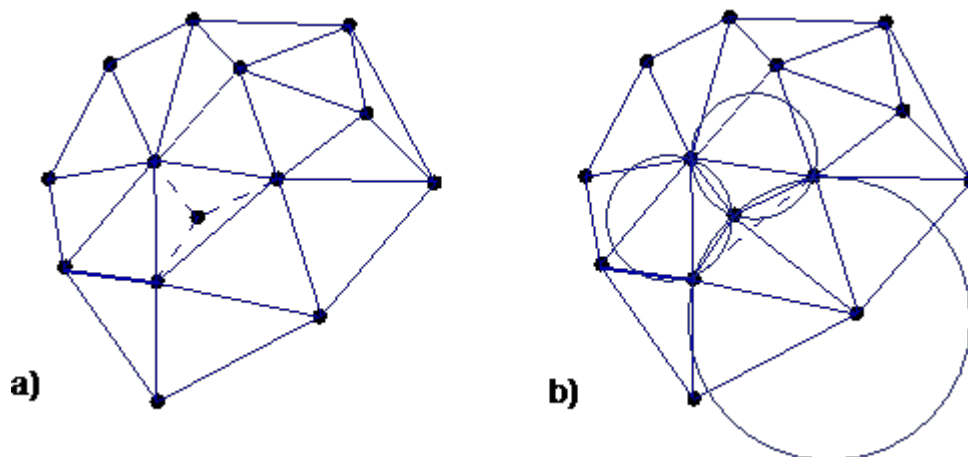
Figure 3.8 *a) insertion of the new point b) edge swapping based on empty circle criterion*

3.3.4.3   Edge flipping algorithm

The edge flipping algorithm is based on the local optimization of an initial triangulation. For each quadrilateral formed by a convex pair of triangles the diagonal is chosen with regard to a local optimization criterion.

Possible optimization criteria:

- Empty circle criterion: a Delaunay triangulation or constrained Delaunay triangulation is obtained. In this case a global optimum is reached;
- Minimize the maximum internal angle for both triangles: gives a MinMax triangulation only locally optimized;
- Optimize a local cost function: leads to data dependent triangulations (only locally optimal triangulation) (Barth (1994)).

Other properties that can be optimized with edge flipping are for instance vertex degree or total edge length, but a global optimum is not guaranteed. If the algorithm is used to obtain a CDT, the constrained edges simply are not tested because they can not be swapped.

3.3.4.3.1   Edge insertion:

This algorithm solves the problem of minimizing the maximum angle in time $O(n^2 log(n))$ exactly. Like the edge flipping algorithm, it starts from an arbitrary initial triangulation. It incidentally inserts candidate edges on a vertex of a worst triangle. The crossed edges are removed, and the remaining regions are re-triangulated. If the triangulation gets worse the added edge is rejected. This algorithm can also be used to find an interpolating surface with minimal slope in time $O(n^3)$ (Bern and Plassmann (2000)).

3.3.4.3.2   Blue, red and green refinement

This is another popular method for mesh refinement. Basically, an element with an aspect ratio out of bound gets divided into two elements by inserting a new vertex on the midpoint of the longest edge. This is the so-called green refinement. The new vertex has to be connected on both sides of the edge with the opposing vertices of the neighbouring elements.

If one of the two resulting triangles still has a bad aspect ratio, the element gets divided again in the same manner resulting in three new elements. This is called the blue refinement. It is the goal of the blue and green refinement to halve the longest side of the original triangle and whereas degeneration of the mesh in repeated mesh refinement steps is prevented. Finally, the red refinement connects all midpoints of the original edges which leads to four new elements.

3.3.5   Use of QGIS plugin BASEmesh for grid generation

In order to provide a free and open source solution for the creation of computational meshes (Pre-Processing) and to visualize simulation results (Post-Processing) the plugin BASEmesh for the free and open source geographic information system (GIS) software QGIS was developed.

We hope to enforce this powerful free GIS-solution in future to allow for a workflow which is completely based on non-commercial software. A further advantage of this workflow is that the involved programs are portable and are available for Windows-OS as well as Linux-OS.

3.3.5.1 Introduction

The plugin BASEmesh is a tool for creating computational meshes, based on the advanced mesh generator TRIANGLE by Jonathan R. Shewchuk. Furthermore, features for loading and editing existing meshes are provided. At the moment, BASEmesh only supports the automated creation of meshes consisting of triangular elements. Meshes containing quadrilateral elements, however, can be imported, manually edited and exported.

Beside 2D meshes, the plugin is also able to convert 1D meshes of different formats into each other (at the moment conversions are possible between BASEchain and HEC-RAS). The meshes created by BASEmesh can directly be used for computations with the basic simulation environment software BASEMENT. Post-processing, i.e. the visualization of simulation results, is not directly supported by BASEmesh. Still, there are various options for displaying and processing BASEMENT results in QGIS, which is briefly explained in Section 3.3.5.9 and detailed in the Post-Processing tutorial.

The manual at hand introduces the features of BASEmesh and the basics of its usage. No information is given about the usage of GIS software in general. Some basic knowledge in applying GIS software might be acquired using QGIS' excellent documentation. To get in touch with BASEmesh and QGIS fast and easy, a step by step tutorial is provided as well and can be downloaded as separate document. This manual provides the reader with an overview of the capabilities of BASEmesh, the installation procedure, and a short description of the BASEmesh components that enable creating, importing, editing, and exporting computational meshes.

3.3.5.1.1 Installation

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

To install BASEmesh, follow these steps:

(1) Start QGIS

(2) Load the QGIS plugin manager by choosing *Manage and Install Plugins. . .* in the menu *Plugins* in the QGIS main toolbar

(3) Go to *Settings* (you should now see the connection to the official QGIS-plugin repository)

(4) Click on *Add. . .* and give a name, e.g. 'BASEmesh repository'

(5) Enter the repository address: http://people.ee.ethz.ch/~basement/qgis_plugins/ qgis_plugins.xml (do not copy paste this address, because it might include line breaks)
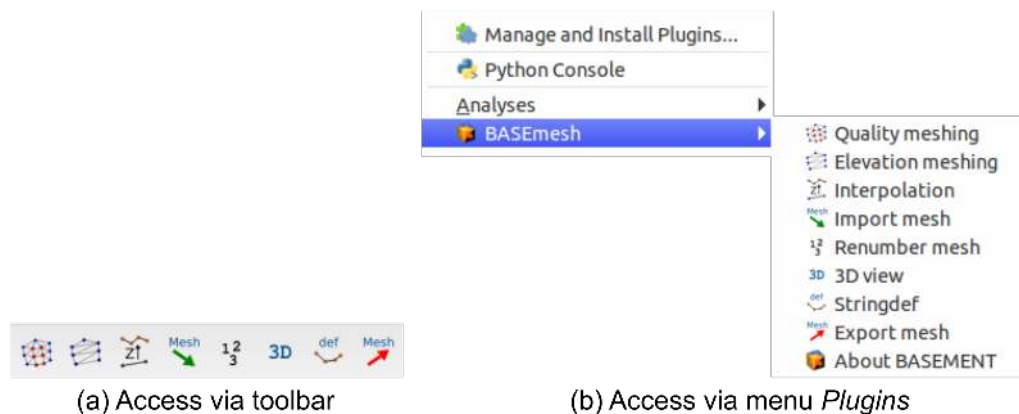
Figure 3.9 *BASEmesh representation in QGIS after successful installation*

(6) Press *OK*

(7) The additional repository should now be visible (make sure that the *Status* is *connected*)

(8) Go to *All* in the menu of the plugin manager and search for 'BASEmesh'

(9) Choose the BASEmesh plugin (if several are available, choose the one with the highest version number) and press *Install plugin*

If everything worked, you should be able to access BASEmesh via the toolbar and via the menu *Plugins* (see Figure 3.9).

3.3.5.1.2 Components of BASEmesh

The philosophy of creating a computational mesh within BASEmesh is as follows (see Figure 3.10):

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

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

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

In addition to mesh creation the possibility of manipulating an existing mesh is provided as well in BASEmesh (see Figure 3.10). Starting from a mesh you have created previously or you have imported to QGIS, you can edit the following:

Material indices This attribute is used in BASEMENT to distinguish between different materials which can be used to define friction, soil composition etc.
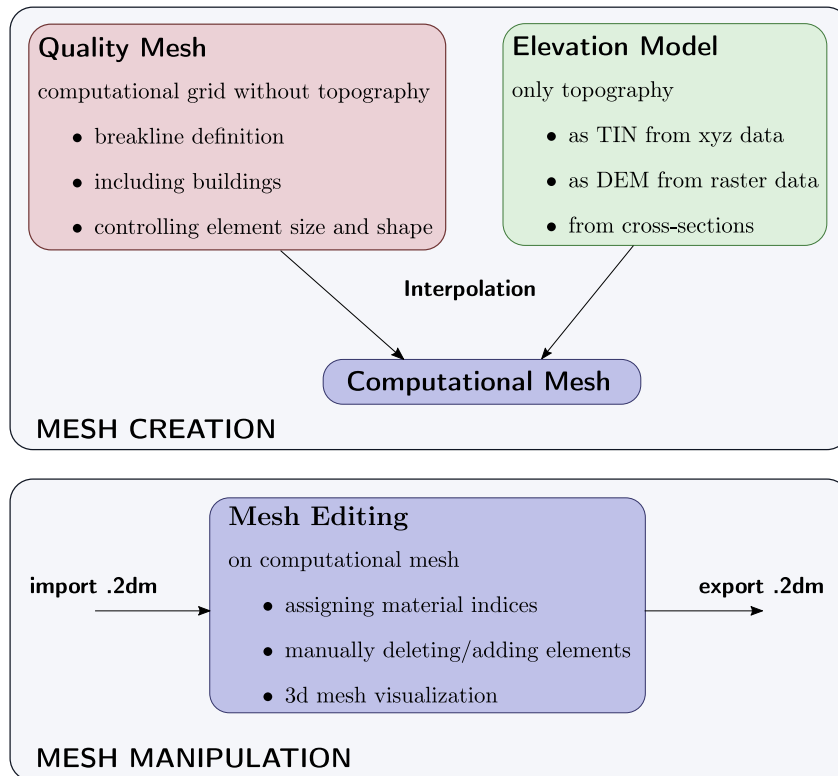
Figure 3.10 *Pre-processing components of BASEmesh*

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

Renumber your mesh After applying manual changes to your mesh, you need to make sure that the internal representation of the mesh is correct. To this end BASEmesh provides a tool called *Renumber Mesh*, where all the manual changes are incorporated into a valid mesh structure (see Section 3.3.5.1.3).

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

Node 4	
NODE_ID	4
X	635000
Y	278000
Z	486.5

Element 3	
ELEMENT_ID	3
NODE1	8
NODE2	5
NODE3	9
NODE4	-1
MATID	1
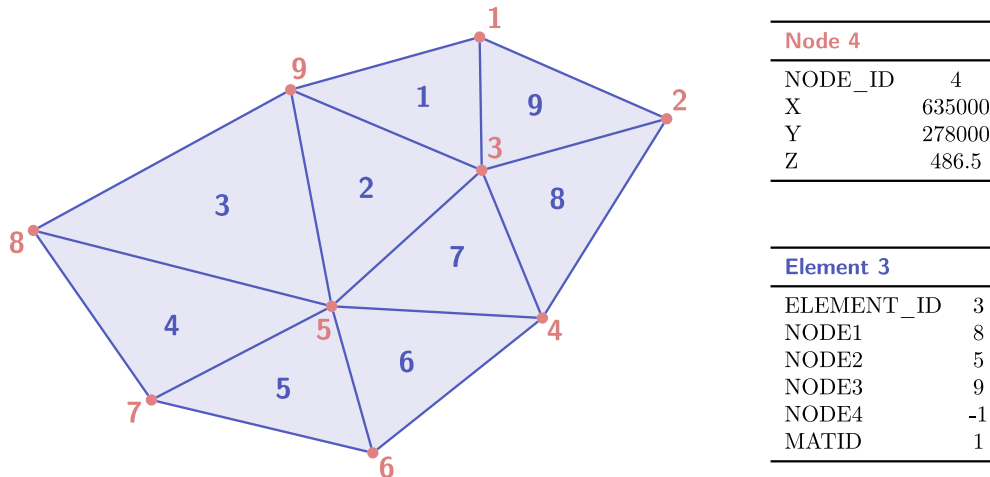
Figure 3.11 *Example of internal mesh representation with nodes as point layer (red) and elements as polygon layer (blue).*

3.3.5.1.3   Mesh respresentation in BASEmesh

Meshes generated by BASEmesh follow a specific structure. When manually editing a mesh it is of high importance to maintain this structure, otherwise it is not possible to export the mesh to a 2dm - grid which can be used for computational purposes. The mesh structure of BASEmesh is defined as follows (see Figure 3.13):

- In BASEmesh, meshes are represented by two vector layers: (1) '*meshName*_nodes' containing only the nodes of the mesh and (2) '*meshName*_elements' containing only the elements of the mesh.

- '*meshName*_nodes' contains point features (2-dimensional geome). The attributes for each feature are *NODE_ID*, *X*, *Y*, and *Z*. Please note thate the elevation information is only contained in the attribute table of the vector file.

- '*meshName*_elements' contains polygon features (2-dimensional). The attributes for each feature are *ELEMENT_ID*, *NODE1*, *NODE2*, *NODE3*, *NODE4*, and *MATID*. The attributes *NODEx* stand for the *NODE_ID*'s within the layer '*meshName*_nodes' and define the mesh connectivity.

- The *MATID* attribute is used to set material indices for each element within the mesh. BASEMENT uses these indices to set up zones with different characteristics in the mesh, e.g. to set different friction values in the river bed and in the flood plain of the domain.

- In case of triangular elements the attribute *NODE4* is -1. In case of quadrilateral elements, however, the *NODE4* attribute has to be specified.

Within QGIS, it is possible to create elements without defining the attributes *NODEx*. In this case, the tool *Renumber Mesh* must be used to update the connectivity of the mesh before exporting to a 2dm - grid.
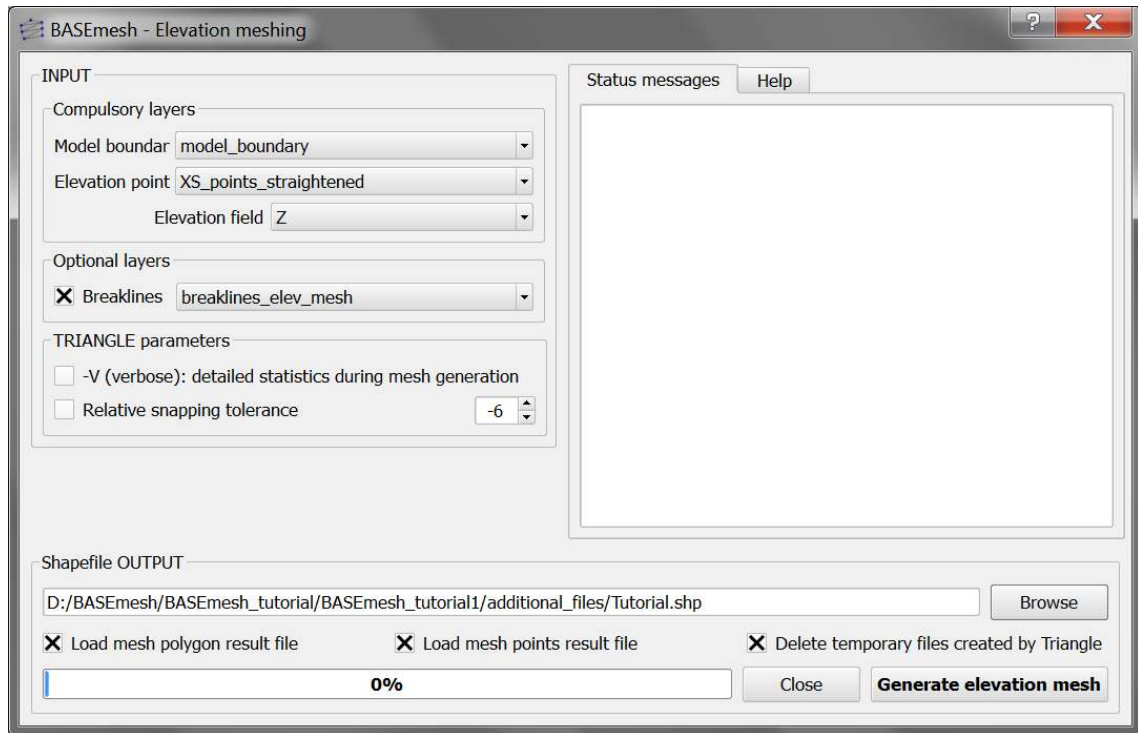
Figure 3.12 *User interface for the creation of elevation meshes.*

3.3.5.2 Elevation Meshing

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

1. *Elevation points* with x,y,z-coordinates. These elevation points may be taken from DTMs, cross section profiles, field measurements and other data sources. As input, all elevation points need to be combined into one single point shapefile.

2. *Breaklines* defining distinct interruptions of the the surface slopes (e.g. dyke crests or river side walls). Breaklines are given as line shapefile and connect elevation points with each other. During mesh generation, no triangles crossing the breaklines will be created. Be aware that all end points or vertices of a breakline must have corresponding elevation points!

3. *Model boundary* defining the extents of the elevation mesh. It must be given as polygon shapefile, which connects elevation points with each other. Be aware that all vertices of the model boundary polygon must have corresponding elevation points!

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

3.3.5.3 Quality Meshing

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

1. Specification of all geometric information about the computational domain:

   - *Model boundary*: extent of the computational domain.
   - *Break lines*: distinct interruptions of the the surface slopes (dyke crest, river side walls, . . .) which shall be preserved in the computational mesh.
   - *Holes*: parts within the mesh which are excluded from modelling (e.g. buildings). These parts are defined by special points (layer region_points) surrounded by breaklines.
   - *Vertices*: enforced geometric points in the mesh (e.g. measurement points).

2. Generation of triangles respecting specific quality criteria using the free program TRIANGLE. Parameters of major importance are:

   - *Maximum area* constraints: definition of the mesh density using *maximum area* contraints for the triangular mesh elements. The *maximum area* is defined as attribut in the layer region_points and holds for a specific *Region* surrounded by breaklines.
   - *Dividing constraints*: With this attribut in the layer *Breaklines* one can enforce a certain number of mesh elements along a breakline. This is of major importance for the use of inner boundaries in BASEMENT, where an equal number of mesh elements at the uptream and downstream interface is required.
   - *Minimum triangle angle*: no elements with angles smaller than the minimum angle specified are generated (smaller angles lead to less elements, while larger angles lead to more elements).
   - *Relative snapping tolerance*: defines, how far two point coordinates may be located apart to still be considered at the same location. The default value is 10E-6. Increasing this tolerance can help to avoid problems due to improper snapping of vertices (line or polygon features) and points in OGIS.

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

3.3.5.4 Interpolation of Elevation Data

An important task in mesh generation is the interpolation of elevation data onto the nodes of the quality mesh. As a result, the final computational mesh is obtained, which then can
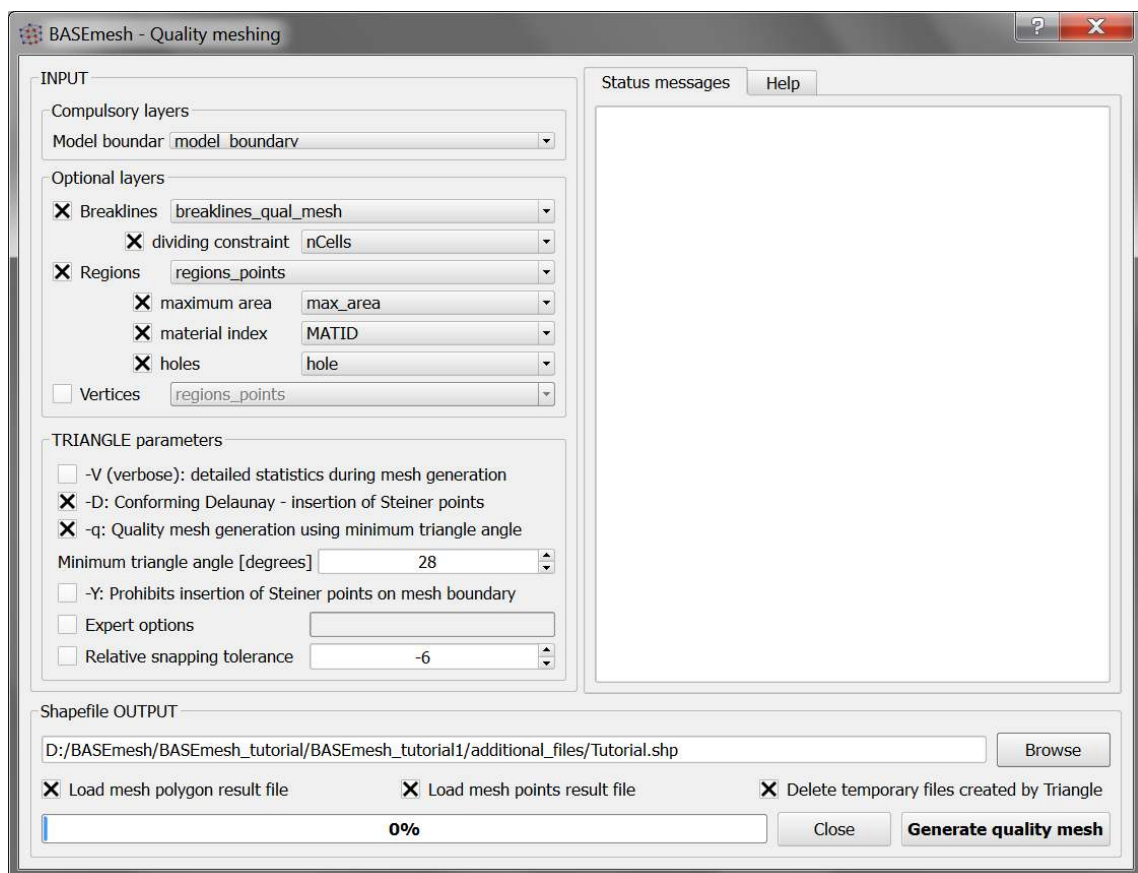
Figure 3.13 *User interface for the creation of quality meshes.*
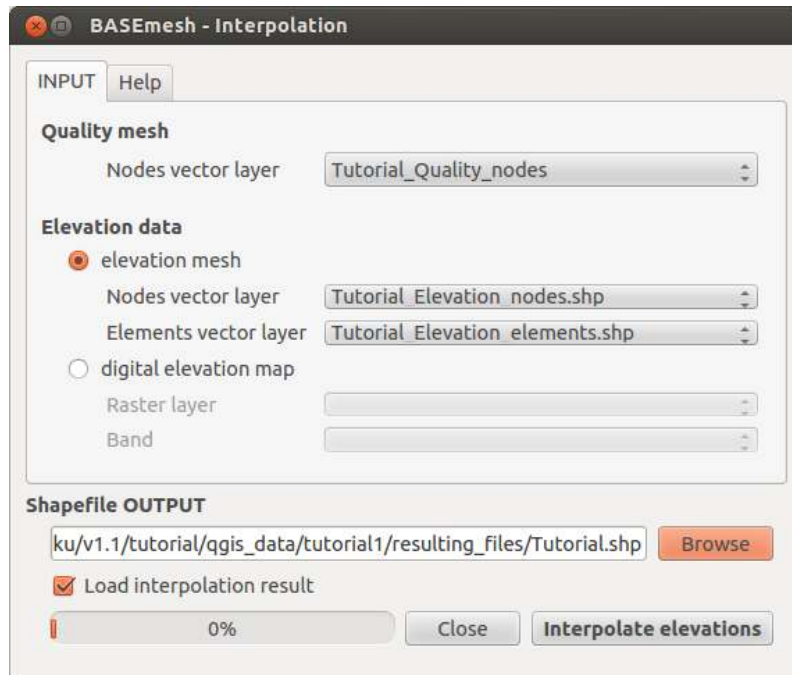
Figure 3.14 *User interface for the interpolation of topographical data onto quality meshes.*

be exported and used for simulations. The elevation data serving as input can be provided in two different elevation model types:

1. *Elevation mesh* triangulated from pointwise elevation data (TIN). The routine identifies the coordinates of each quality mesh node and determines any underlying elevation mesh element. Two methods are used for data interpolation:

   a) If an underlying elevation mesh element is found, the elevation of the quality mesh node is interpolated at its x-y-coordinates. This is the normal case, since the elevation mesh usually covers the whole computational domain. Nodes interpolated with this method are marked by a 1 in the element - field of the node attribute table. If the quality mesh node is located at the exact coordinates of an elevation mesh node, its height value is preserved exactly.

   b) If no underlying elevation mesh element is found, the quality node elevation is set to that of the nearest node of the elevation mesh. This is the case if quality mesh nodes lie outside the domain covered by the elevation mesh or when holes are present in the elevation mesh. It may lead to incorrect quality mesh ndoe elevations. Hence, it is recommended to choose a bigger domain for elevation meshing than for quality meshing. Nodes interpolated this way are marked by a 0 in the element - field of the result attribute table and are named 'with special treatment' in the QGIS status messages.

2. *Digital elevation map* as raster data which contains the topography as DTM. The raster elevation data is directly mapped on the computational mesh nodes without interpolation. If no corresponding raster cell is found, the elevation is set to '-999'.
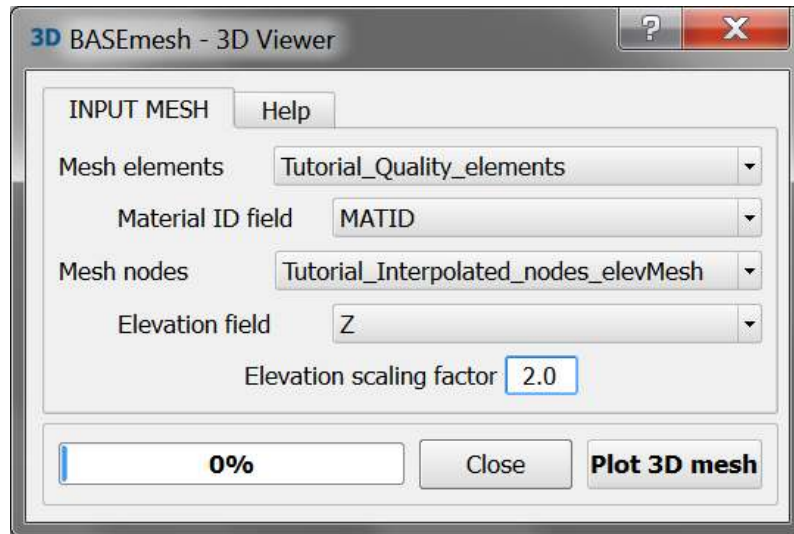
Figure 3.15 *User interface for 3D view.*

3.3.5.5   Visualizing meshes in 3D

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

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

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

- zooming in and out: scroll using the wheel of your mouse

- shifting: drag and drop using the wheel button of your mouse

- rotating: drag and drop using left-click of your mouse (the orbit centre is located in the centre of your mesh)

3.3.5.6   Renumbering Meshes

An unstructered mesh generally consists of a list of nodes, a list of elements and the mesh connectivity logic behind. The elements are formed by three or four nodes and dispose of a certain orientation (here: clockwise orientation). If a mesh is generated via BASEmesh using *quality meshing* or *elevation meshing*, then the mesh connectivity is always in a valid state. However, if the mesh is modified afterwards, e.g. by deleting or adding elements/nodes or swapping edges, then the mesh connectivity becomes invalid. Therefore, after modifications, the mesh connectivity has to be updated. This task is performed using the *Renumber Mesh* tool:
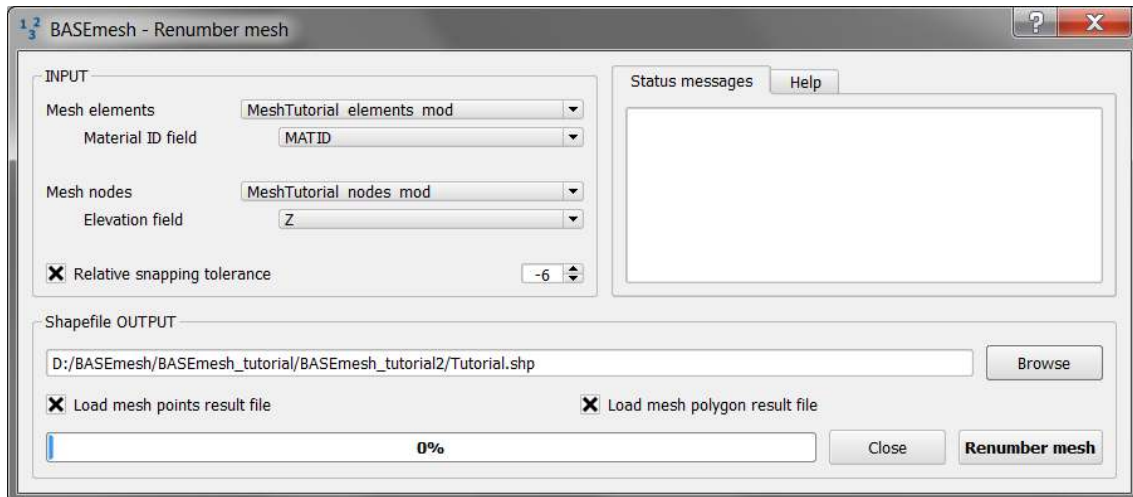
Figure 3.16 *User interface for renumbering meshes.*

1. The mesh connectivity is checked and updated. For modified or added elements, the corresponding node ID-numbers are searched by comparisons of x-y coordinates. Furthermore, the clockwise element orientation is established by re-ordering the element nodes if necessary.

2. Nodes and elements are renumbered in a way that their ID-numbers begin ascending from '1' without gaps.

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

3.3.5.7 Finding Node IDs for *stringdef*

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

1. For each line feature with a non-empty stringdef-field, all nodes that are located exactly on that line, are listed in a text file (BASEMENT-like stringdef block). The content of the stringdef-field represents the stringdef name.

2. The stringdef line features are favorably included in the layer where the breaklines of the quality mesh are defined. To distinguish between regular breaklines and stringdef lines, the stringdef-field can be used (empty vs. non-empty).

3. The upstream direction of the generated stringdefs is *right*. Therefore the line feature has to be drawn from the left riverbank to the right riverbank.
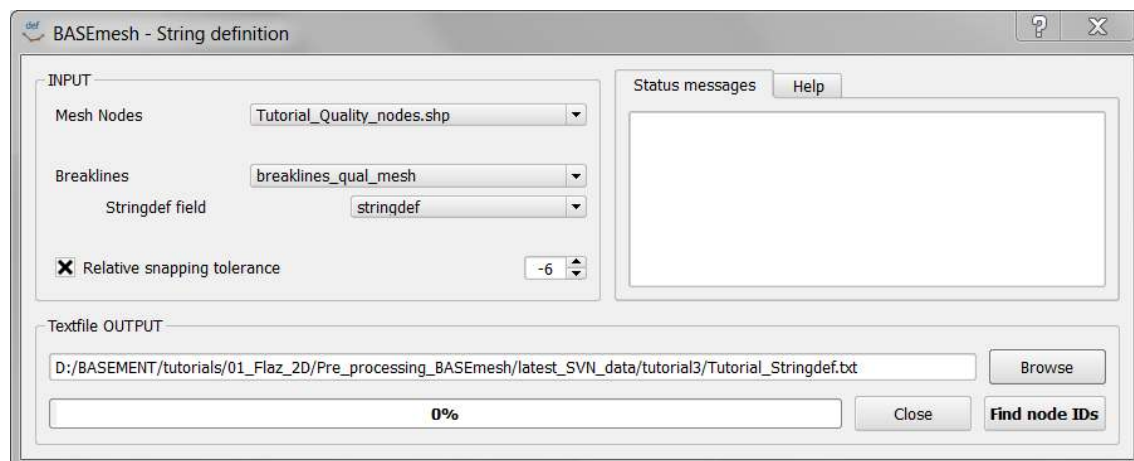
Figure 3.17 *User interface for Stringdef.*

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

3.3.5.8   Import and Export of mesh - files

2D mesh files:

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

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

1D mesh files:

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

- BASEchain geometry files (*.bmg*), and

- HEC-RAS geometry files (*.g01*).

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

Import mesh

After specification of the locations of the input and output files, the 2D / 1D mesh-file can be imported into QGIS. The mesh data are stored in shape-format within QGIS.
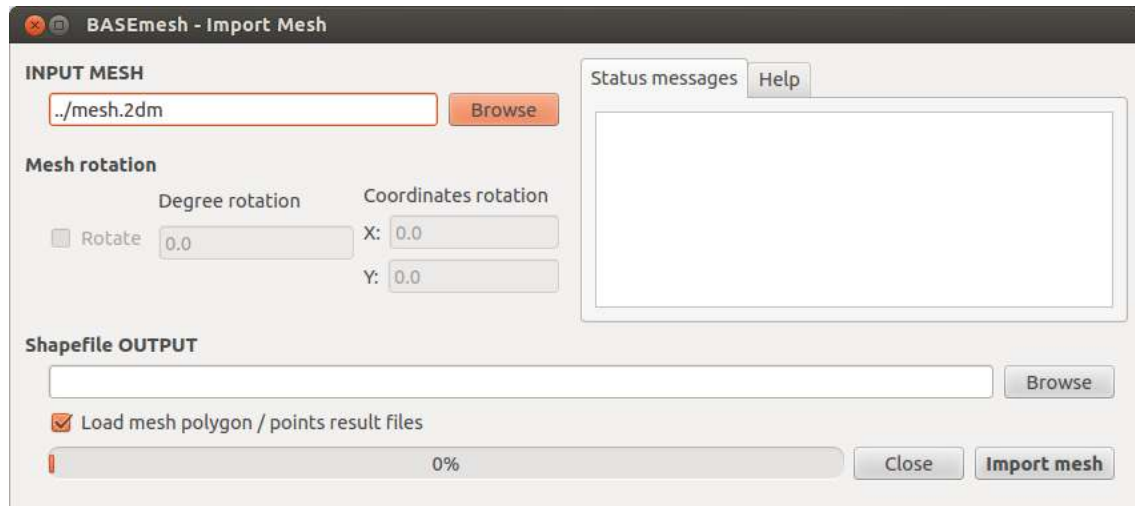
Figure 3.18 *Importing an existing mesh from a .2dm-file.*

- 2D: Two shapefiles are generated with (a) element data (suffix: '_elements.shp') and (b) node data ('_nodes.shp'), containing the mesh connectivity information and material indices.

- 1D: Two shapefiles are generate with (a) cross section data (suffix: '_crossSections.shp') and (b) point data ('_points.shp'), containing the cross section data and the cross section points.

Export mesh

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

- 1D: Export into the 1D mesh-formats requires the specification of the file format and the cross section shapefile.
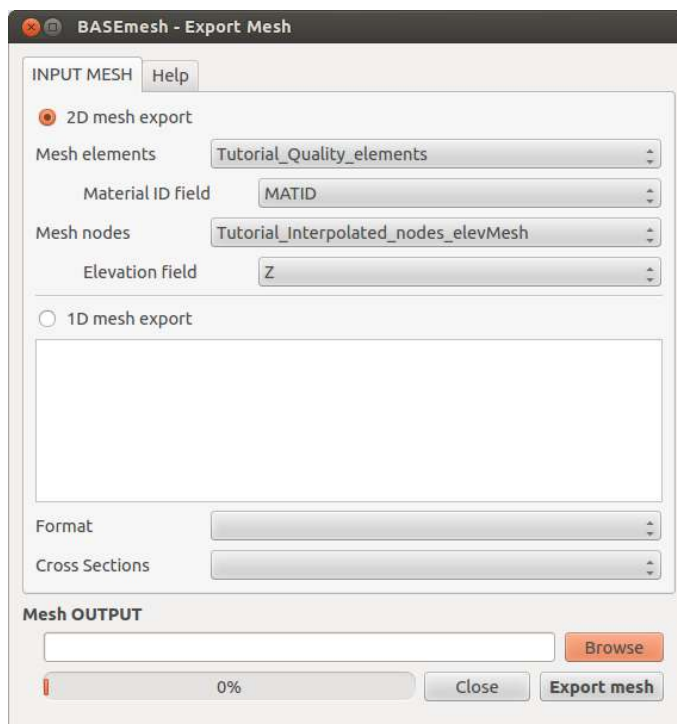
Figure 3.19 *Exporting a mesh from QGIS to file.*

3.3.5.9   Visualization of Simulation Results using QGIS (Post-Processing)

3.3.5.9.1   Visualization with the Crayfish plugin

Fast and easy visualizations of time dependent BASEMENT simulation results can be achieved using the Crayfish plugin developed by Lutra Consulting. The plugin is based on a fast engine which can deal with large and time dependent data sets in an efficient way. The plugin allows for plotting simulations results as adjustable color maps and offers additional visualization options like velocity vectors. Furthermore time dependent 2D results can be exported as animation.

How to do the Post-Processing using the Crayfish plugin is explained in detail in the Post-Processing tutorial.

3.3.5.9.2   Visualization with QGIS operations

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

1. **Shapefiles:** BASEMENT has the option to write results files in the shape-format with each time step as column in the attribute table. These data files can be directly loaded into QGIS and visualized/colorized. A disadvantage of this visualization method is, however, that there is no smooth color transition between the discrete elements.

2. **Raster data:** Output data of BASEMENT (either as shapefiles or as ASCII-x-y-z data files) can be rasterized in QGIS. Hereby, the pointwise data values are interpolated on a raster. To rasterize the pointwise data values you can use the QGIS
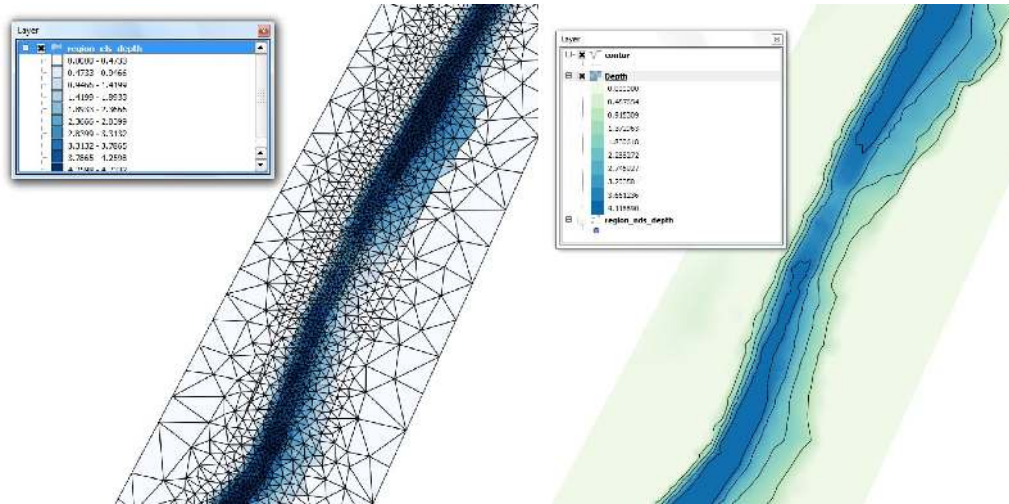
***Figure 3.20** Screenshot of data visualization using shapefiles (left) and raster data (right).*

option *Raster → Interpolation*, which offers multiple interpolation algorithms, like TIN or inverse distance weighting. To colorize the raster data, open the properties dialog and select the option singleband pseudocolor and choose your preferred color map. Raster data are generally nice to work with, since GIS-applications have many and powerful tools to cope with raster data in various ways, e.g. to create contour lines or cross section profiles.

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

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

4

Model Setup

4.1 General

Since Version 2.0, BASEMENT is not a pure console application anymore. The Graphic User Interface (GUI) enhances the usability, but it is still possible to work with BASEMENT on a console.

It is now possible to define the whole Command File using the GUI. This avoids Syntax Errors within the Command File compared to the versions where the Command File was created manually. Furthermore, the *Command File Editor* validates the current state of the Input File and indicates whether there remain some Errors or Warnings.

Grid generation for 1-D simulations has never been easier. The *1-D Grid File Editor* allows the definition of cross sections and provides a graphical visualisation of the whole river reach and also every single cross section. There are many additional tools to work on cross sections which facilitate the setup of a 1-D geometry.

The intention of this document is to help the user working with the Graphic User Interface. The common basic behaviour is explained in Section 4.2. The Use of the Command File Editor is described in Section 4.3 and the Explanation of the 1-D Grid File Editor can be found in Section 4.4. In Section 4.5, interactive Visualisation and Manipulation tools during the runtime of a simulation are shown.

In some situations it may be desired to run the program without GUI and without user interaction, e.g. to execute several program runs via batch script over night or when running the program on a high-performance machine. Further details concerning the batch execution mode are explained in the Section 1.1.
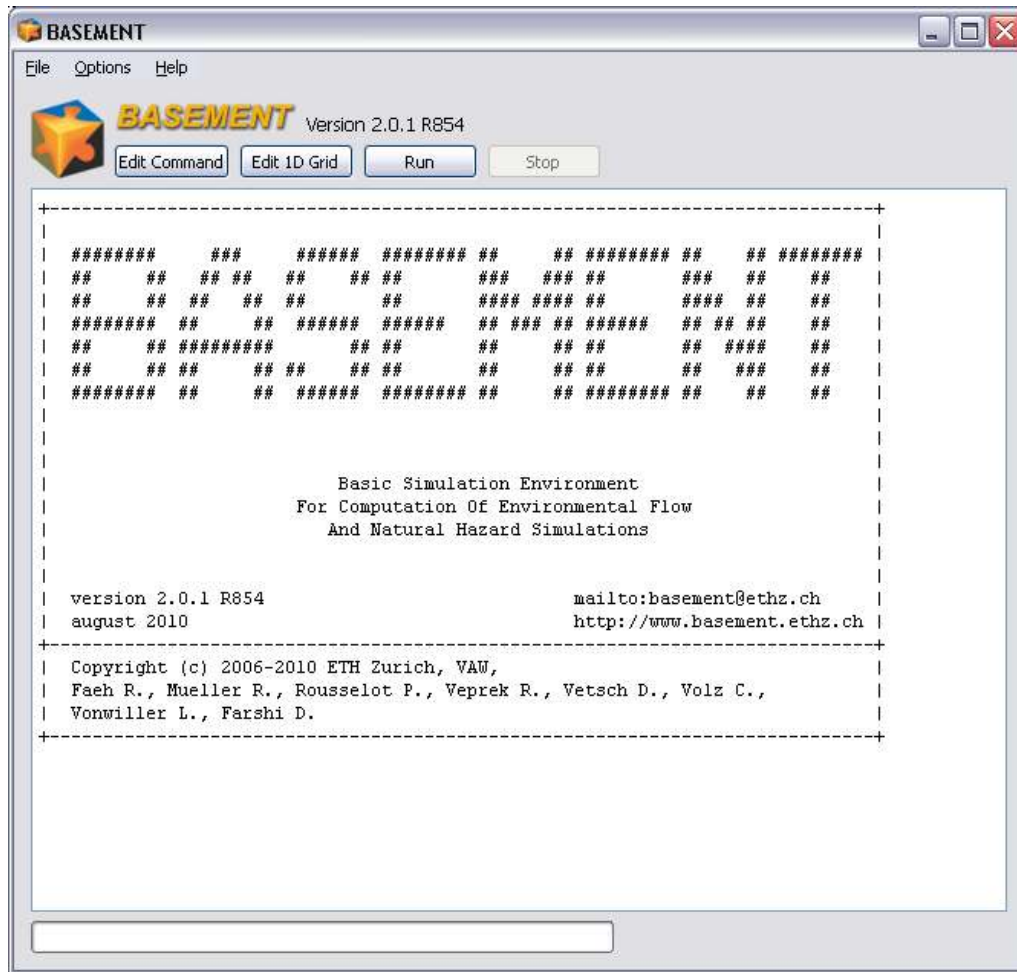
Figure 4.1 *BASEMENT Main Window with menu bar, action buttons, splash screen and a progress bar.*

4.2 The Graphical User Interface (GUI)

4.2.1 The BASEMENT Main Window

Starting BASEMENT will first produce a Graphical User Interface as shown in Figure 4.1. On the top of the screen, a menu bar consisting of the menus *File*, *Options* and *Help* is available. Below the menu bar, near the BASEMENT logo, there are four action buttons named *Edit Command*, *Edit* 1D *Grid*, *Run* and *Stop*. The main window displays a splash screen indicating the current version of the Software. A yet empty progress bar is located below the main window.

The menu File allows for opening either an existing command file (Open Command) or an existing 1 D Grid (Open 1D Grid) within the BASEMENT Editor. Selecting Quit from the File menu will exit the Application.

The menu Options hosts the setting for the Log level. Different Log levels are available:

- Log level 4 will show all possible Log output,
- Log level 3 will suppress debug messages,

- Log level 2 will print warnings and errors only,

- Log level 1 will show error messages only.

The Log output is divided into two tabs, one to show general Log information and debug messages, another one to print warnings and errors. In case of an error the simulations stops and BASEMENT exits. The error messages can then be found in the error-Log file.

The splash screen can be shown anytime again by selecting About BASEMENT from the *Help* menu.

The action buttons have the following behaviour:

- *Edit Command* opens the currently active Command File within the Command File Editor. If no Command File has been opened yet, the Editor will be empty. A description of the Command File Editor is given in Section 4.3.

- *Edit 1D Grid* opens the currently active 1D Grid definition within the 1-D Grid File Editor. If no 1-D Grid has been opened yet, the Editor will be empty. A description of the 1-D Grid File Editor is given in Section 4.4.

- *Run* is only active when a Command File is active in the background. Pressing this button will start a simulation according to the settings in the Command File.

- *Stop* is only active if a simulation is currently being carried out. Pressing this button will stop the current simulation.

4.2.2   General Topics of Editing Files

4.2.2.1   BASEMENT Editor

Both, the Command File Editor and the 1-D Grid File Editor share a common composition. The Window is subdivided into three parts: the Input Structure to the left, the Main View to the right and the Validation Messages at the lower right corner (Figure 4.2).

The Input Structure shows the tree of all Blocks defined and allows the selection of a specific Block. The Main View shows all relevant Information for the currently selected Block like available Sub-Blocks/Tags and the values for all chosen Tags. Graphical visualization of e.g. a Cross-section or a time series can also be seen.

The Validation Messages indicate by their Color whether the Input for the selected Block contains any Errors or Warnings.

4.2.2.2   Edit Blocks and Tags

4.2.2.2.1   Add Blocks

Select a Block from the Input Structure. Use the drop down menu near the *Add Block* button in the Main View to see all available Blocks. Select one of them and press the *Add Block* button to add a new Block.
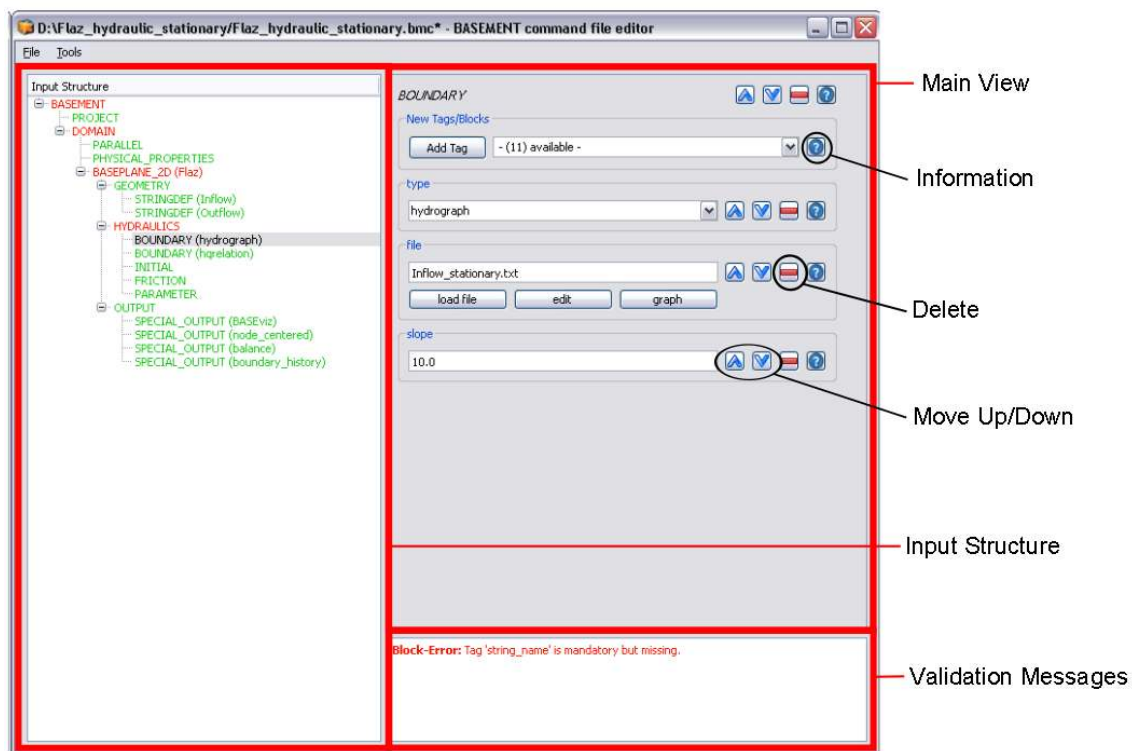
Figure 4.2 *Main Structure the BASEMENT Command File Editor.*

4.2.2.2.2   Define Tags

Select a Block from the Input Structure. Use the drop down menu near the *Add Tag* button in the main view to see all available Tags. Select one of them and press the *Add Tag* button to add a new tag. The Tag will appear in the main view and needs to be given a precise value. Depending on the Tag, a value can be chosen from a drop down menu or has to be defined by the user.

4.2.2.2.3   Remove Blocks or Tags

To remove a Block or a Tag, use the Delete button as indicated in Figure 4.2. For a block, this is located at the top of the main view near the Blocks name. For a Tag, the button is to the right of the precise value of a Tag. There will be a confirmation message whether a Block or Tag really shall be removed.

4.2.2.3   Automatic Validation

Validation Messages indicate immediately whether the current choice of Blocks and Tags is valid (Figure 4.2). In case of errors, the Block and its parent Blocks in the Input Structure are red. In case of warnings, the Block and its parent Blocks in the Input Structure are brown. Blocks without warnings or errors are green.

Selecting a red or brown Block in the Input Structure will show all errors and warnings as Validation Messages.

An Input without red messages has no errors and can be used for simulation.

4.2.2.3.1 Mandatory Blocks and Tags

Certain Blocks and Tags may be mandatory to define. If a mandatory Block or Tag is missing, an error message like *Block Error: Tag "XY" is mandatory but missing* is displayed until the Block or Tag is defined.

4.2.2.3.2 Default Values

Some Tags have default values. They are usually not mandatory to define. Defining a Tag with an associated default value will show the specific Tag value in the Input Field for the Tag where it can also be changed to a user defined value.

4.2.2.4 Use the Built-in Help Function

Every Block and Tag is documented within the Reference Manual R IV. This same information is also available within the BASEMENT Editor. Clicking on the Information button with a question mark (Figure 4.2) will produce a pop up window with a description of the usage and some examples. The Information button for a block is located at the top of the main view near the Blocks name. For a Tag, the button is to the right of the precise value of a Tag. The documentation for Tags also includes information about whether it is mandatory, default values, value range, etc.

4.3 Edit Command File

4.3.1 The "BASEMENT Command File Editor"

From the BASEMENT main window (Figure 4.1) the user gets via the *Edit Command* button to the *Command File Editor* which is shown in Figure 4.3. The Input Structure (as it is built up with Blocks) is on the left-hand side. The Main View on the right-hand side gives more details of the selected Block with the corresponding Tags and the possibility to add Tags and/or Blocks (see Section 4.2).
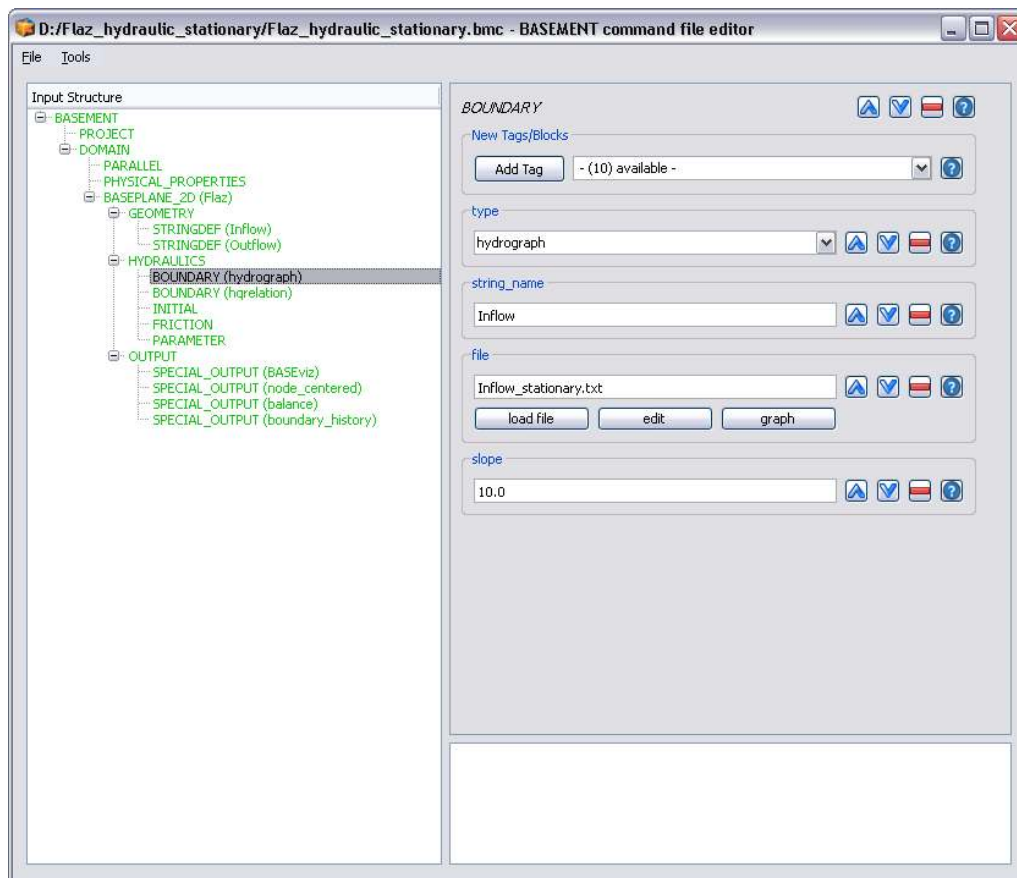
Figure 4.3 *BASEMENT Command File Editor*

4.3.2 Create New or Edit Existing Command File

A new Command File is created form the menu bar in the *Command File Editor*. Choose *File* on the menu bar and select *New*. The whole command file consisting of Blocks and Tags can be built up from scratch. For further details on how to add Blocks and define Tags see Section 4.2.2.2 in this manual.

If you want to open an existing command file you can do this from the *Command File Editor* or directly from the BASEMENT *Main Window*. From the *Command File Editor* choose *File* on the menu bar and then select *Open*. Then you browse and choose your file as usually. Don't forget to save your changes made in the command file before you run a simulation.

4.3.3 Tools

4.3.3.1 Edit Raw

Another way to edit a command file is to edit basically the command file in a raw text mode. For this purpose choose *Tools* on the menu bar and select *Edit Raw*. The Editor window will pop up as shown in Figure 4.4. Long-standing users of BASEMENT will recognize the input structure of the former command file. In the lower part of the window the Input is validated and possible parse errors are indicated. For the sake of completeness it is mentioned here that the command file can still be built up and edited with a simple text editor.
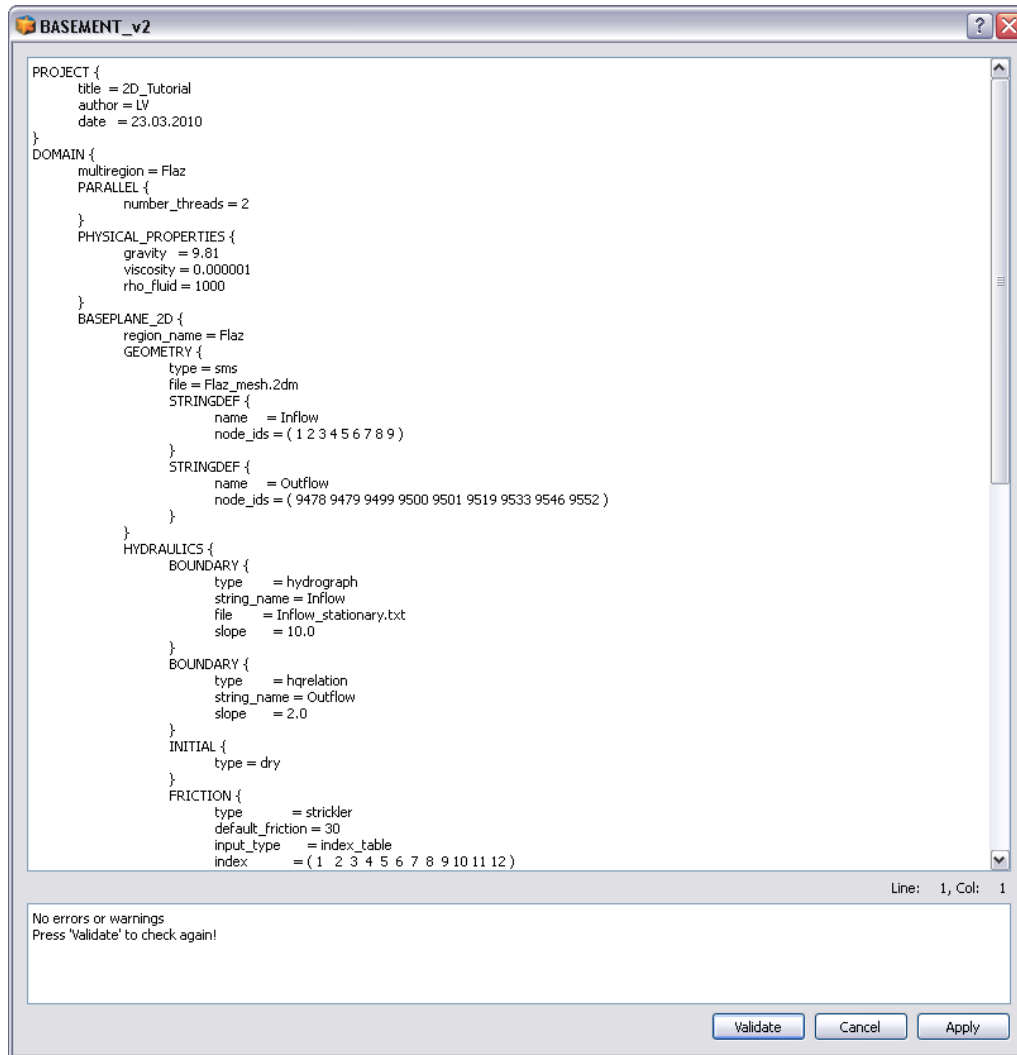
Figure 4.4 *Command File Editor: Raw Edit Window.*
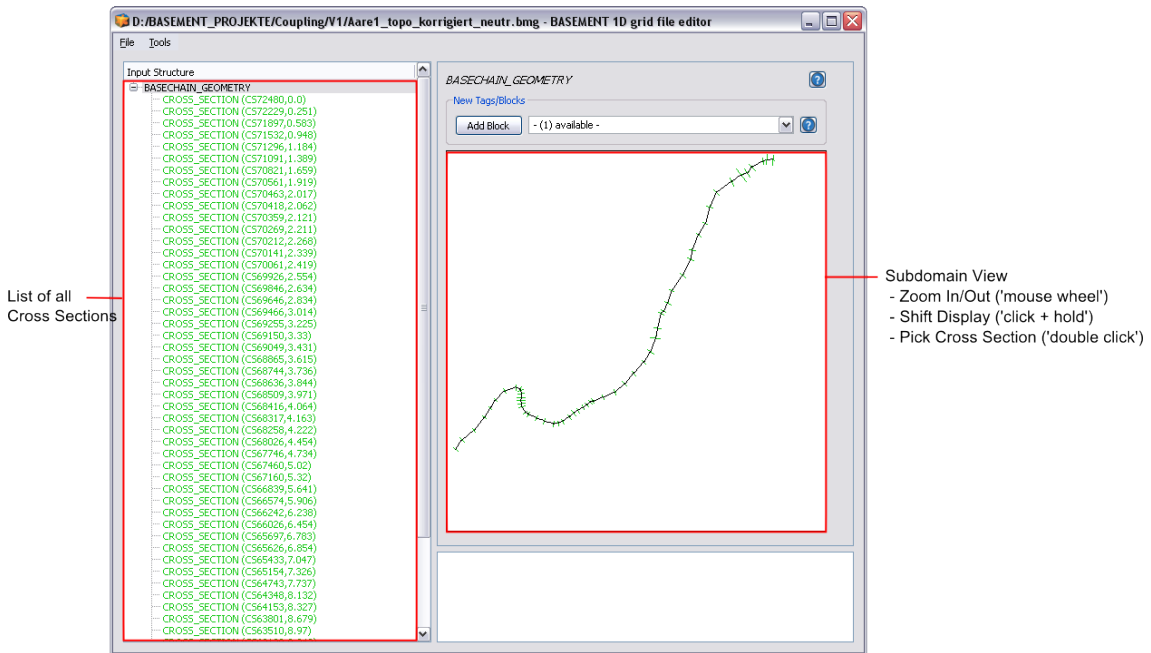
Figure 4.5 *Grid File Editor – Subdomain View*

4.3.3.2   Expand All

The command Expand All (Ctrl+E) opens the whole input structure on the left-hand-side of the Command File Editor. It provides a faster access to a single block or tag.

4.4   Edit 1-D Grid

4.4.1   The "BASEMENT 1-D Grid File Editor"

When the Grid File Editor is opened a new window pops up (see Figure 4.5). In the left part of the window a list of all cross sections is shown where the cross sections can be seen and selected. In the right part of the window a visualization of the whole subdomain with all cross sections is drawn and new cross sections can be created with the *Add Block* option. The subdomain view allows zooming and shifting of the display and the selection of a specific cross section by double clicking. If a cross section is selected then the view changes to the cross section view.

The real (usually curved) shape of the stream can only be illustrated if all cross sections are geo-referenced and if all corresponding data is set (the *orientation_angle* and the *left_point_global_coordinates* must be set for each cross section). If these data are not given than the cross sections are drawn along a straight line.

4.4.2   Create New or Edit Existing Geometry File

If a specific cross section is selected or a new cross section is created, than a profile view of the selected cross section is shown (see Figure 4.6). With this visualization of the profile one can easily check for input errors in the geometrical definition of the cross section profile.
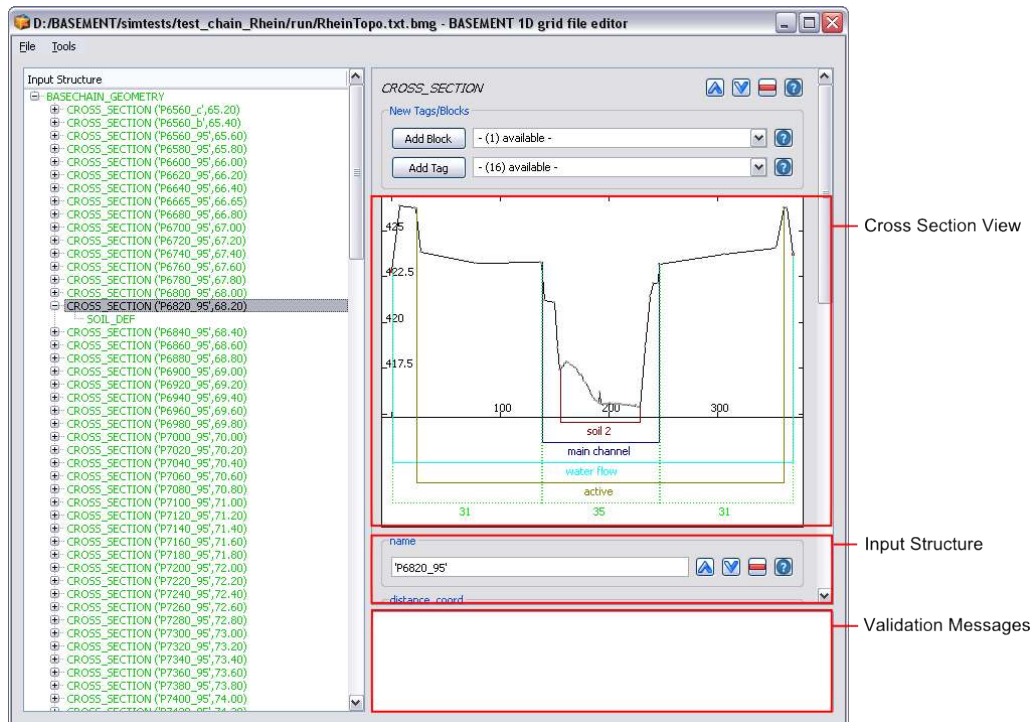
Figure 4.6 *Grid File Editor – Cross Section View*

Furthermore, the most important cross section parameters are indicated visually with different colours, like e.g. the definition of the main channel, the range of the soils, the friction parameters, the cross section fixpoints, etc. Again, one can visually check if these parameters are set up correctly and thus easily detect type errors. New input tags can be added and the validation message box shows warnings or errors if some problematic inputs have been made.

For details on how to set up a new cross section and for information about the various cross section parameters see the 1-D tutorial, hydrodynamics and sediment transport at the river Thur.

4.4.3 Tools

BASEMENT supports several tools which support the user in creating and modifying the 1D grid file. In the following sections some information about the usage and the methods of these tools are given. Please be aware that some of the offered tools are still in a beta-status.

4.4.3.1 Edit Raw

Another way to edit a grid file is to edit the grid file in raw text mode. For this purpose choose *Tools* on the menu bar and select *Edit Raw*. The Editor window will pop up as shown in Figure 4.7 and Figure 4.4. In the lower part of the window the Input is validated and possible parse errors are indicated. For the sake of completeness it is mentioned here that the grid file can still be built up and edited with a simple text editor.
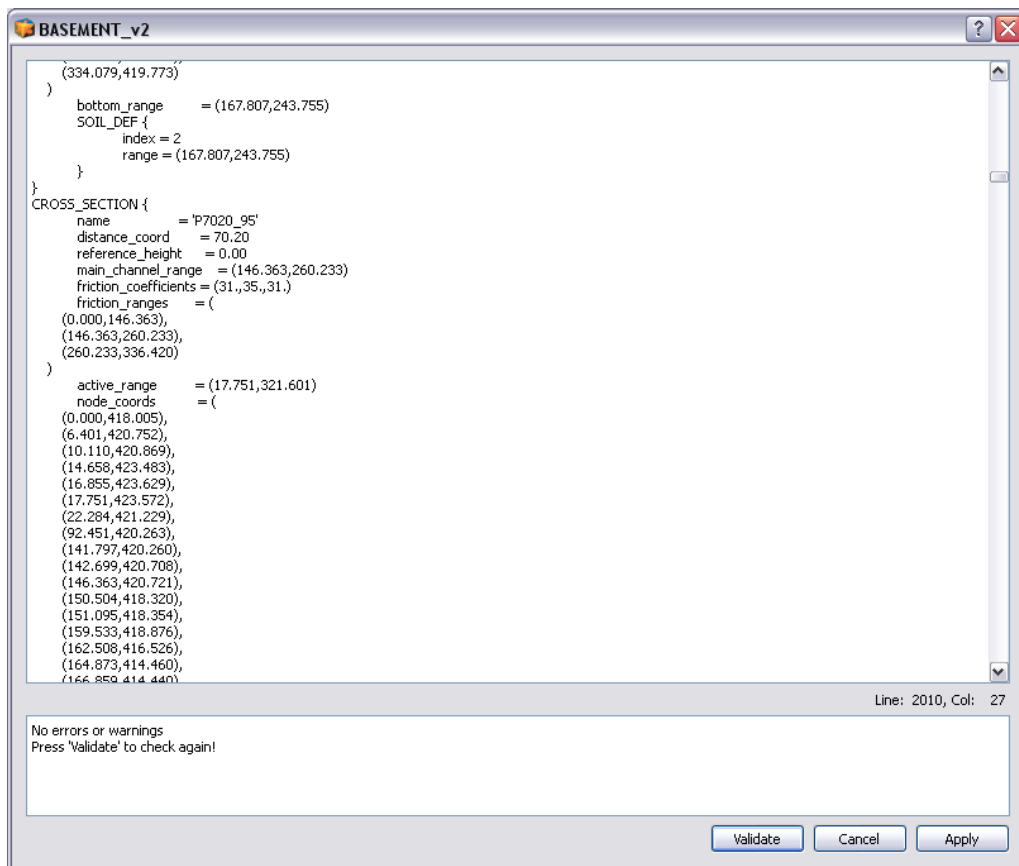
Figure 4.7 *Grid File Editor: Raw Edit Window*
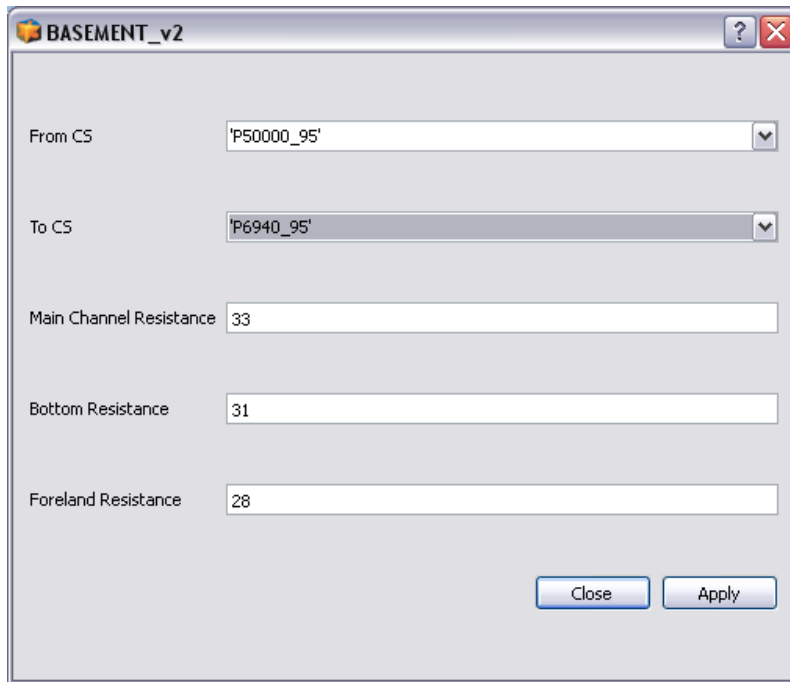
Figure 4.8 *Friction assignment*
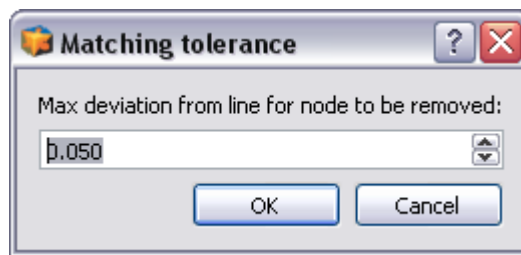
Figure 4.9 *Node removal dialog for setting up the tolerance*

4.4.3.2 Friction

With this tool the friction value of the main channel, the forelands and the bottom can be assigned to a range of consecutive cross sections. The names of the first and the last cross section of the range have to be given.

4.4.3.3 Remove Nodes

Cross sections often consist of a large number of nodes and slices which consumes significant computational performance. If multiple nodes lie on a straight line within the cross section there is redundant information present which can be removed without reducing the accuracy of the computations. Therefore identifying and removing these redundant nodes is a frequent and recommended task before running the simulations. BASEMENT offers a tool which performs this task automatically without need for costly manual operations.

To access this tool open the *Tools* menu and click on the *Remove nodes option*. The following dialog opens:

In this dialog you can define the maximum tolerance by which a node may deviate from the straight line of its two neighbours. If the node lies within this tolerance the node removal is
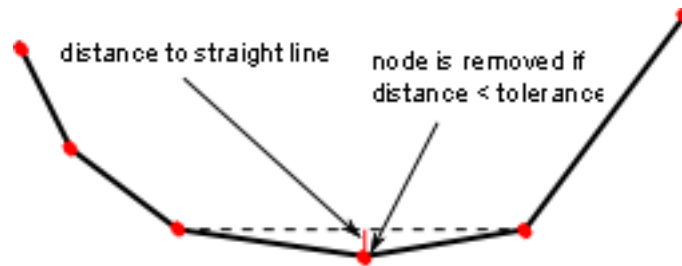
Figure 4.10 *Schematic sketch of node removal*

applied. If you set the tolerance to small values only nodes are removed which are almost exactly situated on the line. If you increase the tolerance more nodes will be removed but some more information about the cross section profile may get lost. Usually one should try different tolerances until the best compromise between computational performance and accuracy is found. Also, the algorithm can be applied multiple times.

The applied algorithm loops all nodes of the cross sections and checks if a node is situated on a straight line between its two neighboured nodes (considering the given tolerance). If this is the case the node is removed from the profile (see Figure 4.10). Nodes which are used as fixpoint or which are explicitly referenced by a slice_index range are excluded from the algorithm and cannot be removed automatically.

Guess Active Range

This tool provides a definition of the active range where it is not yet defined. For this purpose the lowest point of the cross section is searched and then the highest points to the left and the right of it (usually the dam crests) are set as limits of the active range. The definition of the active range can always be changed manually by the user in the interface. As an example on how the active range is set see Figure 4.11.

4.4.3.4 Guess Fixpoints

This tool provides the definition of some fix points, which are needed for a correct interpolation of new cross sections between existing cross sections. So this should be done before an interpolation is executed. The fix points are displayed in red. The points which are automatically set as fix points are:

- The limits of soils
- The limits of the main channel
- The limits of the active range
- The midpoint of the main channel.

It is recommended to check the points visually and add other important points, especially on the break lines. For the interpolation all involved cross sections must have the same number of fix points.

4.4.3.5 Interpolation

First of all, before an interpolation of 1-D cross sections can be performed some information is needed about the spatial alignment of all cross sections in the x-y plane.
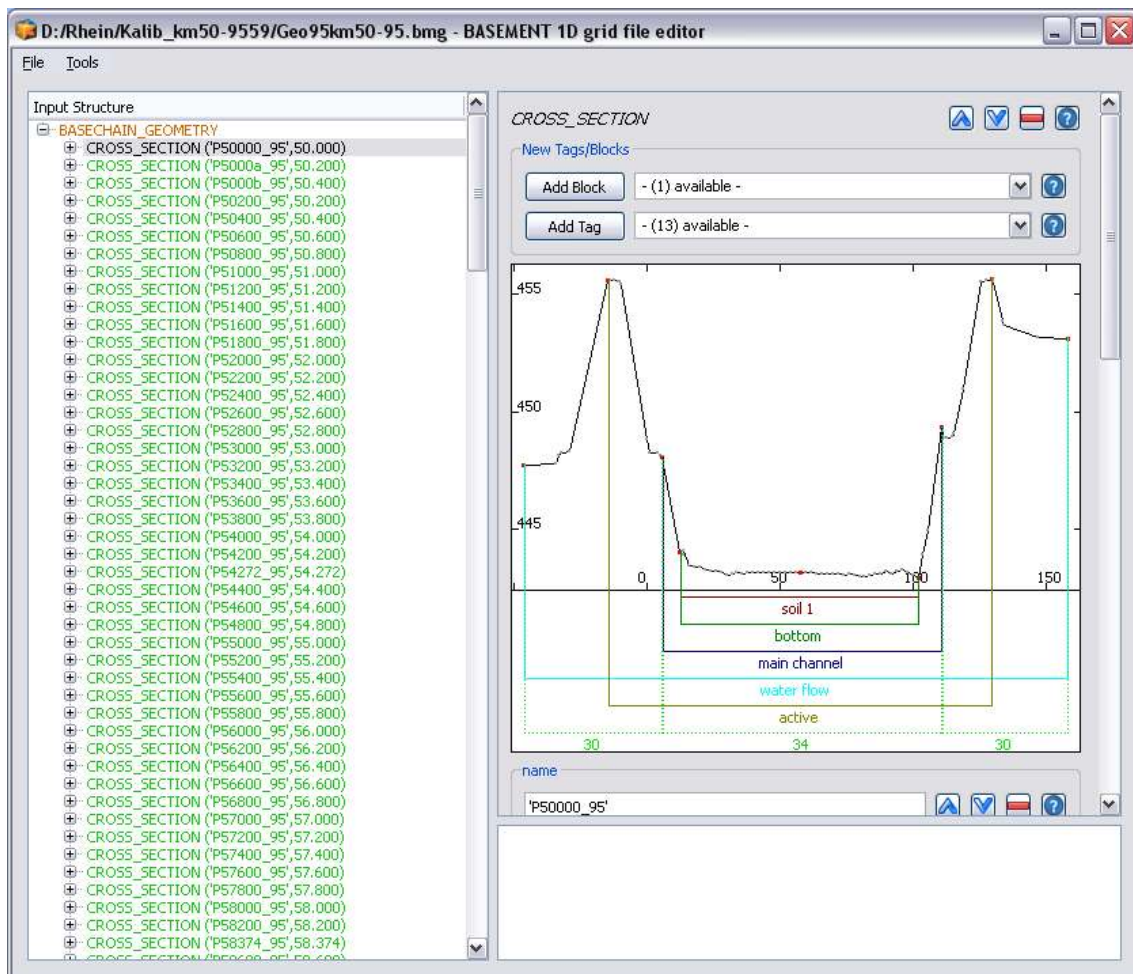
Figure 4.11 *In red the guessed fix points for cross section interpolation*
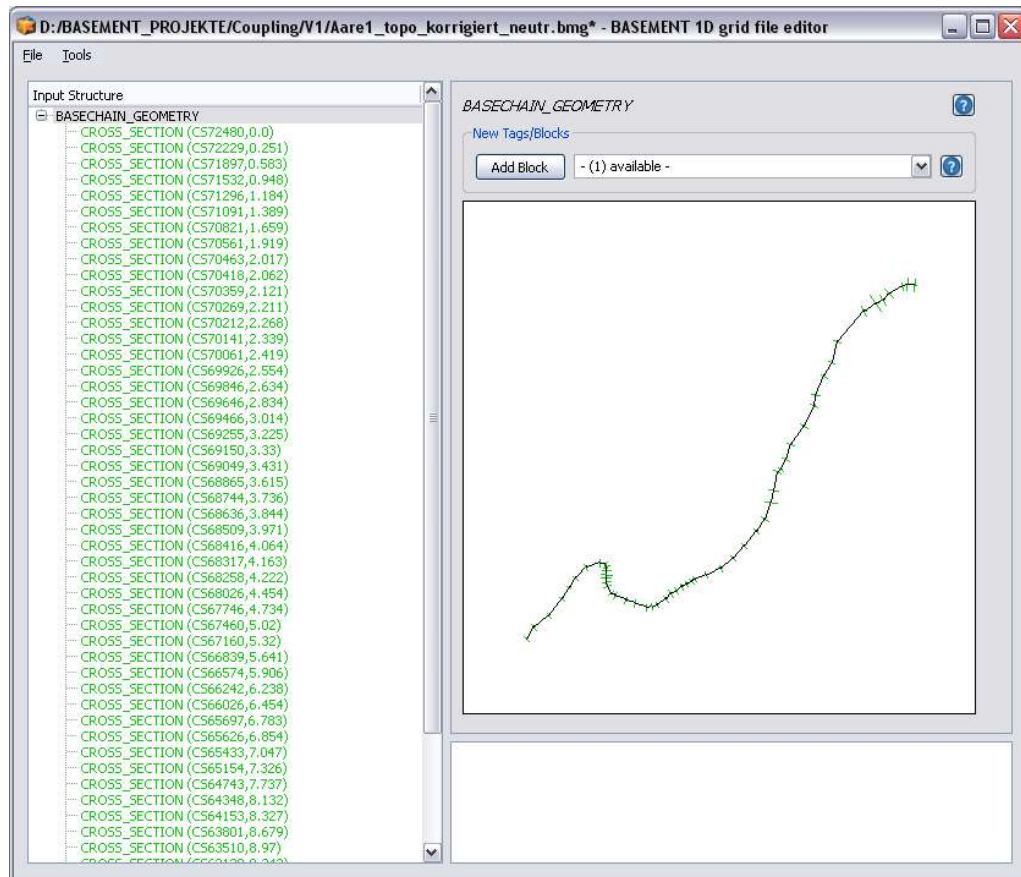
Figure 4.12 *Curved alignment of the cross sections.*

There are two main tags which determine the spatial orientation of the cross section which are crucial for the interpolation algorithm. The *orientation_angle* provides the information which is needed for the orientation of the cross sections. This is the angle between the normal vector of the cross section and the vector in x-direction (1,0). Consequently, in a fully straight channel in x-direction all cross sections would have the angle 0°. If the orientation angle is not given it is set to this value. The other essential tag is the *left_point_global_coordinates*. This parameter sets the global (real world) x,y,z-coordinates of the outer left point of the cross section. This parameter in combination with the orientation angle delivers all needed information about the spatial configuration of the cross section. If this coordinates are not given the value of *distance_coord* is used for x and the elevation of the first point on the left for z. y is set to negative distance of the first point on the left from the middle of the cross section. If both parameters are set for all cross sections, one can see the curved alignment of the stream in the right-hand visualization (see Figure 4.12).

The algorithm of the cross section interpolation is briefly sketched in the following. To grasp the meaning of the different parameters it is helpful to understand the basics of this interpolation algorithm.

This interpolation algorithm bases on the creation of spline curves (a spline is a special polynomial function which is often used for smooth interpolations between given points). For each fixpoint of the cross sections such a spline curve is determined which connects all the corresponding fixpoints with each other in a smooth way. Therefore every cross section must have the same number of fix points. Furthermore, the spline curves have the special
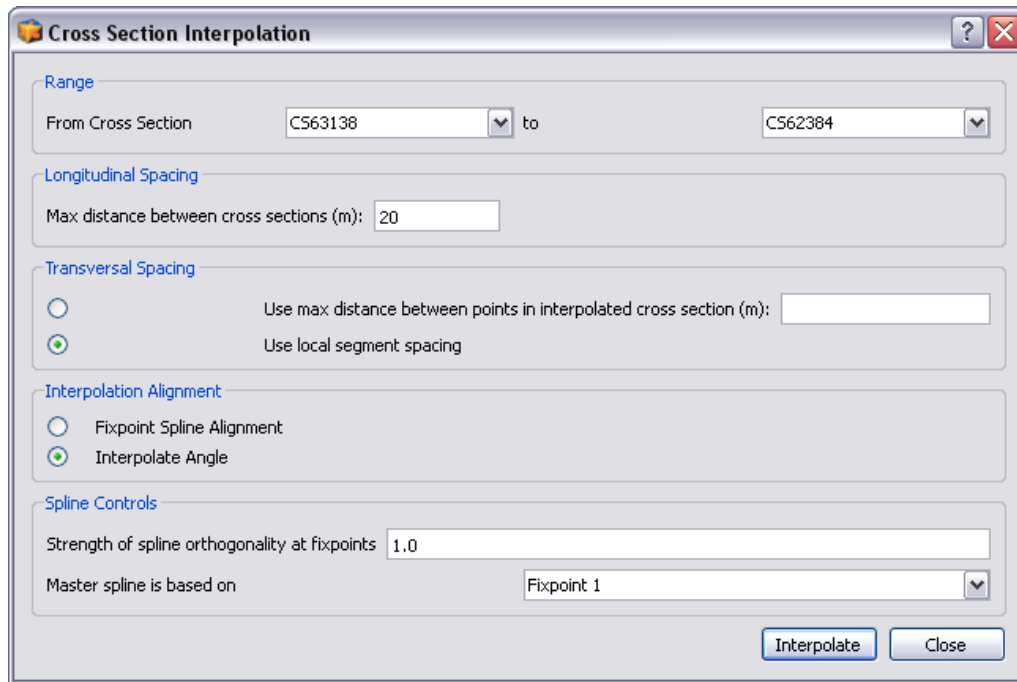
Figure 4.13 *Setup dialog for the cross section interpolation*

property that they are aligned orthogonal to each cross section profile.

After the spline curves have been calculated, the positions of the new interpolated cross sections are determined in given intervals along the spline curves. As soon as these positions are known, the cross sections are created orthogonal to the tangent direction of the master spline. To determine the fixpoints of the new cross sections the intersections of this orthogonal cross section line with all spline curves (of the other fixpoints) are calculated. Finally the new cross section points are determined in between the fixpoints in a given transversal distance interval. The elevations of the cross section points are finally determined using a weighting procedure between the elevations of the left and the right cross section.

In Figure 4.13 the setup dialog for the interpolation is shown. The different parameters are explained briefly in the following. By clicking on *Interpolate* the interpolation of the cross sections finally starts if all data is available.

First of all the *Range* of the interpolation must be defined. This is done by specifying two subsequent cross sections which are chosen from a list of all existing cross sections in the drop down menus.

Another important parameter is the *Longitudinal spacing* which determines the resolution of the interpolated grid. Enter the maximum distance between two interpolated cross sections in [m]. If you choose a small value than many cross sections in small distances will be generated, if you choose a large value only few cross sections in large distances will be generated. The optimal choice depends on the type of simulation.

Furthermore, also the *Transversal spacing* can be specified. It determines the spacing of the points in the newly generated cross section profile. There are two possible choices here two determine the transversal spacing. You can either explicitly specify the maximum distance in [m] using the first option. Alternatively, by using the second option, the distance is chosen automatically from the left and right cross sections by the interpolation algorithm

(local segment spacing).

The alignment of the new cross sections in the x-y plane can be determined with two different methods in the *Interpolation Alignment section*. The *Fixpoint Spline Alignment* means that the cross sections are always oriented orthogonal to the master spine's tangent direction. On the other hand, using the option *Interpolate Angle* the orientation of the new cross section is determined by interpolation of the orientation angles of the left and the right cross sections. This latter option is recommended in strongly curved streams in order to prevent overlapping cross sections.

Finally in the *Spline controls* section some parameters of the spline calculations can be adapted to special needs. The *Strength of spline orthogonality* parameter determines if the spline always must be completely orthogonal to the cross sections or not. In strongly curved streams some relaxation from strict orthogonality (different from 1.0) may lead to nicer shaped spline curves. Some variations and iterative testing with this parameter may improve the interpolation result in such situations. And finally also the fixpoint which determines the master spline can be chosen. The master spline thereby is the spline which determines the orientation of the interpolated cross sections.

Please note: In order to generate a 2D mesh from a given 1D mesh, this interpolation option can be very helpful in combination with the *Export DTM* option.

4.4.3.6   Export DTM for BASEplane

This tool enables the user to convert a 1-D BASECHAIN_GEOMETRY (Figure 4.14) into a digital terrain model (DTM) for further processing in SMS and BASEplane. The main application for this tool is to be found in combination with the Interpolation tool (see chapter 0): In a first step the cross sections are interpolated with the Interpolation tool in order to get a smooth river topography. In a next step the DTM is exported with the *Export DTM for BASEplane* tool (on the menu bar choose *Tools*). The generated DTM can be imported in SMS. Although the file is of .2dm type it can be easily converted into scatter points (DTM) in SMS. Then it can be used for the interpolation of the elevation information on any computational mesh.

Basically a computational mesh can be obtained directly from the *Export DTM for BASEplane* tool, if the interpolated cross sections are chosen in a close and optimal distance to each other. Nevertheless it is suggested to generate the mesh properly in SMS and to consider the generated DTM just as a terrain model from which to get the elevation information.
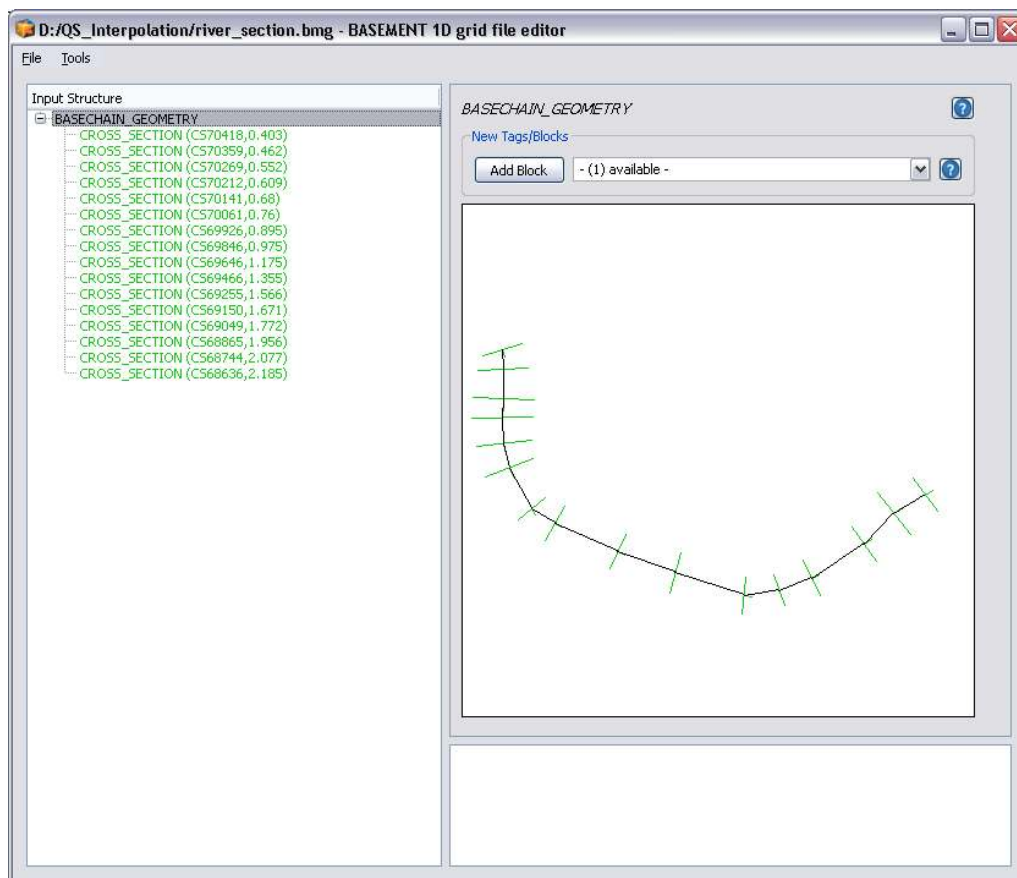
Figure 4.14 *GUI of the BASEMENT 1D grid file editor. The 1-D BASECHAIN_GEOMETRY can be exported with Export DTM for BASEplane under the menu bar Tools.*
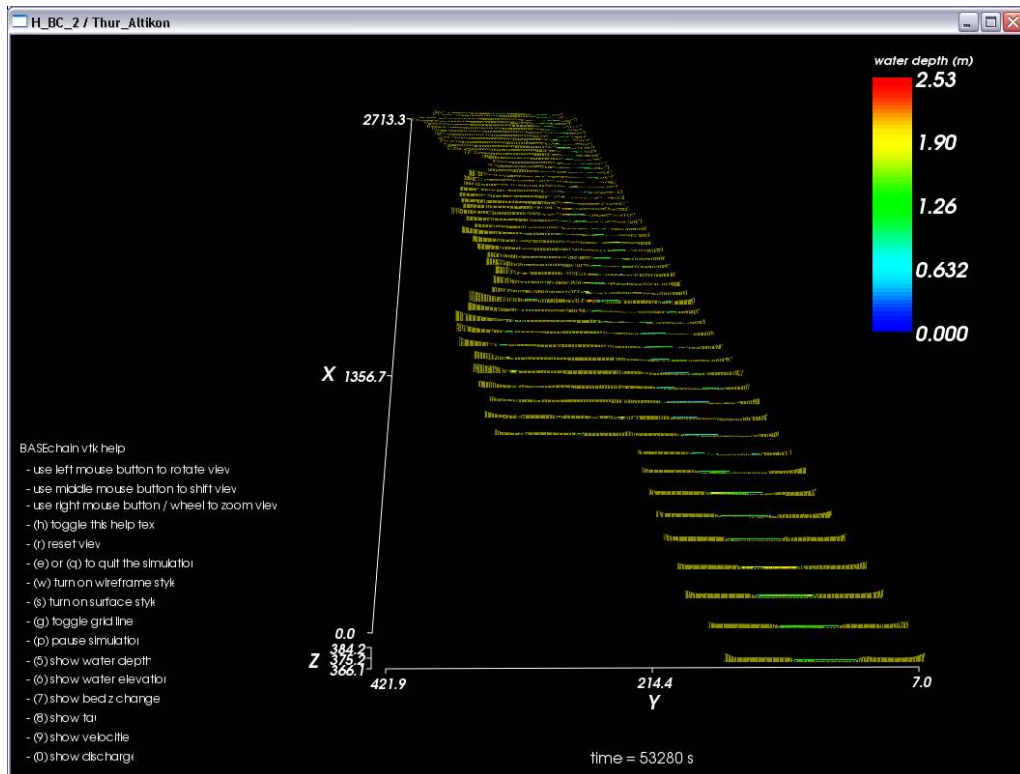
Figure 4.15 *Visualization of BASEchain with BASEviz*

4.5   Built-In GUI Tools

In this section some built-in BUI tools are explained and information about the usage is provided. Built-in GUI Tools will pop up if a certain tag is activated by the user.

4.5.1   Interactive Visualization during run time using BASEviz

BASEviz is a small and lightweight visualization tool which can be used to visualize simulation results during run time. To activate the visualization tool, a *SPECIAL_OUTPUT* block of 'BASEviz' type must be created within the parent OUTPUT block. Then the BASEviz window will appear automatically by starting the simulation. The output can be visualized interactively using the mouse and keyboard keys according to the legend shown in the BASEviz window (see Figure 4.15 and Figure 4.16). The view can be changed and the displayed variables can be selected. This visualization tool allows to easily check for a correct simulation setup and to stop a simulation run if some evident problems arise. Furthermore, it is possible to dump the rendered images from the visualization window in a JPEG image for a given time interval.

- BASEviz for 1-D simulations with BASEchain:

  All 1-D cross sections with its multiple slices are plotted one after the other along the x-axis. The water elevation is plotted within each cross section slice according to its present value.
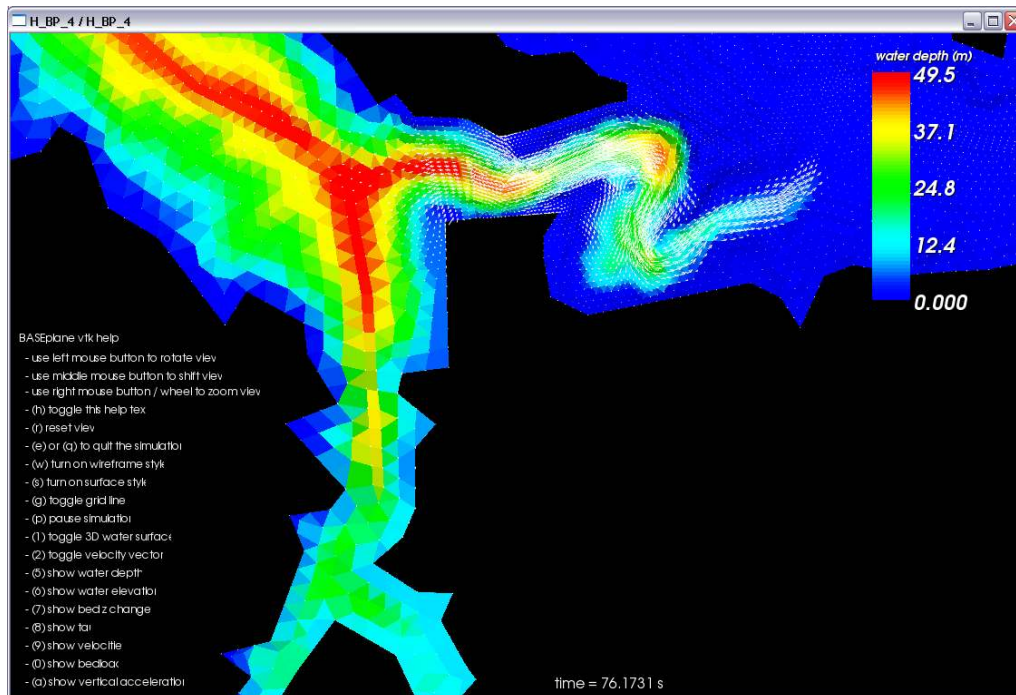
Figure 4.16 *Visualization of BASEplain with BASEviz*

- BASEviz for 2-D simulations with BASEplane:

  The unstructured 2-D mesh is plotted in combination with a contour plot of a chosen output flow variable. Optionally, velocity vectors can be added to the data visualization.

BASEviz is based on the visualization libraries of the Visualization ToolKit (VTK, http://www.vtk.org) which makes use of OpenGL for rendering.

4.5.2 Manual Controller Interface (HID)

In order to create a controller, a new CONTROLLER block is generated in the DOMAIN block. The HID controller provides an interface for the manual operation (Figure 4.17). The control window will pop up automatically after starting the simulation with the start button. In the CONTROLLER block several manipulated variables and controlled variables can be defined. The manipulated variables will appear on the left hand side of the controller interface, whereas the monitored variables will show up on the right hand side (Figure 4.17). In this example two manipulated variables (height of a weir and height of a gate) and two monitored variables (discharge over the weir and through the gate) are selected. With the cursor the slide can be moved within the predefined range of the manipulated variable. Additionally the target value can be entered directly into the white text box (Target). As the simulation proceeds the impact on the monitored variable is visualized on the chart on the right hand side. Additionally the output and the impact of your control measures can be visualised with BASEviz (Section 4.5.1).
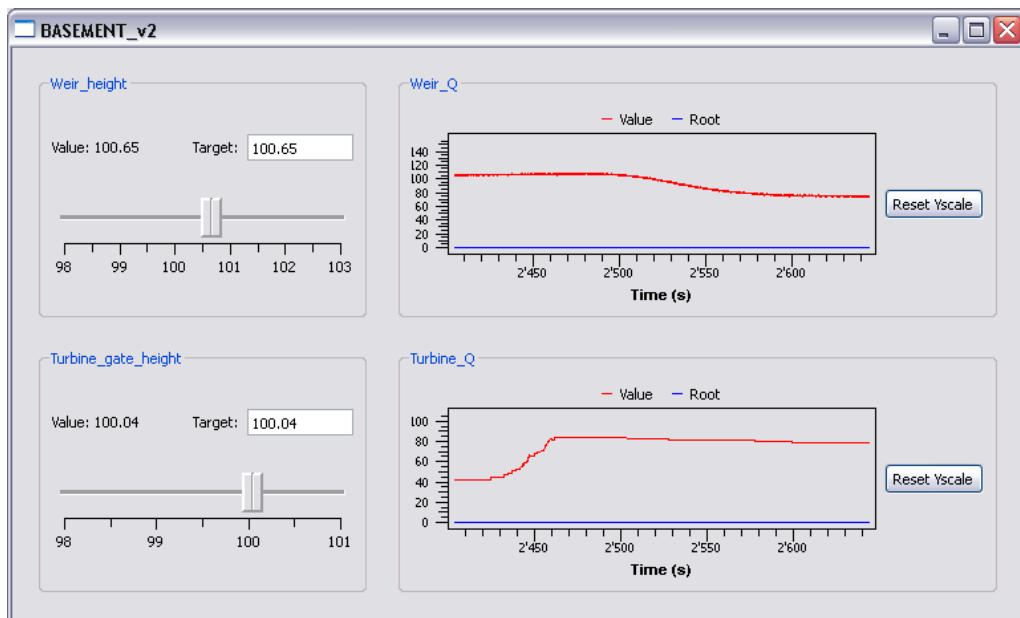
Figure 4.17 *Interface for the manual control and monitoring of the selected variables.*

5

Advanced features

5.1 Parallelization

5.1.1 Overview

5.1.1.1 Needs for parallel computing

Parallel computing is currently getting increasingly important for numeric scientific and engineering applications and this trend will become even more significant in the near future. Where parallel computers in the past were mainly limited affordable and manageable by large research institutions, today nearly all new available computers provide capacities for parallel computing. The famous law of Moore, which predicts an increase in computer power about a factor 2 every two years, is expected to be valid also in the near future due to parallel computing (Manferdelli et al., 2008).

In parallel computing the increase of execution speed is mainly achieved by sharing the overall work load between several processors instead of further accelerating single processors by an increase of tact rates. These changes in computer hardware architectures raise the needs for new parallel programming concepts. Programs originally developed and optimized for execution on a single processor cannot automatically benefit from the availability of multiple processors. Specific software techniques and algorithms must be developed and applied to exploit the additional performance provided by parallel hardware. The software BASEMENT therefore needed to be adapted and optimized for the use of parallel computers.

Some typical hydrodynamic simulation scenarios with high demands on performance are

- multidimensional flow simulations with large computational domains consisting of a very large number of elements, as well as simulations of large river networks,
- simulations of river flow over long time periods (especially with regard to morphological changes) and
- real-time simulations of river flow and flood wave propagation.
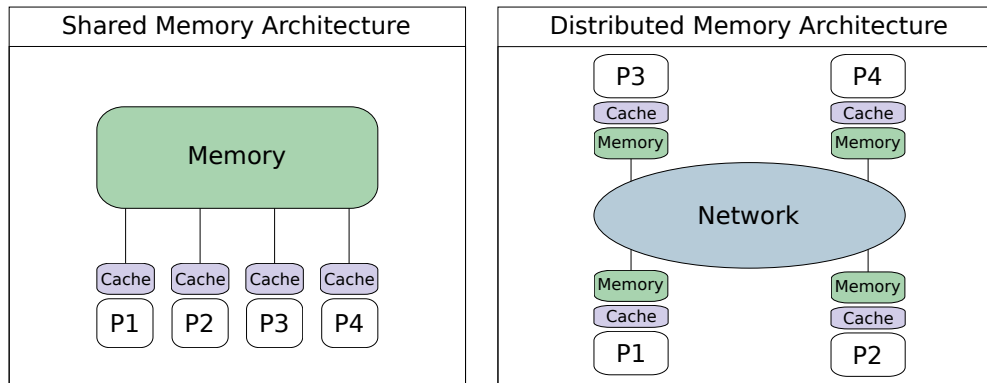
Figure 5.1 *Shared memory architectures (left) and distributed memory architectures (right) with four cores*

5.1.1.2 Parallel computer architectures

Types of parallel computing can be differentiated in many aspects, concerning hardware, software concepts and various other criteria. Frequently, parallel computers are classified in two groups, which differ in the way memory is organized between the multiple processors.

In so called "distributed memory" architectures each core possesses its own memory unit, which cannot be accessed by any of the other cores. Distributed memory systems usually consist of large computer clusters which are connected via a network (right side of Figure 5.1). Such a parallelization concept enables the use of very large numbers of processors and is employed in high performance computing (HPC). But software programming, maintenance and debugging for distributed memory systems generally is very time-consuming and costly. Complex message passing interfaces via network are necessary for data exchange and synchronizations between the processors. Out of these reasons distributed memory systems, or combinations of distributed and shared memory systems, so called hybrid systems, are not further discussed here.

On the other hand, "shared memory" architectures have only one single memory unit which is shared by multiple cores (Figure 5.1, left side). The parallelization of the software BASEMENT is implemented for such shared memory architectures. This concept enables fast and easy communication between all cores through global access to memory. Furthermore, making benefit of parallel computing often is significantly easier and less time-consuming, especially for already existing sequential applications.

Such parallel computers with shared memory, often termed multi-core machines, are highly available today at low costs. The general trend in processor development is from multi core to many core systems with up to even more than hundred cores in cc-NUMA systems. As a consequence, a major drawback of shared memory parallelization, namely not being portable to distributed memory systems, seems acceptable for an application like BASEMENT which is not focused on high performance computing.

5.1.2 Parallelization issues on shared memory systems

5.1.2.1 Levels of parallelization

Parallelization in shared memory systems is based on multi-threading concepts. Threads are a way for a program to divide its work load into several independently running parts
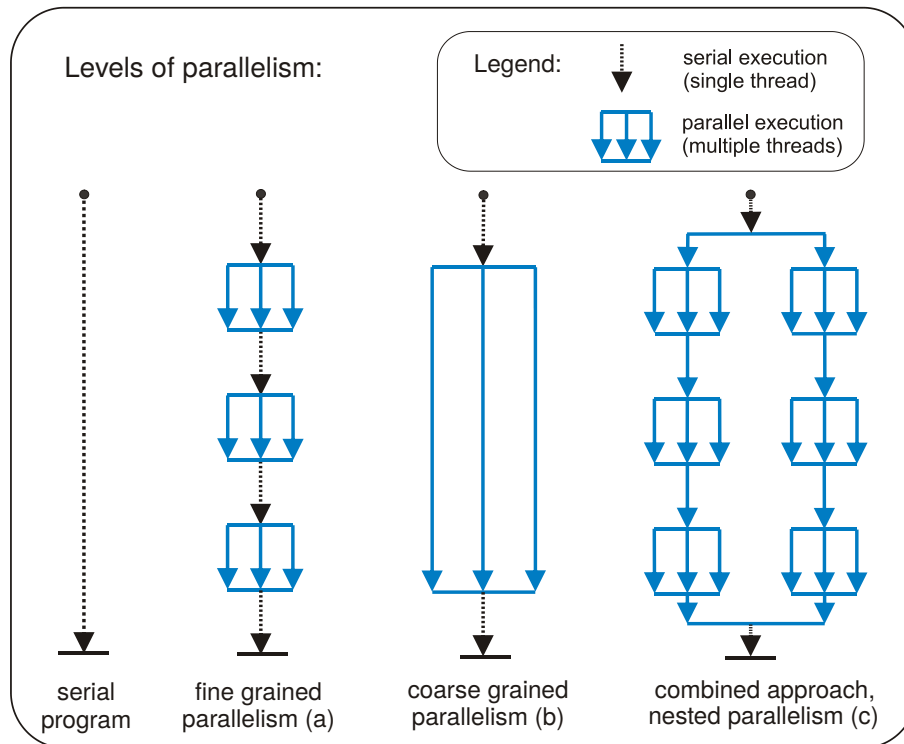
Figure 5.2 *Illustration of different levels of parallelism*

which are finally mapped on the available cores. Threads can be started ("forked") and terminated ("joined") by an application in order to exploit parallelism provided by multiple cores.

The distribution of work load on a series of threads can be made on a small scale loop level ("fine grained parallelism") or on a larger scale procedural or module level ("coarse grained parallelism"). See Figure 5.2 for illustration of these different concepts. Parallelizing on each of these levels has certain advantages and disadvantages. At the moment BASEMENT mainly implements a fine grained parallelism concept. In principle, both concepts can also be combined (nested parallelism).

Numerical applications like BASEMENT spend most of their execution time in rather small parts of the code. These parts are the main calculation loops which iterate over all elements and edges of the computational grid. Fine grained or loop level parallelization aims to exploit parallelism by parallelizing these time consuming loops. If the serial running program arrives at such a loop, its iterations are divided among a certain number of threads and are executed in parallel. Afterwards, a synchronization of the threads is performed so that the program can continue its serial execution (see Figure 5.2). This parallelization approach usually requires no crucial changes to the source code and is especially suitable for parallelizing already existing serial programs. All time consuming loops can be parallelized step by step, thereby incrementally increasing the parallel performance of the program. Due to the small changes in the source code the robustness of a tested serial program is maintained.

Whereas most loops can easily be parallelized, some loops contain flow dependences where the calculation results depend on an ordered succession of all iterations, which is generally not maintained during parallel execution. Such flow dependences can often be resolved by reorganising some parts of the loop. Details on locating and removing flow dependences in

loops can be found in literature (e.g. Chandra et al. (2001)).

5.1.2.2 Factors influencing parallel performance

Parallel performance is often measured as speedup S(n), which is defined as the ratio between "serial execution time" and "parallel execution time" and states how much faster the parallel application runs compared to the serial one.

The theoretical maximum achievable speedup is limited to the number of cores n. The actual speedup is determined by several factors whose influences largely depend on size and type of the simulation. Furthermore the scalability indicates wether the speedup increases linearly with increasing numbers of processors or at a slower rate. The key factors influencing the speedup and according parallel programming aspects are briefly discussed in the following section.

- **Parallel overhead**

  Parallel overhead is created at many locations and times in the program. Threads must be forked and joined, loop iterations must be divided among threads and threads must be synchronized. In the worst case, the parallel overhead can even exceed the performance gains from parallel execution. This is especially a problem in case of small loops with small workload, e.g. in case of initialisation loops. Such loops are not suited for parallelization and should better be executed in serial or, if possible, should be integrated into larger loops.

  For simulations with loops of rather small workload but very high number of time steps, it is important to prevent parallel overhead from frequently joining and forking threads. This was achieved in Basement by integrating the whole time loop into a single parallel region.

  The significance of parallel overhead in a parallel application depends largely on the computational costs of the simulation domain. Simulations with high computational costs, e.g. simulations with large numbers of elements, are less negatively affected.

- **Coverage**

  In order to obtain a good parallel scalability it is important to parallelize large portions of the code resulting in a so called high "coverage". A high coverage is of importance, because the negative impacts of remaining serial parts in the code increase as more and more cores are employed. The achievable speedup finally becomes limited by these serial parts (see "Amdahl's law", Chandra et al. (2001)).

- **Load balance**

  The overall execution time of a parallel loop is determined by the thread with the slowest execution time. If the work load is not properly balanced between the multiple threads performance will suffer. A good approach for parallelizing a loop often is a simple static scheduling, which means that all iterations are divided in parts of equal size before the threads start execution. But in case that the computational costs of the iterations differ largely, a static scheduling may not be optimal. E.g., in 2-D simulations with dry and wet regions, the flow equations must only be solved entirely for the wet or transitional elements but not for the dry elements.

Beside the problem of moving boundaries of the wetted domains, it is also important to note that hydrodynamic models may lead to imbalanced load distributions in case of local, highly unsteady processes like wave propagations.

The load balance can sometimes be optimized in such cases by the use of dynamic scheduling, which divides the work load dynamically among the threads. Threads with computational cheap iterations will dynamically receive additional iterations, thereby taking load from threads with costly iterations. Dynamic scheduling can improve performance in case of load imbalance, but leads also to an increased overhead and suffers from bad data locality. Consequently, the decision between static or dynamic balancing is largely problem dependent.

- **Synchronization**

  Synchronization between threads can become time consuming if some threads must wait for other threads to finish execution. Quite similar, in some cases threads must be prevented from mutual accessing the same memory locations to prevent read/write conflicts and data races. Such parts of the code must be protected by setting mutual exclusions using locks. These code parts can only be accessed by one thread a time and may cause other threads having to wait until access is granted.

  Such bottlenecks caused by synchronisations and mutual exclusions should be avoided and minimized as far as possible. They can sometimes be prevented by reorganising parts of the algorithms. Prevention of memory conflicts can sometimes also be achieved by privatization of the problematic shared variables, i.e. global variables are replaced by thread private variables.

- **Locality**

  When parallelizing a program attention should be paid on data locality aspects. Modern cache based processors optimize execution speed by caching data which speeds up data load and store cycles. Caches are very efficient if a processor can use the cached data (the local data) for many operations instead of having to reload them from memory or from caches of other processors. In general, better data locality is guaranteed if the work load is distributed statically among the threads instead of using a dynamic schedule.

  Additional problems limiting the parallel performance may arise when OpenMP parallelized programs are executed on cc-NUMA architectures with large number of cores due to general aspects of the underlying hardware architecture (Chapman et al. (2008)).

5.1.3   Parallelization with OpenMP

5.1.3.1   Overview

As mentioned before, parallel programming for shared memory systems bases on multi-threading software concepts. Thread programming can be done on a rather complex and time consuming low level by explicitly specifying and controlling all threading options. In recent years high level concepts for thread based parallelization were developed which ease parallelization and do not require low level thread programming skills. Among these high level concepts it can be differentiated between implicit parallelism using parallel programming languages or automatic parallelization by compilers and explicit

parallelism where the developer controls the parallelism with support of parallel libraries and application programmer interfaces (APIs).

In cooperation with leading software and computer enterprises a parallel API called OpenMP ("Open Multi Processing") was developed which is today the de facto standard for parallelizing scientific and engineering applications on shared memory systems. OpenMP is used for the parallelization of BASEMENT.

5.1.3.2   Characteristics of OpenMP

OpenMP consists of a set of compiler directives and library functions. With these compiler directives the developer describes the parallelism in the code. Therefore a compiler is needed which supports these OpenMP directives. OpenMP is currently supported by many C/C++ and Fortran compilers on a variety of platforms. In the table below some aspects pro and contra parallelization with OpenMP are listed (see also Kuhn et al. (2000) for a comparison of OpenMP and threading in C/C++).

Table 5.1 OpenMP parallelization – pro and contra.

Pro	Contra
First successes in parallelization are relatively easy to achieve. Compiler directives reduce the needed programming efforts and implementation details are left to the compiler.	No full control over the implementation details is possible. Bugs in libraries can be difficult to isolate.
Rather small increases in the size of the source code due to the use of compiler directives. Good readability of the code is maintained and the algorithms are not buried by large blocks of added parallelization instructions.	Compiler generated code portions and library calls can complicate debugging of the parallel program.
OpenMP is standardized and portable on different platforms.	OpenMP does not support the whole whole range of threading concepts (e.g. no support for semaphores for synchronization).
The source code can be compiled as serial program without support of OpenMP (compiler directives are treated as comments). This may ease debugging of new implemented application features.	

5.1.3.3   Parallel directives in OpenMP

The basic parallelization construct in OpenMP is the so called "parallel region". The source code placed within a parallel region is executed in parallel on a number of threads. At the end of a parallel region an implicit barrier is automatically set to synchronize the threads.
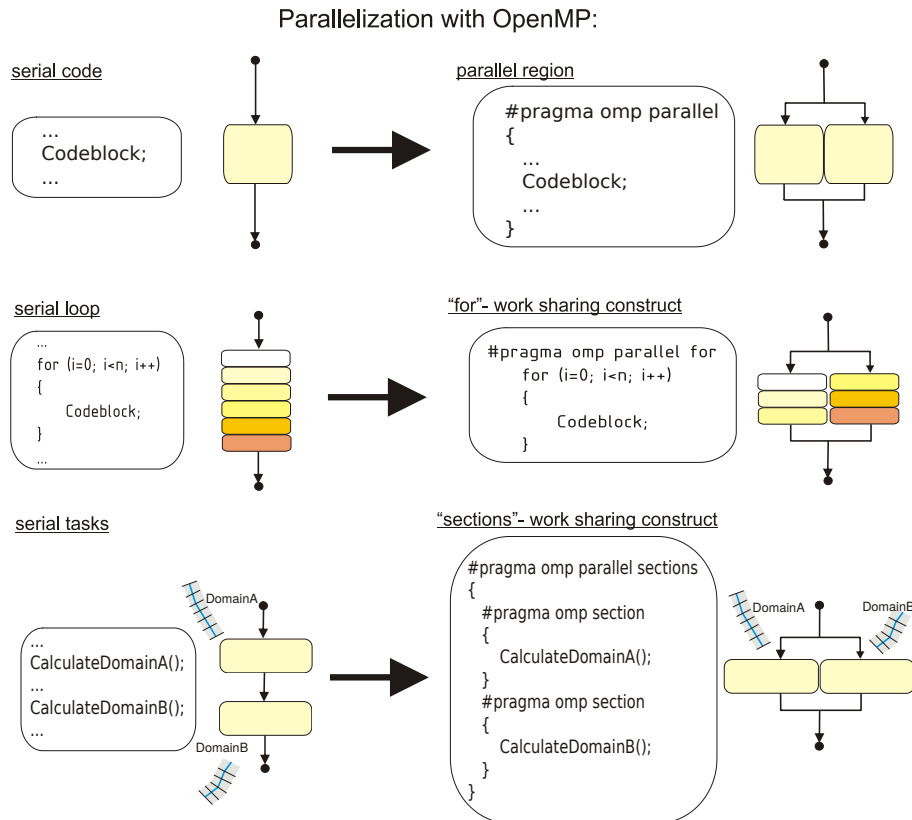
Figure 5.3 *Compiler directives for basic parallelization constructs in OpenMP*

To ease the distribution of work load among the threads of a parallel region, OpenMP supports work sharing constructs which automate such tasks, see Figure 5.3. With optional parameter clauses it is possible for the user to define the behaviour of these compiler directives in more detail. A complete OpenMP reference is available online at http://www.openmp.org ("Online reference of openmp"). For special needs it is also possible to share the work load among the threads fully flexible by individually addressing each thread.

In OpenMP several compiler directives are supported for synchronization issues which cover the most frequent tasks. Such constructs are barriers, locks, critical sections and atomic constructs. Beside these standard constructs, fully flexible synchronisation operations can be implemented for special tasks using global flags in memory. The choice of the best suited constructs for synchronisation and mutual exclusion is problem depended and involves differing efforts in parallel programming.

Some general parameters controlling the parallelism can be set in OpenMP at the program start or during runtime with library calls or by setting environment variables. Such parameters include, for example, the number of parallel threads or the choice between static or dynamic scheduling. This enables the flexibility to let the user determine the number of threads used for parallelization or to optimize the parallel speedup by varying the type of schedule.

synchronization / mutual exclusion

```
#pragma omp parallel
{
   ...
   #pragma omp critical section
   {
      ...
   }
   ...
} // implicit barrier
```

thread A   thread B

critical
section

wait until
critical section
is accesible

critical
section

wait until
thread B has
finished
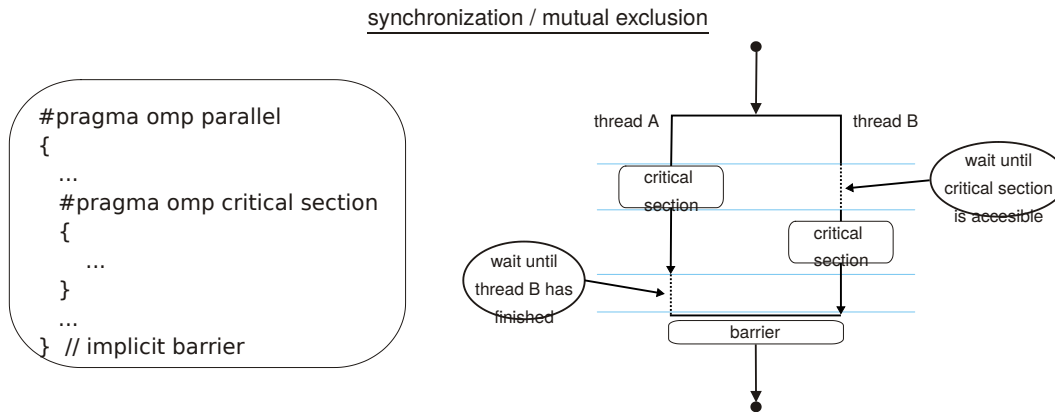
barrier

Figure 5.4 *Critical sections and barriers for synchronization operations in OpenMP*

5.2 Model Coupling

5.2.1 Introduction

In addition to the simulation of single sub-domains using BASEchain (1-D) or BASEplane (2-D), the software BASEMENT also provides the possibility to connect sub-domains for combined numerical simulations. Such coupled simulations can range from simple configurations up to simulations of river networks with integrated river junctions / bifurcations or integrated 1-D/2-D modelling. In Figure 5.5 a river network of multiple sub-domains with several coupling interfaces is illustrated. The coupling mechanisms thereby allow to couple hydrodynamic simulations as well as morphological simulations with sediment transport and suspended load.

Some typical applications of coupled simulations are:

- A step wise modeling approach to the overall problem using smaller parts of the whole domain. This approach has the advantages of reduced complexity and reduced execution and calibration times. Also extensions of existing and calibrated models can be easily made with coupled simulations without the need to redesign the existing models.

- Simulations with hydraulic structures (like weirs or gates) within the domain of interest can be realized by using multiple sub-domains, which are coupled via these hydraulic structures.

- Coupled simulations can be helpful for mixed-dimensional modeling approaches, e.g. for cases where large scale 1-D simulations shall be combined with detailed modeling of local areas in 2-D. Thereby, the advantage of efficient and robust modeling in 1-D is combined with the capability to simulate 2-D flow characteristics. Also, the required efforts for data acquisition and data preparation can be minimized using mixed-dimensional modeling approaches.

5.2.2 Coupling Types

The implemented coupling types are briefly sketched below.
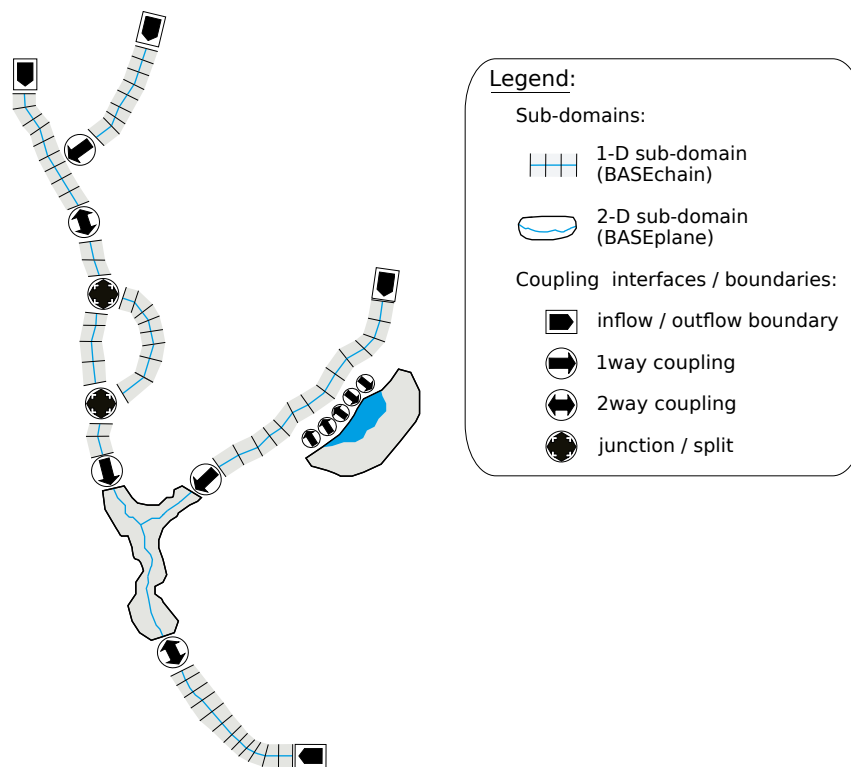
Figure 5.5 *River network with multiple BASEchain (1-D) and BASEplane (2-D)
sub-domains and several coupling interfaces.*

- **Sequestial, Riemann (1D)**

  Single sub-domains can be combined sequentially via coupling interfaces at the
  upstream or downstream boundaries (Figure 5.6). This can also be done for
  sub-domains with mixed dimensionalities (1-D / 2-D, 2-D / 1-D), see Figure 5.7.

  These **sequential** coupling types can be used to combine sub-domains over their
  boundary conditions or external sources. For example, a weir outflow boundary can
  be combined with an input hydrograph of a downstream boundary.

  Beside sequential couplings, also so called **Riemann** couplings can be used (at the
  moment only 1D) which set a Riemann solver between the sub-domains and allow
  flow in any direction. This coupling type requires special 'connection' boundaries at
  the coupling interfaces.

- **Junctions / Bifurcations / ConfluenceWSE (1D)**

  Coupling interfaces for river **junctions** or river **bifurcations** allow a simplified
  modelling of conjunctions of river branches within a 1-D river network (Figure 5.8).

  Beside these both coupling types also a **confluenceWSE** coupling can be used which
  tries to establish a common water surface elevation (WSE) at the confluence point
  (at the moment only 1D). This coupling allows flow in any direction and requires
  special 'connection' boundaries at the coupling interfaces.

- **Lateral coupling.**

  For integrated 1-D and 2-D modelling, a 1-D sub-domain can be coupled laterally
  with a 2-D sub-domain. The coupling takes place along the river channel via multiple
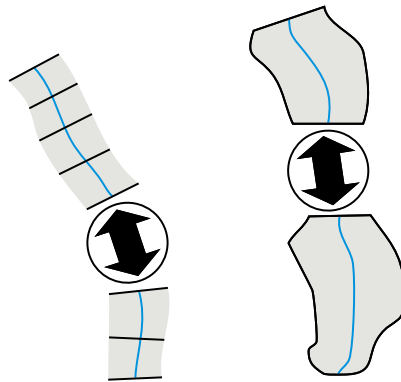
Figure 5.6 *1-D / 1-D coupling (left), 2-D / 2-D coupling (right)*
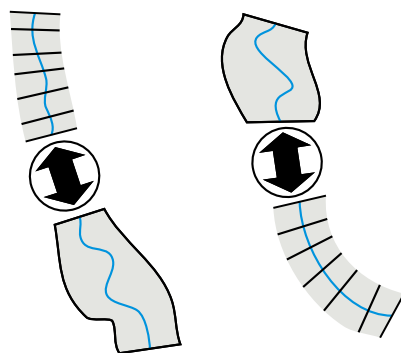
Figure 5.7 *1-D / 2-D coupling (left), 2-D / 1-D coupling (right)*
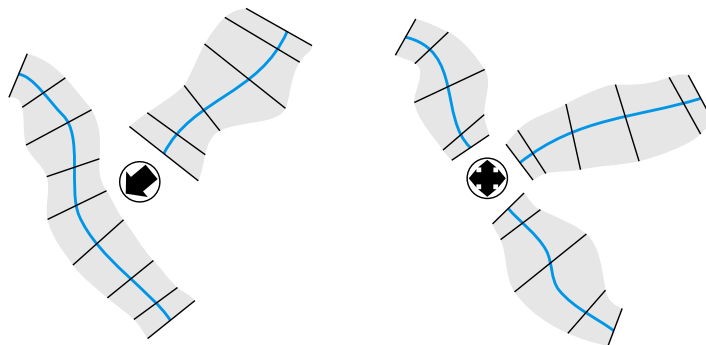
Figure 5.8 *junction /bifurcation / confluence WSE (1-D)*
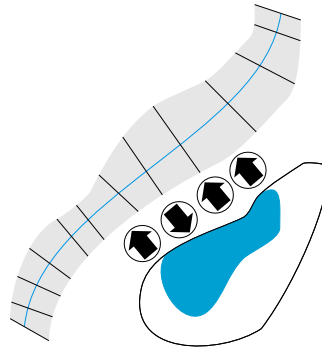
Figure 5.9 *lateral coupling*

coupling interfaces which connect cross sections (1-D) with corresponding mesh elements (2-D), see Figure 5.9.

5.2.3 Coupling Mechanisms

5.2.3.1 Explicit coupling of sub-domains

Coupling of sub-domains is implemented as an explicit coupling approach, which means that data is exchanged explicitly between the sub-domains at certain time intervals. This approach is simpler to implement than an implicit approach, especially regarding the coupling of sub-domains with mixed dimensionalities. However, in comparison to an implicit coupling approach, special care must be taken to achieve robust and stable combined simulations.

5.2.3.2 One-way coupling and two-way coupling

A simple way to couple two sub-domains is to exchange data only in one direction from upstream to downstream. Such a situation is termed as 1-way coupling from here on. It has the advantage that the upstream sub-domain can run independently from the downstream sub-domain and the flow variables are passed over at some time intervals to the downstream sub-domain. But being a one-directional coupling, no information from downstream can travel upstream. Therefore, this type of coupling is restricted to cases where no backwater effects from downstream take place or such influences can be neglected.

In contrast, a two-way coupling enables mutual interactions between the sub-domains by providing mutual data exchange. In two-way coupled sub-domains, backwater effects from downstream can influence the upstream sub-domain. Instead of executing the sub-domains sequentially from upstream to downstream direction, here the sub-domains are executed simultaneously. The two-way coupling approach has the difficulty that no unique flow variables are present at the coupling cross sections, as water levels from upstream and downstream direction may differ for a given time. In principle, iterations between the sub-domains are required and must be performed until the differences of the variables at the coupling cross section do no longer change within subsequent iteration steps. Although a rather small number of iterations has to be expected (as reported by Miglio et al. (2005)), these iterations lead to large additional computational efforts. Therefore these iterations are not performed here. As the time steps in the explicit approach are usually very small, the

differences between the upstream and downstream variables are rather small and iterations may be neglected without substantial loss of accuracy. But in cases of crucial and abrupt changes in the flow variables, oscillations may result.

A combination of both concepts can be used in the coupled river simulation. One-way coupled sub-domains are executed sequentially from upstream to downstream direction, whereas two-way coupled sub-domains are treated as being a single sub-domain within the execution sequence.

5.2.4   Definitions of Exchange Conditions

5.2.4.1   General remarks

Data can be exchanged between the sub-domains by coupling interfaces using boundary conditions and source terms. The following table shows the exchange variables grouped by the direction of the exchange.

Table 5.2 *Possible exchange conditions between sub-domains.*

direction of exchange	type of coupling	exchange variables
in downstream direction	boundary conditions & sources	Q : discharge $q_{b,g}$: bed load C_g : concentration
in upstream direction	boundary conditions & sources	z_s : water surface elevation

In order to enable simple, flexible and efficient coupled simulations, some assumptions are made here:

- It is assumed that flow directions at the coupling interfaces of the river network are known a priori and do not change during the simulation (with the exception of special coupling types, like the lateral coupling).

- The cross sections (1-D) or mesh elements (2-D) of the coupling interfaces should ideally be located at the same or nearby locations and have the same geometries. This is necessary to reduce possible errors around the coupling interfaces due to the disregard flow taking place in between and to avoid discontinuities due to abrupt changes in the geometries.

- It is assumed that the flow is orthogonal over the boundaries, i.e. the directional x- and y flow components in 2-D are not exchanged separately.

- In 2-D coupling only summarized or averaged data are exchanged, instead of exchanging data separately for each edge or element. This approach simplifies the coupling setup since no restrictions are set regarding the geometries and number of cells at the boundaries or sources.

5.2.4.2   Exchange conditions for mixed-dimensional sub-domains

Exchange via boundary/sources:

Table 5.3 *Exchange conditions for mixed-dimensional coupling*
(i=index of 2-D edge or 2-D element)

Echange variable	Exchange equations	Nr of exchange terms
Discharge	$1D \rightarrow 2D$: $q_i^{2D} = \omega_i Q^{1D}$ (ω_i=area/length weighting or conveyance weighting) $2D \rightarrow 1D$: $Q^{1D} = \sum_i^n q_i^{2D}$	1
Water surface	$1D \rightarrow 2D$: $z_{S,i}^{2D} = z_S^{1D}$ $2D \rightarrow 1D$: $z_S^{1D} = \frac{1}{n} \sum_i^n z_{S,i}^{2D}$	1
Bed load	$1D \rightarrow 2D$: $q_{b,g,i}^{2D} = \omega_i q_{b,g}^{1D}$ (ω_i=area/length weighting) $2D \rightarrow 1D$: $q_{b,g}^{1D} = \sum_i^n q_{b,g,i}^{2D}$	g=1..ng
Suspended load	not available yet in 2D	g=1..ng

5.2.4.3   Exchange conditions for river junctions in 1-D river networks

Within a river network locations are encountered where river branches flow together or where a river bifurcates into several branches. The flow characteristics at such conjunctions generally are multidimensional. Therefore the preferable modelling approach to achieve a good accuracy is to simulate a 2-D sub-domain. But if such a situation shall be modelled with 1-D sub-domains than special coupling concepts are required.

Two different approaches are implemented in BASEMENT (see Figure 5.10). These approaches allow no more than three sub-domains being part of a junction. If a larger numbers of river branches are to be modelled, they must be approximated by multiple junctions, placed in small distances.

Following the first approach a junction can be regarded as region where three different river branches meet and mutually exchange data (a). A control volume is defined to which mass and momentum conservation principles can be applied. A simple approach is here balancing discharges and assuming equal water surface elevations along the junction.

Table 5.4 *Exchange conditions for river junctions*

	Exchange conditions	Nr. of equations
Discharge	$Q_{up1} + Q_{up2} = Q_{down}$	1
Bed load	$Q_{up1,bed,g} + Q_{up2,bed,g} = Q_{down,bed,g}$	g=1..ng
Suspension	$Q_{up1}C_{up1,g} + Q_{up2}C_{up2,g} = Q_{down}C_{down,g}$	g=1..ng
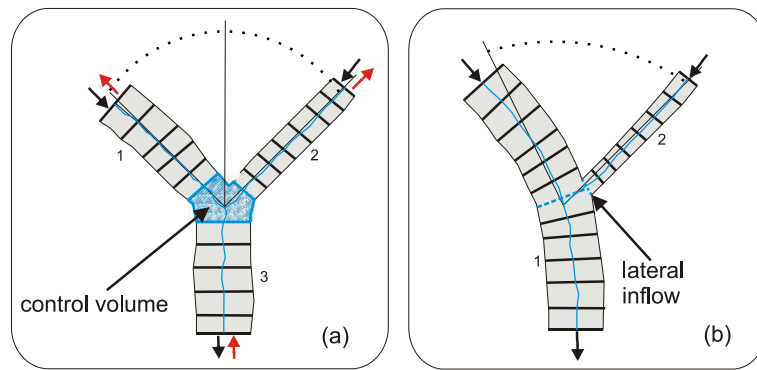
Figure 5.10 *Modeling of a river junction with two different approaches (black arrows indicate a confluence of river branches, red arrows a bifurcation): (a) = river junction with 3 different river branches, (b) = junction as a lateral inflow of a tributary.*

The second approach is to regard the junction as a lateral inflow of a tributary into a river at a specified location (b). The discharge (and sediment) is passed from the tributary to the river as lateral inflow via source term. Additionally, the water level at the inflow cross section can be passed in return to the tributary. Despite its simplicity this approach can be suited well to simulate simple river junctions in 1-D.

5.2.4.4 Exchange conditions for river bifurcations in 1-D river networks

In case of modeling a river branch which bifurcates into two branches, the upstream discharge (and sediment) must be distributed among the two downstream sub-domains. The distribution factor ϕ among the downstream sub-domains has to be chosen according the local conditions. The downstream water elevations of the two downstream sub-domains are averaged and then passed in upstream direction.
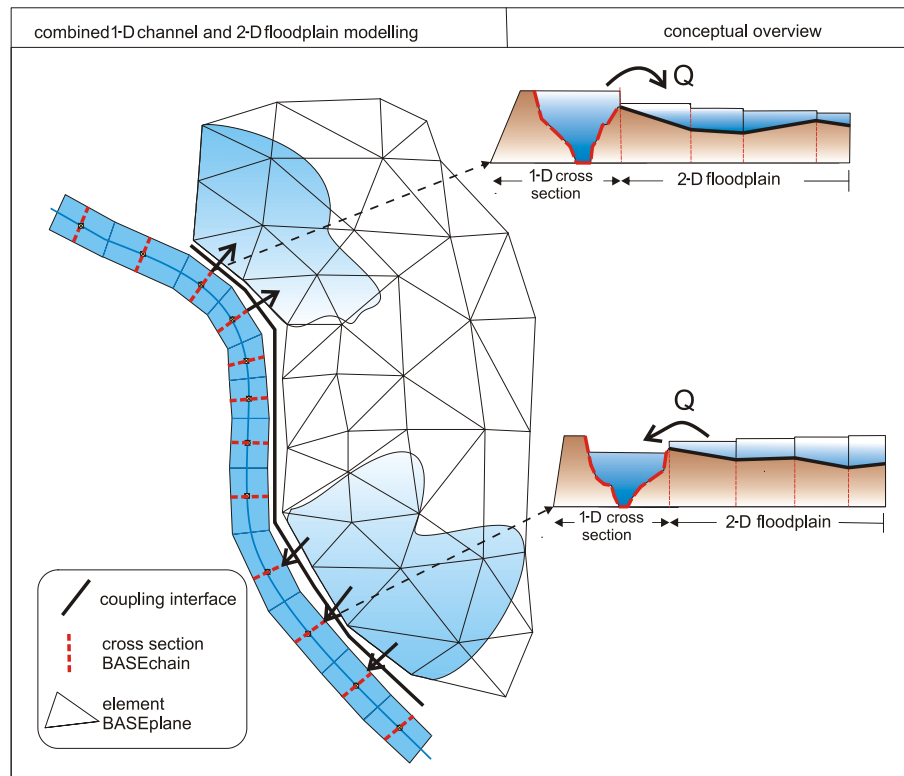
Figure 5.11 *Conceptual overview of combined 1-D river flow and 2-D floodplain modeling*

Table 5.5 *Exchange conditions for river bifurcations*

	Exchange conditions	Nr. of equations
Discharge	$Q_{up} = \phi Q_{down1} + (1 - \phi)Q_{down2}$	1
Bed load	$Q_{up,b,g} = \phi Q_{down1,b,g} + (1 - \phi)Q_{down2,b,g}$	g=1..ng
Suspension	$Q_{up}C_g = \phi Q_{down1}C_{down1,g} + (1 - \phi)Q_{down2,g}C_{down2,g}$	g=1..ng

5.2.4.5   Exchange conditions for combined 1-D and 2-D modelling

The combined 1-D river flow and 2-D floodplain modelling bases mainly on the approach presented by Beffa (2002). A conceptual overview is given in Figure 5.11 which illustrates river cross sections of the BASEchain sub-domain and the 2-D mesh of a floodplain modelled with a BASEplane sub-domain.

The coupling interfaces between the sub-domains are implemented as one-way couplings via source terms. As a consequence, only discharges are exchanged between the sub-domains. The exchange between the sub-domains is calculated as weir flow over the dykes of the 1-D cross section or as weir flow over the edges of 2-D sub-domain. The weir level is chosen as the higher elevation of the dyke or the corresponding edge. As weir width in the weir formula the length of the 2-D boundary edge is taken. Exchange of discharge is possible in both directions, either from the river into the floodplains or backwards depending on the water elevations in the 1-D cross section and the corresponding 2-D element.

To enable a flexible coupling approach it is possible to connect a 1-D cross section with multiple 2-D elements (1:n-relation). For each 2-Element, in contrast, only one connection to a 1-D cross section is possible (1:1-relation). The coupling interfaces are defined using a list, from which the connections are automatically extracted and generated during a pre-processing step.

Table 5.6 *Exchange conditions for lateral coupling*

	Exchange conditions	Nr. of equations
Discharge	$1D \rightarrow 2D$: $Q = \delta\,\mu\dfrac{2}{3}b_{weir}\sqrt{2g}h^{3/2}$ if $(z_{S,1D} \geq z_{S,2D})$ (side weir, δ= side weir reduction factor)	1
	$2D \rightarrow 1D$: $Q = \mu\dfrac{2}{3}b_{weir}\sqrt{2g}h^{3/2}$ if $(z_{S,1D} < z_{S,2D})$ (weir overfall over edge)	

The 1-D dyke crest elevation, where the water overtops, as well as the 1-D water surface elevation are interpolated between the cross section at the location of the 2-D edge. This procedure shall increase the accuracy of the lateral exchange modeling. It is assumed hereby, that the dyke-crest elevation and the water surface elevation vary linearly between two adjacent 1-D cross-sections.

5.2.4.6 Data exchange for morphological simulations with multiple grain classes

In morphological, coupled simulations the possibility exists that sub-domains can have differing grain compositions. The handling of data exchange for such cases is not trivial and unclear. But such situations may arise in coupled large-scale simulations where grain classes get finer along course of the river. A flexible approach is adopted here which allows the usage of differing compositions as well as different numbers of grain classes of the sub-domains.

For data exchange the bed loads of each grain class are mapped on the grain classes of the receiving sub-domain. The mapping is achieved by three successive steps as illustrated in Figure 5.12. The sediment mass balance is thereby fulfilled.

5.2.5 Synchronization Concept

5.2.5.1 General remarks on synchronization

For the coupling of sub-domains a synchronization mechanism must be implemented which directs the execution of the sub-domains and controls the data exchanges at the appropriate times. The type and complexity of the synchronization effort thereby generally depends on the degree of spatial and temporal compatibility of the sub-domains. Especially in case of combined 1-D and 2-D simulations the spatial extends and time step sizes can vary considerable.

Mainly two different coupling concepts are often encountered for the selection of the time step sizes of the sub-domains.
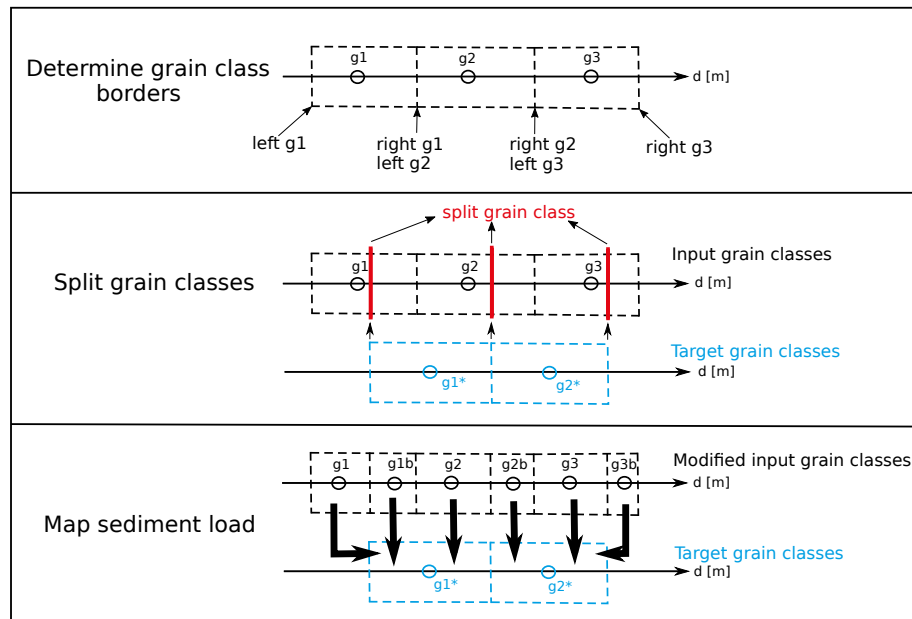
Figure 5.12 *Mapping of grain compositions from one sub-domain to another*

- All sub-domains are executed in a synchronous manner with an equal time step size. To guarantee stable execution the chosen time step size ("global time step") is set to the minimum time step size of all sub-domains, which is determined by stability conditions (CFL criterion). But due to the fact that the sub-domain time step sizes can vary considerable, such a restriction on the minimum time step size can lead to inefficient small time step sizes resulting in large computational efforts.

- In contrast, all sub-domains can be executed asynchronous with different time step sizes ("local time steps"), which are chosen according the sub-domain's optimal time step size. This approach does not suffer the computational inefficiency due to small time step sizes. But generally more synchronization efforts are required and data exchange between the sub-domains requires interpolations and can become cumbersome especially for complex interfaces like junctions or bifurcations.

Here, another approach is selected, a local-time stepping approach, lying in between these concepts and combining efficiency and simplicity.

5.2.5.2 "Local time stepping" approach

This approach bases on the method of local time stepping (LTS) as presented by Osher and Sanders (1983) and Sanders (2008). But in contrast to these methods, LTS is applied here to whole sub-domains instead of single grid elements. Different local time step sizes are allowed for the sub-domains instead of using one global time step for all sub-domains. This enables efficient computations by preventing very small time steps of single sub-domains to dominate the time step sizes of the other sub-domains. But restrictions are set for the time step sizes in a way to ensure that the sub-domains always reach common time levels. At these common time levels data can be exchanged easily without the need for interpolations.

Hierarchical levels L are introduced and attributed to each sub-domain. These levels categorize the sub-domains into groups of common time step sizes. These levels are thereby
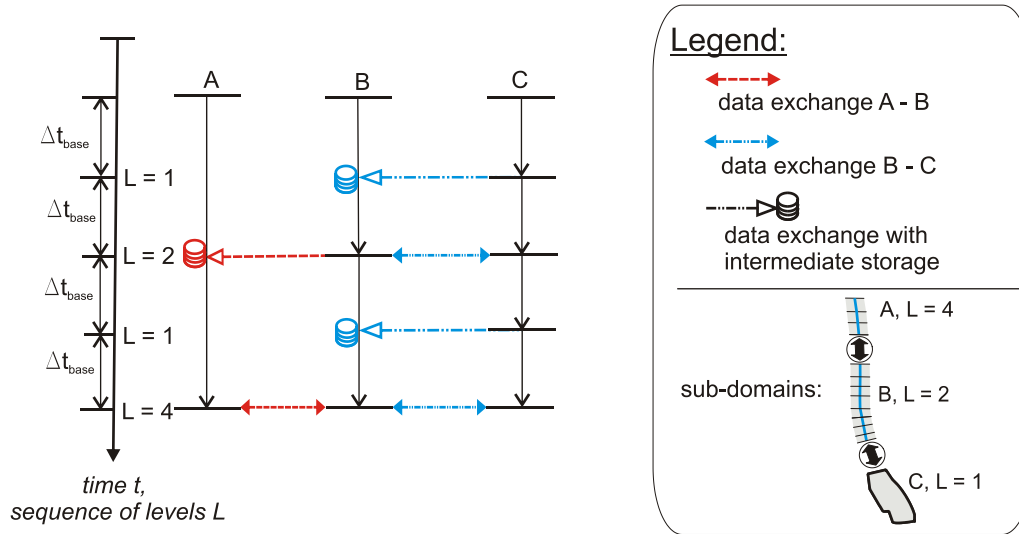
Figure 5.13 *LTS-synchronization for 3 sub-domains with different time step sizes. The sub-domain C with the smallest time step size determines the base time step. Sub-domains A and B run for multiples of 4 and 2 of the base time step size.*

chosen as power-of-two multiples of the base time step size Δt_{base} . This base time step is selected as the minimum time step size of all coupled sub-domains. The attribution of levels L to a sub-domain i depends on the relation of its present time step size to the base time step size and is determined as:

$$2^k \leq \frac{\Delta t_i}{\Delta t_{base}} < 2^{k+1} \rightarrow L_i = 2^k, k = 0..n$$

where L_i is the level attributed to the sub-domain i , k indicates the level and n is the number of levels. Each sub-domain determines its own local time step size as its level L_i multiplied with the base time step size $\Delta t_i = L_i \Delta t_{base}$.

The execution of the sub-domains takes place in loops over level sequences. One loop sequence of the LTS synchronization is sketched in Figure 5.13 for three sub-domains with different time step sizes ($\Delta t_A > \Delta t_B > \Delta t_C$) and levels L_i.

For example, in case that the maximum level of a sub-domain is 8, a level sequence of $m = [\,1,2,1,4,1,2,8\,]$ is executed, where m equals the present level of the loop. Each sub-domain is executed only if its level L_i is smaller or equal to the present level m. These sub-domains are then advanced for a time step size of $\Delta t_i = L_i \Delta t_{base}$. Data exchange between adjacent sub-domains takes place only when the sub-domains have reached a common level. If adjacent sub-domains have different time levels then the exchanged data must be stored intermediately to guarantee conservation principles. The data is finally passed over when the sub-domains reach a common time level. After the end of the loop of the level sequence, all sub-domains have been executed at least once and have finally have reached a common final time $t_{new} = t_{old} + n\Delta t_{base}$. From this starting point the levels L_i are assigned again to the sub-domains and the procedure is repeated.

The selection of the base time level is done at the beginning of each level loop. To account for the possibility that the minimum time step could change during the loop iterations, due to changed flow conditions, the base time level can be reduced by a factor $F \leq 1$ for stability reasons.

5.2.6   External Coupling

5.2.6.1   Introduction

The term "external coupling" means the coupling between the program BASEMENT and an external program. This may be e.g. a rainfall-run off model which delivers input data for a river reach or it may be a standalone groundwater model which makes use of the stream water elevations computed by BASEMENT. As described in the model coupling section, one can also distinguish here between one-way coupling and two-way coupling.

One-way coupling

Two different scenarios can be distinguished here:

- An external Program may receive data which is sent by BASEMENT. Therefore the external program must be defined as an external sub-domain in the command file. In the OUTPUT block an output must be defined and connected with this external sub-domain. While executing, BASEMENT sends data as soon as it is computed using the output routines to the specified external sub-domain. The external program must fetch the data using TCP/IP routines and must take care of the synchronization, i.e. it must always wait until new data is available.

- Another scenario is to send input data to BASEMENT. Thereby the external program again has to be defined as an external sub-domain in the coupling process and it must be connected with other sub-domains using boundary conditions. Then, the external program can send its data in XML format to BASEMENT using TCP/IP routines. BASEMENT takes care of the synchronization within the coupling process and always waits until new data is available before executing.

Two-way coupling

It is possible to couple an external program with BASEMENT with mutual data exchange. Again, the external program must be defined as an external sub-domain in the coupling process. Furthermore, the external program must implement a synchronization mechanism in order to check if the needed data is available. Differing from previous versions, BAEMENT now does NOT apply the local-time-stepping algorithm (LTS) to the external coupling. Data exchange takes place, when either BASEMENT or the external program is ahead in time.

For example:

- The external program is executed until it is ahead in time compared to BASEMENT. Then it has to send its current time and input data to BASEMENT. Afterwards, it waits and checks for incoming data over TCP/IP.

- BASEMENT waits until the external program is ahead in time. While waiting, it checks for incoming data over TC/IP. If the incoming data shows that the external program is ahead in time, then BASEMENT runs as long as its current time is behind the time of the external program. Afterwards, if BASEMENT run-time exceeds the current time of the external program, it sends its current time and data to the external program.
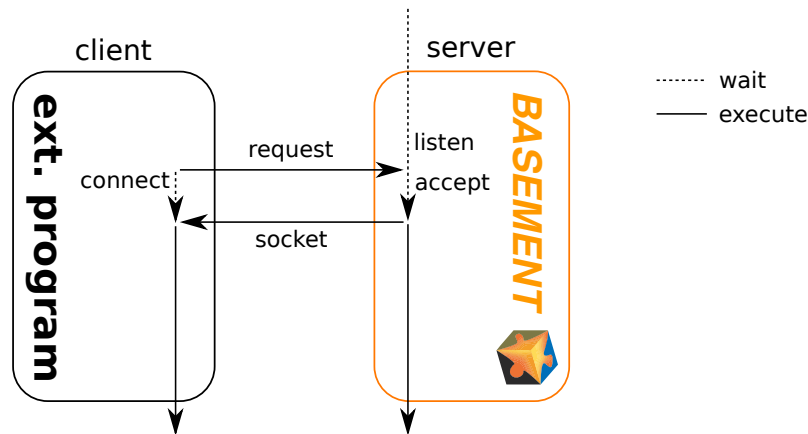
Figure 5.14 *Connection request from external program (client) to BASEMENT (server)*

Note

Please be aware that the external coupling approach is still in an experimental stage. In addition, the usage of this coupling requires programming efforts and knowledge in TCP/IP programing and XML parsing. External coupling may require the implementation of special boundary conditions in the BASEMENT model. For example, coupling with a groundwater model requires leakage boundaries for water exchange to be set. If you want to make use of such a coupling type, you may contact the developer team regarding the implementation of appropriate boundary conditions in the model.

5.2.6.2   Data exchange over TCP/IP

The data exchange between BASEMENT and the external program takes place using TCP/IP communication. This has the advantages that it is generally faster than communications via files and enables the coupling between different computers via intra- or internet, even using different operating systems.

Create connection

The communication requires an IP-address and a port-number as identifier and takes place using TCP-sockets. Usually BASEMENT runs as the server application and must be started first. Then, it waits for an incoming connection request. After the incoming request, a connection is established with the external program and the connection information is sent to the external program (socket descriptor). In case of multiple external programs, BASEMENT waits until all connections are established before it starts the computations.

Data packet

The data is wrapped in "data packets" using the common XML-format, whereby the data values and several additional attributes must be specified. All communications between the programs take place by sending data packets. Therefore, also additional information, like e.g. the time, or the time step size, must be included in the data packet. It is also possible to send or receive multiple data packets for different data types or boundaries. The XML-tag has the following structure

```
<Data attribute1="..." attribute2="...">...<\Data>
```
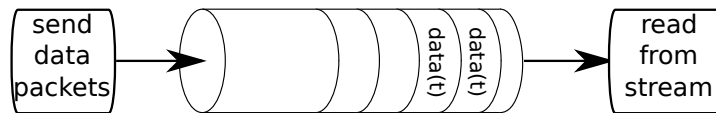
***Figure 5.15** Sending and receiving data from socket stream (FIFO pipe)*

The data values within the XML tag can either be written as ascii or binary data. If ascii format is used, a semicolon separates multiple data values from each other, e.g.

`<Data>10.0;10.0;10.0</Data>`

The following attributes can be set:

Attribute	Values	Obligatory
size	number of data values	Yes
time	time of data values in sec	Yes
type	[Q,h,v,...] type of data	Yes
encoding	[ascii, binary]	No (default=ascii)
byte_order	[little_endian]	No
boundary	name of boundary condition of data values	Only if data is sent to BASEMENT
timestep	current time step of model in sec	Only for local time stepping

Data communication

The data communication via sockets can be compared to data exchange via file-streams. The data packets are inserted into a pipe and the other side of the connection reads the contents after the FIFO concept (First In First Out). The receiving part of the connection must parse the contents of the pipe and extract the data packets. The time attribute of the data packets indicate the time level of the other program required for the synchronization. If the time levels of the data packets are behind the program's time or if no data is in the pipe, than the program must wait and continuously check for incoming data.

5.3   Flow Control in River Systems

5.3.1   Introduction

The flow characteristics of a river system are not only governed by the character of a channel, the morphology and topography, but also by regulations for hydropower stations and lakes. Such regulations commonly demand that a certain water level is maintained or
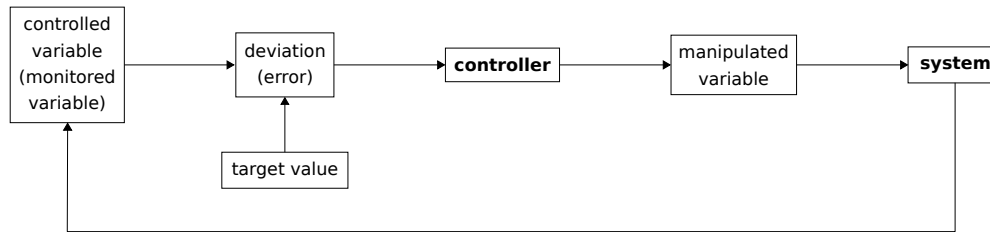
Figure 5.16 *Basic Control Cycle*

impose certain limits on the maximum discharge. The exertion of control structures has commonly a significant direct impact on certain river sections or even on the whole river system. The setting of the control structures over time cannot be defined in advance, but depends on the reaction to a change of the whole river system.

The numerical simulation of regulations is very helpful to properly judge such river systems, as it allows assessing and optimizing the effect of individual regulations of control structures on the whole system. This is of great importance as efficient flood control demands an optimal use of existing retention structures.

Therefore, the automatic steering of control structures has been added to BASEMENT, covering 1-D and 2-D simulations as well. The chosen approach allows the simultaneous combination of different controlled and manipulated variables. Controlled variables can be either water surface elevations or discharges. Here not only fixed values can be defined, but also series in time or values depending on the current flow in the river system. As manipulated variable, settings of weir or gates and an abstract outflow hydrograph has been implemented.

Within the present implementation, the determination of the control structure settings have been strongly abstracted, which allow a very flexible integration of further controlling algorithms in the future. As reference, a classical Proportional-Integral-Derivative (PID) controller has been implemented. By combining various control and manipulated variables within a single controller, BASEMENT now offers the possibility to simulate complex series of weirs over coupled regions.

5.3.2   Concept of Flow Control

There are many cases where the behaviour of boundary conditions such as weirs or gates depends on the actual state of the river system and cannot be described by a simple time-dependent boundary setting. An example would be adjusting the weir height in order to maintain a specific water level in front of the weir. This process is commonly denoted as controlling. The basic controller has a controlled variable, such as water surface elevation, which is desired to be kept at a certain level, i.e. its target value. The deviation from the current value of the controlled variable and its target value, also denoted as error, is then fed into the controller. The controller reads the deviation and calculates the new value for its manipulated variable. An example for a manipulated variable would be a weir height. The new value for the manipulated variable is then fed into the system, i.e. the hydraulic simulation, which finally affects the controlled variable.

In Basement, the system is represented by the simulation, the controller is a mathematical function $\boldsymbol{f}(.)$, determining the values of the manipulated variables $\boldsymbol{u}(t)$ from the values of

the monitored variables $\boldsymbol{m}(t)$. This can be expressed using the following mathematical expression:

$$\boldsymbol{u}(t) = \boldsymbol{f}(\boldsymbol{m}(t))$$

Logically, there can be multiple monitored and manipulated variables.

5.3.2.1   Monitored Variable

A monitored variable is defined by

$$m_i(t) = \nu_i(t - \tau_i) - \nu_{target,i}$$

Here, ν_i can be either a water surface elevation, measured on a specific cross-section (1-D) or element (2-D), or a water flow over a cross section (1-D) or a STRINGDEF (2-D). $\nu_{target,i}$ describes the target value, or in case of a *feed forward* controller, the equilibrium state. τ_i is a delay time controlling when the information of the measured variable is fed into the controller.

5.3.2.2   Manipulated Variable

A *manipulated variable* refers to a boundary condition and can be a weir height, a gate level or an outflow (in case of 1-D hydrographs as downstream boundary).

5.3.3   Controller Types

5.3.3.1   PID-Controller

One possible approach to describe the mathematical function $\boldsymbol{f}(.)$ is a PID (proportional-integral-derivative) controller. This type of controller relates a monitored variable to a manipulated variable by three additive controller elements:

$$u_i(t) = \underbrace{K_{p,ij}m_j(t)}_{P-Element} + \underbrace{\int_0^t K_{I,ij}m_j(t')dt'}_{I-Element} + \underbrace{K_{D,ij}\frac{d}{dt}m_j(t)}_{D-Element}$$

Internally, the PID-controller is implemented in its differential form (i.e. the change of is calculated in each time step). The three required variables are K_P, K_I and K_D. The correct definition of these variables is very crucial to the proper operation of the controller. Which values should be used is highly dependent on the system and therefore requires some care and experience.

The P-element represents an adjustment proportional to the deviation and therefore only limits the deviation, but does not bring the system back into the state where no deviation exists. For this reason, the I-Element integrates the deviation and consequently, the system can be forced into its equilibrium state. If the response of the I-Element is too strong compared to the P-Element, the system oscillates. If the values of both P and I elements are too small, the reaction of the system is very slow or even too weak to re-establish the

given targets. The D-Element depends on the change of the monitored variable and is used to quickly adapt the manipulated variables in case of a fast change.

More on the determination of correct PID coefficients can be found in the article by Fäh and Kühne (1987). Recommendations on how to choose the coefficients are given in the integrated help of the software.

6

References

Barth, T.J. (1994). Aspects of Unstructured Grids and Finite Volume Solvers for the Euler and Navier-Stokes Equations. von Karman Institute Lecture Series for Fluid Dynamics (1994-05).

Beffa, C. (2002). Integration ein- und zweidimensionaler Modelle zur hydrodynamischen Simulation von Gewässersystemen. *Int. Symposium Moderne Methoden und Konzepte im Wasserbau*, ETH Zürich.

Bern, M. and Eppstein, D. (1995). Mesh generation and optimal triangulation. *Computing in euclidean geometry*, Du and Hwang, F. eds., 47–123. *World Scientific*, Singapore.

Bern, M. and Plassmann, P. (2000). Mesh Generation. *Handbook of Computational Geometry*, Sack, J. and Urrutia, J. eds., *Elsevier Science*, Amsterdam, The Netherlands.

Chandra, R., Dagum, L., Kohr, D., Maydan, D., McDonald, J. and Menon, R. (2001). Parallel Programming in OpenMP. *Morgan Kaufmann Publishers Inc*, San Francisco, CA, USA.

Chapman, B., Jost, G. and van der Pas, R. (2008). Using OpenMP - Portable Shared Memory Parallel Programming. *The MIT Press*, Cambridge, Massachusetts.

Fäh, R. and Kühne, A. (1987). Numerische Simulation automatischer Stauregelungen bei Laufwasserkraftwerken. (p. 9. Heft 5/6, Ed.) *Wasser, Energie, Luft*, 79(5/6): 93–98.

Kuhn, B., Petersen, P. and O'Toole, E. (2000). OpenMP vs. threading in C/C++. *In European Workshop on OpenMP Concurrency: Pract. Exper*, No. 12: 1165–1176. *John Wiley & Sons, Ltd.*, New Jersey.

Manferdelli, J., Govindaraju, N. and Crall, C. (2008). Challenges and Opportunities in Many-Core Computing. *Proceedings of the IEEE.*, No. 96: 808–815.

Miglio, E., Perotto, S. and Saleri, F. (2005). Model coupling techniques for free surface flow problems: Part I. *Nonlinear Analysis: Theory, Methods & Applications*, 63: 1885–1896.

Online reference of openmp.

Sanders, B.F. (2008). Integration of a shallow water model with a local time step. *Journal of Hydraulic Research*, 46(4): 466–475.

REFERENCE
MANUAL

of BASEMENT

Contents

<div style="text-align: right">

1

</div>

Mathematical Models

1.1 Governing Flow Equations

1.1.1 Saint-Venant Equations

1.1.1.1 Introduction

The BASEchain module is based on the Saint Venant Equations (SVE) for unsteady one dimensional flow. The validity of these equations implies the following conditions and assumptions:

- Hydrostatic distribution of pressure: this is fulfilled if the streamline curvatures are small and the vertical accelerations are negligible
- Uniform velocity over the cross section and horizontal water surface across the section
- Small slope of the channel bottom, so that the cosine of the angle of the bottom with the horizontal can be assumed to be 1
- Steady-state resistance laws are applicable for unsteady flow.

The flow conditions at a channel cross section can be defined by two flow variables. Therefore, two of the three conservation laws are needed to analyze a flow situation. If the flow variables are not continuous, these must be the mass and the momentum conservation laws (Cunge et al., 1980).

1.1.1.2 Conservative Form of SVE

1.1.1.2.1 Mass Conservation

For the control volume illustrated in Figure 1.1, the conservation of mass is formulated assuming the mass density ρ is constant (incompressible flow). This leads basically to a

<div style="text-align: right">

3

</div>
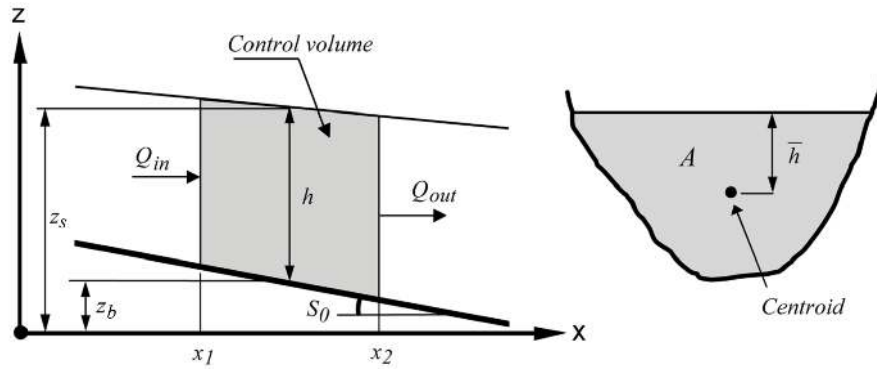
Figure 1.1 *Definition Sketch*

conservation of Volume. The temporal change in Volume equals the difference between inflowing and outflowing Volume (eq. 1.1).

$$\frac{\mathrm{d}}{\mathrm{d}t} \int\limits_{x_1}^{x_2} A \,\mathrm{d}x + Q_{out} - Q_{in} - q_l(x_2 - x_1) = 0 \tag{1.1}$$

where:

A	$[m^2]$	wetted cross section area
Q	$[m^3/s]$	discharge
q_l	$[m^2/s]$	lateral discharge per meter of length (specific discharge)
V	$[m^3]$	volume
x	$[m]$	distance
t	$[s]$	time

Applying Leibnitz's rule and integrating with the mean value theorem

$$\frac{\mathrm{d}}{\mathrm{d}t} \int\limits_{x_1}^{x_2} A \,\mathrm{d}x = \int\limits_{x_1}^{x_2} \frac{\partial A}{\partial t} \,\mathrm{d}x = \frac{\partial A}{\partial t}(x_2 - x_1)$$

and then dividing by (x_2 - x_1) and making use of $\dfrac{Q_{out} - Q_{in}}{(x_2 - x_1)} = \dfrac{\partial Q}{\partial x}$, we obtain the differential form of the continuity equation:

$$\frac{\partial A}{\partial t} + \frac{\partial Q}{\partial x} - q_l = 0 \tag{1.2}$$

1.1.1.2.2   Momentum Conservation

Newton's second law of motion says: The change in momentum equals to the Sum of all external Forces. The momentum is defined as

$$p = m\,u$$

$$\frac{\mathrm{d}p}{\mathrm{d}t} = m\,a = \sum F$$

where:

p	$[kg\ m/s]$	momentum
m	$[kg]$	mass
u	$[m/s]$	velocity
a	$[m/s^2]$	acceleration
F	$[N]$	force

Making use of the Reynolds transport theorem (Chaudhry, 1993) and referring to the control volume in Figure 1.1 one obtains a conservative formulation for the left part of the momentum equation

$$\frac{\mathrm{d}p}{\mathrm{d}t} = \sum F = \frac{\mathrm{d}}{\mathrm{d}t}\int_{x_1}^{x_2} u\rho A\,dx + u_2\rho A_2 u_2 - u_1\rho A_1 u_1 - u_x\rho q_l(x_2 - x_1) \tag{1.3}$$

where:

u_x	$[m/s]$	velocity in x direction (direction of flow) of lateral sources
ρ	$[kg/m^3]$	mass density

Further simplification is achieved by applying Leibnitz's rule and writing $Q = A\,u$ and $Q/A = u$ resulting in:

$$\sum F = \int_{x_1}^{x_2} \rho\frac{\partial Q}{\partial t}\,dx + \rho\frac{Q^2}{A}\bigg|_{out} - \rho\frac{Q^2}{A}\bigg|_{in} - u_x\rho q_l(x_2 - x_1) \tag{1.4}$$

Applying the mean value theorem $\displaystyle\int_{x_1}^{x_2} \rho\frac{\partial Q}{\partial t}\,dx = \frac{\partial Q}{\partial t}(x_2 - x_1)\rho$

and dividing both sides by $\rho(x_2 - x_1)$ and by using

$$\left(\frac{Q^2}{A}\bigg|_{out} - \frac{Q^2}{A}\bigg|_{in}\right)\frac{1}{(x_2 - x_1)} = \frac{\partial}{\partial x}\left(\frac{Q^2}{A}\right)$$

leads to the following formulation:

$$\frac{\sum F}{\rho(x_2 - x_1)} = \frac{\partial Q}{\partial t} + \frac{\partial}{\partial x}\left(\frac{Q^2}{A}\right) - q_l u_x \tag{1.5}$$

For the determination of $\sum F$ all external forces acting on the control volume have to be considered. These are:

- the pressure force upstream and downstream of the control volume: $F_1 = -\rho g A_1\overline{h_1}$ and $F_2 = \rho g A_2\overline{h_2}$ (hydrostatic pressure is $p = \rho gh$, the force is then $F = pA$)

- the weight of water (gravitational force) in x-direction: $F_3 = \rho g\displaystyle\int_{x_1}^{x_2} AS_B\,dx$

- and the frictional force: $F_4 = \rho g\displaystyle\int_{x_1}^{x_2} AS_f\,dx$

where

S_B	$[-]$	bottom slope
S_f	$[-]$	friction slope
g	$[m/s^2]$	gravity

All these forces are now put into the sum of eq. 1.5. For the pressure forces, we get directly a differential form as

$$\frac{\sum F}{\rho(x_2 - x_1)} = \frac{\rho g A_2 \overline{h_2} - \rho g A_1 \overline{h_1}}{\rho(x_2 - x_1)} = g \frac{\partial}{\partial x}(A\overline{h})$$

For the gravitational force and the friction force, the mean value theorem is applied:

$$\int_{x_1}^{x_2} A(S_B - S_f)\,\mathrm{d}x = A(S_B - S_f)(x_2 - x_1)$$

This results in the following momentum equation:

$$\frac{\partial Q}{\partial t} + \frac{\partial}{\partial x}\left(\frac{Q^2}{A}\right) - q_l u_x = -g\frac{\partial}{\partial x}(A\overline{h}) + gA(S_B - S_f) \tag{1.6}$$

There is still an unknown $\overline{h}$ on the right hand side which should possibly be eliminated. Based upon geometrical considerations and using the Leibniz rule, the pressure Term can be expressed as the following not obvious relation. This can be proven mathematically even under the consideration that changes in the channel width are not negligible.

$$-g\frac{\partial}{\partial x}(A\overline{h}) = -gA\frac{\partial h}{\partial x} \tag{1.7}$$

Now, the unknown water depth h can be eliminated using the transformation

$$h = z_S - z_B \quad \text{and} \quad \frac{\partial h}{\partial x} = \frac{\partial z_S}{\partial x} - \frac{\partial z_B}{\partial x} = \frac{\partial z_S}{\partial x} + S_B$$

Inserting this into eq. 1.7, resp. eq. 1.6 leads to a formulation of the momentum equation where we have a term with the gradient of the water surface elevation z_S combining the pressure forces and the gravitational force. Note that the bottom slope S_B vanished.

$$\frac{\partial Q}{\partial t} + \frac{\partial}{\partial x}\left(\frac{Q^2}{A}\right) + gA\frac{\partial z_S}{\partial x} + gAS_f - q_l u_x = 0 \tag{1.8}$$

If only the cross sectional area where the water actually flows (and therefore contributes to the momentum balance) shall be used, and by introducing a factor β accounting for the velocity distribution in the cross section (Cunge et al. (1980)), eq. 1.9 is obtained:

$$\frac{\partial Q}{\partial t} + \frac{\partial}{\partial x}\left(\beta\frac{Q^2}{A_{red}}\right) + gA_{red}\frac{\partial z_S}{\partial x} + gA_{red}S_f - q_l u_x = 0 \tag{1.9}$$

where A_{red} [m^2] is the reduced area, i.e. the part of the cross section area where water flows.

If Strickler values are used to define the friction

$$\beta = \frac{A \sum\limits_{i} k_{str_i}^2 h_i^{7/3} b_i}{\left(\sum\limits_{i} k_{str_i} h_i^{5/3} b_i \right)^2} \tag{1.10}$$

If the equivalent roughness height is used

$$\beta = \frac{A \sum\limits_{i} k_{s_i}^2 h_i^2 b_i}{\left(\sum\limits_{i} k_{s_i} h_i^{3/2} b_i \right)^2} \tag{1.11}$$

1.1.1.3 Source Terms

With the given formulation of the flow equation there are 4 source terms:

For the continuity equation:

- The lateral in- or outflow q_l

For the momentum equation:

- The bed slope

$$W = gA\frac{\partial z_S}{\partial x} \tag{1.12}$$

- The bottom friction:

$$Fr = gA_{red}S_f \tag{1.13}$$

- The influence of lateral in- or outflow:

$$q_l u_x \tag{1.14}$$

However, in BASEchain, the influence of the lateral inflow on the momentum equation is neglected.

1.1.1.4 Closure Conditions

1.1.1.4.1 Determination of Friction Slope

The relation between the friction slope S_f and the bottom shear stress is:

$$\frac{\tau_B}{\rho} = gRS_f \tag{1.15}$$

As the unit of τ_B/ρ is the square of a velocity, a shear stress velocity can be defined as:

$$u_* = \sqrt{\frac{\tau_B}{\rho}} \tag{1.16}$$

The velocity in the channel is proportional to the shear flow velocity and thus:

$$u = c_f\sqrt{gRS_f} \tag{1.17}$$

where c_f is the dimensionless Chézy coefficient. It is defined as $c_f = C/\sqrt{g}$, where C is the Chézy coefficient [m$^{1/2}$/s].

If u is replaced by Q/A:

$$\frac{Q}{A} = c_f\sqrt{gRS_f} \tag{1.18}$$

results, with

$$S_f = \frac{Q|Q|}{gA^2c_f^2R} \tag{1.19}$$

Introducing the conveyance K :

$$K = \frac{Q}{\sqrt{S_f}} = Ac_f\sqrt{Rg} \tag{1.20}$$

$$S_f = \frac{Q|Q|}{K^2} \tag{1.21}$$

The dimensionless friction coefficient c_f can be determined based on a power-law approach using Manning-Strickler friction coefficient k_{str} or based on log-law approach using equivalent sand roughness k_s of Nikuradse.

Power Friction Law

The power friction law according to Manning-Strickler is widely used in practice. Therfore channel roughness is defined using Strickler's k_{str} or Manning's n. For conversion a simple relation holds:

$$k_{str} = \frac{1}{n}$$

The dimensionless friction coefficient c_f is calculated as

$$c_f = \frac{k_{str}R^{1/6}}{\sqrt{g}} \tag{1.22}$$

Logarithmic Friction Laws

The following approaches are implemented to determine the coefficient c_f:

Chézy:

$$c_f = 5.75 \log \left(12 \frac{R}{k_s} \right) \tag{1.23}$$

Yalin:

$$c_f = \frac{1}{\kappa} \ln \left(11 \frac{R}{k_s} \right) \tag{1.24}$$

Bezzola:

This approach uses the roughness sublayer y_R as relevant roughness height. Usually for rivers $y_R \approx 1.0 d_{90}$ is a good approximation. This approach takes small relative roughness heights into account Bezzola (2002).

$$
\begin{aligned}
c_f &= 2.5 \sqrt{1 - \frac{y_R}{R}} \ \ln \left(10.9 \frac{R}{y_R} \right), & \text{for} \quad \frac{R}{y_R} &> 2 \\
c_f &= 1.25 \sqrt{\frac{R}{y_R}} \ \ln \left(10.9 \frac{R}{y_R} \right), & \text{for} \quad \frac{R}{y_R} &\leq 2 \\
c_f &= 1.5, & \text{for} \quad \frac{R}{y_R} &< 0.5
\end{aligned}
\tag{1.25}
$$

Darcy-Weissbach:

$$c_f = \sqrt{\frac{8}{f}} \quad \text{with} \quad f = \frac{0.24}{\log \left(\frac{12R}{k_s} \right)} \tag{1.26}$$

In the case where friction is determined based on the bed composition of the mobile bed, friction can be determined based on the local characteristic grain size d_{90}:

$$k_{str} = \frac{\text{factor}}{\sqrt[6]{d_{90}}} \quad \text{default value of factor} = 21.1 \tag{1.27}$$

or

$$k_s = \text{factor} \cdot d_{90} \quad \text{default value of factor} = 3 \tag{1.28}$$

The default values of the factors above correspond to a natural river bed with well graded bed material and exposed coarse components.

1.1.1.5 Boundary Conditions

At the upper and lower end of the channel it is necessary to know the influence of the region outside on the flow within the computational domain. The influenced area depends on the propagation velocity of a perturbation. The propagation velocity in standing water is

$$c = \sqrt{gh} \tag{1.29}$$

If a one dimensional flow is considered this propagation takes place in two directions: upstream $(-c)$ and downstream $(+c)$. These velocities must then be added to the flow velocities in the channel, giving the upstream (C_-) and downstream (C_+) characteristics:

$$C_- = \frac{\mathrm{d}x}{\mathrm{d}t} = u - c \tag{1.30}$$

$$C_+ = \frac{\mathrm{d}x}{\mathrm{d}t} = u + c \tag{1.31}$$

With these functions it is possible to determine which region is influenced by a perturbation and which region influences a given point after a given time.

In particular it can be said that if $c < u$ the information will not be able to spread in upstream direction, thus the condition in a point cannot influence any upstream point, and a point cannot receive any information from downstream. This is the case for a supercritical flow.

In contrast, if $c > u$, which is the case for sub-critical flow, the information spreads in both directions, upstream and downstream. This fact substantiates the necessity and usefulness of information at the boundaries. As there are two equations to solve, two variables are needed for the solution.

If the flow conditions are sub-critical, the flow is influenced from downstream. Thus at the inflow boundary one condition can be taken from the flow region itself and only one additional boundary condition is needed. At the outflow boundary, the flow is influenced from outside and so one boundary condition is needed.

If the flow is supercritical, no information arrives from downstream. Therefore, two boundary conditions are needed at the inflow end. In contrast, as it cannot influence the flow within the computational domain, it is not useful to have a boundary condition at the downstream end.

Table 1.1 *Number of needed boundary conditions*

Flow type	Inflow	Outflow
Sub critical flow ($Fr < 1$)	1	1
Supercritical flow ($Fr > 1$)	2	0

At the inflow boundary the given value is usually Q. If the flow is supercritical the second variable A is determined by a flow resistance law (slope is needed!).

At the outflow boundary there are several possibilities to provide the necessary information at the boundary:

- determine an out flowing discharge by a weir or a gate;
- set the water surface elevation as a function of time;
- set the water surface elevation as a function of the discharge (rating curve).

1.1.2 Shallow Water Equations

1.1.2.1 Introduction

Mathematical models of the so-called *shallow water* type govern a wide variety of physical phenomena. For reasons of simplicity, the shallow water equations will from here on be abbreviated as SWE. An important class of problems of practical interest involves water flows with a free surface under the influence of gravity. It includes:

- Tides in oceans
- Flood waves in rivers
- Dam break waves

The validity of the SWE implies the following conditions and assumptions:

- Hydrostatic distribution of pressure: this is fulfilled if the vertical accelerations are negligible.
- Small slope of the channel bottom, so that the cosine of the angle between the bottom and the horizontal can be assumed to be 1.
- Steady-state resistance laws are applicable for unsteady flow.

A key assumption made in derivation of the approximate shallow water theory concerns the first aspect, the hydrostatic pressure distribution. Supposing that the vertical velocity acceleration of water particles is negligible, a hydrostatic pressure distribution can be assumed. This eventually allows for integration over the flow depth, which results in a non-linear initial value problem, namely the shallow water equations. They form a time-dependent two-dimensional system of non-linear partial differential equations of hyperbolic type.

There are two approaches for the derivation of shallow water equations:

- Integrating the three-dimensional system of Navier-Stokes equations over flow depth
- Direct approach by considering a three-dimensional control volume

In the following the derivation of the depth integrated mass and momentum conservation equations from the Reynolds-averaged 3-D Navier-Stokes equation is briefly presented. Following boundary conditions are imposed:

1. At the top of water surface:

   Kinematic boundary condition (this condition describes that no flow across the water surface can take place):

   $$w_s = \frac{\partial z_S}{\partial t} + u_s \frac{\partial z_S}{\partial x} + v_s \frac{\partial z_S}{\partial y} \qquad (1.32)$$

   Dynamic boundary condition:

   $$\tau = (\tau_{Sx}, \tau_{Sy}) \quad \text{and} \quad \mathrm{P} = \mathrm{P}_{atm} \qquad (1.33)$$

2. At the bottom of water body:

   Kinematic boundary condition (this condition describes that no flow through the bed surface can take place):

$$w_B = u_B \, \frac{\partial z_B}{\partial x} + v_B \, \frac{\partial z_B}{\partial y} \tag{1.34}$$

Derivation of mass conservation

The derivation makes use of Leibniz's integration rule, which is used here to remove the partial derivatives from the integral. It can be written generally as

$$\int\limits_{z_B(x,y)}^{z_S(x,y)} \frac{\partial f(x,y,z)}{\partial x} \, dz = \frac{\partial}{\partial x} \int\limits_{z_B(x,y)}^{z_S(x,y)} f(x,y,z) \, dz + f(x,y,z_B) \, \frac{\partial z_B}{\partial x} - f(x,y,z_S) \, \frac{\partial z_S}{\partial x}$$

The 3-D Reynolds-averaged mass conservation equation is integrated over the flow depth from the bed bottom z_B to the water surface z_S.

$$\int\limits_{z_B}^{z_S} \left[\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} \right] \, dz = 0 \tag{1.35}$$

Applying Leibniz's rule on the first term, the velocity gradient in x-direction, leads to

$$\int\limits_{z_B}^{z_S} \frac{\partial u}{\partial x} \, dz = \frac{\partial}{\partial x} \int\limits_{z_B}^{z_S} u \, dz + u_B \frac{\partial z_B}{\partial x} - v_s \frac{\partial z_S}{\partial x}$$

whereas u_B, u_S and v_B, v_S are the velocities at the bottom and at the surface in x- and y-directions respectively. The second term of eq. 1.35 is treated analogous.

The third term can be evaluated exactly by applying the fundamental theorem of calculus. It results in the difference of the vertical velocity at the surface and the bottom:

$$\int\limits_{z_B}^{z_S} \frac{\partial w}{\partial z} \, dz = w_S - w_B$$

Assembling these terms, one can identify and eliminate the kinematic boundary conditions as stated above.

$$\frac{\partial}{\partial x} \int\limits_{z_B}^{z_S} u \, dz + \frac{\partial}{\partial y} \int\limits_{z_B}^{z_S} v \, dz + \underbrace{u_B \frac{\partial z_B}{\partial x} + v_B \frac{\partial z_B}{\partial y} - w_B}_{= 0} - \underbrace{u_S \frac{\partial z_S}{\partial x} - v_S \frac{\partial z_S}{\partial y} + w_S}_{= \frac{\partial z_S}{\partial t} = \frac{\partial h}{\partial t}} = 0$$

Evaluating the integrals over the depth as

$$\int\limits_{z_B}^{z_S} u \, dz = \overline{u}h$$

finally leads to the depth integrated formulation of mass conservation:

$$\frac{\partial h}{\partial t} + \frac{\partial(\overline{u}h)}{\partial x} + \frac{\partial(\overline{u}h)}{\partial y} = 0$$

Derivation of momentum conservation

The Reynolds-averaged 3-D momentum equation in x-direction of the Navier-Stokes equation is integrated over the depth

$$\int\limits_{z_B}^{z_S} \left[\frac{\partial u}{\partial t} + \frac{\partial u^2}{\partial x} + \frac{\partial uv}{\partial y} + \frac{\partial uw}{\partial z} \right] \, dz = \int\limits_{z_B}^{z_S} \left[-\frac{1}{\rho}\frac{\partial p}{\partial x} + \frac{1}{\rho}\frac{\partial \tau_{xx}}{\partial x} + \frac{1}{\rho}\frac{\partial \tau_{xy}}{\partial y} + \frac{1}{\rho}\frac{\partial \tau_{xz}}{\partial y} \right] \, dz \quad (1.36)$$

whereas the momentum equation in y-direction can be treated in an analogous manner.

The first term, the time derivative of the x-velocity, is transformed with Leibniz's rule and integrated over the depth.

$$\int\limits_{z_B}^{z_S} \frac{\partial u}{\partial t} \, dz = \frac{\partial}{\partial t}\int\limits_{z_B}^{z_S} u \, dz + u_B\frac{\partial z_B}{\partial t} - u_S\frac{\partial z_S}{\partial t} = \frac{\partial \overline{u}h}{\partial t} + u_B\frac{\partial z_B}{\partial t} - u_S\frac{\partial z_S}{\partial t}$$

The Leibniz rule is also applied on the advective terms as follows:

$$\int\limits_{z_B}^{z_S} \frac{\partial u^2}{\partial x} \, dz = \frac{\partial}{\partial x}\int\limits_{z_B}^{z_S} u^2 \, dz + u_B^2\frac{\partial z_B}{\partial x} - u_S^2\frac{\partial z_S}{\partial x} \quad , \quad \int\limits_{z_B}^{z_S} \frac{\partial uv}{\partial y} \, dz = \frac{\partial}{\partial y}\int\limits_{z_B}^{z_S} uv \, dz + u_Bv_B\frac{\partial z_B}{\partial y} - u_Sv_S\frac{\partial z_S}{\partial y}$$

And the fundamental theorem of calculus allows the evaluation of the fourth term on the left hand side.

$$\int\limits_{z_B}^{z_S} \frac{\partial uw}{\partial z} dz = u_Sw_S - u_Bw_B$$

All terms of the left hand side of eq. 1.36 now can be assembled, and again, the kinematic boundary conditions are identified and can be eliminated. The left hand side is reduced to three remaining terms.

$$\overline{u}h + \frac{\partial}{\partial x}\int\limits_{z_B}^{z_S} u^2 \, dz + \frac{\partial}{\partial y}\int\limits_{z_B}^{z_S} uv \, dz + u_B\underbrace{\left(\frac{\partial z_B}{\partial t} + u_B\frac{\partial z_B}{\partial x} + v_B\frac{\partial z_B}{\partial x} - w_B\right)}_{=\,0} -$$

$$u_S \underbrace{\left(\frac{\partial z_S}{\partial t} + u_s \frac{\partial z_S}{\partial x} + v_S \frac{\partial z_S}{\partial y} + w_S \right)}_{= 0}$$

With the assumption of a hydrostatic pressure distribution, the pressure term on the right hand side can be evaluated. Furthermore, the water surface elevation is replaced by the bottom elevation and the depth.

$$\frac{1}{\rho} \int\limits_{z_B}^{z_S} \frac{\partial p}{\partial x} \, dz = gh \frac{\partial z_S}{\partial x} = gh \left(\frac{\partial z_B}{\partial x} + \frac{\partial h}{\partial x} \right)$$

The depth integration of the first two shear stresses of the right hand side yields the depth integrated viscous and turbulent stresses, which require additional closure conditions to be evaluated.

$$\frac{1}{\rho} \int\limits_{z_B}^{z_S} \frac{\partial \tau_{xx}}{\partial x} \, dz + \frac{1}{\rho} \int\limits_{z_B}^{z_S} \frac{\partial \tau_{yx}}{\partial y} \, dz = \frac{1}{\rho} \frac{\partial \overline{\tau}_{xx} h}{\partial x} + \frac{1}{\rho} \frac{\partial \overline{\tau}_{xy} h}{\partial y}$$

The third shear stress term can be integrated over the depth and introduces the bottom and surface shear stresses at the domain boundaries, which again need additional closure conditions. The surface shear stresses τ_{Bx}, e.g. due to wind flow over the water surface, are neglected from here on.

$$\frac{1}{\rho} \int\limits_{z_B}^{z_S} \frac{\partial \tau_{zx}}{\partial z} \, dz = \frac{1}{\rho} (\tau_{Sx} - \tau_{Bx})$$

Putting the terms together one obtains

$$\frac{\partial \overline{u} h}{\partial t} + \frac{\partial}{\partial x} \int\limits_{z_B}^{z_S} u^2 \, dz + \frac{\partial}{\partial y} \int\limits_{z_B}^{z_S} uv \, dz + gh \frac{\partial h}{\partial x} = -gh \frac{\partial z_B}{\partial x} - \frac{1}{\rho} \tau_{Bx} + \frac{1}{\rho} \frac{\partial \overline{\tau}_{xx} h}{\partial x} + \frac{1}{\rho} \frac{\partial \overline{\tau}_{xy} h}{\partial y}$$

The depth integrals of the advective terms still need to be solved. By dividing the velocities in a mean velocity $\overline{u}$ and a deviation from the mean u', similar to the Reynolds averaging procedure, the advective terms can be evaluated as follows. The depth integration introduces new dispersion terms, which describe the effects of the non-uniformity of the velocity distribution.

$$u = \overline{u} + u' \Rightarrow \frac{\partial}{\partial x} \int\limits_{z_B}^{z_S} u^2 \, dz = \frac{\partial \overline{u}^2 h}{\partial x} + \frac{\partial \overline{u'u'} h}{\partial x} = \frac{\partial \overline{u}^2 h}{\partial x} + \frac{1}{\rho} \frac{\partial D_{xx} h}{\partial x}$$

In the end, after dividing the equations by the water depth, the depth integrated x-momentum equation of the SWE is obtained in the following formulation:

$$\frac{\partial \overline{u}}{\partial t} + \overline{u} \frac{\partial \overline{u}}{\partial x} + \overline{v} \frac{\partial \overline{u}}{\partial y} + g \frac{\partial h}{\partial x} = -g \frac{\partial z_B}{\partial x} - \frac{1}{\rho h} \tau_{Bx} + \frac{1}{\rho h} \frac{\partial [h(\overline{\tau}_{xx} + D_{xx})]}{\partial x} + \frac{1}{\rho h} \frac{\partial [h(\overline{\tau}_{xy} + D_{yx})]}{\partial y}$$

Shallow Water Equations

Conclusive, as shown before, the complete set of SWE is derived in the form:

$$\frac{\partial h}{\partial t} + \frac{\partial(\overline{u}h)}{\partial x} + \frac{\partial(\overline{v}h)}{\partial y} = 0 \tag{1.37}$$

$$\frac{\partial \overline{u}}{\partial t} + \overline{u}\frac{\partial \overline{u}}{\partial x} + \overline{v}\frac{\partial \overline{u}}{\partial y} + g\frac{\partial h}{\partial x} = -g\frac{\partial z_B}{\partial x} - \frac{1}{\rho h}\tau_{Bx} + \frac{1}{\rho h}\frac{\partial[h(\overline{\tau}_{xx} + D_{xx})]}{\partial x} + \frac{1}{\rho h}\frac{\partial[h(\overline{\tau}_{xy} + D_{yx})]}{\partial y} \tag{1.38}$$

$$\frac{\partial \overline{v}}{\partial t} + \overline{u}\frac{\partial \overline{v}}{\partial x} + \overline{v}\frac{\partial \overline{v}}{\partial y} + g\frac{\partial h}{\partial y} = -g\frac{\partial z_B}{\partial y} - \frac{1}{\rho h}\tau_{By} + \frac{1}{\rho h}\frac{\partial[h(\overline{\tau}_{yx} + D_{yx})]}{\partial x} + \frac{1}{\rho h}\frac{\partial[h(\overline{\tau}_{yy} + D_{yy})]}{\partial y} \tag{1.39}$$

where:

h	$[m]$	water depth
g	$[m/s^2]$	gravity acceleration
P	$[Pa]$	pressure
$\overline{u}$	$[m/s]$	depth averaged velocity in x direction
u_S	$[m/s]$	velocity in x direction at water surface
u_B	$[m/s]$	velocity in x direction at bottom (usually equal zero)
$\overline{v}$	$[m/s]$	depth averaged velocity in y direction
v_S	$[m/s]$	velocity in y direction at water surface
v_B	$[m/s]$	velocity in y direction at bottom (usually equal zero)
w_S	$[m/s]$	velocity in z direction at water surface
w_B	$[m/s]$	velocity in z direction at bottom (usually equal zero)
z_B	$[m]$	bottom elevation
z_S	$[m]$	water surface elevation
τ_{Sx}, τ_{Sy}	$[N/m^2]$	surface shear stress in x- and y direction (here neglected)
τ_{Bx}, τ_{By}	$[N/m^2]$	bed shear stress in x- and y direction
$\overline{\tau}_{xx}, \overline{\tau}_{xy}, \overline{\tau}_{yx}, \overline{\tau}_{yy}$	$[N/m^2]$	depth averaged viscous and turbulent stresses
$D_{xx}, D_{xy}, D_{yx}, D_{yy}$	$[N/m^2]$	momentum dispersion terms

For brevity, the over bars indicating depth averaged values will be dropped from here on.

1.1.2.2 Closure Conditions

1.1.2.2.1 Turbulence

The turbulent and viscous shear stresses can be quantified in accordance with the Boussinesq eddy viscosity concept, which can be expressed as

$$\tau_{xx} = 2\rho v\frac{\partial u}{\partial x} \quad , \quad \tau_{yy} = 2\rho v\frac{\partial v}{\partial y} \quad , \quad \tau_{xy} = \rho v\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right) \tag{1.40}$$

If the flow is dominated by the friction forces, the total viscosity is the sum of the eddy viscosity (quantity due to turbulence modelling) and molecular viscosity (kinematic viscosity of the fluid): $v = v_t + v_m$.

Turbulent eddy viscosity may be dynamically calculated as $v_t = \kappa u_* h/6$ with the friction velocity $u_* = \sqrt{\tau_B/\rho}$.

The molecular viscosity is a physical property of the fluid and is constant due to the assumption of an isothermal fluid.

1.1.2.2.2 Bed Shear Stress

The bed shear stresses are related to the depth–averaged velocities by the quadratic friction law

$$\tau_{Bx} = \rho \frac{|\boldsymbol{u}|u}{c_f^2} \quad ; \quad \tau_{By} = \rho \frac{|\boldsymbol{u}|v}{c_f^2} \tag{1.41}$$

in which $|\boldsymbol{u}| = \sqrt{u^2 + v^2}$ is the magnitude of the velocity vector. The friction coefficient c_f can be determined by any friction law.

1.1.2.2.3 Momentum Dispersion

The momentum dispersion terms account for the dispersion of momentum transport due to the vertical non-uniformity of flow velocities.

At the moment the momentum dispersion terms are not explicitly modelled here. Usually, in straight channels, these dispersive effects can be accounted for by adapting the turbulent viscosity in the determination of the turbulent stresses (Wu, 2007).

1.1.2.3 Conservative Form of SWE

Various forms of SWE can be distinguished with their primitive variables. The proper choice of these variables and the corresponding set of equations plays an extremely important role in numerical modelling. It is well known that the conservative form is preferred over the non-conservative one if strong changes or discontinuities in a solution are to be expected. In flooding and dam break problems, this is usually the case.

Bechteler et al. (1993) showed that the equation sets in conservative form, with (h, uh, vh) as independent and primitive variables produce best results. The conservative form can be derived by multiplying the continuity equation with u and v and adding to momentum equations in x and y direction respectively. This set of equations can be written in the following form:

$$\boldsymbol{U}_t + \nabla(\boldsymbol{F}, \boldsymbol{G}) + \boldsymbol{S} = 0 \tag{1.42}$$

where $U, F(U), G(U)$ and S are the vectors of conserved variables, fluxes in the x and y directions and sources respectively, given by:

$$\boldsymbol{U} = \begin{pmatrix} h \\ uh \\ \nu h \end{pmatrix} \tag{1.43}$$

$$\boldsymbol{F} = \begin{pmatrix} uh \\ u^2 h + \dfrac{1}{2} g h^2 - \nu h \dfrac{\partial u}{\partial x} \\ u \nu h - \nu h \dfrac{\partial u}{\partial y} \end{pmatrix} \quad ; \quad \boldsymbol{G} = \begin{pmatrix} \nu h \\ u \nu h - \nu h \dfrac{\partial \nu}{\partial x} \\ \nu^2 h + \dfrac{1}{2} g h^2 - \nu h \dfrac{\partial \nu}{\partial y} \end{pmatrix} \tag{1.44}$$

$$\boldsymbol{S} = \begin{pmatrix} 0 \\ gh(S_{fx} - S_{Bx}) \\ gh(S_{fy} - S_{By}) \end{pmatrix} \tag{1.45}$$

where ν is the total viscosity.

1.1.2.4   Source Terms

The eq. 1.45 has two source terms: the bed shear stress $(\tau_B/\rho = ghS_f)$ and the bed slope term (ghS_B). The viscous stresses in the flux of eq. 1.44 are also treated as flux term.

The bed shear stress is the most important physical parameter besides water depth and velocity field of a hydro- and morphodynamic model. It causes the turbulence and is responsible for sediment transport and has a non-linear effect of retarding the flow. When the effect of turbulence grows, the effect of molecular viscosity becomes relatively smaller, while viscous boundary layer near a solid boundary becomes thinner and may even appear not to exist. It means that the bed stress (friction) is equal to the bed turbulent stress.

However, bed stress is usually estimated by using an empirical or semi-empirical formula since the vertical distribution of velocity cannot be readily obtained.

In the one dimensional system of equations the term bed stress can be expressed as gRS_f, where S_f denotes the energy slope. Assuming that the frictional force in a two dimensional unsteady open flow can be estimated by referring to the formulas for one dimensional flows in open channel it can be written:

$$S_f = \frac{u|u|}{g c_f^2 R} \tag{1.46}$$

where u = velocity; c_f = friction coefficient; R = hydraulic radius. It can be easily seen that the above formula can be approximately generalized to the two dimensional system. In one-dimensional flows it is not distinguished between bottom and lateral (side wall) friction, while in two dimensional flows it is often taken a unit width channel $(R = h)$. For the two dimensional system the eq. 1.46 has the following forms

$$S_{fx} = \frac{u\sqrt{u^2 + v^2}}{g c_f^2 R} \quad ; \quad S_{fy} = \frac{v\sqrt{u^2 + v^2}}{g c_f^2 R} \tag{1.47}$$

The coefficient c_f can be calculated by different empirical approaches as in the one dimensional system, e.g. using the Manning or Strickler coefficients. See Section 1.1.1.4 for

further details on the determination of the friction coefficient c_f . If the friction value is calculated from the bed composition, the d_{90} diameter of the mixture is determined. In case of single grain simulations the d_{90} diameter is estimated by $d_{90} = 3 \cdot d_{mean}$.

The bed stress terms need additional closures equation (see Section 1.1.2.2.2) to be determined. Bed slope terms represent the gravity forces

$$S_{B,x} = -\partial z_B/\partial x \quad ; \quad S_{B,y} = -\partial z_B/\partial y \qquad (1.48)$$

The viscous fluxes are treated as source terms (The superscript 'd' refers to the diffusion)

$$\boldsymbol{S}_d = \frac{\partial \boldsymbol{F}^d}{\partial x} + \frac{\partial \boldsymbol{G}^d}{\partial y}$$

where the diffusive fluxes read

$$\boldsymbol{F}^d = \begin{pmatrix} 0 \\ \nu h \dfrac{\partial u}{\partial x} \\ \nu h \dfrac{\partial \nu}{\partial x} \end{pmatrix}$$

,

$$\boldsymbol{G}^d = \begin{pmatrix} 0 \\ \nu h \dfrac{\partial u}{\partial y} \\ \nu h \dfrac{\partial \nu}{\partial y} \end{pmatrix}$$

The total viscosity ν is the sum of the kinematic viscosity and turbulent eddy viscosity. The kinematic viscosity is a physical property of the fluid and is set to a constant value.

The turbulent eddy viscosity can either be set to a constant value or calculated dynamically as Turbulent eddy viscosity may be dynamically calculated as $\nu_t = \kappa u_* h/6$ with the friction velocity $u_* = \sqrt{\tau_B/\rho}$.

1.1.2.5 Boundary Conditions

SWE provide a model to describe dynamic fluid processes of various natural phenomena and find therefore widespread application in science and engineering. Solving SWE needs the appropriate boundary conditions like any other partial differential equations. In particular, the issue of which kind of boundary conditions are allowed is still not completely understood (Agoshkov et al., 1994). However several sets of boundary conditions of physical interest that are admissible from the mathematical viewpoint will be discussed here.

The physical boundaries can be divided into two sets: one *closed* (Γ_c), the other *open* (Γ_o) (Figure 1.2). The former generally expresses that no mass can flow through the boundary. The latter is an imaginary fluid-fluid boundary and includes two different inflow and outflow types.
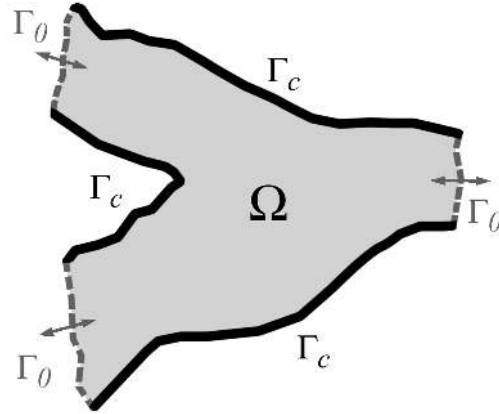
Figure 1.2 *Computational Domain and Boundaries*

1.1.2.5.1   Closed Boundary

The following relations are often described on the closed boundary, say Γ_c :

$$\rho \boldsymbol{u} \cdot \boldsymbol{n} = 0 \quad \frac{\partial \boldsymbol{u}}{\partial \boldsymbol{n}} = 0 \tag{1.49}$$

where

$\boldsymbol{n}$	$[m]$	the normal (directed outward) unit vector on Γ_c
$\boldsymbol{u}$	$[m/s]$	velocity vector $= (u, v)$

1.1.2.5.2   Open Boundary

The number of boundaries of a partial differential equations system depends on the type of the system. From the mathematical point of view, the SWE establish a quasi-linear hyperbolic differential equations system. If the temporal derivatives vanish, the system is elliptical for $F_r \leq 1.0$ and hyperbolical for $F_r \geq 1.0$, where F_r is Froude number.

On the open boundary (Γ_o) the two types inflow and outflow can be respectively distinguished as follows:

$$\Gamma_{in} = (x \in \Gamma_o; \boldsymbol{u} \cdot \boldsymbol{n} < 0) \tag{1.50}$$

$$\Gamma_{out} = (x \in \Gamma_o; \boldsymbol{u} \cdot \boldsymbol{n} > 0) \tag{1.51}$$

Based on the behaviour of the system of equations, the theoretical number of open boundary conditions is listed in Table 1.2 (Agoshkov et al., 1994) (Beffa, 1994):

Table 1.2 *The Correct Number of Boundary Conditions in SWE*

Flow type	Inflow	Outflow
Sub critical flow ($Fr < 1$)	2	1
Supercritical flow ($Fr > 1$)	3	0

However in practical application of boundary conditions, the number of the implemented conditions is often higher or lower than the theoretical criteria (Nujić, 1998).

1.2 Sediment and Pollutant Transport

1.2.1 Fundamentals of Sediment Motion

1.2.1.1 Threshold Condition for Sediment Transport

1.2.1.1.1 Determination of the Critical Shear Stress

The critical shear stress $\tau_{B_{cr}} = \theta_{cr}(\rho_s - \rho)gd_g$ is the threshold for incipient motion of grain class g where the critical Shields parameter θ_{cr} is a function of the shear Reynolds number Re^* (Shields, 1936). The critical Shields parameter θ_{cr} can be set to a constant value, e.g. Meyer-Peter and Müller (1948) proposed a constant Shields parameter of 0.047 for fully turbulent flow ($Re^* > 10^3$). Furthermore, θ_{cr} can be dynamically determined from a transformed Shields diagram as a function of the dimensionless grain diameter D^* ($\theta_{cr} = f(D^*)$) (Figure 1.3).

An approximation of the original Shields diagram was proposed by van Rijn (1984a):

$$
\begin{array}{lll}
\theta_{cr} = 0.24(D^*)^{-1} & for & 1 \le D^* \le 4 \\
\theta_{cr} = 0.14(D^*)^{-0.64} & for & 4 < D^* \le 10 \\
\theta_{cr} = 0.04(D^*)^{-0.1} & for & 10 < D^* \le 20 \\
\theta_{cr} = 0.013(D^*)^{0.29} & for & 20 < D^* \le 150 \\
\theta_{cr} = 0.055 & for & D^* > 150
\end{array}
\tag{1.52}
$$

where the dimensionless grain diameter D^* is defined as

$$
D^* = \left(\frac{(\rho_s - \rho)g}{\rho v^2} \right)^{1/3} d
\tag{1.53}
$$

Another explicit formulation of the Shields curve was proposed by Yalin and Silva (2001):

$$
\theta_{cr} = 0.13D^{*-0.392} \exp(-0.015D^*) + 0.045 \left(1 - \exp(-0.068D^*)\right)
\tag{1.54}
$$

1.2.1.1.2 Influence of Local Slope on Incipient Motion

The investigations on incipient motion by Shields were made for almost horizontal bed. In the case of sloped bed in flow direction or transverse to it, the stability of grains can either be reduced or increased due to gravity. One approach to consider the effects of local slopes on the threshold for incipient motion is to correct the critical shear stresses for incipient motion. Here k_β and k_γ are correction factors for slope in flow direction and in transversal direction and $\tau_{B_{cr},Shields}$ is the critical shear stress for almost horizontal bed as derived by Shields. The corrected critical shear stress then is determined as

$$
\tau_{B_{cr}} = k_\beta k_\gamma \tau_{B_{cr},Shields}
\tag{1.55}
$$
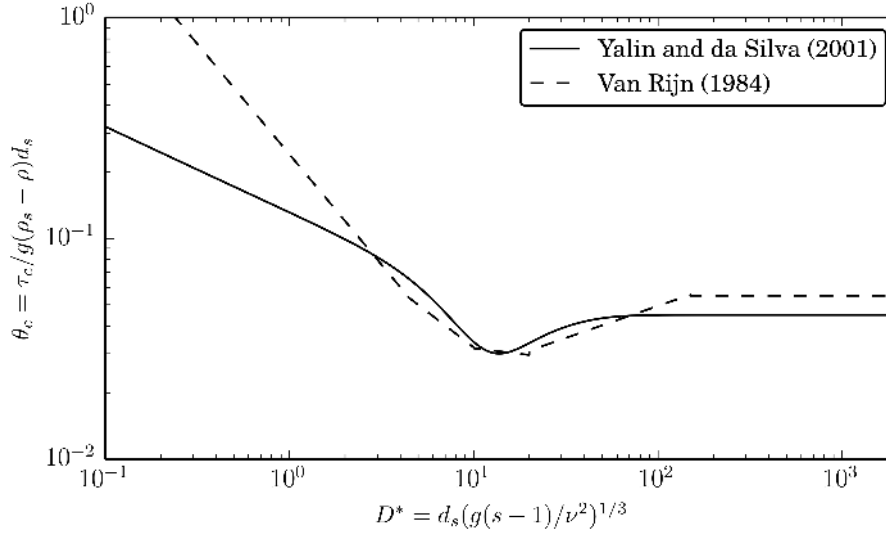
Figure 1.3 *Transformed Shields diagram to determine critical Shields parameter*

The correction factors are calculated as suggested by van Rijn (1989):

$$
k_\beta = \begin{cases} \dfrac{sin(\gamma - \beta)}{sin\gamma} & \text{if slope} < 0 \\[2mm] \dfrac{sin(\gamma + \beta)}{sin\gamma} & \text{if slope} > 0 \end{cases}
$$

$$
k_\gamma = \begin{cases} cos\delta\sqrt{1 - \dfrac{tan^2\delta}{tan^2\gamma}} \end{cases}
$$

$$(1.56)$$

where β is the angle between the horizontal and the bed in flow direction, δ is the slope angle transversal to the flow direction and γ is the angle of repose of the sediment material.

Furthermore, Chen et al. (2010) proposed a correction $k = k_\beta k_\gamma$ as follows

$$
k = \frac{\tau_{B_{cr}}}{\tau_{B_{cr},Shields}} =
$$

$$
\frac{1}{\tan\gamma}\left(\cos^2\left(\frac{\pi}{2} - \beta\right) - 1 + \frac{1}{(1 + \tan^2\beta + \tan^2\gamma)} + \frac{\tan^2\gamma}{(1 + \tan^2\beta + \tan^2\gamma)}\right)^{0.5} +
$$

$$
\cos\left(\frac{\pi}{2} - \beta\right)
$$

$$(1.57)$$

1.2.1.2   Influence of Bed Forms on Bottom Shear Stress

In presence of bed forms, like ripples, sand dunes or gravel banks, additional friction losses can occur due to complex flow conditions around these bed forms and the formation of turbulent eddies. In such cases the dimensionless bottom shear stress θ determined from

the present flow conditions can differ from the effective dimensionless bottom shear stress θ', which is relevant for the transport of the sediment particles. It is usually assumed that the determination of the effective shear stress should be based upon the grain friction losses only and should exclude additional form losses, to prevent too large sediment transport rates. Therefore a reduction factor μ is introduced for the determination of the effective bottom shear stress θ' from the bottom shear stress θ as

$$\mu = \frac{\tau_B'}{\tau_B} = \frac{\theta'}{\theta} \quad \text{with} \quad \left\{ \begin{array}{ll} \mu = 1.0 & \text{(no bed forms)} \\ \mu < 1.0 & \text{(bed forms)} \end{array} \right. \tag{1.58}$$

This reduction factor (also called "ripple factor") can be given a constant value if the bed forms are distributed uniformly over the simulation domain. After Jäggi this factor should be set between 0.8 and 0.85. If there are no bed forms present one can consider that $\theta' = \theta$, i.e. $\mu=1.0$. Generally can be said, the larger the form resistance, the smaller becomes the reduction factor μ.

Another approach is to calculate the reduction factor by introducing a reduced energy slope J', compared to the energy slope J, due to the presence of the bed forms as done by Meyer-Peter and Müller. This approach is in particular suitable if ripples are present at the river bed and finally leads to the following estimation of the reduction factor.

$$\mu = \left(\frac{k_{str}}{k_{str}'} \right)^{3/2} \tag{1.59}$$

Here, k_{str}' corresponds to the definition of the Strickler coefficient for experiments with Nikuradse-roughness (Jäggi, 1995) and can be calculated from the grain sizes using the d90 diameter as detailed in section Section 1.1.1.4. k_{str} is the calibrated Strickler coefficient used in the hydraulic calculations which includes the form friction effects.

1.2.1.3 Bed Armouring

In morphological simulations with fractional sediment transport the forming or destroying of bed armouring layers can be simulated by modelling sorting effects without special features.

But there are also some types of protection layers which need a special treatment, like e.g. grass layers or geotextiles. Furthermore, for single grain simulations the bed material sorting effects cannot be captured and therefore a special treatment is needed if effects of bed armouring shall be considered.

The effects of such a protection layer can be considered using two methods:

- A critical shear stress $\tau_{cr,start}$ of the protection layer can be specified, which must be exceeded at least once before erosion of the substrate can take place. This method is suited for simulations with one or multiple grain classes. Start of erosion: $\tau_B > \tau_{cr,start}$

- Another approach is to define the d90 grain diameter of the bed armouring layer. The dimensionless critical shear stress $\theta_{cr,armour}$ of this bed armour is then estimated as

$$\theta_{cr,armour} = \theta_{cr} \left(\frac{d_{90}}{d} \right)^{2/3}$$

where d_{90} is the specified d90 grain diameter of the bed armour and d and θ_{cr} are the diameter and critical shear stress of the substrate. But be aware that in case the bed armour is eroded once, it cannot be built up again using this approach in single grain computations.

If sediment has accumulated above the protection layer, the armouring condition is not applied until this sediment is totally eroded.

1.2.1.4   Settling Velocities of Particles

The settling velocity w of sediment particles is an important parameter to determine which particles are transported as bed load or as suspended load. Many different empirical or semi-empirical relations for the determination of settling velocities in dependence of the grain diameter have been suggested in literature.

Approach of van Rijn

The sink rate can be determined against the grain diameter after van Rijn (1984b).

$$w = \frac{(s-1)gd^2}{18\nu} \quad for \quad 0.001 < d \le 0.1 \quad mm \tag{1.60}$$

$$w = \frac{10\nu}{d} \left(\sqrt{1 + \frac{0.01(s-1)gd^3}{\nu^3}} - 1 \right) \quad for \quad 0.1 < d \le 1 \quad mm \tag{1.61}$$

$$w = 1.1\sqrt{(s-1)gd} \quad for \quad d \ge 1 \quad mm \tag{1.62}$$

d is the diameter of the grain, ν is the kinematic viscosity and $s = \rho_S/\rho$ the specific density.

Approach of Wu and Wang

A newer approach for the computation of the sink velocity is the one of Wu et al. (2000):

$$w = \frac{M\nu}{Nd} \left[\sqrt{\frac{1}{4} + \left(\frac{4N}{3M^2}(D^*)^3 \right)^{1/n}} - \frac{1}{2} \right]^n \tag{1.63}$$

where:
$M = 53.5e^{-0.65S_p}$
$N = 5.65e^{-2.5S_p}$
$n = 0.7 + 0.9S_p$

S_p is the Corey shape factor, with a value for natural sediments of about 0.7 (0.3 - 0.9).

Approach of Zhang

The Zhang formula (Zhang, 1961) is based on many laboratory data and was developed for naturally worn sediment particles. It can be used in a wide range of sediment sizes in the laminar as well turbulent settling region (Wu, 2007).

$$w = \sqrt{\left(13.95\frac{\nu}{d}\right)^2 + 1.09(s-1)gd} - 13.95\frac{\nu}{d} \tag{1.64}$$

1.2.1.5 Bed load Propagation Velocity

The propagation velocity of sediment material is an important parameter to characterize the bed load transport in rivers. In some numerical approaches for morphological simulations this velocity is a useful input parameter.

Several empirical investigations have been made to measure the velocity of bed load material in experimental flumes. One recent approach for the determination of the propagation velocity is the semi-empirical equation based on probability considerations by Zhilin Sun and John Donahue (Sun and Donahue, 2000) as

$$u_B = 7.5(\sqrt{\theta'} - C_0\sqrt{\theta_{cr}})\sqrt{(s-1)gd} \tag{1.65}$$

where θ' is the dimensionless effective bottom shear stress and θ_{cr} is the critical Shields parameter for incipient motion for a grain of diameter d. Furthermore, C_0 is a coefficient less than 1 and s is specific density of the bedload material. As can be seen, the propagation velocity will increase if the difference of the actual shear stress and the criticial shear stress enlarges as well as in the case of larger grain diameters.

1.2.2 Suspended Sediment and Pollutant Transport

1.2.2.1 One Dimensional Advection-Diffusion-Equation

For a channel with irregular cross section area A (Figure 1.1) the following advection-diffusion equation for to the number of pollutant species or grain size classes ng holds:

$$\frac{\partial(AC_g)}{\partial t} + \frac{\partial(QC_g)}{\partial x} - \frac{\partial}{\partial x}\left(A\Gamma\frac{\partial C_g}{\partial x}\right) - S_g - Sl_g = 0 \quad for \quad g = 1, ..., ng \tag{1.66}$$

Introducing the continuity equation $\frac{\partial A}{\partial t} + \frac{\partial Q}{\partial x} = 0$ in eq. 1.66 one becomes:

$$A\frac{\partial(C_g)}{\partial t} + Q\frac{\partial(C_g)}{\partial x} - \frac{\partial}{\partial x}\left(A\Gamma\frac{\partial C_g}{\partial x}\right) - S_g - Sl_g = 0 \quad for \quad g = 1, ..., ng \tag{1.67}$$

1.2.2.2 Two Dimensional Advection-Diffusion-Equation

According to the number of pollutant species or grain size classes, ng advection-diffusion equations for transport of the suspended material are provided as follows:

$$\frac{\partial}{\partial t}C_g h + \frac{\partial}{\partial x}\left(C_g q - h\Gamma\frac{\partial C_g}{\partial x}\right) - \frac{\partial}{\partial y}\left(C_g r - h\Gamma\frac{\partial C_g}{\partial y}\right) - S_g - Sl_g = 0 \quad for \quad g = 1, ..., ng$$
$$\tag{1.68}$$

where C_g is the concentration of each grain size class and Γ is the eddy diffusivity.

1.2.2.3　Source Terms

For both suspended sediment and pollutant transport there can be a local sediment or pollutant source: Sl_g given by a volume.

For suspended sediment transport an additional source term S_g representing the exchange with the bed has to be considered. This term appears also in the bed load equations if they are applied in combination with suspended load.

This term is calculated by the difference between the deposition rate q_d and the suspension (entry) rate q_{e_k}.

$$S_g = q_{e_g} - q_{d_g} \tag{1.69}$$

The deposition rate is expressed as convection flux of the sink rate:

$$q_{d_g} = w_g C_{d_g} \tag{1.70}$$

w_g is the sink rate of grain class g and C_{d_g} its concentration near the bed. According to a suggestion of Bennett and Nordin (1977), the suspension entry is formulated in line with the deposition rate employing empirical relations:

$$q_{e_g} = w_g \beta_g C_{e_g} \tag{1.71}$$

The outcome of this is:

$$S_g = w_g(\beta_g C_{e_g} - C_{d_g}) \tag{1.72}$$

The sink rate w_g can be determined against the grain diameter after one of the relations given in Section 1.2.1.4

The reference concentration for the suspension entry can be calculated as follows after van Rijn (1984b):

$$C_{e_g} = 0.015 \frac{d_g}{a} \frac{T_g^{1.5}}{(D^*)_g^{0.3}} \tag{1.73}$$

where:

T_g　is the dimensionless characteristic number for the bottom shear stress of grain class g.
a　is the reference height above the mean bed bottom.
D_g^*　is the dimensionless diameter of grain class g.

Another approach is the one of Zyserman and Fredsøe (1994):

$$C_{e_g} = \frac{0.331(\theta' - 0.045)^{1.75}}{1 + 0.72(\theta' - 0.045)^{1.75}} \tag{1.74}$$

with the dimensionless effective bottom shear stress

$$\theta' = u_*^2 / [(s-1)gd] \tag{1.75}$$

The reference concentration for the deposition rate is calculated after Lin (1984):

$$C_{d_g} = \left(3.25 + 0.55 \ln \left(\frac{w_g}{\kappa u_*}\right)\right) C_g \tag{1.76}$$

where:

C_g is the mean concentration over depth of suspended particles of grain class g,

u_* is the shear velocity,

κ the Von Karmann constant.

Another approach is the one of Minh Duc (1998):

$$\alpha_c = \frac{(h-\delta)}{\int\limits_\delta^h \left(\frac{h-z}{z}\frac{\delta}{h-\delta}\right)^{\omega/\kappa U_*} dz}$$

The erosion and deposition rates can also be computed using critical shear stresses, as does Xu (1998).

$$q_d = \begin{cases} \omega_s C_a \left(1 - \dfrac{\tau}{\tau_{c,S}}\right) & \tau < \tau_{c,S} \\ 0 & \tau \ge \tau_{c,S} \end{cases}$$

$$\tau_{c,S} = \frac{\rho_s - \rho}{\rho_s}\frac{gh\omega_s C_k}{T_k U}$$

T_k is a calibration parameter which has been suggested to take a value of 0.0018 by Westrich and Juraschek (1985). The reference concentration C_a can be determined after van Rijn (1984b) (eq. 1.73) or Zyserman and Fredsøe (1994) (eq. 1.74). Alternatively, C_a can be set to the actual depth-averaged concentration C_k.

$$q_e = \begin{cases} M \left(\dfrac{\tau}{\tau_{c,E}} - 1\right)^n & \tau > \tau_{c,E} \\ 0 & \tau \le \tau_{c,E} \end{cases}$$

With $\tau_{c,E} = \rho u_{*cr}^2$. M and n are calibration parameters.

1.2.3 Bed Load Transport

1.2.3.1 One Dimensional Bed Load Transport

1.2.3.1.1 Component of the Bed Load Flux

The bed load flux for the one-dimensional case consists of one single component for each grain size, namely the specific bed load flux in stream wise direction q_{B_g}.

1.2.3.1.2   Evaluation of Bed Load Transport due to Stream Forces

In the one-dimensional case, the total specific bed load flux due to stream forces is evaluated as follows:

$$q_{B_g} = \beta_g q_B(\xi_g) \tag{1.77}$$

Details about the transport laws for evaluation of q_B with or without the consideration of the hiding factor ξ_g can be found in Section 1.2.3.4.

1.2.3.1.3   Bed Material Sorting

For each fraction g a mass conservation equation can be written, the so called "bed-material sorting equation":

$$(1-p)\frac{\partial}{\partial t}(\beta_g \cdot h_m) + \frac{\partial q_{B_g}}{\partial x} + s_g - sf_g - sl_{B_g} = 0 \quad for \quad g = 1, ..., ng \tag{1.78}$$

where p = porosity of bed material (assumed to be constant), q_{B_g} = total bed load flux per unit width, sf_g = specific flux through the bottom of the active layer due to its movement and sl_{B_g} = source term per unit width to specify a local input or output of material (e.g. rock fall, dredging). The term s_g describes the exchange per unit width between the sediment and the suspended material (see Section 1.2.2.3).

1.2.3.1.4   Global Mass Conservation

Finally, the global bed material conservation equation is obtained by adding up the masses of all sediment material layers between the bed surface and a reference level for all fractions (Exner-equation) directly resulting in the elevation change of the actual bed level:

$$(1-p)\frac{\partial z_B}{\partial t} + \sum_{g=1}^{ng}\left(\frac{\partial q_{B_g}}{\partial x} + s_g - sl_{B_g}\right) = 0 \tag{1.79}$$

1.2.3.2   Two Dimensional Bed Load Transport

1.2.3.2.1   Components of the Bed Load Flux

The specific bed load flux in x direction is composed of three parts (an analogous relation exists in y direction):

$$q_{B_g,x} = q_{B_g,xx} + q_{B_g,xlateral} + q_{B_g,xgrav} \tag{1.80}$$

where (quantities per unit width) $q_{B_g,xx}$ = bed load transport due to flow in x direction, $q_{B_g,xlateral}$ = lateral transport in x-direction (due to bed load transport in y direction on sloped bed), $q_{B_g,xcurv}$ = transport due to curvature effect in x-direction, and $q_{B_g,xgrav}$ = pure gravity induced transport (e.g. due to collapse of a side slope).

1.2.3.2.2 Evaluation of Equilibrium Bed Load Transport

In the two-dimensional case, bed load transport is evaluated as follows

$$q_{B_g,xx} = \beta_g q_B(\xi_g) \cdot \boldsymbol{e}_x \tag{1.81}$$

The specific bed load discharge q_{B_g} of the gth grain class has to be evaluated by a suitable transport law. Approaches for the bed load discharge $q_{B_g}(\xi_g)$ with or without the consideration of a hiding factor ξ_g are discussed in Section 1.2.3.4.

1.2.3.2.3 Bed Load Direction due to Lateral Bed Slope

Empirical bed load formulas were originally derived for situations where bed slope equals flow direction. However, in case of lateral bed slope with respect to flow direction, bed load direction differs from flow direction due to gravitational influence on the bed material, e.g. moving sediment particle on riverbank. Therefore, bed load direction is corrected for lateral bed slope based on the following approach (e.g. see Ikeda (1982); Talmon et al. (1995)):

$$\tan \varphi_b = -N_l \left(\frac{\tau'_{B_{cr,g}}}{\tau'_B} \right)^\gamma \vec{s} \cdot \vec{n}_q \tag{1.82}$$

where: φ_b = bed load direction with respect to the flow vector $\vec{q}$, N_l = lateral transport factor ($1.2 \leq N_l \leq 2.4$), γ = lateral transport exponent (default $\gamma = 0.5$), $\vec{s}$ = bed slope (positive uphill, negative downhill), $\vec{n}_q$ = is the unit vector perpendicular to $\vec{q}$ pointing in downhill direction ($\vec{s} \cdot \vec{n}_q < 0$) , τ'_B = dimensionless bed shear stress, and $\tau'_{B_{cr,g}}$ = dimensionless critical shear stress of the individual grain class.

1.2.3.2.4 Bed Load Direction due to Curvature Effect

Due to the presence of geometrical curvatures in rivers, the bed load direction may deviate from the depth averaged flow direction. Due to the three dimensional spiral flow motion, the bed load direction tends to point towards the inner side of the curve (Figure 1.4). This curvature effect is taken into account according to an approach proposed by Engelund (1974) , where the deviation angle φ_b (positive counterclockwise and vice versa) from the main flow direction is determined as

$$\tan \varphi_b = N_* \frac{h}{R} \tag{1.83}$$

where h denotes the water depth, N_* denotes a curvature factor, and R denotes the radius of the river bend (positive for curvature in counterclockwise direction and vice versa).

Note that the curvature factor N_* mainly depends on bed roughness. Therefore, $N_* \approx 7$ for natural streams (Engelund, 1974), and values up $N_* \approx 11$ for laboratory channels (Rozovskii, 1957).

1.2.3.2.5 Gravitational Bank Collapse

Gravitational induced riverbank or sidewall failures are significant aspects concerning erosion and transport modelling. Such processes may play an important role in many
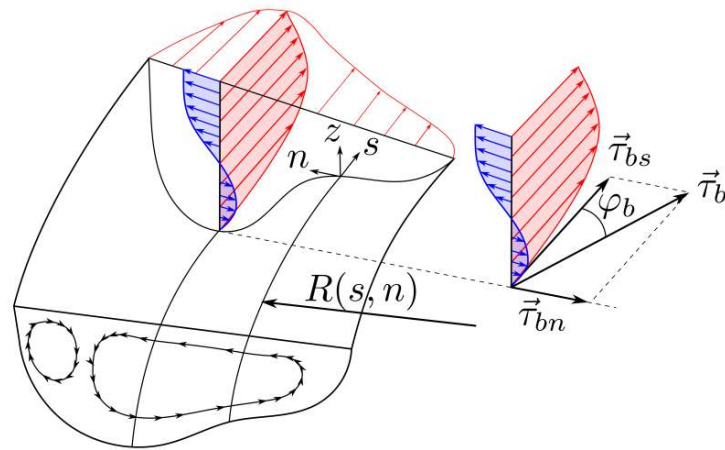
Figure 1.4 *Effect of spiral motion in river bend on bed shear stress τ_b with deviation angle from main flow direction φ_b, adapted from Blanckaert (2011)*

situations, such as meandering streams, river widenings or failures of erodible embankment structures due to overtopping waters. Such slope failure processes take place mostly discontinuous and can deliver significant contributions to the total sum of transported material.

The modes of slope failures can differ largely (falls, topples, slides, etc.) and depend on the soil material, the degree of soil compaction and the pore pressures within the soil matrix. Here, a simplified, geometric approach is applied to be able to consider some aspects of this purely gravitational induced transport.

This approach is based on the idea that a slope is flattened if its angle α becomes steeper than a given critical slope angle γ_{crit} .

$$q_{B_g,xgrav} = \begin{cases} 0 & \text{if } (\alpha \leq \gamma_{crit}) \\ f(\alpha, \gamma_{crit}) & \text{if } (\alpha > \gamma_{crit}) \end{cases} \tag{1.84}$$

The sliding material is moved from the sediment element with higher elevation to the lower situated element. Three characteristic critical slope angles are defined in this approach to have some flexibility in modelling the complex geotechnical aspects. The critical angles can be characterized as:

- critical angle for dry or partially saturated bank material which may greatly exceed the material's angle of repose (up to nearly vertical walls) due to negative pore pressures,
- critical angle for fully saturated and over flown material which is in the range of the material's angle of repose and
- a critical angle for deposited, not-compacted material.

A more physically based geotechnical approach, which takes into account more geotechnical considerations, is planned to be implemented in the future.

1.2.3.2.6   Bed Material Sorting
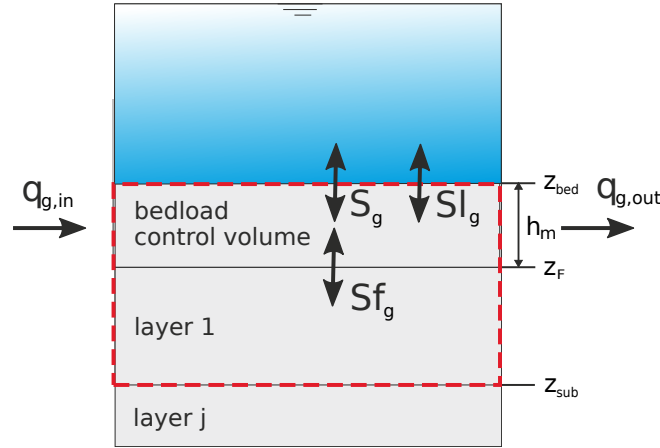
Figure 1.5 *Definition sketch of overall control volume (red) of bed material sorting equation*

The change of volume of a grain class g is balanced over the bed load control volume V_g and the underneath layer volume V_{sub_g} , as it is illustrated in Figure 1.5.

Depending on the bedload in- and outflows, the composition of the grain fractions in the bedload control volume can change. Futhermore three source terms are distinguished:

- External sediment sources or sinks can be specified (Sl_g).
- An exchange of sediment with the water column can take place (S_g).
- The movement of the bedload control volume bottom Z_F can lead to changes of the grain compositions within the bedload control volume and the underneath soil layer (Sf_g). (This is a special kind of source term, because it does not change the overall grain volume within the control volume indicated in Figure 1.5. It is not related with a physical movement of particles.)

For each grain class g a mass conservation equation can be written, the so called "bed-material sorting equation", which is used to determine the grain fractions β_g at the new time level

$$\frac{\partial V}{\partial t} = (1-p) \left[\underbrace{\frac{\partial (\beta_g \cdot h_m)}{\partial t}}_{\text{change of grain volume in bedload control volume}} + \underbrace{\frac{\partial (\beta_{sub_g} \cdot (z_F - z_{sub}))}{\partial t}}_{\text{change of grain volume in layer } 1 = sf_g} \right]$$

$$= \underbrace{-\frac{\partial q_{B_g,x}}{\partial x} - \frac{\partial q_{B_g,y}}{\partial y}}_{\text{fluxes over boundary}} \underbrace{-s_g + sl_{Bg}}_{\text{source terms}}$$

Rearranging this sorting equation leads to following formulation which is used from here on

$$(1-p)\frac{\partial}{\partial t}(\beta_g \cdot h_m) + \frac{\partial q_{B_g,x}}{\partial x} + \frac{\partial q_{B_g,y}}{\partial y} + s_g - sf_g - sl_{B_g} = 0 \qquad for \quad g = 1, ..., ng$$

$$(1.85)$$

where h_m = thickness of bedload control volume, p = porosity of bed material (assumed to be constant), $(q_{B_g,x}, q_{B_g,y})$ = components of total bed load flux per unit width, sf_g = flux through the bottom of the bedload control volume due to its movement and sl_{B_g} = source term to specify a local input or output of material (e.g. rock fall, dredging).

1.2.3.2.7   Global Mass Conservation

The global bed material conservation equation, which is often called Exner-equation, is obtained by adding up the masses of all sediment material layers between the bed surface and a reference level. This is done for all grain fractions and directly results in the elevation change of the actual bed level z_B:

$$(1 - p)\frac{\partial z_B}{\partial t} + \sum_{g=1}^{ng} \left(\frac{\partial q_{B_g,x}}{\partial x} + \frac{\partial q_{B_g,y}}{\partial y} + s_g - sl_{B_g} \right) = 0 \tag{1.86}$$

1.2.3.3   Sublayer Source Term

The bottom elevation of the bed load control volume z_F is identical to the top level of the underneath layer. If z_F moves up, sediment flows into this underneath layer and leads to changes in its grain compositions. The exchange of sediment particles between the bed load control volume and the underlying layer is expressed by the source term:

$$s_{f_g} = -(1 - p)\frac{\partial}{\partial t}((z_F - z_{sub})\beta_{sub_g}) \tag{1.87}$$

1.2.3.4   Closures for Bed Load Transport

In the following a variety of bed load transport formulas are listed which are implemented to calculate the transport capacity. For practical purposes usually a calibration of the used formula is needed and several parameters can be adjusted by the user.

1.2.3.4.1   Meyer-Peter and Müller (MPM & MPM-Multi)

The bed load transport formula of Meyer-Peter and Müller (Meyer-Peter and Müller, 1948) can be written as follows:

$$q_{B_g} = \alpha\sqrt{(s - 1)gd_g^3} \, (\theta_g - \theta_{cr,g})^m \tag{1.88}$$

Herein, α denotes the bed load factor (orginally $\alpha = 8$), m the bed load exponent (orginally $m = 1.5$), q_{B_g} is the specific bed load transport rate of grain class g, θ_g is the effective dimensionless shear stress for grain class g, $\theta_{cr,g}$ is the critical dimensionless shear stress for grain class g , d_g is the diameter of the grain class g, $s = \rho_s/\rho$ and g stands for the gravitational acceleration. Note that by adjusting α to 4.93 and m to 1.6, the bed load formula can be adapted accoring to Wong and Parker (2006).

Meyer-Peter and Müller observed in their experiments that the fist grains moved already for $\theta_{cr} = 0.03$. But as their experiments took place with steady conditions they used a value for which already 50% of the grains where moving. They proposed the value of 0.047.

However for very unsteady conditions one should use values for which the grains really start to move (Fäh, 1997) like the values given by the shields diagram.

The formula of Meyer-Peter and Müller is applicable in particular for coarse sand and gravel with grain diameters above 1 mm (Malcherek, 2001).

The original bed load transport formula is intended for single grain simulations. But an extension of the MPM-Formula for fractional transport is implemented in the program and called MPM-Multi. It uses the hiding function ξ_g proposed by Ashida and Michiue (1971):

$$\xi_g = \begin{cases} [\log{(19)}/\log{(19 d_g/d_m)}]^2 & d_g/d_m \geq 0.4 \\ d_m/d_g & d_g/d_m < 0.4 \end{cases} \tag{1.89}$$

d_g is the grain size diameter of grain class g and d_m the mean diameter of the grain mixture. The dimensionless critical shear stress of grain class becomes:

$$\theta_{cr,g} = \theta_{cr,ref}\xi_g, \tag{1.90}$$

where $\theta_{cr,ref}$ usually is assigned to a fix value (e.g. $\theta_{cr,ref} = 0.047$) or the critical Shields paramter of the mean grain size.

1.2.3.4.2 Ashida and Michiue

The bed load formula for non-uniform sediments according to Ashida and Michiue (Ashida and Michiue, 1971) reads

$$q_{B_g} = 17\sqrt{(s-1)g d_g^3}\,(\theta_g - \xi_g\theta_{cr,ref})(\sqrt{\theta_g} - \sqrt{\xi_g\theta_{cr,ref}}) \tag{1.91}$$

where q_{B_g} is the specific bed load transport rate of grain class g, θ_g is the dimensionless shear stress for grain class g, $\theta_{cr,ref}$ is the reference critical dimensionless shear stress (Ashida and Michiue (1971) proposed $\theta_{cr,ref} = 0.05$), d_g is the diameter of the grain class g, $s = \rho_s/\rho$, g is the gravitational acceleration, and ξ_g is the hiding function according to eq. 1.89.

1.2.3.4.3 Parker

Parker extended his empirical substrate-based bed load relation for gravel mixtures (G. Parker, 1990, Parker et al. (1982)), which was developed solely with reference to field data and suitable for near equilibrium mobile bed conditions, into a surfaced-based relation. The new relation is proper for non-equilibrium processes.

Based on the fact that the rough equality of bed load and substrate size distribution is attained by means of selective transport of surface material and the surface material is the source for bed load, Parker has developed the new relation based on the surface material. An important assumption in deriving the new relation is suspension cut-off size. Parker supposes that during flow conditions at which significant amounts of gravel are moved, it is commonly (but not universally) found that the sand moves essentially in suspension (1 to 6 mm). There for Parker has excluded sand from his analysis. In his free access Excel file, he has explicitly emphasised that the formula is valid only for the size larger than 2 mm. Regarding to the Oak Creek data, the original relation predicted 13% of the bed load as sand. For consistency it has to be corrected for the exclusion of sand and finer material.

$$W_{si}^* = 0.00218\, G[\xi_s \omega \phi_{sg0}] \quad ; \quad W_{si}^* = \frac{Rgq_{bi}}{(\tau_B/\rho)^{3/2} F_i} \tag{1.92}$$

where:

$$\xi_s = \left(\frac{d_i}{d_g}\right)^{-0.0951} \quad ; \quad \phi_{50} = \frac{\tau_{sg}^*}{\tau_{rsg0}^*} \quad ; \quad \tau_{sg}^* = \frac{\tau_B}{\rho Rgd_g} \quad ; \quad \tau_{rsg0}^* = 0.0386$$

$$\omega = 1 + \frac{\sigma}{\sigma_0(\phi_{sg0})}[\omega_0(\phi_{sg0}) - 1] \quad ; \quad \sigma = \sum F_i \left[\frac{\ln(d_i/d_g)}{\ln(2)}\right]^2 \quad ; \quad d_g = e^{\sum F_i \ln(d_i)}$$

ξ_s is a "reduced" hiding function and differs from the one of Einstein. The Einstein hiding factor adjusts the mobility of each grain d_i in a mixture relative to the value that would be realized if the bed were covered with uniform material of size d_i. The new function adjusts the mobility of each grain di relative to the d_{50} or d_g , where d_g denotes the surface geometric mean size.

Although the above formulation does not contain a critical shear stress, the reference shear stress τ_{rsg0}^* makes up for it, in that transport rates are exceedingly small for $\tau_{sg}^* < \tau_{rsg0}^*$. Regarding to the fact that parker's relation is based on field data and field data are often in case of low flow rates, the relation calculates low bed load rates (Marti, 2006).

If this transport formula is used in combination with a local slope correction of the reference shear stress (see Section 1.2.1.1.2) attention must be paid that τ_{rsg0}^* may not become too small or even zero. Since this value is in the denominator of the transport formula, such situations may lead to numerical instabilities. To avoid these problems a minimum value is enforced:

$$\tau_{rsg0}^* = min(\tau_{rsg0}^{*,min}, \tau_{rsg0}^*)$$

1.2.3.4.4   Wilcock and Crowe

Wilcock and Crowe developed a sediment transport model for sand/gravel mixtures (Wilcock, 2003), similar to Parker's model (G. Parker, 1990), and it was developed with a large experimental results dataset. It reference fractional transport rates to the size distribution of the bed surface, rather than the subsurface, making the model explicit and capable of predicting transient conditions. The hiding function incorporated in the model resolves discrepancies obvserved among earlier hiding functions implemented in other transport models, such as the Oak Creek and the Cambridge ones (A.J. Parker G.; Sutherland, 1990). Wilcock and Crowe model (Wilcock, 2003) uses the full grain size distribution of the bed surface, including sand, incorporating a non-linear effect of sand content on gravel transport rate.

$$W_{si}^* = G(\phi_i) \quad ; \quad W_{si}^* = \frac{Rgq_{bi}}{(\tau_B/\rho)^{3/2} F_i} \tag{1.93}$$

where:

$$G\left(\phi_i\right) = \begin{cases} 0.002\,\phi_i^{7.5} & \phi_i < 1.35 \\[2mm] 14\left(1 - \dfrac{0.894}{\phi_i^{0.5}}\right)^{4.5} & \phi_i \geq 1.35 \end{cases}$$

and:

$$\phi_i = \frac{\tau_{sg}^*}{\tau_{ssrg}^*}\frac{d_i}{d_g}^{-b} \qquad\qquad ;\quad \tau_{ssrg}^* = \frac{\tau_B}{\rho R g d_g}$$

$$\tau_{ssrg}^* = 0.021 + 0.015exp(-20F_s) \quad ;\quad b = \frac{0.67}{1 + exp(1.5 - \dfrac{d_i}{d_g})}$$

The non-linear effect of sand content F_s on gravel transport is taken into account in τ_{ssrg}^*. Wilcock and Crowe (Wilcock, 2003) have shown that increasing sand content in the bed active layer of a gravel-bed stream increases the surface gravel mobility. This effect is captured in their relationship between τ_{ssrg}^* (a surrogate for a critical Shields number) and the fraction sand in the active layer F_s. Note that τ_{ssrg}^* decreases as F_s increases, causing an increase of ϕ_i and in turn of the fraction bedload q_{bi}.

1.2.3.4.5   Hunziker (MPM-H)

Hunziker (1995) proposed a bed load formula for fractional bed load transport of graded sediment:

$$q_{Bg} = 5\beta_g[\xi_g(\theta'_{dms} - \theta_{cdms})]^{3/2}\sqrt{(s-1)gd_{ms}^3} \qquad\qquad (1.94)$$

where θ'_{dms} denotes the Shields parameter of the mean grain size of the surface bed material d_{ms} according to eq. 1.95, ξ_g denotes the hiding function applied on the excess shear stress $(\theta'_{dms} - \theta_{cdms})$.

$$\theta'_{dms} = \frac{\tau'_b}{\rho_w\left(s-1\right)d_{ms}} \qquad\qquad (1.95)$$

Note that due to the correction of the excess shear stress $(\theta'_{dms} - \theta_{cdms})$, the transport formula is based on the concept of "equal mobility", i.e. all grain classes start to move at same flow condition. The critical Shields parameter θ_{cdms} of the mean grain size diameter is determined according to

$$\theta_{cdms} = \theta_{ce}\left(\frac{d_{mo}}{d_{ms}}\right)^{0.33} \qquad\qquad (1.96)$$

where θ_{ce} denotes the critical Shields parameter for incipient motion for uniform bed material. Two sediment layers are distinguished: the upper mixing layer which is in interaction with the flow and a subsurface layer below. Here, d_{ms} denotes the mean grain size diameter of surface bed material and d_{mo} denotes the mean grain size diameter of subsurface bed material. This relation (d_{ms}/d_{mo}) can be approximated as a function of the Shields parameter of the mean grain size of the surface bed material as

$$\frac{d_{ms}}{d_{mo}} = 0.0163\theta'^{-1.45}_{dms} + 0.6 \tag{1.97}$$

Finally, the hiding function is determined as

$$\xi_g = \left(\frac{d_g}{d_{ms}}\right)^{-\alpha} \tag{1.98}$$

where α is an empirical parameter depending on the Shields parameter (see also Hunziker and Jaeggi (2002)) according to eq. 1.99, which is limited to a range between -0.4 and 2.0.

$$\alpha = 0.011\theta'^{-1.5}_{dms} - 0.3 \tag{1.99}$$

1.2.3.4.6   Rickenmann

Experiments for bed load transport in gravel beds were performed at VAW ETH Zurich for bed slopes of 0.0004-0.023 by Meyer-Peter and Müller (1948) and for bed slopes of 0.03-0.2 by Smart and Jaeggi (1983) and by Rickenmann (1990). Rickenmann (1991) developed the following bed load transport formula for the entire slope range using 252 of these experiments.

$$\Phi_B = 3.1 \left(\frac{d_{90}}{d_{30}}\right)^{0.2} \theta'^{0.5}(\theta' - \theta_{cr})Fr^{1.1}(s - 1)^{-0.5} \tag{1.100}$$

$$q_B = \Phi_B((s - 1)gd_m^3)^{0.5} \tag{1.101}$$

θ' is the dimensionless shear stress, θ_{cr} the dimensionless shear stress at the beginning of bed load transport, $s = \rho_s/\rho$ the sediment density coefficient, Fr the Froude number and d_m the mean grain size.

1.2.3.4.7   Smart and Jäggi (for single grain and multiple grain classes)

Experiments for bed load transport in gravel beds were performed at VAW ETH Zurich for bed slopes of 0.0004-0.023 by Meyer-Peter and Müller (1948) and for bed slopes of 0.03-0.2 by Smart and Jaeggi (1983) and by Rickenmann (1990). Smart and Jäggi developed a bed load transport formula for steep channels using their own experimental results and the results of Meyer-Peter and Müller.

$$q_B = \frac{4}{s - 1} \left(\frac{d_{90}}{d_{30}}\right)^{0.2} J^{0.6}Ru(J - J_{cr}) \tag{1.102}$$

where s is the sediment density coefficient ($s = \rho_s/\rho$), R is the hydraulic radius, u is the velocity, J is the slope and J_{cr} is the critical slope for the initiation of the bed load transport, which is calculated as

$$J_{cr} = \frac{\theta_{cr}(s - 1)d_m}{R} \tag{1.103}$$

where θ_{cr} is the critical shields parameter (for the initiation of motion) and d_m is the mean grain size. In order to account for the gravitational influence of the local bed slope Smart and Jaeggi (1983) proposed the following reduction of the critical shields parameter:

$$\theta_{cr} = \theta_{cr,Ref}(\cos(\arctan J)) \left(1 - \frac{J}{\tan \psi}\right) \tag{1.104}$$

where J is the local bed slope, ψ the angle of repose and $\theta_{cr,Ref}$ the critical reference shields parameter for the medium grain size defined by the user (Smart and Jaeggi (1983) propose a value of 0.05).

The Smart & Jäggi transport formula is extended to multiple grain classes by applying the original equation to the individual grain classes according to the following approach:

$$q_{B,g} = \frac{4}{s-1} \left(\frac{d_{90}}{d_{30}}\right)^{0.2} J^{0.6} Ru(J - J_{cr,g}) \tag{1.105}$$

Compared to the original eq. 1.102 the transport rate for each grain class $q_{B,i}$ is calculated with the critical slope $J_{cr,g}$ for the initiation of motion of the grain class i according to

$$J_{cr,g} = \frac{\theta_{cr,g}(s-1)d_i}{R}$$

where $\theta_{cr,g}$ is the critical shields parameter for grain class g, d_g is the diameter of the grain class g.

With the term $\alpha = (d_{90}/d_{30})^{0.2}$ the original equation intends to account for the influence of the grain class distribution. According to Smart and Jaeggi (1983) this term is in the range of $1.06 \le \alpha \le 1.53$. If this term is to be neglected Smart and Jaeggi (1983) recommend substituting $\alpha = 1.05$. The influence of the grain class distribution is considered in the hiding and exposure approach according to Ashida and Michue (Ashida and Michiue, 1971; Parker, 2008) in eq. 1.106 and eq. 1.107.

$$\zeta_g = \begin{cases} 0.85 \left(\dfrac{d_g}{d_m}\right)^{-1} & \text{for} \quad \dfrac{d_g}{d_m} \le 0.4 \\[4mm] \left(\dfrac{\log(19)}{\log\left(19\dfrac{d_g}{d_m}\right)}\right)^2 & \text{for} \quad \dfrac{d_g}{d_m} > 0.4 \end{cases} \tag{1.106}$$

$$\theta_{cr,g} = \zeta_g \theta_{cr} \tag{1.107}$$

The critical shields parameter θ_{cr} is calculated according to eq. 1.104.

1.2.3.4.8 Wu

Wu et al. (2000) developed a transport formula for graded bed materials based on a new approach for the hiding and exposure mechanism of non-uniform transport. The hiding and exposure factor is assumed to be a function of the hidden and exposed probabilities, which are stochastically related to the size and gradation of bed materials. Based on this concept, formulas to calculate the critical shear stress of incipient motion and the fractional

bed-load transport have been established. Different laboratory and field data sets were used for these derivations.

The probabilities of grains d_g hidden and exposed by grains d_i is obtained from

$$p_{hid_g} = \sum_{i=1}^{ng} \beta_i \frac{d_i}{d_g + d_i}, \quad p_{exp_g} = \sum_{i=1}^{ng} \beta_i \frac{d_g}{d_g + d_i} \tag{1.108}$$

The critical dimensionless shields parameter for each grain class g can be calculated with the hiding and exposure factor η_g and the shields parameter of the mean grain size θ_{cr_m} as

$$\theta_{cr_g} = \theta_{cr_m} \underbrace{\left(\frac{p_{exp_g}}{p_{hid_g}} \right)^m}_{\eta_g} \tag{1.109}$$

The transport capacity now can be determined with Wu's formula in dimensionless form as

$$\Phi_{B_g} = 0.0053 \left[\frac{\theta'}{\theta_{cr_g}} - 1 \right]^{2.2} \tag{1.110}$$

Finally the bed load transport rates calculates for each grain fraction as

$$q_{b_g} = \beta_g \sqrt{(s-1)g d_g^3} \, \Phi_{B_g} \tag{1.111}$$

As results of their data analysis the authors recommend to set $m = -0.6$ and $\theta_{cr_m} = 0.03$ to obtain best results.

If this transport formula is used in combination with a local slope correction of the critical shear stress (see Section 1.2.1.1.2) attention must be paid that θ_{cr_g} may not become too small or even zero. Since this critical dimensionless shear stress is in the denominator of the transport formula, such situations may lead to numerical instabilities. To avoid these problems a minimum value for θ_{cr_g} is enforced.

$$\theta_{cr_g} = min(\theta_{cr}^{min}, \theta_{cr_g})$$

1.2.3.4.9   Van Rijn

van Rijn (1984a) developed a bed load formula for grain sizes between 0.2 and 2 mm according to eq. 1.112.

$$q_B = 0.053 \sqrt{(s-1)g} \, \frac{d_{50}^{1.5} T^{2.1}}{D_*^{0.3}} \tag{1.112}$$

Here D^* is the dimensionless grain diameter according to eq. 1.53 and T is the non-dimensional excess bed shear stress or the transport stage number, defined as

$$T = (u_*/u_{*cr})^2 - 1 \tag{1.113}$$

where u_* is the effective bed shear velocity determined as

$$u_* = u\sqrt{g}/C'_h \tag{1.114}$$

with $C'_h = 18\log\left(4h/d_{90}\right)$.

u_{*cr} is the critical bed shear velocity, u is the mean flow velocity, h is the water depth, d_{50} and d_{90} are characteristic grain diameters of the bed material.

1.2.3.4.10   Engelund and Hansen

Engelund and Hansen (1972) proposed a bed load transport formula for uniform bed material

$$q_B = 0.05\sqrt{(s-1)g}\; c_f^2\theta^{2.5}d_f^{1.5} \tag{1.115}$$

where d_f denotes the mean fall diameter of the bed material and θ the Shields parameter. Note that this rather simple bed load formula does not consider critical Shields parameter.

1.2.3.4.11   Power Law

The power law bed load formula is a very simple approach. Therfore, no critical shear stress is taken into account and bed load transport only depends on the flow velocity u.

$$q_b = au^b \tag{1.116}$$

Coefficient a and exponent b are used as calibration parameter.

1.2.3.5   Abrasion in 1D

The reduction of the grain diameters by mechanical stress can be described by the approach of Sternberg (1875). It postulates, that the mass loss dM of a grain, which is transported over a distance of dx, is proportional to the achieved work $M \cdot g \cdot dx$. If the equation $(-dM = c \cdot M \cdot dx)$ is integrated over the distance from x_0 to x (where the mass of the grain at x_0 is M_0) one obtains the relation for the reduction of the mass of the grain:

$$M(x) = M_0 e^{-c(x-x_0)} \tag{1.117}$$

The abrasion coefficient of Sternberg c is determined empirically.

1.2.3.6   Sediment Boundary Conditions

The necessity of hydraulic boundary conditions for SWE and Saint-Venant equation were explained in Section 1.1.1.5 and Section 1.1.2.5 respectively. In case of bed load transport, boundary conditions are also needed to solve the bed load transport formula and calculate the transport capacity. Boundary conditions are defined upstream and downstream of the channel (i.e. at the inflow and outflow cross section). If no boundary condition is defined, a wall is assumed at the boundary and sediment transport will not occur.

1.2.3.6.1   Upstream boundary condition

The bed load input type is given by the upstream boundary condition. Three types of upstream boundary condition are available:

- sediment_discharge: based on a sediment hydrograph describing the bed load inflow at an upstream boundary in time.

- transport_capacity: based on a given mixture, the sediment inflow is calculated for every element by calculating the equilibrium transport capacity. In this case, it is required that the sediment inflow is defined on edges and an inflow hydrograph (set as hydraulic boundary condition) has to be defined.

- IOUp: this upstream boundary condition grants a constant bed level at the inflow. The same amount of sediment leaving the first computational cell in flow direction enters the cell from the upstream bound.

1.2.3.6.2   Downstream boundary condition

The IODown is the only downstream boundary condition available for sediment transport. It corresponds to the definition of IOup, where all sediment entering the last computational cell will leave the cell over the downstream boundary.

1.2.4   Enforced Bed Movement

In case the effect of bed movements on hydraulic variables want to be investigated, we can enforce grid nodes to move vertically with time. The movements are user defined, thus they are not coupled to any morphological calculations. These changes in the nodal elevation can be regarded as *enforced erosion or aggradation.*

The elevation update of the moving nodes is done right after the calculation of the hydraulic variables. The water depth and flow velocities are not touched, i.e. no change in the mass or impulse balance. Moving node(s) simply leads to a changed water surface in the next time step.

$$w_{S,t+1}^{forced} = z_{B,t}^{forced} + h_t \tag{1.118}$$

1.2.5   Random Bed Perturbation

Random bed perturbation can be applied to the bed topography during morphological simulations. Therefore, bed elevation is perturbed by

$$z_b^t = z_b^{t-1} + \varepsilon \tag{1.119}$$

where z_b^{t-1} denotes the bed elevation from the previous time step and ε is the perturbation offset. Bed perturbation is applied on all cells using randomly $-\varepsilon$, $+\varepsilon$, or zero.

1.3   Sub-surface flow

1.3.1   Introduction

Seepage analysis is an important part of geotechnical engineering and is, for example, required for design and stability evaluations of earth embankment structures. Solving the Richard's equation accounts for the flow in the saturated zone as well as in the unsaturated zone and allows for an accurate modelling of water infiltration into the soil.

Empirical constitutive models consisting of a retention curve and a relative hydraulic conductivity function allow for approximating the multi-phase flow in the unsaturated zone with a single partial differential equation. Thereby the assumption is made that the air phase is always continuous and at atmospheric pressure, which is often said to be accurate enough for most practical applications (Lam et al., 1987).

1.3.2   Governing equations

The Richard's equation is a non-linear partial differential equation, which can be formulated in form of an advection-diffusion equation. The soil moisture content θ [-] in the equation is defined as the effective water saturation $\theta = (\theta_0 - \theta_R)/(\theta_S - \theta_R)$, with $\theta_0 =$ water content, $\theta_R =$ residual water content and $\theta_S =$ saturated water content (=porosity). The other main variable is the pore pressure of the water within the embankment body which is described in a pressure head formulation as $h = p/(\rho g)$ [L].

The 3-D Richard's equation is applied in a moisture formulation for θ as primary variable as

$$\frac{\partial \theta}{\partial t} - \nabla(K\nabla z) - \nabla(D\theta) = Q$$

$$D = K\frac{\partial h}{\partial \theta}$$

(1.120)

It is implemented also in a mixed moisture and pressure head formulation for θ and h as

$$\frac{\partial \theta}{\partial t} - \nabla(K\nabla z) - \nabla(D\nabla h) = Q$$

$$D = K$$

(1.121)

where D is a diffusivity and K is the soil conductivity which is calculated as $K = k_r(\theta)k_f$, being the product of the dimensionless relative conductivity k_r [-] and the hydraulic soil conductivity k_f [L/T].

These formulations of the Richard's equation are made dimensionless for the computations using the cell size Δx as length scale and $\Delta x/\Delta t$ as velocity scale, leading to a "mesh speed" of $c=1$ in the model.

1.3.3   Constitutive relationships

Empirical closures for the retention curve $h = f(\theta)$ and the relative permeability function $k_r = f(\theta)$ are required to close Richard's equation. The retention function in soil sciences describes how much water is retained in the soil by the capillary pressure and can be seen as a description of the pore distributions of the soil. The relative conductivity function describes the water mobility within the unsaturated zone depending on the moisture contents. It equals 1.0 in the saturated zone and reduces to smaller values in the unsaturated zone, mainly due to longer flow paths. Two different empirical relationships are employed here, the approach after Brooks and Corey (1964) and Mualem (1976) (BCM) and a modified version of the van Genuchten (1980) and Mualem model (VGM).

For the BCM model the following relations are used

$$h(\theta) = h_s \theta^{-1/\lambda}$$

$$k_r = (h/h_S)^{-(4\lambda+2)}$$

$$\frac{\partial h}{\partial \theta} = -\frac{h_S}{\lambda} \theta^{-1/\lambda - 1}$$

(1.122)

with λ [-] being a soil parameter.

For the modified VGM model the functions and derivate are as follows

$$h(\theta) = \frac{-1}{\alpha} \left[-\varkappa \left(\frac{\beta}{\theta} \right) \right]^{1/n}$$

$$k_r = \sqrt{\theta} \left(\frac{1 - \varkappa^m \theta/\beta}{1 - \varkappa^m 1/\beta} \right)^2$$

$$\frac{\partial h}{\partial \theta} = \frac{1-m}{\beta m \alpha} \frac{\theta^{-1/m - 1}}{\beta} - \varkappa \left(\frac{\beta}{\theta} \right)^{-m}$$

(1.123)

with $\varkappa(x) = 1 - x^{1/m}$ and $m = 1 - 1/n$. The empirical constants α [1/L] and n [-] describe the soil properties. The modified Version of the VGM model is used instead of the original VGM model in order to prevent infinite slopes $\partial h/\partial \theta$ at the transition to the saturated zone.

Following Ginzburg, the primary variable θ is used for the unsaturated zone as well as for the saturated zone. Therefore the retention curve $h(\theta)$, which is defined for the unsaturated zone only, is extrapolated linearly into the saturated zone as

$$h(\theta) = (\theta - 1) \frac{\partial h}{\partial \theta} \bigg|_{\theta=1} + h_S \quad \theta \geq 1 - 0$$

(1.124)

with $\partial h/\partial \theta$ being the gradient of the retention curve at the transition to the saturated zone ($\theta = 1.0$) and h_S [L] being the air entry pressure head, at which air can enter the pores when the soil is drained (= measure of the largest pore sizes).

This approach has the advantage that no special treatment and no change of variables regarding the saturated/unsaturated zones are necessary. However, it leads to an artificial

compressibility error in unsteady simulations. This error is neglected here, but can be mitigated in principle using sub-iterations (Ginzburg et al., 2004).

1.4 Morphodynamics and Vegetation

1.4.1 Introduction

Vegetation, as main biotic component of riverine environments, has a key role on shaping river morphology at a wide range of spatial and temporal scales. Above-ground biomass (plant canopy) affects the flow field altering turbulence structure and hydraulic roughness, while below-ground biomass (plant roots) modify sediment properties increasing the soil cohesion. Morphodynamic processes in turn affect vegetation survival mainly by causing burial and uprooting.

The hydrological regimes plays also an important role on the spatial distribution of vegetation by controlling seed dispersal and ensuring water and nutrients for plant growth. In riparian systems, the water table level usually represents the main source of water determining, in combination with the fluvial disturbance, a strong control on plant species distribution and composition on the elevation gradient. As a result, each species is characterized by a specific range of elevations in which its optimal conditions are met. The timescale at which vegetation reaches its equilibrium value may vary across environments and among species, allowing for the co-existence of vegetation patterns with different biomass densities.

1.4.2 Mathematical model

The mathematical model accounting for the main feedback between river morphodynamics and vegetation and implemented in BASEMENT has been developed by a joint research between the Laboratory of Hydraulics, Hydrology and Glaciology (VAW) at ETH Zurich and the Department of Civil, Environmental and Mechanical Engineering at University of Trento (Italy). Model and results are presented in Bertoldi et al. (2014). This approach is built upon three main cornerstones that describe:

- the vegetation biomass dynamics depending on species-specific parameters;
- the feedback mechanisms between river morphodynamic processes and vegetation;
- the mean water table level (groundwater) computed by adopting a simple spatial interpolation method.

1.4.3 Vegetation dynamics

In Figure 1.6, the blue line represents the water discharge through time and the dotted brown line corresponds to the discharge at which the Shields parameter θ reaches its critical value θ_{cr}, needed for the onset of sediment transport. Vegetation growth ($\frac{\partial B}{\partial t} > 0$) is enabled only during the time frame T_v, where $\theta \leq \theta_{cr}$, while is inhibited during T_m, in which sediment transport takes place ($\theta > \theta_{cr}$). During T_m, if uprooting occurs, vegetation biomass is assumed to instantaneously decrease to $B = 0$, otherwise it does not change until the beginning of subsequent vegetation growth phase (T_v).
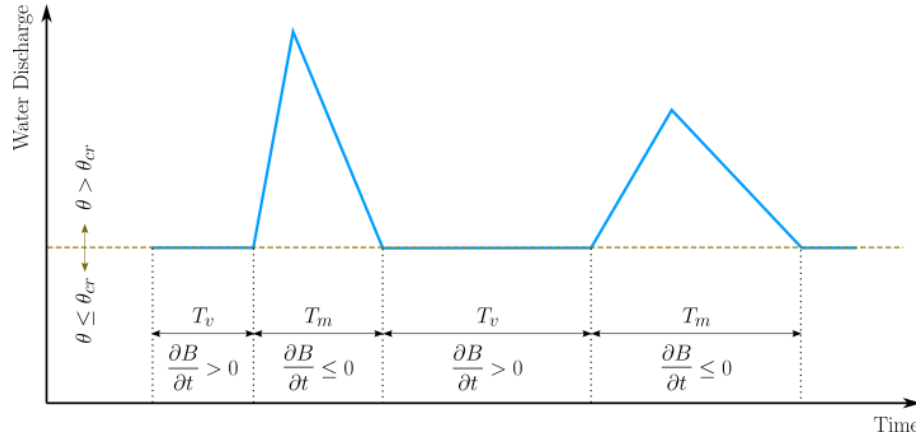
Figure 1.6 *Schematic illustration of the time frame concept used in the vegetation module.*

Vegetation is described with a dimensionless biomass density B in which both above-ground and below-ground are lumped. Within each computational cell a value of B is defined with respect to a reference dimensional value B_{max} [kg m^{-2}], representing its maximum carrying capacity. Following Marani et al, we can define an equilibrium biomass B_{eq}, normalized by its maximum, as

$$B_{eq} = \frac{1}{exp[\lambda_1(z - z_0)] + exp[-\lambda_2(z - z_0)]} \tag{1.125}$$

in which z is the bed elevation, the parameters λ_1 and λ_2 define the rate at which vegetation fitness diminishes away from it maximum, while z_0 represents the optimal elevation above the mean water table level z_w. If $\lambda_1 = \lambda_2$, B_{eq} represents a bell-shaped function with its maximum at the elevation $z_w + z_0$, such as $B_{eq}(z_w + z_0) = B_{max}$. Adopting different values of the parameters λ_1, λ_2 and z_0 correspond to modeling different species or type of vegetation that are adapted to grow in specific range of elevations. For instance, specialized vegetation would have higher values of $\lambda_{1,2}$, while species with greater tolerance to drought can be characterized by higher value of z_0 (more distant to the mean water level z_w).

Vegetation growth is governed by a logistic differential equation (non-linear ordinary differential equation)

$$\frac{dB}{dt} = \sigma B(t)\left[1 - \frac{B(t)}{B_{eq}(z)}\right] \tag{1.126}$$

where σ represents the vegetation grow rate $[s^{-1}]$. The timescale of vegetation growth is significantly higher when compared to the morphological timescale. The time needed for a riparian species to reach its maximum expansion (i.e. $B = B_{max}$), under optimal conditions, ranges from years to decades, while the timescale at which bed level changes occur, for example during a flood event, ranges from hours to days. In addition, rivers often experience only a few events that causes riverbed changes during the year. To allow for computationally feasible simulations while including all those timescales, vegetation dynamics and morphodynamic processes have been decoupled. Vegetation growth is enabled during the time frame (T_v) in which morphological changes do not occur (e.g. during low flow period), whereas it is inhibited during high flows, where sediment transport and morphodynamic processes take place (time frame T_m, see Figure 1.6). Since during T_v

riverbed does not change, we can significantly reduce computational time by decreasing T_v and increasing the grow rate of vegetation (σ), without affecting hydro-morphodynamic processes.

From a computational point of view, eq. 1.126 is integrated by using a classical Euler method with an integration timestep that equals the timestep computed for morphodynamics.

1.4.4 Feedback

1.4.4.1 Vegetation effects on hydro-morphodynamics

Vegetation increases the hydraulic roughness depending on the abundance and type. By adopting the Manning-Strickler approach for calculation of the friction factor c_f, we assume

$$K_s = K_{s,g} + (K_{s,g} - K_{s,v})\frac{B}{B_{max}} \tag{1.127}$$

where K_s $[m^{1/3}\ s^{-1}]$ is the Strickler coefficient and $K_{s,g}$ and $K_{s,v}$ refer to the values attributed to the grain and the vegetation, respectively. $K_{s,v}$ is a lumped variable encompassing a variety of plant characteristics (e.g. stiffness, bending ability, crown area and leaves density) and different plant life stages.

The modifications of the flow field have a profound effect on sediment transport as well. In vegetated flows the friction factor incorporates not only the friction located at the bottom of the river but also the drag generated by the vegetation. Therefore we assume that the shear stresses τ_g, responsible for sediment transport, decreases as a function of the grain-related roughness $K_{s,g}$. Although this approach can be generalized to any transport formula, the model only accounts for the Meyer-Peter and Müller formula (no other formulas are allowed with vegetation module activated). In term of dimensionless shear stress θ (Shield parameter), it can be derived as

$$\frac{\theta}{\theta'} = \left(\frac{K_{s,g}}{K_{s,v}}\right)^2 \tag{1.128}$$

where θ' represents the effective dimensionless shear stress applied at the bottom of the river. Notably, this strategy is similar to the correction factor applied to limit sediment transport in case of bed forms with the Meyer-Peter and Müller formula.

The presence of vegetation also changes the sediment properties, increasing the sediment cohesion and reinforcing the soil matrix. This results in a reduction of the sediment mobility and a consequent modification of the threshold for the onset of sediment transport (in case of bed load). Similarly to the added roughness by the vegetation, this can be modeled as

$$\theta_{cr} = \theta_{cr,g} + (\theta_{cr,v} - \theta_{cr,g})\frac{B}{B_{max}} \tag{1.129}$$

in which θ_{cr} represents the critical value of the Shields number that has to be exceeded to have sediment transport. $\theta_{cr,v}$ is the value attributed in presence of vegetation, while $\theta_{cr,g}$ is the typical value for gravel. This effect is usually considered to be dependent on the below-ground vegetation biomass, i.e. plant roots, and its characteristics. Here, we

assume that the biomass density B is representative of the overall plant biomass, with no distinction between above-ground (plant canopy) and below-ground biomass.

1.4.4.2 Hydro-morphodynamic effects on vegetation

Hydro-morphological processes have a significant impact of vegetation distribution determining plant removal by uprooting, which occurs when the resistance provided by the plant roots equals the pull-out forces on the canopy. Recent studies suggest that plants are able to increase their resistance to uprooting by growing large root systems. Therefore, we assume that vegetation, during morphological phases (T_m), can be uprooted by flow erosion when the bed level changes Δz exceed a threshold value z_{upr}. The latter represents a lumped estimation of the rooting depth.

1.4.5 Range of the parameters used in previous works

Model parameters used in previous publications (Bertoldi et al. (2014), Li and Millar (2011), Zen et al. (2016)) are reported in Table 1.3.

Table 1.3 *Vegetation Parameters*

Parameter	Range [min, max]	References
$\lambda_1 \; [m^{-1}]$	[0, 6]	Bertoldi et al. (2014), Zen et al. (2016)
$\lambda_2 \; [m^{-1}]$	[0.4, 100]	Bertoldi et al. (2014), Zen et al. (2016)
$z_0 \; [m]$	[0.5, 1.25]	Bertoldi et al. (2014), Zen et al. (2016)
$K_{s,v} \; [m^{1/3}s^{-1}]$	[9, 17]	Bertoldi et al. (2014), Zen et al. (2016), Li and Millar (2011)
$\theta_{cr,v} \; [-]$	[0.055, 0.21]	Bertoldi et al. (2014), Zen et al. (2016), Li and Millar (2011)

1.4.6 Reconstruction of the position of the mean water table

The key assumption used for the calculation of the position of the mean water table is that the water table level z_w instantaneously match the river water stage, which holds for gravel bed substrates in proximity to the river. z_w is computed by using the Inverse Distance Weighting method (IDW), a popular deterministic model adopted in spatial interpolation. The basic assumption of the model is that the values of any given pair of points are related to each other, but their similarities are inversely related to the distance between their locations (Lu and Wong, 2008). IDW assumes that the unknown value of the variable in location $S_j = (x_j, y_j)$ can be estimated by the observed value at sample location $S_i = (x_i, y_i)$. Here, the unknown variable is represented by the water table elevation in dry cells, $\hat{z}_w$, while its observed value is the water surface elevation in submerged cells, z_w. Formally, given the number of known locations N, the model reads

$$\hat{z}_w(x_j, y_j) = \sum_{i=1}^{N} \alpha_i z_w(x_i, y_i) \tag{1.130}$$

where the estimated value $\hat{z}_w(x_j, y_j)$ is a linear combination of the weights α_i and the

known value $z_w(x_i, y_i)$. α_i can be written as

$$\alpha_i = \frac{d_{j,i}^{-\gamma}}{\sum\limits_{i=1}^{N} d_{j,i}^{-\gamma}} \qquad (1.131)$$

with

$$\sum_{i=1}^{N} \alpha_i = 1 \qquad (1.132)$$

in which $d_{j,i}$ is the distance between S_i and S_j and γ a parameter used to account for the so-called distance-decay effect. IDW, in fact, assumes that the local influence of a know variable decreases with the distance $(d_{j,i})$, where greater weights α_i are given to the points closest to the location S_j. As such, γ measures the rate at which the weights decrease with the distance. If $\gamma = 0$, there is no decrease with the distance and the method results on a mean of the known variables, while at higher values only the immediate surrounding points from the location of the unknown variable affect the prediction.

Because of the decreasing similarities with the distance between any pair of points, to speed calculation, we can exclude the more distant points from the location of the unknown variable S_j. This is obtained by specifying a radius r [m] centered in such location, within which the algorithm searches the known values of the variable (z_w) for the interpolation. In addition, the search can be also limited by a specific number of points surrounding the prediction location (default = 3).

2

Numerics Kernel

2.1 General View

There is great improvement in the development of numerical models for free surface flows and sediment movement in the last decade. The presented number of publications about these subjects proves this clearly. The SWE and the sediment flow equation are a nonlinear, coupled partial differential equations system. A unique analytical solution is only possible for idealised and simple conditions. For practical cases, it is required to implement the numerical methods. A numerical solution arises from the discretisation of the equations. There are different methods to discretize the equations such as:

- Finite difference.............(FD)
- Finite volume................(FV)
- Finite element...............(FE)
- Characteristic Method......(CM)

It is normally distinguished between temporal and spatial discretisation of continuum equations. The latter can be undertaken on different forms of grids such as Cartesian, non-orthogonal, structured and unstructured, while the former is usually done by a FD scheme in time direction, which can be explicit or implicit. The explicit method is usually used for strong unsteady flows.

In FD methods the partial derivations of equations are approximated by using Taylor series. This method is particularly appropriate for an equidistant Cartesian mesh. In FV methods; the partial derivations of equations are not directly approximated like in FD methods. Instead of that, the equations are integrated over a volume, which is defined by nodes of grids on the mesh. The volume integral terms will be replaced by surface integrals using the Gauss formula. These surface integrals define the convective and diffusive fluxes through the surfaces. Due to the integration over the volume, the method is fully conservative. This is an important property of FV methods. It is known that in order to simulate discontinuous transition phenomena such as flood propagation, one must use conservative

numerical methods. In fact, 40 years ago Lax and Wendroff proved mathematically that conservative numerical methods, if convergent, do converge to the correct solution of the equations. More recently, Hou and LeFloch proved a complementary theorem, which says that if a non-conservative method is used, then the wrong solution will be computed, if this contains a discontinuity such as a shock wave (Toro, 2001).

The FE methods originated from the structural analysis field as a result of many years of research, mainly between 1940 and 1960. In this method the problem domain is ideally subdivided into a collection of small regions, of finite dimensions, called finite elements. The elements have either a triangular or a quadrilateral form (Figure 2.1) and can be rectilinear or curved. After subdivision of the domain, the solution of discrete problem is assumed to have a prescribed form. This representation of the solution is strongly linked to the geometric division of sub domains and characterized by the prescribed nodal values of the mesh. These prescribed nodal values must be determined in such way that the partial differential equations are satisfied. The errors of the assumed prescribed solution are computed over each element and then the error must be minimised over the whole system. One way to do this is to multiply the error with some weighting function $\omega(x, y)$, integrate over the region and require that this weighted-average error is zero. The finite volume method has been used in this study; therefore it is explained in more detailed.

In this chapter it will be reviewed how the governing equations comprising the SWE and the bed-updating equation, can be numerically approximated with accuracy. The first section studies the fundamentals of the applied methods, namely the finite volume method and subsequently the hydro- and morphodynamic models are discussed. In section on the hydrodynamic model, the numerical approximate formulation of the SWE, the flux estimation based on the Riemann problem and solver and related problems such as boundary conditions will be analyzed. In the section on morphodynamics, the numerical determination of the bed shear stress and the numerical treatment of bed slope stability will be discussed as well as numerical approximation of the bed-updating equation.

2.2 Methods for Solving the Flow Equations

2.2.1 Fundamentals

2.2.1.1 Finite Volume Method

The basic laws of fluid dynamics and sediment flow are conservation equations; they are statements that express the conservation of mass, momentum and energy in a volume enclosed by a surface. Conversion of these laws into partial differential equations requires sufficient regularity of the solutions.

This condition of regularity cannot always undoubtedly be guaranteed. The case of occurring of discontinuities is a situation where an accurate representation of the conservation laws is important in a numerical method. In other words, it is extremely important that these conservation equations are accurately represented in their integral form. The most natural method to achieve this is obviously to discretize the integral form of the equations and not the differential form. This is the basis of the finite volume (FV) method. Actually the finite volume method is in fact in the classification of the weighted residual method (FE) and hence it is conceptually different from the finite difference method. The weighted function is chosen equal unity in the finite volume method.
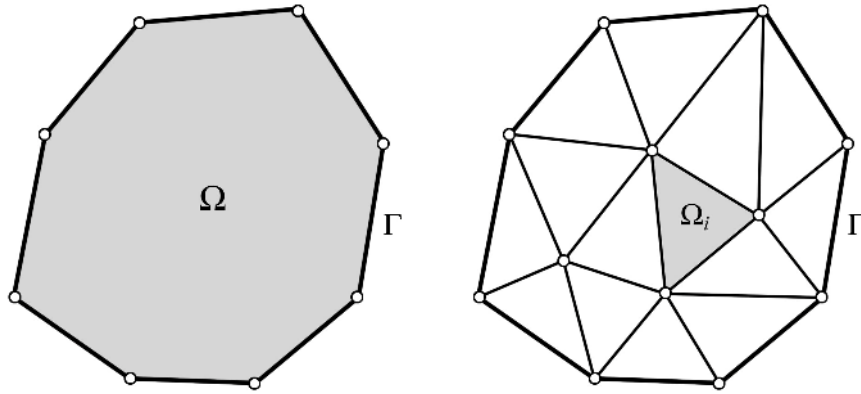
Figure 2.1 *Triangular Finite Elements of a Two-Dimensional Domain*

In this method, the flow field or domain is subdivided, as in the finite element method, into a set of non-overlapping cells that cover the whole domain on which the equations are applied. On each cell the conservation laws are applied to determine the flow variables in some discrete points of the cells, called nodes, which are at typical locations of the cells such as cell centres (cell centred mesh) or cell-vertices (cell vertex mesh) (Figure 2.2).

Obviously, there is basically considerable freedom in the choice of the cell shapes. They can be triangular, quadrilateral etc. and generate a structured or an unstructured mesh. Due to this unstructured form ability, very complex geometries can be handled with ease. This is clearly an important advantage of the method. Additionally the solution on the cell is not strongly linked to the geometric representation of the domain. This is another important advantage of the finite volume method in contrast to the finite element method.

2.2.1.2   The Riemann Problem

Formally, the Riemann problem is defined as an initial-value problem (IVP):

$$
\left.
\begin{array}{l}
\mathbf{U}_t + \mathbf{F}_x(\mathbf{U}) = 0 \\[4pt]
\mathbf{U}(x,0) = \left\{
\begin{array}{lll}
\mathbf{U}_L & \forall & x < 0 \\
\mathbf{U}_R & \forall & x > 0
\end{array}
\right.
\end{array}
\right\}
\tag{2.1}
$$

Here eq. 1.36 and eq. 1.37 are considered for the x-split of SWE. The initial states $\mathbf{U}_L$, $\mathbf{U}_R$ at the left or right side of an triangle edge

$$
\boldsymbol{U}_L = \begin{pmatrix} h_L \\ u_L h_L \\ \nu_L h_L \end{pmatrix}
$$

$$
\boldsymbol{U}_R = \begin{pmatrix} h_R \\ u_R h_R \\ \nu_R h_R \end{pmatrix}
$$

are constant vectors and present conditions to the left of axes $x = 0$ and to the right $x = 0$, respectively (Figure 2.3).

There are four possible wave patterns that may occur in the solution of the Riemann
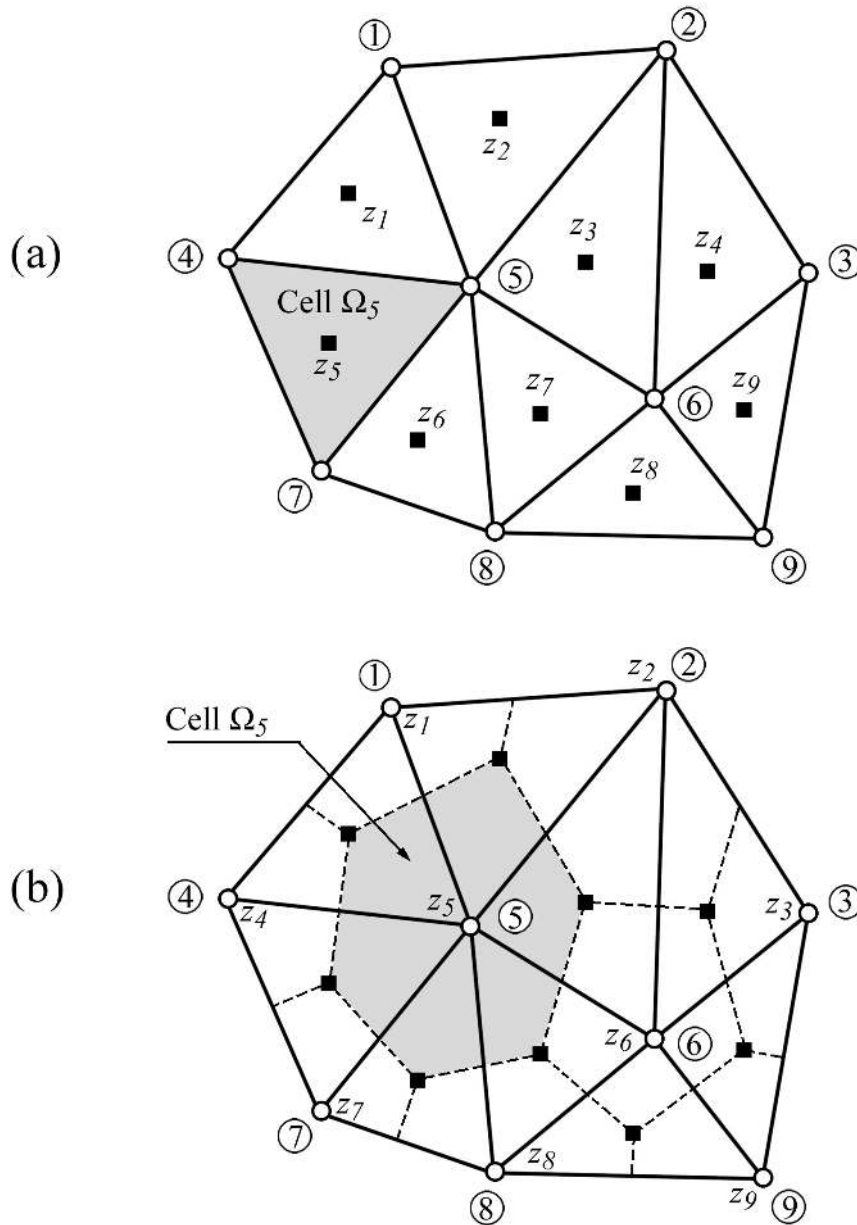
Figure 2.2 *Two-Dimensional Finite Volume Mesh: (a) Cell Centered mesh (b) Cell Vertex mesh*
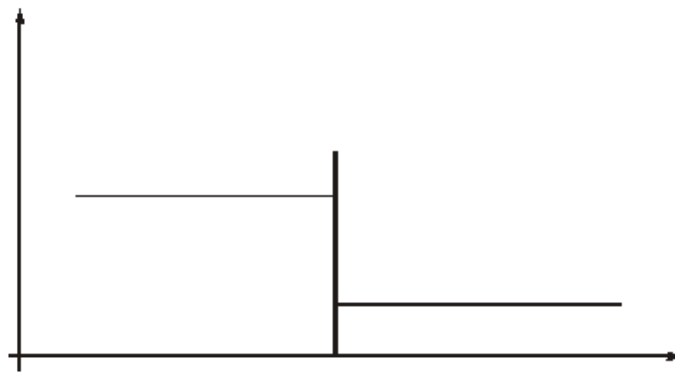
Figure 2.3 *Initial Data for Riemann Problem*
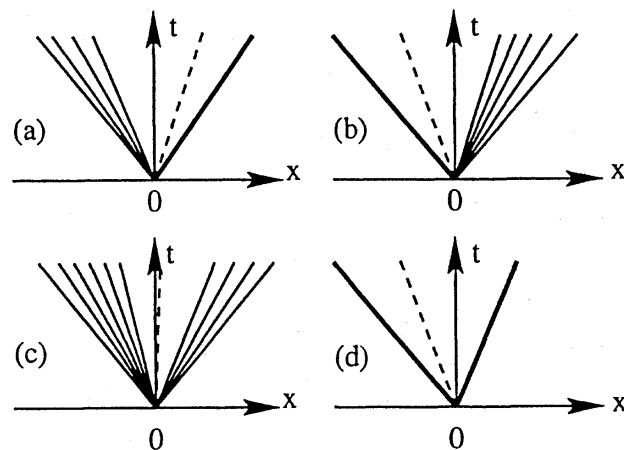
Figure 2.4 *Possible Wave Patterns in the Solution of Riemann Problem*

problem. These are depicted in Figure 2.4. In general, the left and right waves are shocks and rarefactions, while the middle wave is always a shear wave. A shock is a discontinuity that travels with Rankine-Hugoniot shock speed. A rarefaction is a smoothly varying solution which is a function of the variable x/t only and exists if the eigenvalues of $\boldsymbol{J} = \partial \boldsymbol{F}(\boldsymbol{U})/\partial \boldsymbol{U}$ are all real and the eigenvectors are complete. The water depth and the *normal* velocity are both constant across the shear wave, while the *tangential* velocity component changes discontinuously (see Toro (1997) for details about Riemann problem).

By solving the IVP of Riemann (eq. 2.1), the desired outward flux $\boldsymbol{F}(\boldsymbol{U})$ which is at the origin of local axes $x = 0$ and the time $t = 0$ can be obtained.

2.2.1.3   Exact Riemann Solvers

An algorithm, which solves the Riemann initial-value problem is called Riemann solver. The idea of Riemann solver application in numerical methods was used for the first time by Godunov (1959). He presented a shock-capturing method, which has the ability to resolve strong wave interaction and flows including steep gradients such as bores and shear waves. The so called Godunov type upwind methods originate from the work of Godunov. These methods have become a mature technology in the aerospace industry and in scientific disciplines, such as astrophysics. The Riemann solver application to SWE is more recent. It was first attempted by Glaister (1988) as a pioneering attack on shallow water flow problems in 1-D cases.

In the algorithm pioneered by Godunov, the initial data in each cell on either side of an interface is presented by piecewise constant states with a discontinuity at the cell interface (Figure 2.5). At the interface the Riemann problem is solved exactly. The exact solution in each cell is then replaced by new piecewise constant approximation. Godunov's method is conservative and satisfies an entropy condition (Delis et al., 2000). The solution of Godunov is an exact solution of Riemann problem; however, the exact solution is related to a simplified problem since the initial data is approximated in each cell.

To compute numerical solutions by Godunov type methods, the exact Riemann solver or approximate Riemann solver can be used. The choice between the exact and approximate Riemann Solvers depends on:
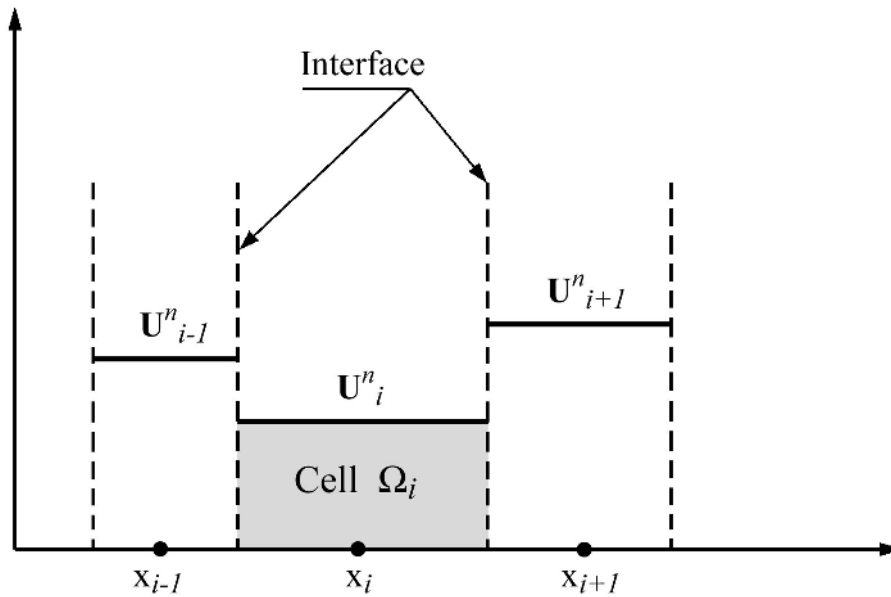
Figure 2.5 *Piecewise constant data of Godunov upwind method*

- Computational cost
- Simplicity and applicability
- Correctness

Correctness seems to be the overriding criterion; however the applicability can be also very important. Toro argued that for the shallow water equations, the argument of computational cost is not as strong as for the Euler equations. For the SWE approximate Riemann solvers may lead to savings in computation time of the order of 20%, with respect to the exact Riemann solvers (Toro, 2001). However, due to the iterative approach of the exact Riemann solver on SWE, its implementation of this will be more complicated if some extra equations, such as dispersion and turbulence equations, are solved together with SWE. Since for these new equations new Riemann solvers are needed, which include new iterative approaches, complications seem to be inevitable. Therefore, an approximate solution which retains the relevant features of the exact solution is desirable. This led to the implementation of approximate Riemann solvers which are non-iterative and therefore, in general, more efficient than the exact solvers.

2.2.1.4   Approximate Riemann Solvers

Several researchers in aerodynamics have developed approximate Riemann solvers for the Euler equations, such as Flux Difference Splitting (FDS) by Roe (1981), Flux Vector Splitting (FVS) by van Leer (1982) and approximate Riemann Solver by Harten et al. (1983); Osher and Solomon (1982). The first two approximate Riemann solvers have been frequently used in aerodynamics as well as in hydrodynamics.

In addition to the exact Riemann solver, the HLL and HLLC approximate Riemann solvers have been implemented in the code.

2.2.1.4.1   HLL Riemann Solver
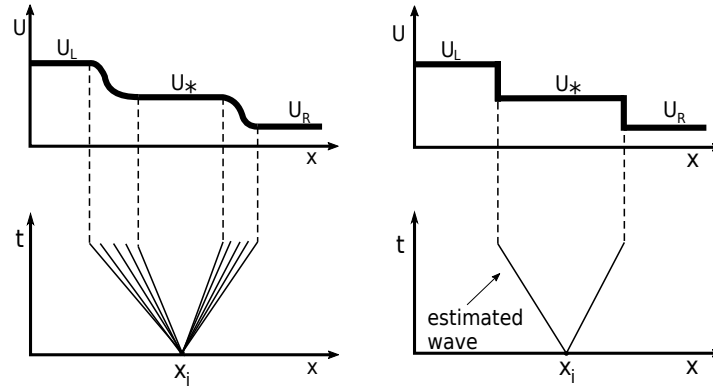
Figure 2.6 *Principle of the HLL Riemann Solver. The analytical solution (left) is replaced by an approximate one with a constant state U_* separated by waves with estimated wave speeds (right)*

The HLL (Harten, Lax and van Leer) approximate Riemann solver devised by Harten et al. (1983) is used widely in shallow water models. It is a Godunov-type scheme based on the two-wave assumption. This approach assumes estimates for the wave speeds of the left and right waves. The solution of the Riemann problem between the two waves is thereby approximated by a constant state as indicated in Figure 2.6. This two-wave assumption is only strictly valid for the one dimensional case. When applied to two dimensional cases, the intermediate waves are neglected in this approach.

The HLL solver is very efficient and robust for many inviscid applications such as SWE.

By applying the integral form of the conservation laws in appropriate control volumes the HLL-flux is derived. The numerical flux over an edge is sampled between three different cases separated by the left and right waves. The indices L and R stand for the left and right states of the local Riemann problem.

$$\boldsymbol{F}_{i+1/2} = \begin{cases} \boldsymbol{F}_L & if \quad 0 \leq S_L, \\ \boldsymbol{F}^{hll} = \dfrac{S_R\boldsymbol{F}_L - S_L\boldsymbol{F}_R + S_RS_L(\boldsymbol{U}_R - \boldsymbol{U}_L)}{S_R - S_L} & if \quad S_L \leq 0 \leq S_R, \\ \boldsymbol{F}_R & if \quad 0 \geq S_R. \end{cases} \quad (2.2)$$

Furthermore, the left and right wave speed velocities are estimated as

$$S_L = u_L - \sqrt{gh_L}q_L \; ; \; S_R = u_R + \sqrt{gh_R}q_R \qquad (2.3)$$

where $q_K(K = L, R)$ are

$$q_K = \begin{cases} \sqrt{\frac{1}{2}\left[\frac{(h_* + h_K)h_*}{h_K^2}\right]} & if \quad h_* > h_K, \\ 1 & if \quad h_* \leq h_K. \end{cases} \qquad (2.4)$$

The quantity h_* is an estimate for the exact solution of the water depth in the star region

obtained using the depth positivity condition. It reads as

$$h_* = \frac{1}{2}(h_L + h_R) - \frac{1}{4}(u_R - u_L)(h_L - h_R)/(\sqrt{gh_L} + \sqrt{gh_R}) \qquad (2.5)$$

In case of dry-bed conditions, the wave speed estimations are modified as follow:

$$S_L = \begin{cases} u_R - 2\sqrt{gh_R} & if \quad h_L = 0, \\ usual estimate & if \quad h_L > 0, \end{cases}$$

$$S_R = \begin{cases} u_L + 2\sqrt{gh_L} & if \quad h_R = 0, \\ usual estimate & if \quad h_R > 0. \end{cases} \qquad (2.6)$$

2.2.1.4.2   HLLC Riemann Solver

A modification and improvement of the HLL approximate solver was proposed by Toro (1994). This so called HLLC approximate Riemann solver also accounts for the impact of intermediate waves, such as shear waves and contact discontinuities, in two dimensional problems. In addition to the estimates of left and right wave speeds, the HLLC solver also requires an estimate for the speed of the middle wave. This middle wave than divides the region between the left and right waves into two constant states.

The numerical flux over an edge is sampled regarding four different cases as

$$\boldsymbol{F}_{i+1/2} = \begin{cases} \boldsymbol{F}_L & if \quad 0 \le S_L, \\ \boldsymbol{F}_{*L} = \boldsymbol{F}_L + S_L(\boldsymbol{U}_{*L} - \boldsymbol{U}_L) & if \quad S_L \le 0 \le S_*, \\ \boldsymbol{F}_{*R} = \boldsymbol{F}_R + S_R(\boldsymbol{U}_{*R} - \boldsymbol{U}_R) & if \quad S_* \le 0 \le S_R, \\ \boldsymbol{F}_R & if \quad 0 \ge S_R. \end{cases} \qquad (2.7)$$

The states $\boldsymbol{U}_{*L}, \boldsymbol{U}_{*R}$ are obtained, as proposed by Toro (1997), from the relations

$$\boldsymbol{U}_{*L} = h_L \left(\frac{S_L - u_L}{S_L - S_*} \right) \begin{bmatrix} 1 \\ S_* \\ \nu_L \end{bmatrix} \quad ; \quad \boldsymbol{U}_{*R} = h_R \left(\frac{S_R - u_R}{S_R - S_*} \right) \begin{bmatrix} 1 \\ S_* \\ \nu_R \end{bmatrix} \qquad (2.8)$$

S_L, S_* and S_R are the estimated wave speeds for the left, middle and right waves. S_L and S_R are estimated according to eq. 2.3 or eq. 2.6 for the dry bed case. with h_* given by eq. 2.5.

The middle wave speed S_* is calculated as proposed by Toro (1997) in a suitable way for taking into account dry-bed problems as

$$S_* = \frac{S_L h_R (u_R - S_R) - S_R h_L (u_L - S_L)}{h_R (u_R - S_R) - h_L (u_L - S_L)} \qquad (2.9)$$
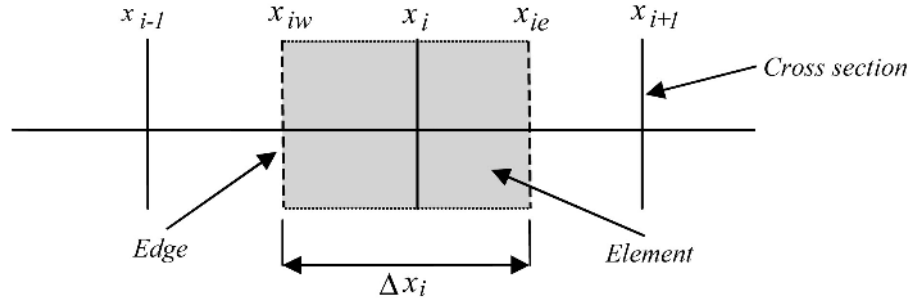
Figure 2.7 *Definition sketch*

2.2.2 Saint-Venant Equations

2.2.2.1 Spatial Discretisation

The time Discretisation is based on the explicit Euler schema, where the new values are calculated considering only values from the precedent time step. The spatial Discretisation of the Saint Venant equations is carried out by the finite volume method, where the differential equations are integrated over the single elements.

It is assumed that values, which are known at the cross section location, are constant within the element. Throughout this section it therefore can be stated that:

$$\int\limits_{x_{iL}}^{x_{iR}} f(x) \; dx \approx f(x_i)(x_{iR} - x_{iL}) = f_i \Delta x_i \tag{2.10}$$

where x_{iR} and x_{iL} are the positions of the edges at the east and the west side of element i, as illustrated in Figure 2.7.

2.2.2.2 Discrete Form of Equations

2.2.2.2.1 Continuity Equation

The eq. 2.10 is integrated over the element:

$$\int\limits_{x_{iL}}^{x_{iR}} \left(\frac{\partial A}{\partial t} + \frac{\partial Q}{\partial x} - q_l \right) dx = 0 \tag{2.11}$$

The different parts of the equation are discretized as follows:

$$\int\limits_{x_{iL}}^{x_{iR}} \frac{\partial A_i}{\partial t} \; dx \approx \frac{\partial A_i}{\partial t} \Delta x_i \approx \frac{A_i^{t+1} - A_i^t}{\Delta t} \Delta x_i \tag{2.12}$$

$$\int\limits_{x_{iL}}^{x_{iR}} \frac{\partial Q_i}{\partial x} \; dx = Q(x_{iR}) - Q(x_{iL}) = \Phi_{c,iR} - \Phi_{c,iL} \tag{2.13}$$

$$\int\limits_{x_{iL}}^{x_{iR}} q_l \ \mathrm{d}x \approx q_{iR}(x_{iR} - x_i) + q_{iL}(x_i - x_{iL}) \tag{2.14}$$

$\Phi_{c,iR}$ and $\Phi_{c,iL}$ are the continuity fluxes calculated by the Riemann solver and q_{iR} and q_{iL} the lateral sources in the corresponding river segments.

For the explicit time discretisation the new value of A will be:

$$A_i^{t+1} = A_i^t - \frac{\Delta t}{\Delta x_i}(\Phi_{c,iR} - \Phi_{c,iL}) - \frac{\Delta t}{\Delta x_i}(q_{iR}(x_i - x_{i,R}) + q_{iL}(x_{iL} - x_i)) \tag{2.15}$$

2.2.2.2.2   Momentum Equation

Assuming that the lateral in- and outflows do not contribute to the momentum balance, thus neglecting the last term of eq. 1.9 and integrating over the element, the momentum equation becomes:

$$\int\limits_{x_{iL}}^{x_{iR}} \left(\frac{\partial Q}{\partial t} + \frac{\partial}{\partial x}\left(\beta \frac{Q^2}{A_{red}}\right) + \sum Sources \right) \mathrm{d}x = 0 \tag{2.16}$$

The different parts of the equation are discretized as follows:

$$\int\limits_{x_{iL}}^{x_{iR}} \frac{\partial Q_i}{\partial t} \ \mathrm{d}x \approx \frac{\partial Q_i}{\partial t} \ \Delta x_i \approx \frac{Q_i^{t+1} - Q_i^t}{\Delta t} \ \Delta x_i \tag{2.17}$$

$$\int\limits_{x_{iL}}^{x_{iR}} \frac{\partial Q_i}{\partial x} \ \mathrm{d}x = \beta \frac{Q^2}{A_{red}}\bigg|_{x_{iR}} - \beta \frac{Q^2}{A_{red}}\bigg|_{x_{iL}} = \Phi_{m,iR} - \Phi_{m,iL} \tag{2.18}$$

$\Phi_{m,iR}$ and $\Phi_{m,iL}$ are the momentum fluxes calculated by the Riemann solver.

For the explicit time Discretisation the new value of Q will be:

$$Q_i^{t+1} = Q_i^t - \frac{\Delta t}{\Delta x_i}(\Phi_{m,iR} - \Phi_{m,iL}) + \sum Sources \tag{2.19}$$

2.2.2.3   Discretisation of Source Terms

2.2.2.3.1   Bed Slope Source Term

The bed slope source term:

$$W = gA\frac{\partial z_S}{\partial x} \tag{2.20}$$

is discretized as follows:

$$\int\limits_{x_{iL}}^{x_{iR}} gA_{red}\frac{\partial z_S}{\partial x}\,dx \approx gA_{redi}\frac{\partial z_S}{\partial x}\bigg|_{x_i}\Delta x_i \approx gA_{redi}\left(\frac{z_{S,i+1}-z_{S,i-1}}{x_{i+1}-x_{i-1}}\right)\Delta x_i \qquad (2.21)$$

With the subtraction of the bed slope source term, eq. 2.19 becomes

$$Q_i^{t+1} = Q_i^t - \frac{\Delta t}{\Delta x_i}(\Phi_{m,ir} - \Phi_{m,il}) - \Delta t\, gA_{redi}\left(\frac{z_{S,i+1}-z_{S,i-1}}{x_{i+1}-x_{i-1}}\right) \qquad (2.22)$$

2.2.2.3.2   Friction Source Term

The friction source term:

$$Fr = gA_{red}S_f$$

is simply calculated with the local values:

$$\int\limits_{x_{iL}}^{x_{iR}} gA_{red}S_{fi}\,\mathrm{d}x \approx gA_{redi}S_{fi}\Delta x_i \qquad (2.23)$$

and

$$S_{fi} = \frac{Q_i^t|Q_i^t|}{(K_i^t)^2} \qquad (2.24)$$

This computation form however leads to problems if the element was dry in the precedent time step, because in this case A, and thus K, become very small, and S_{fi} very large. This leads to numerical instabilities. For this reason a semi-implicit approach has been applied, which considers the discharge of the present time step:

$$S_{fi} = \frac{Q_i^{t+1}|Q_i^t|}{(K_i^t)^2} \qquad (2.25)$$

Consequently instead of simply subtracting the source term from eq. 2.22, the following operation is executed on the new discharge Q_i^{t+1}:

$$Q_i^{t+1} = \frac{Q_i^{t+1}}{1 + \dfrac{\Delta t|Q_i^t|gA_i^t}{(K_i^t)^2}} \qquad (2.26)$$

With this approach the discharge tends to zero for small conveyances.

2.2.2.3.3   Hydraulic Radius / Conveyance

The hydraulic radius and the conveyance of a cross section are calculated by different ways for different types of cross sections, depending on the geometry which is specified by the user. The cross section can be simple or composed by a main channel and flood plains. Additionally it can have a bed bottom.
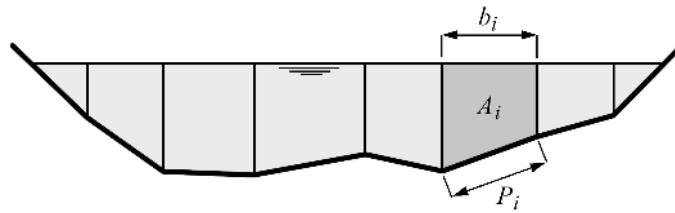
Figure 2.8 *Simple cross section without definition of bed*
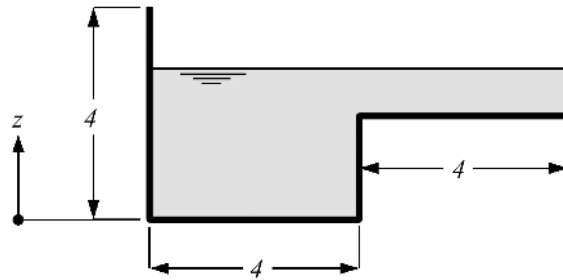
Figure 2.9 *Conveyance computation of a channel with a flat zone*

A simple cross section is used when only a range for the main channel is defined and no flood plain or bottom is specified (Figure 2.8). For an arbitrary simple cross section and a given water surface elevation, the corresponding hydraulic radius R is calculated by the total wetted area A divided by the total wetted perimeter P.

In order to get the conveyance K, a representative friction coefficient c_f for the wetted part of the cross section needs to be determined. This is done by averaging the raw friction values k_i (e.g. Strickler, Manning, Chezy, etc.) over the wetted part of the cross section, weighted with the wetted perimeter. The averaged friction value $\bar{k}$ is then used to calculate the friction coefficient dependent on the function of the friction type. In general form, $c_f = f(\bar{k})$.

$$R = \frac{\sum A_i}{\sum P_i} \tag{2.27}$$

$$K = c_f \sqrt{gR} A \tag{2.28}$$

If there are slices with nearly horizontal ground, such as flood plains for example, this computation mode can lead to jumps in the water level-conveyance graph (see Figure 2.9 and Figure 2.10 for a dimensionless example). It is dangerous however to store the conveyance in a table, from which values will be read by interpolation. If the conveyance is calculated by adding the conveyances of the single slices, the jump can be avoided, but this method leads to an underestimation of the conveyance for water levels higher than the step.

In order to overcome this problem and deal with such cross sections, there is the possibility to define a main channel and (right and left) flood plains, which will be treated differently. Such a cross section is illustrated in Figure 2.11. In this case the conveyance and the discharge are calculated separately for each part of the cross section. The total conveyance results by summation.

Cross section with definition of bed (Figure 2.12)
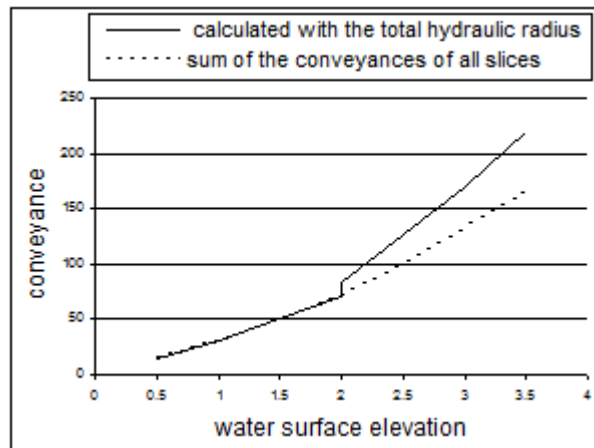
Figure 2.10 *Conveyance computation of a channel with a flat zone*
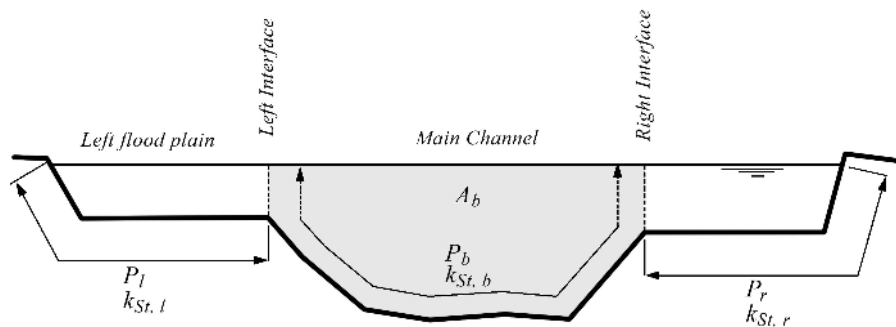
Figure 2.11 *Cross section with flood plains and main channel*
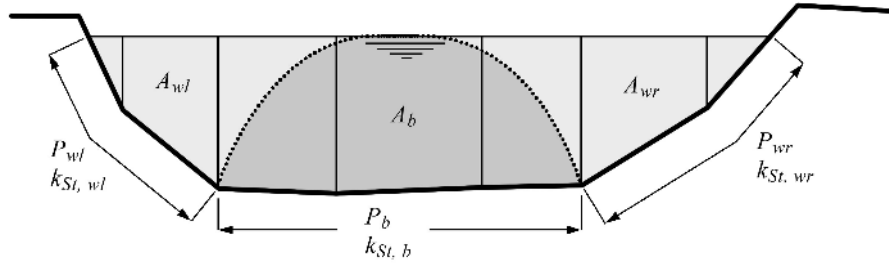
Figure 2.12 *Cross section with definition of a bed*

Additionally to the distinction of flood plains and main channel, a bed bottom can be specified for both simple and composed cross sections. In this case for the computation of the hydraulic radius of the main channel are considered the distinct influence areas of bottom, walls and, if existing, interfaces to the flood plains. In Figure 2.12 is illustrated the case without flood plains or a water level below them.

The conveyance in this case is calculated by the following steps:

$$A_b = R_b P_b \tag{2.29}$$

$$A_{tot} = A_{wl} + A_{wr} + A_b \tag{2.30}$$

$$R_b = \frac{A_{tot}}{k_{stb}^{3/2}\left(\dfrac{P_{wl}}{k_{StWl}^{3/2}} + \dfrac{P_b}{k_{Stb}^{3/2}} + \dfrac{P_{wr}}{k_{StWr}^{3/2}}\right)} \tag{2.31}$$

$$K_b = cf\sqrt{gR_b}A_b \tag{2.32}$$

$$U_{mb} = \frac{K_b}{A_b}\sqrt{S} \tag{2.33}$$

$$Q_b = U_{mb}A_b \tag{2.34}$$

$$Q_w = U_{mb}\left(\frac{A_{tot}}{1.05} - R_b P_b\right) \tag{2.35}$$

$$Q = Q_b + Q_w \tag{2.36}$$

$$K = Q/\sqrt{S} \tag{2.37}$$

$$K = K_b + K_w = \frac{Q_b}{\sqrt{S}} + \frac{Q_w}{\sqrt{S}} = \frac{K_b}{A_b}R_b P_b + \frac{K_b}{A_b}\left(\frac{A_{tot}}{1.05} - R_b P_b\right) = K_b + K_b\left(\frac{A_{tot}}{1.05A_b} - 1\right) \tag{2.38}$$
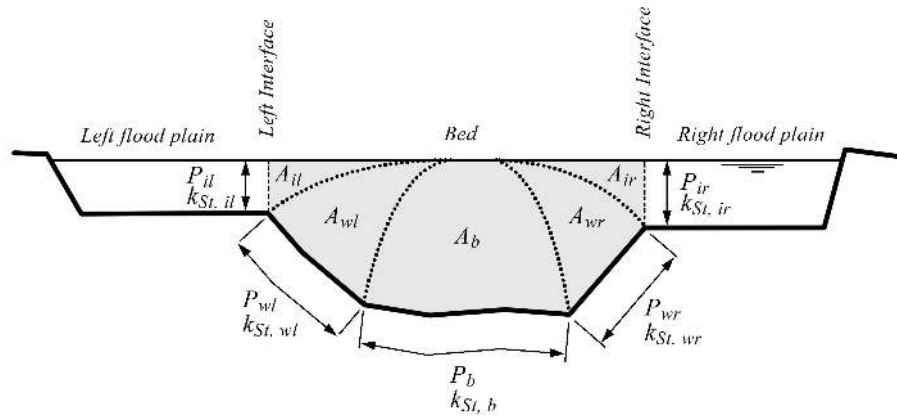
Figure 2.13 *Cross section with flood plains and definition of a bed bottom*
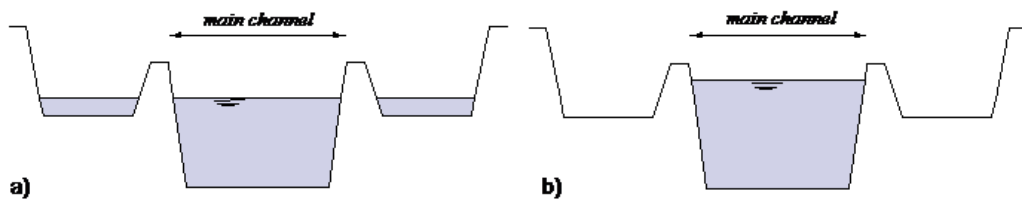
Figure 2.14 *Water level in a cross section with internal levees: a) without consideration of the levees, b) with consideration of the levees*

Figure 2.13 shows the case of a water level higher than the flood plains.

In this case two more partial areas are distinguished for the computation of the hydraulic radius of the bed and:

$$R_b = \frac{A_{tot}}{k_{Stb}^{3/2} \left(\dfrac{P_{il}}{k_{Stil}^{3/2}} + \dfrac{P_{wl}}{k_{Stwl}^{3/2}} + \dfrac{P_b}{k_{Stb}^{3/2}} + \dfrac{P_{wr}}{k_{Stwr}^{3/2}} + \dfrac{P_{ir}}{k_{Stir}^{3/2}} \right)} \qquad (2.39)$$

2.2.2.3.4   Simple approach for consideration of internal levees

From the wetted surface A provided by the equation solution, the corresponding water surface elevation z has to be determined. All further hydraulic variables in the cross section are computed from this value. A special problem occurs if there are levees in the channel. The very simple 1-D approach leads to the situation in illustration a) of Figure 2.14. The reality however would correspond more to illustration b).

The first approach can lead to problems i.e. for the computation of bed load, as part of the shear stress is lost. For this reason the regions out of the main channel are considered only when the water level exceeds the top level of the levees. When the whole main channel is full and the wetted area increases, the water surface elevation remains constant until the flood plains are "filled". Only then the water surface elevation will continue to rise, like illustrated in Figure 2.15. The second approach is used as default, but it can be switched-off.
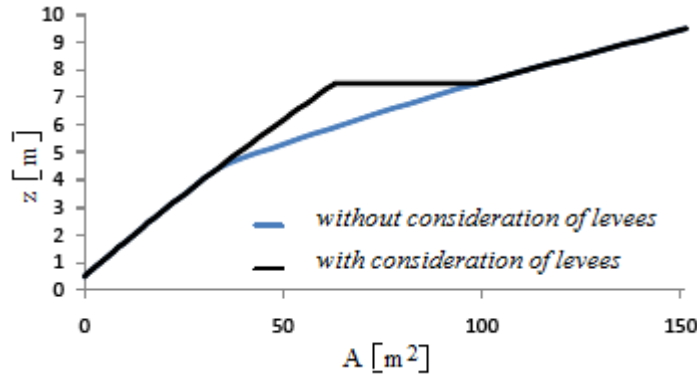
Figure 2.15 *Function $z(A)$ with and without consideration of the levees*

2.2.2.4 Discretisation of Boundary conditions

While normally the edges are placed in the middle between two cross sections, which are related to the elements, at the boundaries the left edge of the first upstream element and the right edge of the last downstream element are situated at the same place as the cross sections themselves. Thus there is no distance between the edge and the cross section.

2.2.2.4.1 Inlet Boundary

The boundary condition is applied to the left edge of the first element, where it serves to determine the fluxes over the element side. If there is no water coming in, these fluxes are simply set to 0.

If there is an inlet flux it must be given as a hydrograph. The given discharge is directly used as the continuity flux, whereas for the computation of the momentum flux, A_{red} and β are determined in the first cross section (which has the same location).

$$\Phi_{c,1R} = Q_{bound} \tag{2.40}$$

$$\Phi_{m,1R} = \beta_1 \frac{(Q_{bound})}{A_{red1}} \tag{2.41}$$

For the computation of the bed source term in the first cell, the values in the cell are used instead of the lacking upstream values:

$$W_1 = A_{red1} \left(\frac{z_{S,2} - z_{S,1}}{x_2 - x_1} \right) \tag{2.42}$$

2.2.2.4.2 Outlet Boundary

- weir and gate:

  The wetted area A_N^{n-1} of the last cross section at the previous time step is used to determine the water surface elevation $z_{S,N}$. With $z_{S,N}$, the wetted area of the weir or gate and finally the out flowing discharge Q are calculated, which are used for the computation of the flux over the outflow edge:

$$\Phi_{c,NL} = Q_{weir} \tag{2.43}$$

$$\Phi_{m,NL} = \frac{(Q_{weir})^2}{A_{weir}} \tag{2.44}$$

- z(t) and z(Q):

    If the given boundary condition is a time evolution of the water surface elevation or a rating curve, the discharge Q is taken from the last cell and time step. In the case of a rating curve it is used to determine the water surface elevation z_S . The given elevation is used to calculate the target A and β in the last cross section.

    To satisfy the desired water level, the required inflow from the reservoir into the simulation domain is calculated from:

$$Q_{res}(t) = Q_{in}(t) - \frac{\Delta x}{\tau}(A_{target} - A(t)) \tag{2.45}$$

    This approach converges to the target water surface elevation within the characteristic time τ.

$$\Phi_{c,NL} = Q_{res} \tag{2.46}$$

$$\Phi_{m,NL} = \beta_{bound}\frac{(Q_{res})^2}{A_{bound}} \tag{2.47}$$

- In/out:

    With the boundary condition in/out the flux over the outflow edge is just equal to the inflow flux of the last cell:

$$\Phi_{c,Ll} = \Phi_{c,NR} \tag{2.48}$$

$$\Phi_{m,NL} = \Phi_{m,NR} \tag{2.49}$$

    For the computation of the bed source term in the last cell, the values in the cell are used instead of the lacking downstream values:

$$W_N = A_{redN}\left(\frac{z_{S,N} - z_{S,N-1}}{x_N - x_{N-1}}\right) \tag{2.50}$$

2.2.2.4.3   Inner Boundaries

- Inner Weir:

    The inner weir uses a slightly other approach than the boundary weir. If the weir crest is higher than the water surface elevation in the neighbouring elements, the weir acts as a wall.

If one or both of the neighbouring water surface elevations are above the weir crest, the weir formula for discharge

$$Q = \frac{2}{3}\mu\sigma_{uv}w\sqrt{2g(z_{upstream} - z_{weir})^3} \qquad (2.51)$$

is used. This is the classical Poleni formula for a sharp crested weir with an additional factor σ_{uv} which accounts for the reduction in discharge due to incomplete weir flow. If only one side of the weir has a water surface elevation above the weir crest, then a complete weir flow is given with $\sigma_{uv} = 1$. w is the width of the weir.

As soon as the water surface elevation tops the weir crest level on both sides of the weir, the incomplete case is active and the reduction factor σ is calculated according to the Diagram Figure 2.32.

As for the momentum, there is no momentum due to velocity accounted for in the downstream direction. This behaviour acts as if all kinetic energy is dissipated over a weir.

2.2.2.4.4   Moving Boundaries

Boundaries are always considered to be on an edge. A moving boundary appears if one of the cells, limited by the edge, is dry. In this case some changes have to be considered for the computations:

- Flux over an internal edge:

  If in one of the cells the water depth at the lowest point of the cross section is lower than the dry depth h_{min} , the energy level in the other cell is computed. If this is lower than the water surface elevation in the dry cell the flow over the edge is 0. Otherwise it is calculated normally.

- Bed source:

  For the computation of the bed source term in the case of the upstream or downstream cell being dry, the local values are used as in the following example with an upstream dry cell:

$$W_i = A_{red,i}\left(\frac{z_{S,i+1} - z_{S,i}}{x_{i+1} - x_i}\right) \qquad (2.52)$$

2.2.2.5   Solution Procedure

a) General solution procedure of BASEchain in detail (Figure 2.16 and Figure 2.17)

b) Time loop (Figure 2.18)

c) Hydrodynamic equations (Figure 2.19)

d) Morphodynamic equations (Figure 2.20)

e) Suspended load equations (Figure 2.21)
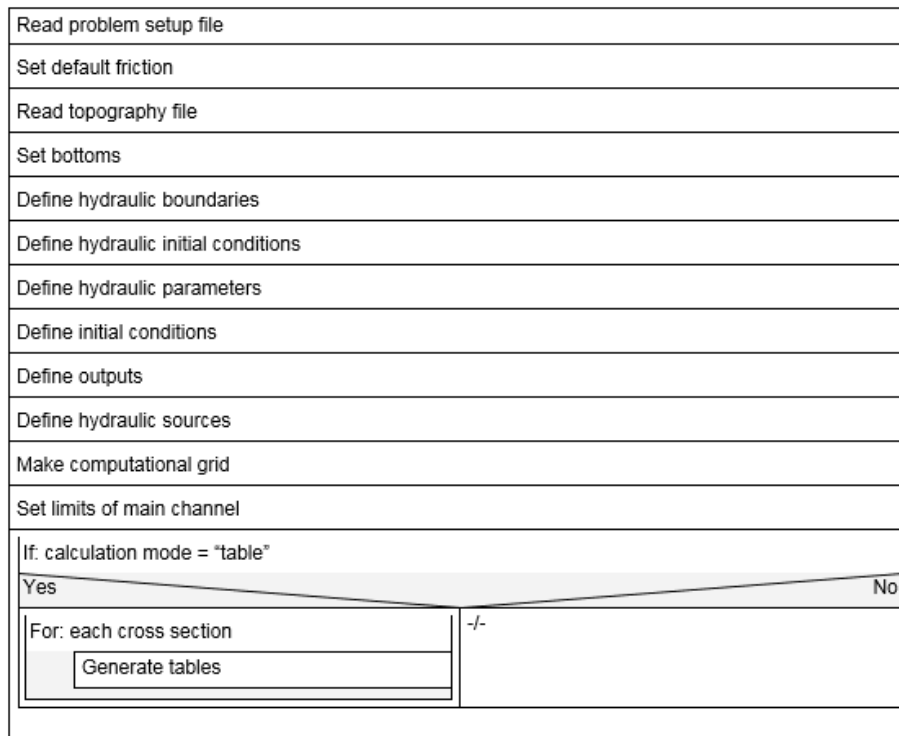
Figure 2.16 *General solution procedure of BASEchain*

2.2.3 Shallow Water Equations

2.2.3.1 Discrete Form of Equations

Numerical methods are used to transform the differential and the integral equations into discrete algebraic equations. Based on the mentioned reasons, the FV method has been used in the present work for the discretisation of SWE. The eq. 1.36 can be rewritten in the following integral form:

$$\int_\Omega \boldsymbol{U}_t \, \mathrm{d}\Omega + \int_\Omega \nabla \cdot (\boldsymbol{F}, \boldsymbol{G}) \, \mathrm{d}\Omega + \int_\Omega \boldsymbol{S} \, \mathrm{d}\Omega = 0 \tag{2.53}$$

in which Ω equals the area of the calculation cell (Figure 2.1) Using the Gauss' relation eq. 2.53 becomes:

$$\int_\Omega \boldsymbol{U}_t \, \mathrm{d}\Omega + \int_{\partial\Omega} (\boldsymbol{F}, \boldsymbol{G}) \cdot \boldsymbol{n}_s \, \mathrm{d}l + \int_\Omega \boldsymbol{S} \, \mathrm{d}\Omega = 0 \tag{2.54}$$

Assuming U_t and S are constant over the domain for first order accuracy, it can be written:

$$\boldsymbol{U}_t + \frac{1}{\Omega} \int_{\partial\Omega} (\boldsymbol{F}, \boldsymbol{G}) \cdot \boldsymbol{n}_s \, \mathrm{d}l + \boldsymbol{S} = 0 \tag{2.55}$$

The eq. 2.54 can be discretized by a two-phase scheme namely predictor corrector scheme as follows:

If: sediment transport is active (bed load or suspended load)	
Yes / No	
Define general sediment parameters	-/-
Define grain classes	-/-
Define mixtures	-/-
Define soils	

If: bedload is active	
Yes / No	
Set bedload specific parameters	
Set default transport type factors	
Define bed load boundaries	
Set bed load fluxes	
Set transport capacities	

If: suspended load is active	
Yes / No	
Initialize suspended load	
Define suspended load boundaries	
Set advection-diffusion fluxes	

If sediment exchange with soil is active:	
Yes / No	
Crate exchange source term	

Print topography output
Print initial conditions to output files
Generate the needed sources

Figure 2.17 *General solution procedure of BASEchain*
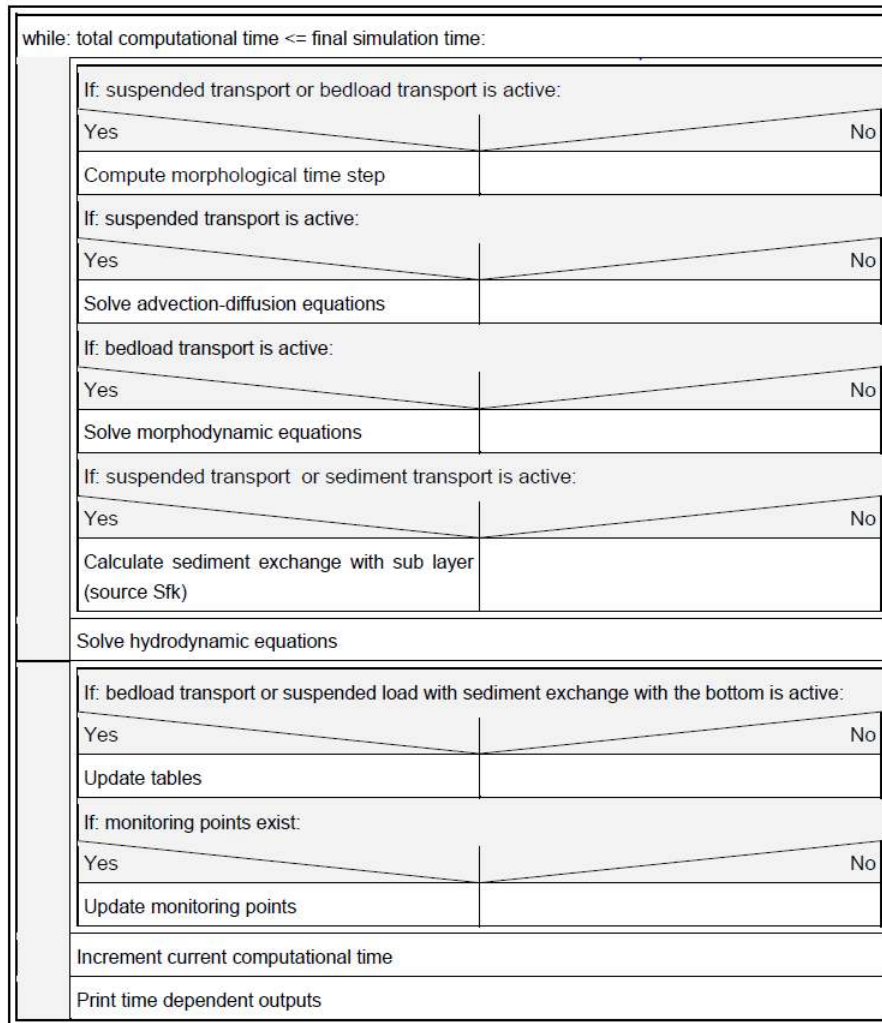
Figure 2.18 *Time loop*
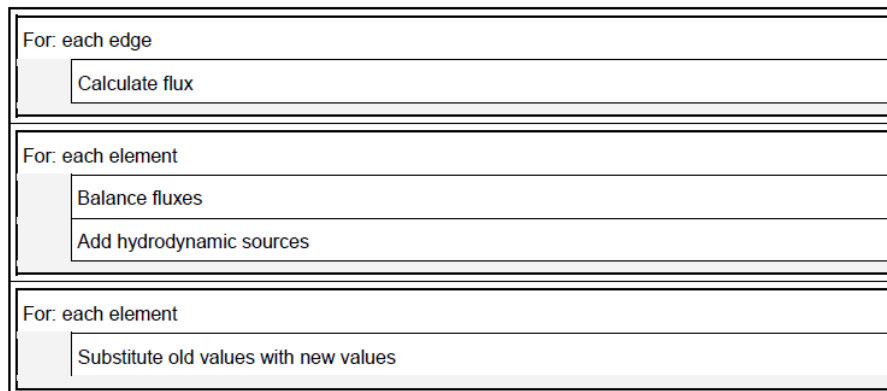
Figure 2.19 *Hydrodynamic equations*

For: each element
  Calculate transport width of cross section
  Reset transport capacities

For: each edge
  Calculate sediment fluxes

For: each element
  Calculate local sources
  Balance global material conservation
  Add local sources
  Determine bed level change for cross section
  For: each slice
    Update layer positions
  For: each grain class
    Balance material sorting equation
    Add local sources
    Calculate sediment exchange with sub layer (source Sf)
    Add exchange with sub layer
  For: each slice
    Correct all   ☐

Figure 2.20 *Morphodynamic equations*

For: each element
  Calculate wetted width of cross section

For: each edge
  Calculate advection and diffusion fluxes

For: each element
  Calculate exchange sources (Sk))
  Calculate local sources (Slk)
  Calculate new concentrations
  Solve Exner equation
  Solve grain sorting equation
  Update cross section geometry
Advance all values
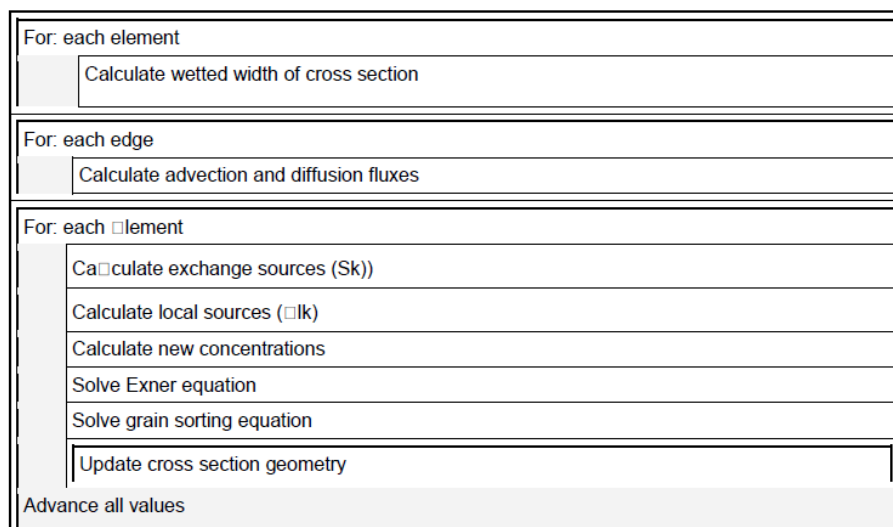
Figure 2.21 *Suspended load equations*

$$U_i^{n+1} = U_i^n - \frac{\Delta t}{\Omega} \sum_{j=1}^{3} (\boldsymbol{F}, \boldsymbol{G})_{i,j}^n \times \boldsymbol{n}_j l_j - \Delta t \boldsymbol{S}_i \tag{2.56}$$

where

m	= number of cell or element sides
$(\boldsymbol{F}, \boldsymbol{G})_{i,j}$	= numerical flux through the side of cell
$n_{s,i}$	= unit vector of cell side

The advantage of two-phase scheme is the second order accuracy in time marching. In the FV method, the key problem is to estimate the normal flux through each side of the domain, namely $((\boldsymbol{F}, \boldsymbol{G}) \cdot \boldsymbol{n}_s)$. There are several algorithms to estimate this flux. The set of SWE is hyperbolic and, therefore, it has an inherent directional property of propagation. For instance, in 1-D unsteady flow, information comes from both, upstream and downstream, in sub critical cases, while information only comes from upstream in supercritical cases. Algorithms to estimate the flux should appropriately handle this property. The *Riemann solver*, which is based on characteristics theory, is such an algorithm. It is the solution of Riemann Problem. The Riemann solver under the FV method formulation is especially suitable for capturing discontinuities in sub critical or supercritical flow, e.g. a dam break wave or flood propagation in a river.

2.2.3.1.1   Flux Estimation

Considering the integral term of flux in eq. 2.55, it can be written:

$$\int_{\partial\Omega} (\boldsymbol{F}(U), \boldsymbol{G}(U)) \cdot \boldsymbol{n}_s \, \mathrm{d}l = \int_{\partial\Omega} (\boldsymbol{F}(U) \cos\theta + \boldsymbol{G}(U) \sin\theta) \, \mathrm{d}l \tag{2.57}$$

in which $\boldsymbol{n}_s = (\cos\theta, \sin\theta)$ is the outward unit vector to the boundary of domain Ω_i (see Figure 2.22). Based on the rotational invariance property for $\boldsymbol{F}(U)$ and $\boldsymbol{G}(U)$ on the boundary of the domain, it can be written according to Toro (1997):

$$(\boldsymbol{F}(U), \boldsymbol{G}(U)) = \boldsymbol{T}^{-1}(\theta) \boldsymbol{F}(\boldsymbol{T}(\theta) U) \tag{2.58}$$

where θ is the angle between the vector $\boldsymbol{n}_s$ and x-axis, measured counter clockwise from the x-axis (see Figure 2.22).

$$\boldsymbol{T}(\theta) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & \sin\theta \\ 0 & -\sin\theta & \cos\theta \end{pmatrix} \tag{2.59}$$

$\boldsymbol{T}^{-1}(\theta) = $ inverse of $\boldsymbol{T}(\theta)$

Using eq. 2.58, eq. 2.55 can be rewritten as:

$$U_t + \frac{1}{\Omega} \int_{\partial\Omega} \boldsymbol{T}^{-1}(\theta) \boldsymbol{F}(\boldsymbol{T}(\theta) U) \, \mathrm{d}l + \boldsymbol{S} = 0 \tag{2.60}$$

The quantity $\boldsymbol{T}(\theta) U$ is transformed of U, with velocity components in the normal and tangential direction. For each cell in the computational domain, the quantity U, thus
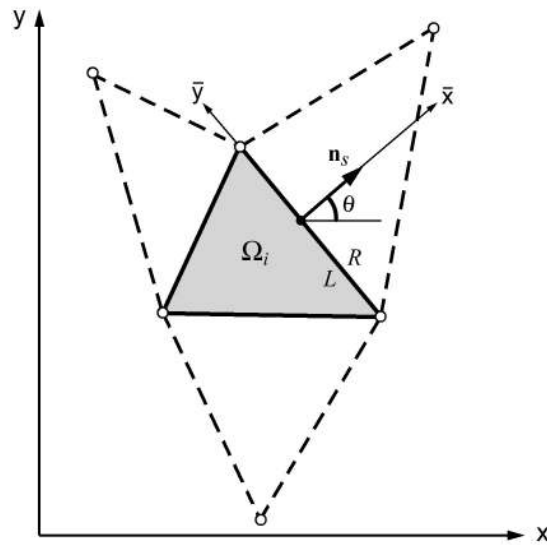
Figure 2.22 *Geometry of a Computational Cell Ω_i in FV*

$\boldsymbol{T}(\theta)\boldsymbol{U}$ may have different values, which results in a discontinuity across the interface between cells. Therefore, the two-dimensional problem in eq. 2.53 or eq. 2.55 can be handled as a series of local Riemann problems in the normal direction to the cell interface ($\bar{x}$) by the eq. 2.60.

Applying the foregoing, the flux computations over the edges are preformed in three successive steps:

- First, the vector of conserved variables $\boldsymbol{U}$ is transformed into the local coordinate system at the edge with the operation $\boldsymbol{T}(\theta)\boldsymbol{U}$.

- A one-dimensional, local Riemann problem is formulated and solved in the normal direction of the edge. From this calculation results the new flux vector over the edge $\boldsymbol{F}[\boldsymbol{T}(\theta)\boldsymbol{U}]$.

- The flux vector, formulated in the local coordinate system at the edge, is transformed back to Cartesian coordinates with $\boldsymbol{T}^{-1}\boldsymbol{F}[\boldsymbol{T}(\theta)\boldsymbol{U}]$. The Sum of the fluxes of all edges of an element gives the total fluxes in x- and y directions.

2.2.3.1.2 Flux Correction

When the solution is advanced and the continuity and momentum equations are updated in each cell, there may be occurring situations in which more water is removed from an element than is actually stored in the element (overdraft). Such overdraft is mostly experienced in situations with strongly varying topography and low water depths, e.g. near wet-dry interfaces on irregular beds. To guarantee positive depths in all elements, a correction of the depths and volumetric fluxes is applied in such situations following an approach based on Begnudelli and Sanders (2006). However, in some rare cases the overdraft cannot be corrected and therefore mass continuity is not guaranteed.

The overdraft element i having a negative water depth receives water from its surrounding element k if two conditions are fulfilled. The element k must previously have taken water from the overdraft element and it must have water available. The corrections of the depths and volumetric fluxes of the neighbouring elements k are then calculated as

$$h_k^{corr} = h_k + \omega_k h_i \frac{A_i}{A_k}$$

$$Flux_k^{corr} = Flux_k + \omega_k h_i \frac{A_i}{\Delta t} \tag{2.61}$$

where h_k is the water depth and $Flux_k$ is the volumetric Flux of the neighbouring element. h_i is the (negative) water depth of the overdraft element and ω_k is a weighting factor which is obtained by weighting the volumetric fluxes of all corrected neighbouring elements k.

$$\omega_k = \frac{Flux_k}{\sum\limits_k Flux_k} \tag{2.62}$$

In case of element k does not have enough water available the overdraft is partly compensated. Subsequently all the weights have to be recalculated and a new correction attempt is made. After the correction of the neighbouring elements, the water depth of the overdraft element is set to zero.

2.2.3.2 Discretisation of Source Terms

In eq. 2.56 there are different possibilities for the evaluation of the source term $\boldsymbol{S}_i$. It can be evaluated either with the variables of the old time step as $\boldsymbol{S}_i(\boldsymbol{U}_i)$, which is often referred to as unsplitted scheme, or it can be evaluated with the advanced values $\boldsymbol{U}_i^{n+1/2}$, which already include changes due to the numerical fluxes computed during this time step as $\boldsymbol{S}_i(\boldsymbol{U}_i^{n+1/2})$. The use of the advanced values for the source term calculation is chosen here because it gives better results (Toro, 2001). Therefore eq. 2.56 is split in following way

$$\boldsymbol{U}_i^{n+1/2} = \boldsymbol{U}_i^n - \frac{\Delta t}{\Omega} \sum_{j=1}^{3} (\boldsymbol{F}, \boldsymbol{G})_{i,j}^n \cdot \boldsymbol{n}_j \mathrm{d}l_j$$

$$\boldsymbol{U}_i^{n+1} = \boldsymbol{U}_i^{n+1/2} + \Delta t \boldsymbol{S}_i(\boldsymbol{U}_i^{n+1/2}) \tag{2.63}$$

But, as explained in the following, the friction source term $\boldsymbol{S}_{i,fr}$ receives a special treatment.

2.2.3.2.1 Friction Source Term

When treating the friction source terms, a simple explicit Discretisation may cause numerical instabilities if the water depth is very small, because the water depth is in the denominator. Such problematic situations may occur in particular at drying-wetting interfaces. To circumvent the numerical instabilities, the frictions terms are treated in a semi-implicit way. Therefore the friction source term is calculated with the unknown value $\boldsymbol{U}_i^{n+1}$ at the new time level as

$$\boldsymbol{U}_i^{n+1} = \boldsymbol{U}_i^{n+1/2} + \Delta t \boldsymbol{S}_{i,fr}(\boldsymbol{U}_i^{n+1}) \tag{2.64}$$

Considering the generalized c_f friction coefficient and after some algebraic manipulations, one obtains:
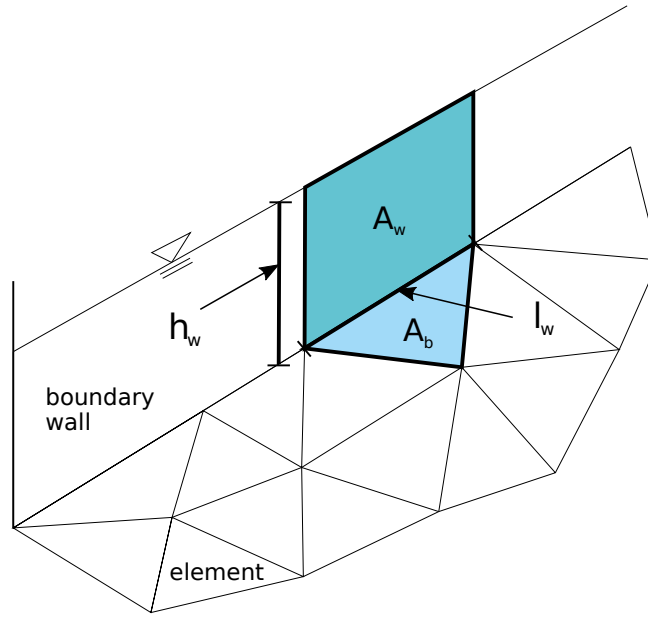
Figure 2.23 *Wetted perimeter of a boundary element with wall friction*

$$U_i^{n+1} = \frac{U_i^{n+1/2}}{1 + \Delta t \dfrac{\sqrt{(u_i^n)^2 + (\nu_i^n)^2}}{c_{fi}^2 R_i}} \tag{2.65}$$

Hydraulic Radius

The calculation of the friction source term requires a definition of the hydraulic radius R_i in the element i. The hydraulic radius is defined here as water depth in the element ($R_i = h_i$).

Wall friction

In cases where an element is situated at a boundary wall of the domain, the influence of the additional wall friction on the flow can optionally be considered, as illustrated in Figure 2.23. The friction slope is extended to include additional wall friction effects. The method is similar to the approach of Brufau and Garcia-Navarro and Vazquez-Cendon (2000) but differs in the type the different friction parts are added together. In this implementation the friction values of the bed c_f and the wall c_{fw} can be chosen differently.

The friction slope in x-direction is calculated as

$$S_{i,fr,x} = \frac{u\sqrt{u^2 + \nu^2}}{g}\left(\frac{1}{\bar{c}_f^2 R_w}\right) \tag{2.66}$$

with R_w being the hydraulic radius of the wall friction at the boundary edge w. The first term in eq. 2.66 defines the friction losses due to bed friction and the second term defines the additional friction losses due to the flow along the boundary wall. The friction slope in y-direction is derived in an analogous way.

The hydraulic radius R_w at the wall boundary edge i is calculated as

$$R_w = \frac{V_{water}}{A_w} = \frac{A_b h}{A_b + \sum_i l_{w,i} h} \tag{2.67}$$

where I_w is the length of the element's edge located at the wall boundary, A_b is bottom area of the element and A_w is the wetted area of the wall. The average friction coefficient $\bar{c}_f$ at the boundary edge is calculated as

$$\bar{c}_f = \frac{c_f A_b + c_{fw} \sum_i A_{w,i}}{A_b + \sum_i A_{w,i}}$$

with c_{fw} as the friction coefficient of the wall and $\sum_i A_{w,i}$ as the sum of all wetted wall areas.

For the determination of the bottom shear stress for sediment transport computations, this additional wall friction component is not taken into account.

2.2.3.2.2   Source Term for viscous and turbulent Stresses

The kinematic and turbulent stresses are treated as source term. For the derivatives, the divergence theorem from Gauss is used similarly to the ordinary fluxes. This approach allows a derivative to be calculated as a sum over averaged values on an edge. A potential division by zero cannot occur. In the following, a cantered scheme for diffusive fluxes on an unstructured grid based on the approach of Mohamadian et al. (2005) is described.

For the Finite Volume Method, the diffusive source terms are integrated over an element:

$$\iint_\Omega \boldsymbol{S}_d \, \mathrm{d}\Omega = \iint_\Omega \frac{\partial \boldsymbol{F}^d}{\partial x} + \frac{\partial \boldsymbol{G}^d}{\partial y} \, \mathrm{d}\Omega \tag{2.68}$$

By using the divergence theorem, the diffusive flux integrals are becoming boundary integrals

$$\iint_\Omega \frac{\partial \boldsymbol{F}^d}{\partial x} + \frac{\partial \boldsymbol{G}^d}{\partial y} \, \mathrm{d}\Omega = \oint_{\partial\Omega} \boldsymbol{F}^d \cdot \boldsymbol{n} + \boldsymbol{G}^d \cdot \boldsymbol{n} \, \mathrm{d}s \tag{2.69}$$

The boundary integral is discretized by a summation over the element edges (index e)

$$\oint_{\partial\Omega} \boldsymbol{F}^d \cdot \boldsymbol{n} + \boldsymbol{G}^d \cdot \boldsymbol{n} \, \mathrm{d}s = \sum_e (\boldsymbol{F}_e^d \cdot \boldsymbol{n}_e + \boldsymbol{G}_e^d \cdot \boldsymbol{n}_e) \, \mathrm{d}s_e \tag{2.70}$$

The diffusive fluxes $\boldsymbol{F}_e^d$ and $\boldsymbol{G}_e^d$ on the element edges are calculated by a centred scheme

$$\boldsymbol{F}_e^d = \frac{1}{2}(\boldsymbol{F}_R^d + \boldsymbol{F}_L^d), \quad \boldsymbol{G}_e^d = \frac{1}{2}(\boldsymbol{G}_R^d + \boldsymbol{G}_L^d) \tag{2.71}$$

where R and L stand for a value right and left of the edge. The diffusive fluxes on the edges read then as

$$
\boldsymbol{F}_e^d = \frac{1}{2}
\begin{pmatrix}
0 \\
\left(\nu h \dfrac{\partial u}{\partial x}\right)_R + \left(\nu h \dfrac{\partial u}{\partial x}\right)_L \\
\left(\nu h \dfrac{\partial v}{\partial x}\right)_R + \left(\nu h \dfrac{\partial v}{\partial x}\right)_L
\end{pmatrix}
\tag{2.72}
$$

and

$$
\boldsymbol{G}_e^d = \frac{1}{2}
\begin{pmatrix}
0 \\
\left(\nu h \dfrac{\partial u}{\partial y}\right)_R + \left(\nu h \dfrac{\partial u}{\partial y}\right)_L \\
\left(\nu h \dfrac{\partial v}{\partial y}\right)_R + \left(\nu h \dfrac{\partial v}{\partial y}\right)_L
\end{pmatrix}
\tag{2.73}
$$

where $\nu = \nu_m + \nu_t$ is the sum of the molecular (kinematic) and turbulent eddy viscosity. For this approach, the velocity derivatives at the element canters are used as right and left approximation near the edge. The values for the water depth h right and left of an edge are reconstructed using the water surface elevation of the adjacent elements. The turbulent eddy viscosity ν_t can be either set to a constant value or calculated dynamically for each element. Using the dynamic case, the values for ν_t are taken from the right and left element of an edge.

All that remains is to calculate the derivatives of the velocity components at the element centres. For Finite Volume Methods, this is an easy task using again the divergence theorem. The derivative of a general scalar variable φ on an arbitrary element is given by

$$
\left(\frac{\partial \varphi}{\partial x}\right)_{Elem} = \frac{1}{\Omega} \int\limits_{\Omega} \frac{\partial \varphi}{\partial x} \ d\Omega \approx \frac{\sum\limits_{e} \varphi_e \Delta y_e}{\Omega}
\tag{2.74}
$$

$$
\left(\frac{\partial \varphi}{\partial y}\right)_{Elem} = \frac{1}{\Omega} \int\limits_{\Omega} \frac{\partial \varphi}{\partial y} \ d\Omega \approx \frac{\sum\limits_{e} \varphi_e \Delta x_e}{\Omega}
\tag{2.75}
$$

where Ω is the area of the element and e stands for an edge. φ_e is a value on the edge. As in finite volume methods, most variables are defined on an element, φ_e has to be calculated as average of the neighbouring elements:

$$
\varphi_e = \frac{1}{2}(\varphi_R + \varphi_L)
\tag{2.76}
$$

The spatial differences Δy_e and Δx_e are the differences of the edge's node-coordinates in x and y direction. For this method to work, it is important to have the same direction of integration along the elements edges either clockwise or counter-clockwise.

As a result, a viscous term from eq. 2.72 is computed as

$$\left(\nu h \frac{\partial u}{\partial x}\right)_L \approx \nu_L h_L \frac{\sum_e u_e \Delta y_e}{\Omega_L} \tag{2.77}$$

with $u_e = 0.5(u_{eL} + u_{eR})$.

The depth-averaged turbulent viscosity ν_t can either be set to a constant value or it is calculated for every element using the formula

$$\nu_t = \frac{\kappa}{6} u_* h \tag{2.78}$$

Where $\kappa = 0.4$ is the von Karman constant and u_* is the shear velocity which is defined as

$$u_* = \sqrt{c_f(u^2 + v^2)} \tag{2.79}$$

Where c_f is the bed friction coefficient derived from the same Manning- or Strickler-value as defined for bed friction.

2.2.3.2.3   Bed Slope Source Term and Bed Slope Calculation

The irregularity of the topography plays an important role in real world applications and often can have great impacts on the final accuracy of the results. A Discretisation scheme with the elevations defined in the nodes of a cell leads to an accurate representation of the topography. Special attention thereby is needed with regard to the C-property which is discussed in Section 2.2.3.3.

The numerical treatment of the bed slope source term here is formulated based on Komaei's method (Komaei and Bechteler, 2004). Regarding eq. 2.55, it is required to compute the integral of the bed slope source term over a element Ω_i .

$$\iint\limits_\Omega \boldsymbol{S}_B \, \mathrm{d}\Omega = \iint\limits_\Omega \begin{pmatrix} 0 \\ ghS_{Bx} \\ ghS_{By} \end{pmatrix} \mathrm{d}\Omega = g \iint\limits_\Omega h \begin{pmatrix} 0 \\ -\dfrac{\partial z_B(x,y)}{\partial x} \\ -\dfrac{\partial z_B(x,y)}{\partial y} \end{pmatrix} \mathrm{d}\Omega \tag{2.80}$$

Assuming that the bed slope values are constant over a cell, eq. 2.80 can be simplified to:

$$\iint\limits_\Omega \boldsymbol{S}_B \, \mathrm{d}\Omega = g \left(\iint\limits_\Omega h \, \mathrm{d}\Omega \right) \begin{pmatrix} 0 \\ S_{Bx} \\ S_{By} \end{pmatrix} = g Vol_{water} \begin{pmatrix} 0 \\ S_{Bx} \\ S_{By} \end{pmatrix} \tag{2.81}$$

In order to evaluate the above integral, it is necessary to compute the bed slope of a cell and the volume of the water over a cell. Since the numerical model allows the use of triangular cells as well as quadrilateral cells in hybrid meshes, these both cases need to be distinguished.

Triangular cells

The bed slope of a triangular cell can be computed by using the finite element formulation as given by Hinton and Owen (1979). It is assumed that z_b varies linearly over the cell (Figure 2.24):
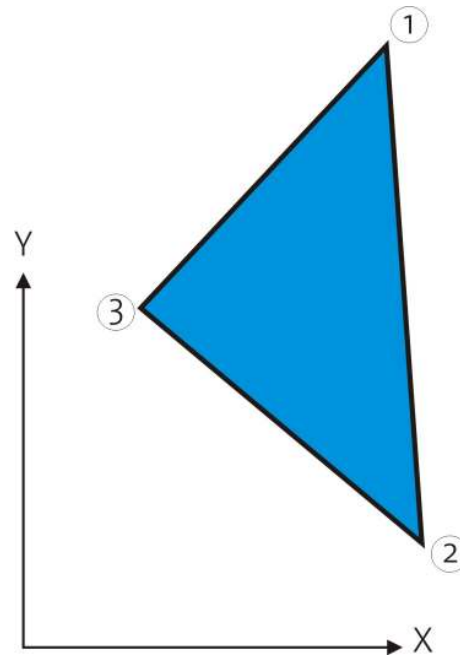
***Figure 2.24** A Triangular Cell*

$$z_B(x, y) = \alpha_1 + \alpha_2 x + \alpha_3 y \tag{2.82}$$

in which $\alpha_2 = \dfrac{\partial z_B}{\partial x}; \; \alpha_3 = \dfrac{\partial z_B}{\partial y}$

The constants α_1, α_2 and α_3 can be determined by inserting the nodal coordinates and equating to the corresponding nodal values of z_B. Solving for α_1, α_2 and α_3 finally gives

$$z_B(x, y) = \frac{1}{2\Omega} \left[(a_1 + b_1 x + c_1 y) z_{b,1} + (a_2 + b_2 x + c_2 y) z_{b,2} + (a_3 + b_3 x + c_3 y) z_{b,3} \right] \tag{2.83}$$

where

$$\left. \begin{array}{l} a_1 = x_2 y_3 - x_3 y_2 \\ b_1 = y_2 - y_3 \\ c_1 = x_3 - x_2 \end{array} \right\} \tag{2.84}$$

With the other coefficients given by cyclic permutation of the subscripts in the order 1,2,3. The area Ω of the triangular element is given by

$$2\Omega = \begin{vmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{vmatrix} \tag{2.85}$$

One can compute the bed slopes in x- and y-direction in each cell as
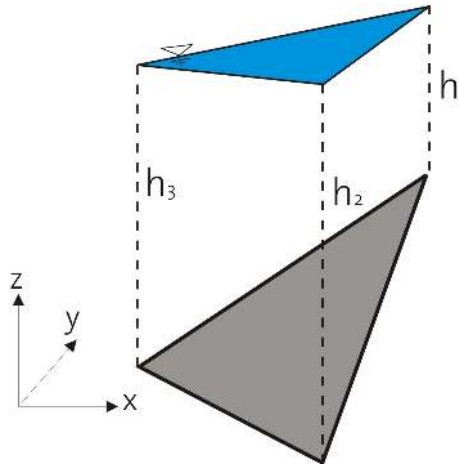
Figure 2.25 *Water volume over a cell*
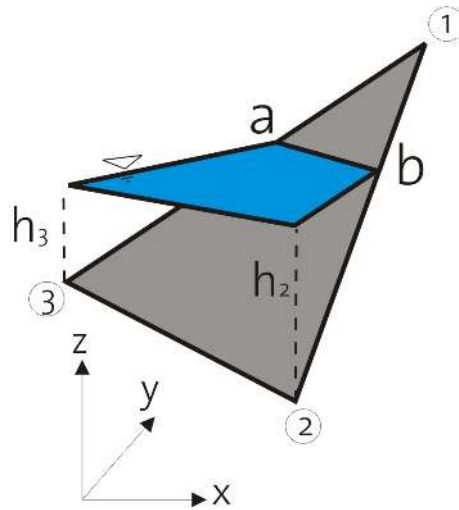
Figure 2.26 *Partially wet cell*

$$
\left.\begin{aligned}
S_{Bx} &= -\frac{\partial z_B(x,y)}{\partial x} = -\frac{1}{2\Omega}(b_1 z_{B,1} + b_2 z_{B,2} + b_3 z_{B,3}) \\
S_{By} &= -\frac{\partial z_B(x,y)}{\partial y} = -\frac{1}{2\Omega}(c_1 z_{B,1} + c_2 z_{B,2} + c_3 z_{B,3})
\end{aligned}\right\}
\tag{2.86}
$$

The water volume over a cell can also be computed by using the parametric coordinates of the Finite Element Method.

$$
Vol_{water} = \left(\frac{h_1 + h_2 + h_3}{3}\right)\Omega
\tag{2.87}
$$

where h_1 , h_2 and h_3 are the water depths at the nodes 1, 2 and 3 respectively (Figure 2.25).

In the case of partially wet cells (Figure 2.26) the location of the wet-dry line (a and b) has to be determined, where the water surface plane intersects the cell surface.

Using the coordinates of a and b, the water volume over the cell can be calculated as:
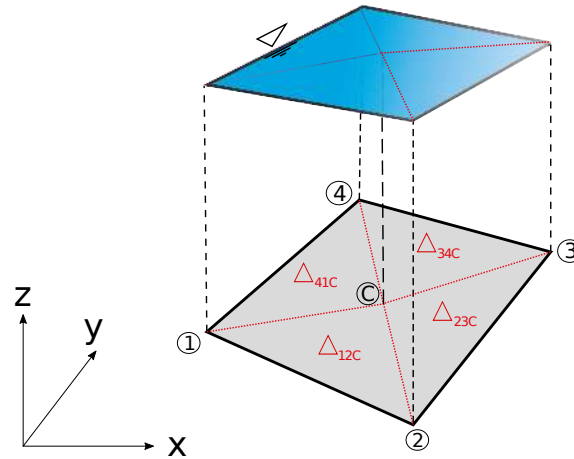
Figure 2.27 *Quadrilateral element and its division into four triangles (Δ_{12C}, Δ_{23C}, Δ_{34C}, Δ_{41C})*

$$Vol_{water} = \Omega_{a32}\frac{h_2 + h_3}{3} + \Omega_{2ba}\frac{h_2}{3} \tag{2.88}$$

In the computation of the fluxes through the edges (1-2) and (1-3), the modified lengths are used. The modified length is computed under the assumption that the water elevation is constant over a cell.

Quadrilateral cells

The numerical treatment of the bed source term calculation greatly increases in complexity if one has to deal with quadrilateral elements with four nodes. In the common case these nodes do not lie on a plane and therefore the slope of the element is not uniquely determined and not trivially computed. Even if the nodes initially lie on a plane, this situation can change if morphological simulations with mobile beds are performed.

To prevent complex geometric algorithms and to avoid the problematic bed slope calculation, the quadrilateral element is divided up into four triangles. Then the calculations outlined before for triangular cells can be applied separately on each triangle.

The four triangles are obtained by connecting each edge with the centroid C of the element. The required bed elevation of this centroid C is thereby estimated by a weighted distance averaging of the nodal elevations as proposed by Valiani et al. (2002):

$$z_C = \frac{\displaystyle\sum_{k=1}^{4} z_k\sqrt{(x_k - x_C)^2 + (y_k - y_C)^2}}{\displaystyle\sum_{k=1}^{4} \sqrt{(x_k - x_C)^2 + (y_k - y_C)^2}} \tag{2.89}$$

where z_C is the interpolated bed elevation of the centroid and k is the index of the four nodes of the quadrilateral element. Following this procedure the water volumes and bed slopes are calculated in the same way as outlined before for each of the triangles. Finally the bed slope term of the quadrilateral element is obtained as sum over the values of the corresponding four triangles.

$$\iint\limits_{\Omega} \boldsymbol{S}_B \, \mathrm{d}\Omega = g \left(\iint\limits_{\Omega} h \, \mathrm{d}\Omega \right) \begin{pmatrix} 0 \\ S_{Bx} \\ S_{By} \end{pmatrix} = g \sum_{k=1}^{4} \left[Vol_{water,\Delta_k} \begin{pmatrix} 0 \\ S_{Bx,k} \\ S_{By,k} \end{pmatrix} \right] \qquad (2.90)$$

This calculation method circumvents the problematic bed slope determination for the quadrilateral element during the calculation of the bed source term.

But for other purposes, like for morphological simulations with bed load transport, a defined bed slope within the quadrilateral element may be needed. For such situations the bed slope is determined by an area-weighting of the slopes of the k triangles. Replacing ϕ with the x- or y-coordinate, one obtains the bed slopes as follows:

$$\frac{\partial z_B(x,y)}{\partial \phi} \approx \frac{1}{A_{Quad}} \sum_{k=1}^{4} \left(\frac{\partial z_{B,k}(x,y)}{\partial \phi} A_{\Delta,k} \right) \quad , \quad \phi = x, y \qquad (2.91)$$

2.2.3.3 Conservative property (C-Property)

The usually applied shock capturing schemes were originally designed for hyperbolic systems without source terms. Such schemes do not guarantee the C-property in the presence of source terms like the bed source term in the shallow water equations. At stagnant conditions, when simulating still water above an uneven bed, unphysical fluxes and oscillations may result from an unbalance between the flux gradients and the bed source terms. In order to guarantee the C-property following condition, the reduced momentum equation for stagnant conditions, must hold true:

$$\oint\limits_{d\Omega} \left(\frac{1}{2} g h^2 \right) n_x \, \mathrm{d}l = g \iint\limits_{\Omega} h S_{Bx} \, \mathrm{d}\Omega|_{\zeta=const}$$

$$\oint\limits_{d\Omega} \left(\frac{1}{2} g h^2 \right) n_y \, \mathrm{d}l = g \iint\limits_{\Omega} h S_{By} \, \mathrm{d}\Omega|_{\zeta=const} \qquad (2.92)$$

Therefore it is necessary to guarantee conservation by an appropriate treatment and discretisation of the flux gradients and the bed source terms. Recent studies provide several approaches for proper source term treatment, but are often either computationally complex or cannot be easily transferred to unstructured meshes. Following the approach of Komaei, the left hand terms $0.5gh_{mod}^2$ are calculated with a modified depth at the edges. This modified depth is calculated as integral over the linearly varying water depth at an edge.

$$h_{mod}^2 = \frac{1}{L} \int\limits_0^L h^2(x) \, \mathrm{d}x = \frac{h_i^2 + h_i h_j + h_j^2}{3} \qquad (2.93)$$

h_i and h_j are the water depths at the edge's left and right nodes, as shown in Figure 2.28.

It can be easily proved that using h_{mod} in the determination of the flux gradients guarantees the C-property on unstructured grids, if the bed source terms are discretized as product of the water volume with the bed slope as shown before. This is exemplified here for a completely wetted triangle in x-direction:
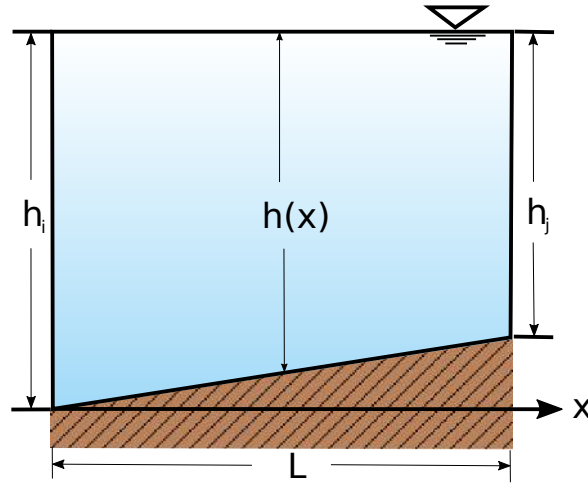
Figure 2.28 *Edge with linearly varying water depth*

$$
g \iint_\Omega h S_{Bx} \, \mathrm{d}\Omega \big|_{\zeta=const} = g Vol_{water} S_{Bx} = g \frac{1}{3}(h_1 + h_2 + h_3) \frac{1}{2}[b_1 z_{B,1} + b_2 z_{B,2} + b_3 z_{B,3}]
$$
$$
= \frac{g}{6}(h_1 + h_2 + h_3)[b_1(\zeta - h_1) + b_2(\zeta - h_2) + b_3(\zeta - h_3)]
$$
$$
= \frac{g}{6}(h_1 + h_2 + h_3)\left[\zeta \underbrace{(b_1 + b_2 + b_3)}_{=0} - b_1 h_1 - b_2 h_2 - b_3 h_3 \right]
$$
$$
= -\frac{g}{6}(h_1 + h_2 + h_3)[b_1 h_1 + b_2 h_2 + b_3 h_3]
$$

$$
-\oint_{d\Omega} \left(\frac{1}{2} g h^2 \right) n_x \, \mathrm{d}l = -\frac{1}{2} g \sum_{k=1}^{n} \left[\int_{x=0}^{l} h(x) \, \mathrm{d}x \right]
$$
$$
= -g\frac{1}{2}\left[\frac{1}{3}(h_1^2 + h_1 h_2 + h_2^2)b_3 + \frac{1}{3}(h_2^2 + h_2 h_3 + h_3^2)b_1 + \frac{1}{3}(h_3^2 + h_3 h_1 + h_1^2)b_2 \right] \tag{2.94}
$$
$$
= -\frac{g}{6}(h_1 + h_2 + h_3)[b_1 h_1 + b_2 h_2 + b_3 h_3]
$$

Both terms lead to the same result and therefore balance exactly for stagnant flow conditions.

2.2.3.4   Discretisation of Boundary Conditions

The hydrodynamic model uses the essential boundary conditions, i.e. velocity and water surface elevation are to be specified along the computational domain. The theoretical background of the boundary condition has already been already discussed in book one "Physical Models". In this part the numerical treatment of the two most important boundary types, namely inlet and outlet will be discussed separately.

2.2.3.4.1   Inlet Boundary

- Hydrograph:

The hydrograph boundary condition is applied to a user defined inlet section which is defined by a list of boundary edges. The velocity vectors are assumed to be perpendicular

to these boundary edges and the inlet section is assumed to be a straight line with uniform water elevation (1-D treatment). If there is an incoming discharge it must be given as hydrograph. Both steady and unsteady discharges can be specified along the inlet section, where water surface elevation and the cross-sectional area are allowed to change with time. For an unsteady discharge, the hydrograph is digitized in a data set of the form:

$$
\begin{array}{cc}
t^1 & Q^1 \\
\cdots & \\
t^n & Q^n \\
\cdots & \\
t^{end} & Q^{end}
\end{array}
$$

where Q^n is the total discharge inflow at time t^n. The hydrograph specified in this way, can be arbitrary in shape. The total discharge is interpolated, based on the corresponding time.

The water elevation at the inlet section is determined by the values of the old time step at the adjacent elements. In case of dry conditions or supercritical flow at the inlet section, the water elevation is calculated according the known discharge. For these iterative h-Q calculations normal flow is assumed and an average bed slope, perpendicular to the inlet section, must be given.

The calculated total inflow discharge is distributed over the inflow boundary edges and the according momentum component is calculated. Thereby only the edges below the water surface elevation receive a discharge. If some edges lie above the water elevation they are treated as walls. To distribute the inflow discharge over the wetted inflow boundary edges following approach is implemented.

The discharge Q_i for each edge i is calculated as fraction of the total discharge Q_{in} using a weighting factor K_i as

$$Q_i = K_i Q_{in} \tag{2.95}$$

This weighting factor K_i can be calculated based on its local conveyance as

$$K_i = \frac{c_{fi}\sqrt{R_i}\sqrt{g}A_i}{\sum\limits_{j=1}^{n}(c_{fj}\sqrt{R_j}\sqrt{g}A_j)} \tag{2.96}$$

where K_i is the discharge at edge i and the index j ranges from the first to the last wetted edge of the inflow boundary edges. R_i , A_i and c_{fi} are the hydraulic radius, the wetted area and the friction factor of the corresponding elements respectively.

Alternatively, the weighting factor K_i can be calculated based on the local wetted areas at the edge i, which finally results in equal inflow velocities over all edges.

$$K_i = \frac{A_i}{\sum\limits_{j=1}^{n} A_j} \tag{2.97}$$
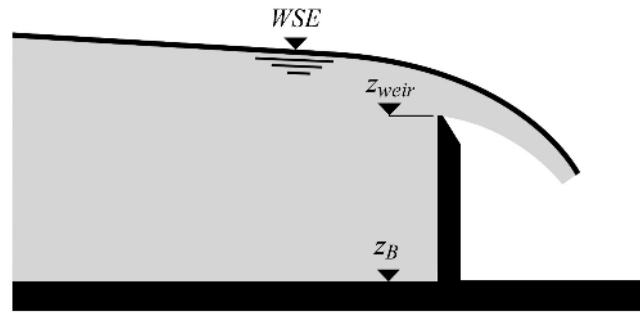
Figure 2.29 *Flow over a weir*

2.2.3.4.2 Outlet Boundary

- Free surface elevation boundary:

As is mentioned in Table 1.2, just one outlet boundary condition is necessary to be defined. This could be the flux, the water surface elevation or the water elevation-discharge curve at the outflow section. There is often no boundary known at the outlet. In this situation, the boundary should be modelled as a so-called free surface elevation boundary. A zero gradient assumption at the outlet could be a good choice. This could be expressed as follows:

$$\frac{\partial}{\partial \mathbf{n}} = 0 \tag{2.98}$$

Although this type of boundary condition has a reflection problem, numerical experiences have shown that this effect is limited just to five to ten grid nodes from the boundary. Therefore it has been suggested to slightly expand the calculation domain in the outlet region to use this type of outlet boundary (Nujić, 1998).

- Weir:

In the other possibility of outlet boundary condition namely defining a weir (Figure 2.29), the discharge at the outlet is computed based on the weir function (Chanson, 1999)

$$q = \frac{2}{3} C \sqrt{2g(h_{up} - h_{weir})^3}$$

where

$$C = 0.611 + 0.08 \frac{h_{up} - h_{weir}}{h_{weir}} \quad ; \quad h_{up} = WSE - z_B \quad and \quad h_{weir} = z_{weir} - z_B$$

Alternatively, instead of calculating the factor C automatically, a constant Poleni factor can be set.

The hydraulic and geometric parameters such as WSE, z_B are the calculated variables on the adjacent elements of the outlet boundary. Here it is assumed that the water surface elevation is constant within the element. The weir elevation z_{weir} is a time dependent parameter. Based on the weir elevation (Figure 2.30) some edges of the outlet are considered as a weir and the others have free surface elevation boundary condition. In order to avoid
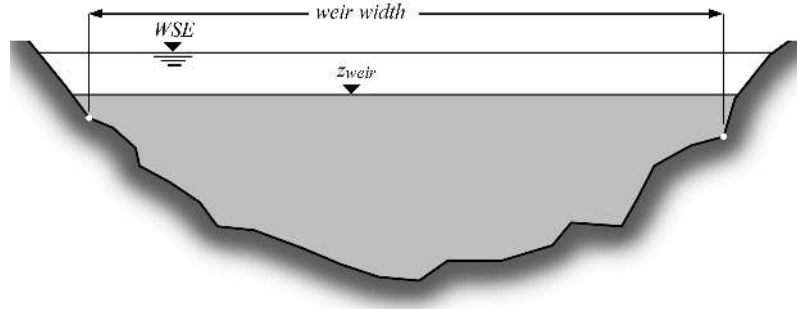
Figure 2.30 *Outlet cross section with a weir*

instabilities due to the water surface fluctuations, the following condition are adopted in the program

if $(h_{up}(t) \leq h_{weir}(t) + kh_{dry}) \Rightarrow$ edge is a wall

if $(h_{up}(t) > h_{weir}(t) + kh_{dry}) \Rightarrow$ edge acts as weir or a free surface elevation boundary

k is a numerical factor and has been set to 3 in this version. Figure 2.30 also shows the effective computational width of a weir in a natural cross section.

- Gate:

The discharge over a gate boundary condition is computed according to

$$q = \mu h_{gate} \sqrt{2gh_0} \tag{2.99}$$

Within this formula, h_0 is the water depth upstream of the gate. h_{gate} denotes the difference between gate level and soil elevation at the gate's location. The factor μ can be defined by the user (constant value or dynamical). According to Bernoullis' equation and assuming the upstream water is at rest, the discharge coefficient is

$$\mu = \frac{\delta}{\sqrt{1 + \dfrac{\delta h_{gate}}{h_0}}} \tag{2.100}$$

where δ is the contraction coefficient of the outflow jet. Assuming a sharp-edged sluice gate the contraction coefficient is calculated by Voigt (1971).

$$\delta = \frac{1}{1 + 0.64\sqrt{1 - (h_{gate}/h_0)^2}} \tag{2.101}$$

The value of μ is usually around 0.6.

The gate formula is only active if the water surface elevation in the element belonging to the boundary edge is higher than the gate elevation. Other possible states of the gate are wall (in case of gate elevation lower than the soil elevation) and zero-gradient (in all other cases).

- h-Q-relation:

A water elevationdischarge relation can be applied as outflow boundary condition. Several outflow boundary edges are therefore defined in a list. The outflow velocity vectors are assumed to be perpendicular to the outflow boundary edges and the outflow section is assumed to be a straight line with a uniform water elevation (1-D treatment). A relation must be given between the outflow water surface elevation and the total outflow discharge. This h-Q relation is digitized in a data set of the form

$$
\begin{array}{cc}
h^1 & Q^1 \\
\cdots & \\
h^n & Q^n
\end{array}
$$

where Q^n is the total outflow discharge for a given water surface elevation h^n. The h-Q-relation specified in this way, can be arbitrary in shape. The total outflow discharge is interpolated, based on the corresponding water surface elevation.

The water surface elevation at the outflow section is determined from the values of the elements adjacent to the boundary edges at the last time step. With this water elevation a total outflow discharge is interpolated using the given h-Q relation. Alternatively, if no h-Q-relation is given, the outflow discharge is calculated under the assumption of normal flow at the outflow section. In this case an average bed slope perpendicular to the outflow cross section must be given.

The calculated total outflow discharge is then distributed over all wetted outflow boundary edges according to a weighting factor based on the local conveyance or the wetted area (see Section 2.2.3.4.1). In contrast, cells which are not fully wetted are set to wall boundary.

- Z-Hydrograph:

Another outflow boundary condition is to specify the time evolution of the water surface at the outflow location. This boundary conditions aims to control the water elevation at the outlet, e.g. at outflows to reservoirs with known water elevations.

A time evolution of the water surface elevation must be given in the form

$$
\begin{array}{cc}
t^1 & WSE^1 \\
\cdots & \\
t^n & WSE^n
\end{array}
$$

If the actual outlet water elevation lies below the desired reservoir elevation, than a wall boundary is set at the outflow. On the other hand, if the actual water elevation lies above the reservoir water elevation, a Riemann solver is applied. The Riemann problem is defined between the outflow edges and a ghost cell outside of the domain with the reservoir water level. (But be aware that the outlet water level is not guaranteed to be identical to the specified water elevation.)

2.2.3.4.3 Inner Boundaries

Inner boundary conditions are used to model hydraulic structures like a weir or a gate within a model domain. Since the Saint Venant equations are not applicable for calculating the flux at these structures, an empirical approach is implemented in BASEMENT.

For the inner boundary in BASEchain a definition of a reference cross section which is located upstream of the weir/gate is mandatory. The reference cross section is the cross section where the water surface is just not affected by the flow acceleration at the weir (Figure 2.31).

In the 1-D model only one weir field with the width of all weir fields can be set up. Thus, an empirical formula for the effective width w_{eff} is implemented to take the reduction of the width due to head losses at piers and abutments into account,

$$w_{eff} = w - 2(n_p c_p + c_a) H_{ref} \tag{2.102}$$

where w is the total width of all the weir fields, n_p is the number of piers, c_p is a coefficient depending on the shape of the piers, c_a is a coefficient depending on the shape of the abutment and H_{Ref} is the energy height above the bottom level at the reference cross section.

- Inner Weir:

The inner weir uses a slightly other approach than the boundary weir. If the weir crest is higher than the water surface elevation in the neighbouring elements, the weir acts as a wall. If one or both of the neighbouring water surface elevations are above the weir crest (1, Figure 2.31), this weir formula is used

$$q = \frac{2}{3} \mu \sigma_{uv} \sqrt{2g(z_{Ref} - z_{weircrest})^3} \tag{2.103}$$

where q is the specific discharge related to the effective width w_{eff}. This formula is the classical POLENI formula for a sharp crested weir with an additional factor σ_{uv} which accounts for the reduction in discharge due to incomplete weir flow. If only one side of the weir has a water surface elevation above the weir crest, then a complete weir flow is given with $\sigma_{uv} = 1$ (Figure 2.31). As soon as the water surface elevation tops the weir crest level on both sides of the weir, the incomplete case is active and the reduction factor σ_{uv} is calculated according to the Diagram in Figure 2.32. z_{Ref} is the water surface elevation at the reference cross section. The weir bottom level (2) is the lowest possible weir crest level (Figure 2.31).

The momentum flux at the weir is a function of the wetted area A_{Weir} above the weir crest. In BASEchain two options of calculation types are implemented:

I) "Standard": The wetted area above the weir crest is calculated as

$$A_{Weir} = (z_{Ref} - z_{weircrest}) w_{eff}$$

II) "Critical": It is assumed that the critical flow depth is at the weir crest. The critical flow depth is calculated as $h_{crit} = 2/3[z_{Ref} - z_{weircrest}]$. The wetted area above the weir crest is calculated as $A_{Weir} = h_{crit} w_{eff}$. For a model calibration the water surface elevation z_{crit} of the critical flow depth above the weir crest can be adjusted by the calibration factor f_{crit}. $z_{correction} = z_{crit} + f_{crit}(z_{Ref} - z_{crit})$. The formula for the calculation of the factor f_{crit} is $f_{crit} = a + bh_{crit}$.
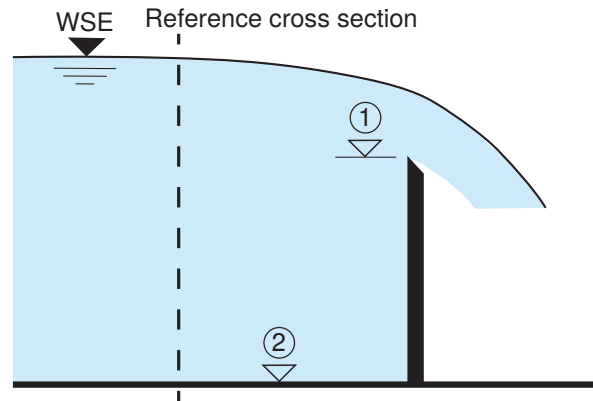
Figure 2.31 *Inner weir with a complete weir flow. (1) weir crest level, (2) weir bottom level, Reference cross section where water surface in unaffected by the flow acceleration at the weir*
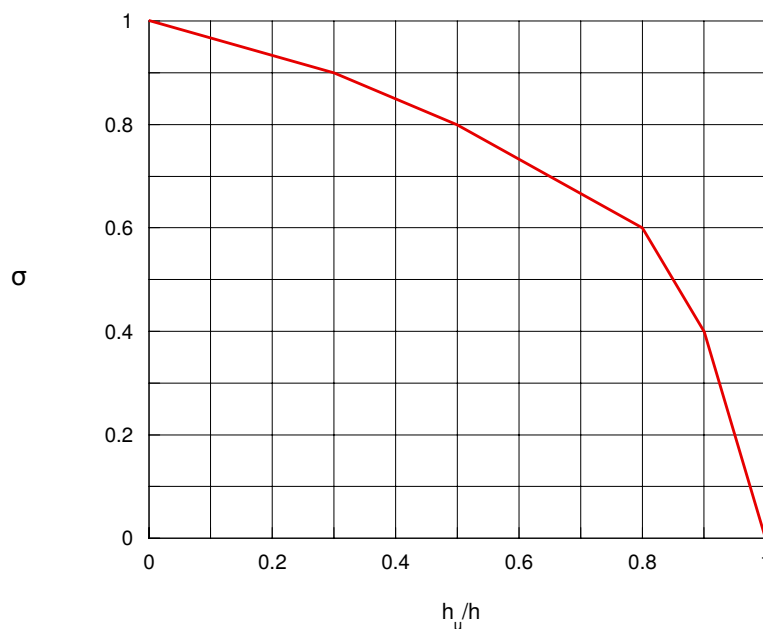
Figure 2.32 *Reduction factor σ_{uv} for an incomplete flow over the weir. h_u is the downstream water depth over the weir crest and h denotes the upstream flow depth over the weir crest*
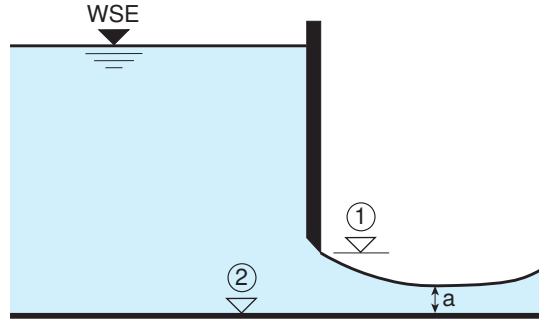
Figure 2.33 *Inner sluice gate. (1) gate level, (2) gate bottom level, a = water depth at vena contracta*

- Inner Gate:

In BASEchain three types of gates are implemented: sluice gate, gate with flap and radial gate with flap. The simplest gate type is the sluice gate which has three modes. Either the gate level (1) is equal or less than the gate bottom level (2, Figure 2.33). The gate is then closed and acts as a wall.

If the gate level is above the gate bottom level, the gate is considered as open. As long as the water surface elevation near the gate is below the gate level, the flux at the inner boundary is calculated as a weir flux with, where the weir level is equal to the gate bottom level. The gate is active as soon as one of the neighbouring water surface elevations is above the gate level (Figure 2.33). Similar to the gate boundary condition, the calculation of the specific discharge is

$$q = \mu h_{gate} \sqrt{2gh_0} \qquad (2.104)$$

The gate opening is defined as $h_{gate} =$ gate level (1) - gate botom level (2) (Figure 2.33), h_0 denotes the water depth at the cross section upstream of the gate. The discharge coefficient μ depends on the contraction factor δ which is the ratio of the water depth a at the vena contracta to the gate opening h_{gate} and is calculated as

$$\mu = \delta / \sqrt{1 + \frac{\delta h_{gate}}{h_0}} \qquad (2.105)$$

Derived from the conjugate depths at a hydraulic jump the criterion for considering the backwater effect at the gate is defined as

$$\frac{h_d}{h_{gate}} > \frac{\delta}{2} \left(\sqrt{1 + \frac{16}{\delta \left(1 + \frac{\delta h_{gate}}{h_0}\right)} \frac{h_0}{h_{gate}}} - 1 \right) \qquad (2.106)$$

where h_d is the water depth downstream of the gate. When the backwater effect has to be considered the discharge coefficient depends also on the water depth downstream of the gate (Bollrich, 2000).
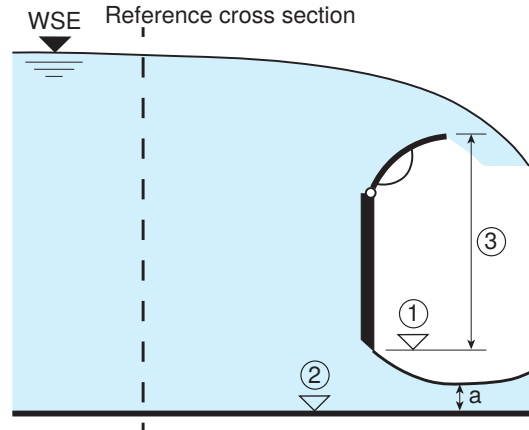
Figure 2.34 *Inner gate with flap. (1) gate level, (2) gate bottom level, (3) gate size which is defined as the difference between flap level and gate level, a = water depth at the vena contracta*

$$\mu = \delta \sqrt{ \left[1 - 2\frac{\delta h_{gate}}{h_0} \left(1 - \frac{\delta h_{gate}}{h_d} \right) \right] - \sqrt{ \left[1 - 2\frac{\delta h_{gate}}{h_0} \left(1 - \frac{\delta h_{gate}}{h_d} \right) \right]^2 + \left(\frac{h_d}{h_0} \right)^2 } - 1 }$$
(2.107)

As for the momentum, the velocity through the gate is taken into account in both, downstream and upstream direction.

The second gate type is the gate with flap. Three modes are equal to the sluice gate:

I) The gate is closed and the water surface elevation is lower than the flap level. The inner boundary acts as a wall.

II) The water surface elevation is lower than the gate level. Then the exact Riemann solver is used.

III) The gate is active. This means the water surface elevation is higher than the gate level but lower than the flap level. In this case the gate flow is calculated by the gate formula.

In case of a closed gate an overflow at the flap is possible. Then the inner boundary acts as a weir and the weir level corresponds to the flap level. The flux calculation complies with the flux calculation of the inner weir. For the weir flow a discharge coefficient depending on the shape of the flaps must be defined. If the gate is active and the water surface elevation is higher than the flap level, the flux is calculated as the sum of the gate flow and the weir flow (Figure 2.34).

The third gate type in BASEchain is the radial gate with flap. The specific discharge at a radial gate is calculated with this formula (Knapp, 1960)

$$q = \psi h_{gate} \sqrt{ 2g \left(H_{Ref} - \frac{h_{gate}}{2} \right) }$$
(2.108)
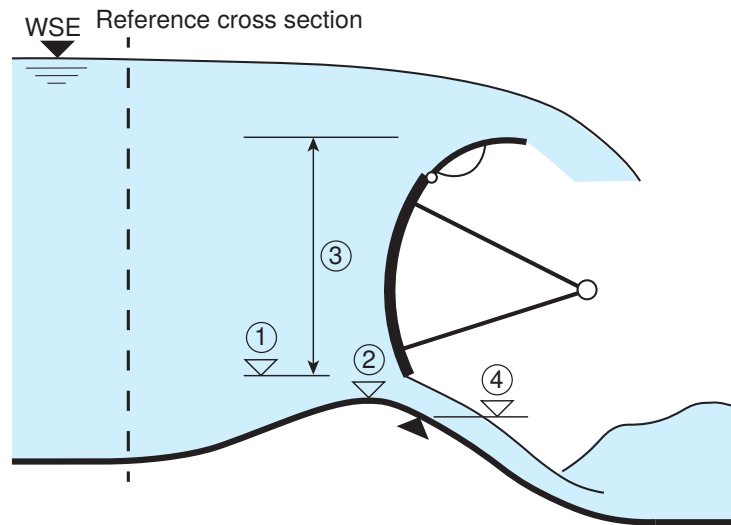
Figure 2.35 *Radial gate with flap at a Jambor sill: (1) gate level, (2) weir bottom level, (3) gate height which is the difference between flap level and gate level, (4) gate bottom level which is the level of the closed gate*

where H_{Ref} is the energy height at the reference cross section above the bottom level and ψ is the discharge coefficient which is defined as $\psi = \alpha(h_{gate}/H_{Ref})^\beta$. The user has to define the calibration parameters α and β. The gate is active, when the gate level is higher than the gate bottom level. The gate height h_{gate} is the difference between gate level (1) and gate bottom level (4) (Figure 2.35).

A radial gate is often combined with a weir sill, a so called Jambor sill. Due to various operational conditions for this gate type several parameters have to be defined. The gate is closed, if the gate level is equal or less than the gate bottom level at the Jambor sill (4, Figure 2.35). In this case the inner boundary acts either as a wall or a as a weir with the overfall at the flaps. This depends on the water surface elevation, whether it is higher or lower than the flap level.

If the water surface elevation is higher than the flap level, the flux is calculated in the same way as the gate with flap, namely as the sum of the gate flow and the weir flow. If the gate is closed and the water surface exceeds the flap level, the inner boundary acts as a weir.

As soon as the radial gate is open and the water surface elevation is lower than the gate level the inner boundary acts as a weir with the weir bottom level (2, Figure 2.35). For the calculation of the weir flux the energy height above the bottom elevation at the reference cross section is used.

- Inner HQ-relation:

The inner HQ-Relation boundary acts similar to the inner gate boundary condition. However, instead of applying the gate-formula to determine the discharge over the inner structure, a water surface discharge relation is applied. By specifying a self-determined HQ-relation for the inner structure, a lot of flexibility is offered for the implementation. The HQ-relation may be used e.g. to simulate a culvert, a bridge or a pipe flow. The quality of the results depends strongly on the provided HQ-table!

For each edge of the upstream stringdef, the water level is taken from the adjacent cell and the HQ-relation is used to determine the corresponding discharge over the edge (scaled to the edge length). The water flows through the inner structure and re-enters the domain at the downstream corresponding edge. The HQ relation is digitized in a data set as

$$
\begin{array}{cc}
h^1 & Q^1 \\
\cdots & \\
h^n & Q^n
\end{array}
$$

where Q^n is the total outflow discharge and h^n is the water surface elevation.

Differing from other inner boundaries, this boundary operates only in a given direction from upstream to downstream (stringdef1 = upstream, stringdef = downstream) and cannot deal with changing flow directions! Furthermore, please note that the inner HQ-relation does not depend upon the z-elevations of the boundary cells and, hence, may be used to act over large distances with arbitrary height differences, as e.g. a long pipe within the domain. At the moment, however, it is not feasible to incorporate information of the downstream water surface elevation, what limits the applicability to culvert or pipe modelling in some scenarios.

If inner boundaries and sediment transport computations are combined, the problem arises that sediment masses are not transported over the inner boundary but stop in front of the inner boundary structure. This behaviour is undesired in some scenarios. A solution to overcome this problem, allowing for sediment continuity, is the use of 'dredge sources' which can be used to let the sediments pass the inner boundary (see Section 2.3.3.3.2).

2.2.3.4.4   Moving Boundaries

Dry, partial wet and fully wet elements

Natural rivers and streams are highly irregular in both plan form and topography. Their boundaries change with the time varying water level. The FV-based model with moving boundary treatment is capable of handling these complex and dynamic flow problems conveniently. The computational domain expands and contracts as the water elevation rises and falls. Obviously, the governing equations are solved only for wet cells in the computational domain. An important step of this method is to determine the water edge or the instantaneous computational boundary. A criterion, h_{min} , is used to classify the following two types of nodes:

1. A node is considered dry, if $z_S \leq z_B + h_{min}$
2. A node is considered wet, if $z_S > z_B + h_{min}$

The determination of h_{min} is tricky, which can vary between 10^{-6} m and 0.1 m. Based on the flow depth at the centre of the element we defined three different element categories (Figure 2.36):

1. dry cells where the flow depth is below h_{min},
2. partial wet cells where the flow depth above h_{min} but not all nodes of the cell are under water and
3. fully wet cells where all nodes included cell centre are under water.
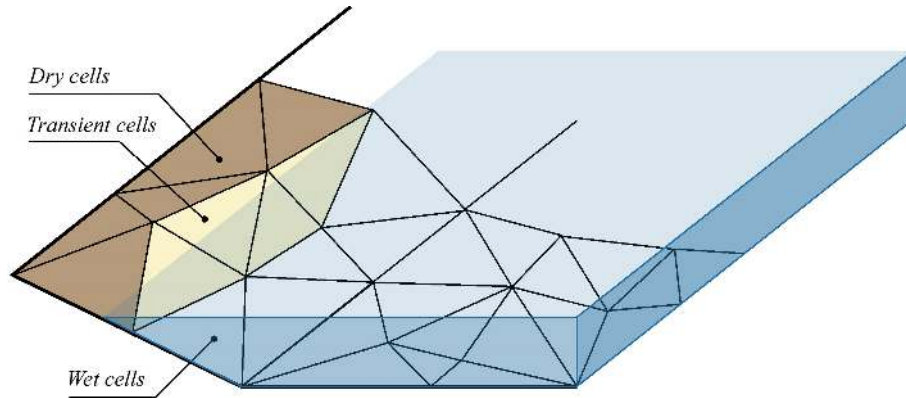
Figure 2.36 *Schematic representation of a mesh with dry, partial wet and wet elements*
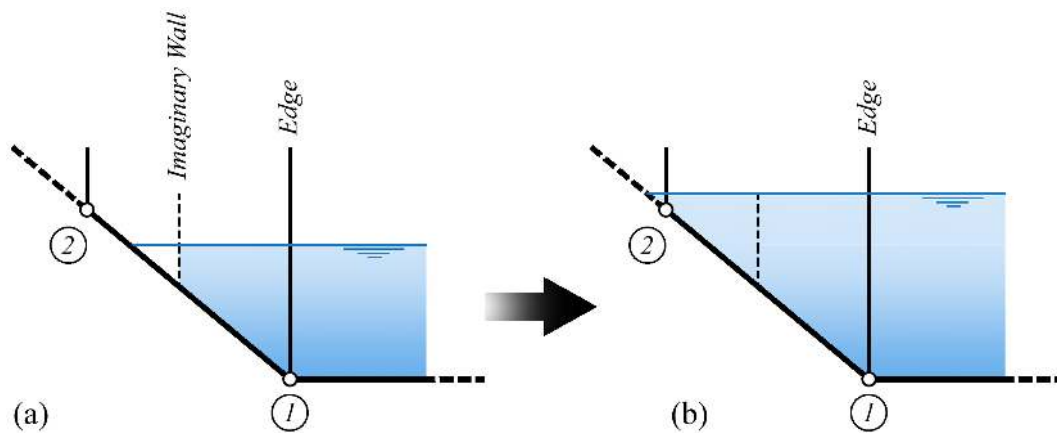
Figure 2.37 *Wetting process of an element*

By comparing the water surface elevation of two adjacent elements (Figure 2.37) and determining which cell is dry it was decided whether there were or not a flux through the edge.

Although the determination of h_{min} is tricky, as it mentioned above, it has been successfully used in the past, has in the range of 0.05 ~ 0.1 m for natural rivers. It can be adjusted to optimize the solution for particular flow and boundary conditions. It is suggested to consider it close to $min(0.1h, 0.1)$.

Another problem related to partially wetted elements is the determination of the final velocities at the end of the time step from the vector of the conserved variables $\mathbf{U}$. To calculate the velocities the conserved variables must be divided by the flow depth as indicated below.

$$\nu_x = \frac{(\nu_x h)}{h}$$

$$\nu_y = \frac{(\nu_y h)}{h}$$

(2.109)

For an element situated at the wetting and drying interface, the outflow of water amounts may lead to very small water depths. Because the water depths are in the denominator,
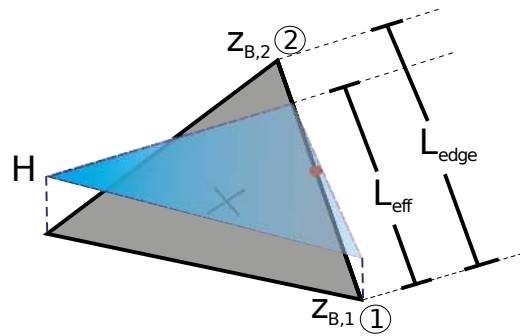
Figure 2.38 *Definition of effective length at partially wetted edge*

instabilities can arise when updating the new velocities. To prevent these instabilities it is checked if the water depth is smaller than the residual h_{min} . In such a case the velocities are set to zero, since the water will not move in such a practical situation.

Flux computations at dry-wet interfaces

When solving the shallow water equations along dry-wet interfaces, special attention is needed and different situations must be distinguished. Some models solve the complete equations only for completely wetted elements, where all nodes are under water. Here, in contrast, the flux computations are also performed for partially wetted elements. This procedure is computationally more costly and has larger programming efforts, but it leads to more accurate results in some situations and it can reduce problems related to the wetting-drying process.

The complete flux computation is performed over a partially wetted edge if two conditions are fulfilled:

- At least one of the both elements adjacent to the edge must be wetted, i.e. its water depth must be above h_{min} .
- At least at one side of the edge, the element's water surface elevation must be above the average edge elevation.

The flux computations over partially wetted edges need to take into account that the flux takes not place over the whole length of the edge (see Figure 2.38). The actually over flown effective length L_{eff} is calculated as follows assuming a constant water level.

$$L_{eff} = \left(\frac{H - z_{B,1}}{z_{B,2} - z_{B,1}} \right) L_{edge} \tag{2.110}$$

Here H is the water surface elevation in the partially wetted element and $z_{B,1}$, $z_{B,2}$ are the nodal elevations of the edge

In Figure 2.39 several possible configurations at dry-wet interfaces are illustrated which need to be treated appropriately. Attention must be paid to correctly reproduce the physics and to preserve the C-property for stagnant flow conditions.

The first case a) shows a wetted left element adjacent to a dry right element, where the left water surface elevation H_L is above the center elevation $z_{B,R}$ of the right element. Here, no
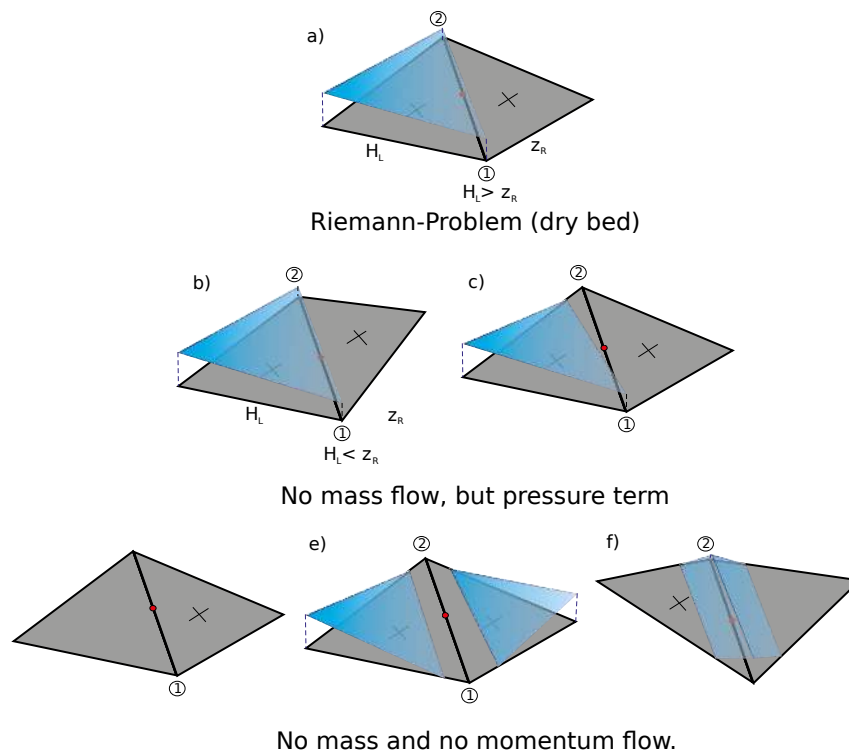
Figure 2.39 *Different cases for flux computations over an edge at wet-dry interfaces*

special treatment is needed and a Riemann problem can be formulated. But the Riemann solver must be capable of treating dry bed conditions in an appropriate way.

Case b) corresponds to an adverse slope at the right, dry element, where the left water surface elevation H_L is below the right bed elevation $z_{B,R}$. This case has recently received attention in the literature, as e.g. by Brufau et al. (2002). It requires a special treatment because applying the Riemann solver in a situation with adverse slopes can produce incorrect results. Some authors suggest to treat the dry-wet interface as a wall or to set the velocities to zero. But these treatments are problematic because they do not always preserve the C-property. Here a simple method is adopted whereas no Riemann problem is formulated at the edge. But instead, only the pressure terms are evaluated which exactly balance the bed source terms, thus guaranteeing a correct behaviour for stagnant flow conditions.

In case c) the water elevation H_L at the left element is below the average edge elevation. In such a situation no Riemann problem is formulated as stated above. But again the pressure term is evaluated here to preserve the C-property.

Finally, in cases d), e) and f) either both elements are dry or the edge is completely dry. In these cases neither mass fluxes nor momentum fluxes need to be evaluated.

2.2.3.5   Solution Procedure

The logical flow of data through BASEplane from the entry of input data to the creating of output files and the major functions of the program is illustrated in Figure 2.40, Figure 2.41 and Figure 2.42 shows the data flow through the hydrodynamic and morphodynamic routines respectively. Program main control data is read first, and then the mesh file,
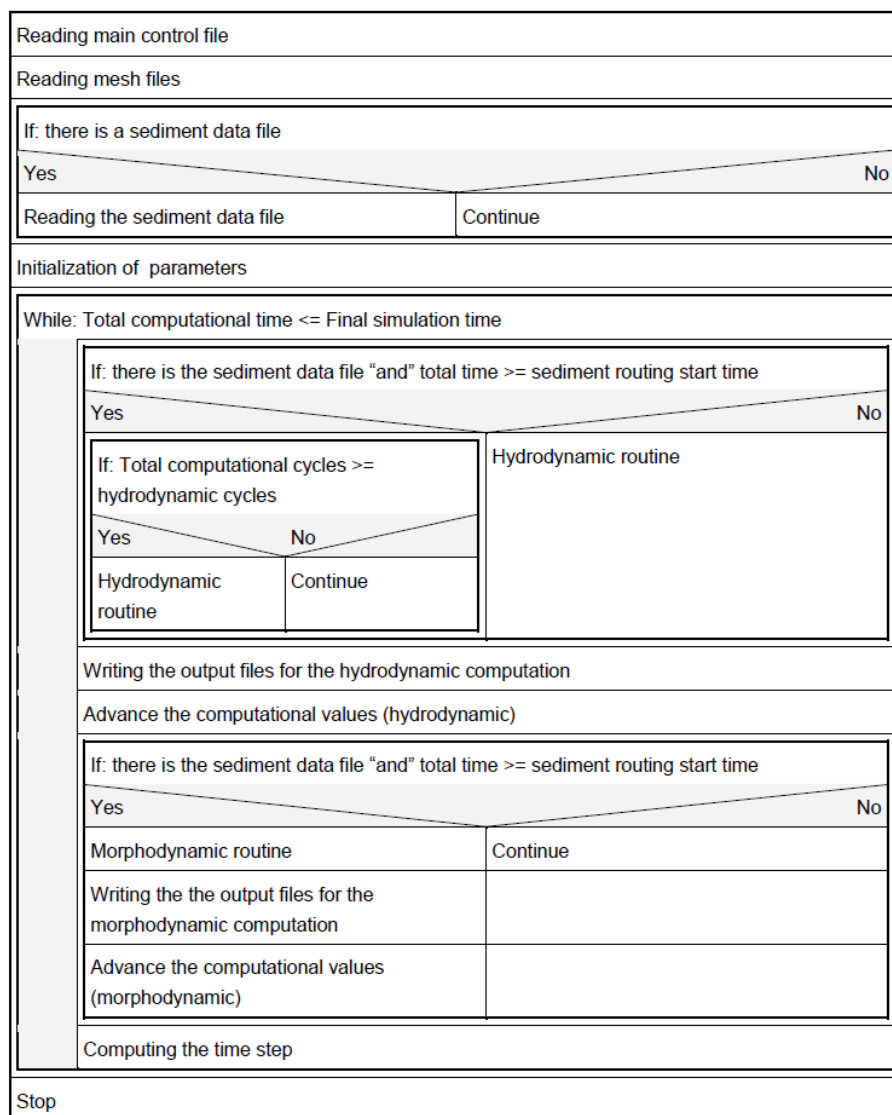
Figure 2.40 *The logical flow of data through BASEplane*

and then the sediment data file if there is one. Initialization of the parameters and computational values is made next. If the sediment movement computation is requested, the hydrodynamic routine will be started in cycle steps defined by user, otherwise the hydrodynamic routine is run. After the hydrodynamic routine, the morphodynamic routine is carried out next, if it is requested. Results can be printed at the end of every time steps or only at the end of selected time steps.

   a) General solution procedure of BASEplane (Figure 2.40)

   b) Hydrodynamic routine in detail (Figure 2.41)

   c) Morphodynamic routine in detail (Figure 2.42)

Setting the boundary conditions

Reconstruction

For: All edges

Computing the edge values

Computing Fluxes

For: All edges

Computation of fluxes

Time Evaluating

For: All elements

Balancing the fluxes

Adding source terms

Computing the conservative values
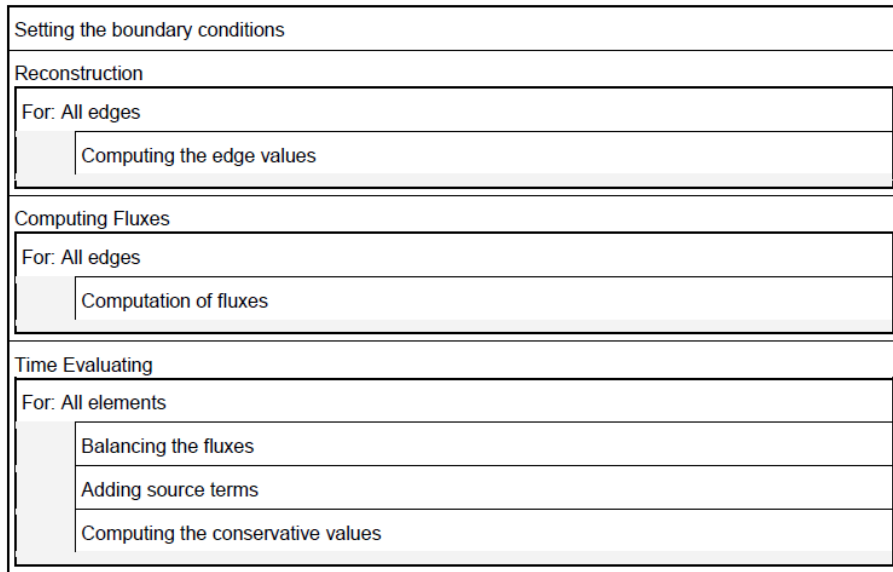
Figure 2.41 *Data flow through the hydrodynamic routine*

Setting the boundary conditions

Computing Fluxes

For: all hydraulic elements

Computation of transport capacities

For: all sediment edges

Computation of bedload fluxes

Time Evaluating

For: all elements

Balancing the bedload fluxes

Adding source terms

Computing the new bed elevation

Computing bedload control volume thickness

For: all grains

Adding source terms

Computing new compositions

Updating the nodal elevations and the element's centre elevation

Figure 2.42 *Data flow through the morphodynamic routine*
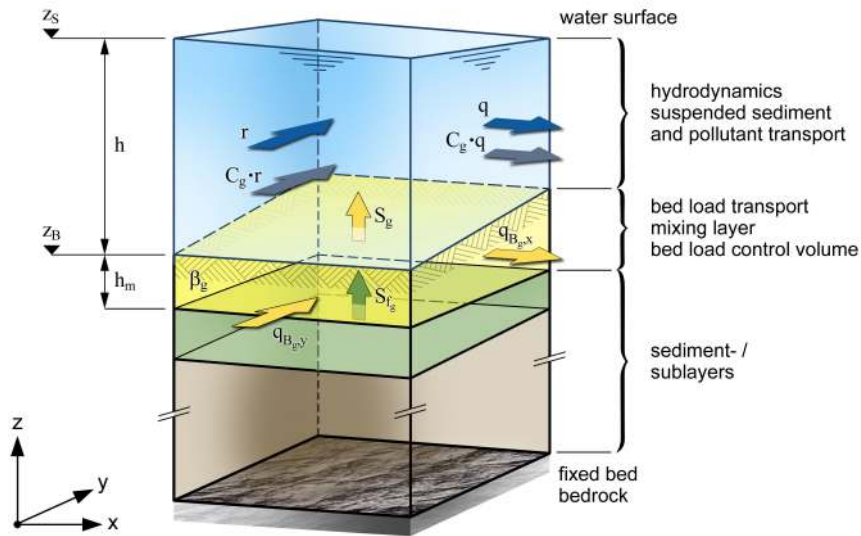
Figure 2.43 *Vertical Discretisation of a computational cell*

2.3 Solution of Sediment Transport Equations

2.3.1 Vertical Discretisation

2.3.1.1 General

A two phase system (water and solids) in which the sediment mixture can be represented by an arbitrary number of different grain size classes is formulated. The continuous physical domain has to be horizontally and vertically divided into control volumes to numerically solve the governing equations for the unknown variables. Figure 2.43 shows a single cell of the numerical model with vertical partition into the three main control volumes: the upper layer for momentum and suspended sediment transport, the active layer for bed load sediment transport as well as bed material sorting and sub layers for sediment supply and deposition.

The primary unknown variables of the upper layer are the water depth h and the specific discharge q and r in directions of Cartesian coordinates x and y. In the active layer, $q_{B_g,x}$ and $q_{B_g,y}$ are describing the specific bed load fluxes (index refers to the g-th grain size class). A change of bed elevation z_B can be gained by a combination of balance equations for water and sediment and corresponding exchange terms (source terms) between the vertical layers.

2.3.1.2 Determination of Mixing Layer Thickness

The bed load control volume is the region where bed load transport occurs and it is assumed to have a uniform grain distribution over the depth. Its extension is well-defined by the bed surface at level z_B and its thickness h_m, which plays an important role for grain sorting processes during morphological simulations with multiple grain classes.

Different methods are implemented for the determination of this thickness h_m. It can be determined either dynamically during the simulation (at the moment only for 2-D

simulations) or it can be given a priori as a constant value for the whole simulation. The latter is used by default with an active layer thickness of 0.1 m.

Borah's approach

With this approach the active layer thickness h_m is different for degradation and for aggradation. In case of deposition, h_m corresponds to the thickness of the current deposition stratum. In case of degradation, h_m is proportional to the bed level decrase with a limitation to account for the situation of an armoured bed (Borah et al. (1982)).

If the bed level increases ($\Delta z_B > 0$):

$$h_m^{n+1} = h_m^n + \Delta z_B \tag{2.111}$$

If the bed level decreases ($\Delta z_B < 0$):

$$h_m = 20\Delta z_B + \frac{d_l}{\sum \beta_{nm}(1-p)} \tag{2.112}$$

where d_l is the smallest non-mobile grain size and $\sum \beta_{nm}$ is the sum of the non-mobile sediment fractions, and p is porosity.

Calculation based on mean diameter d_{90}

Following this approach the new thickness h_m is determined proportional to the characteristic grain size diameter d_{90} in the bed load control volume. The factor of proportionality can be chosen freely.

$$h_m^{n+1} = factor \cdot d_{90} \tag{2.113}$$

But this simple approach does not take into account influences of present bottom shear stresses or present erosion rates. Typically, the factor is between 1 and 3.

2.3.2   One Dimensional Sediment Transport

2.3.2.1   Spatial Discretisation

2.3.2.1.1   Soil Segments

For each cross section slice a different composition of the soil can be specified. A variable number of sediment layers can be defined. Figure 2.44 illustrates by example a possible distribution of soil types in a cross section. Each colour represents for a different grain class mixture. Usually however one soil type will cover several cross section slices, like in Figure 2.46.

The modification of the geometry of the cross section due to sediment transport it is illustrated in Figure 2.46. This can lead to the elimination of layers or to the creation of new ones. The grain class mixture of deposition will be the same over the whole wetted width.

The elevation changes will modify the soil elements which are considered to be wetted, but it is not always obvious when this is the case. For this reason the user can define
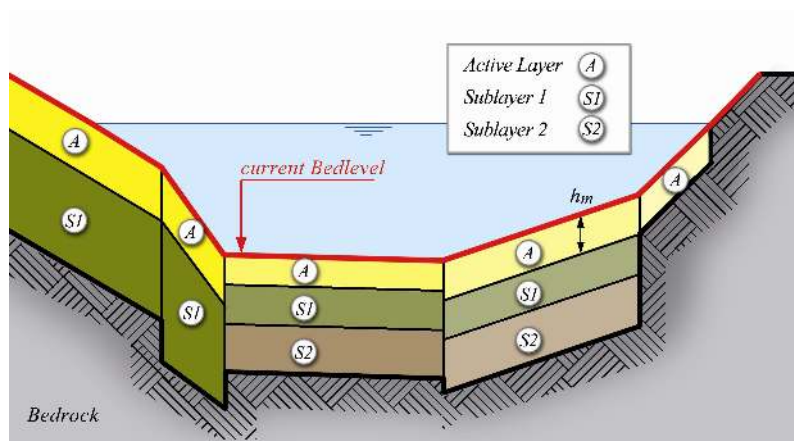
Figure 2.44 *Soil Discretisation in a cross section*
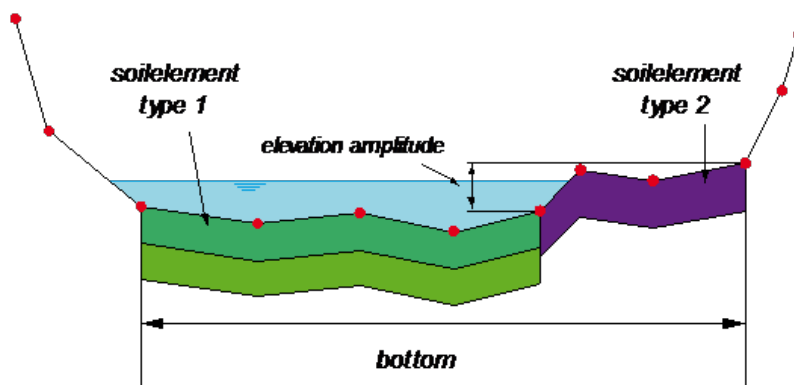
Figure 2.45 *Soil Discretisation in a cross section*
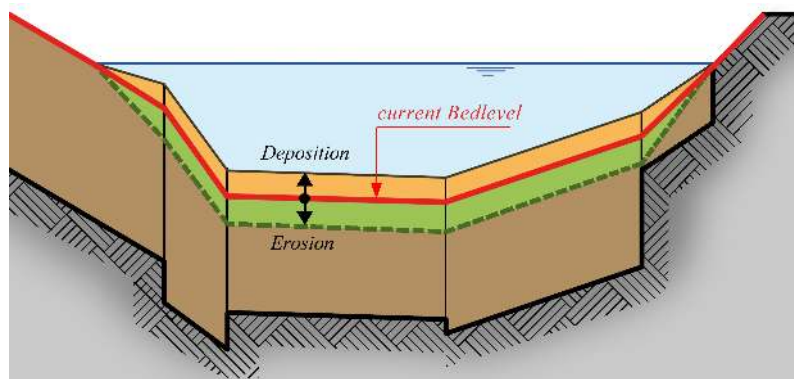
Figure 2.46 *Effect of bed load on cross section geometry*
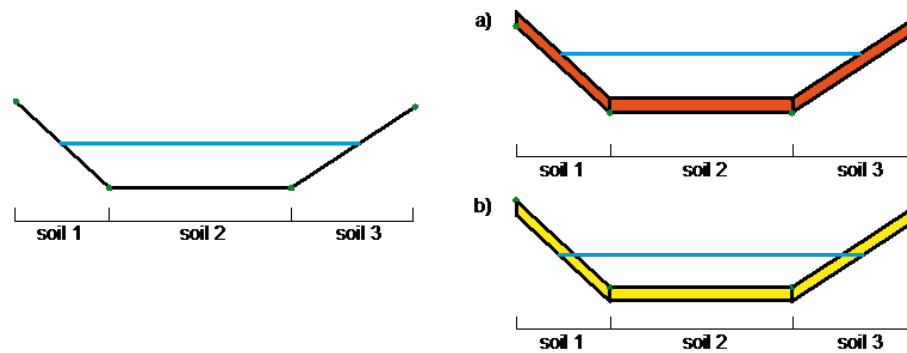
Figure 2.47 *Deposition a) and erosion b) due to suspended load with a wetting fraction of 0.9.*
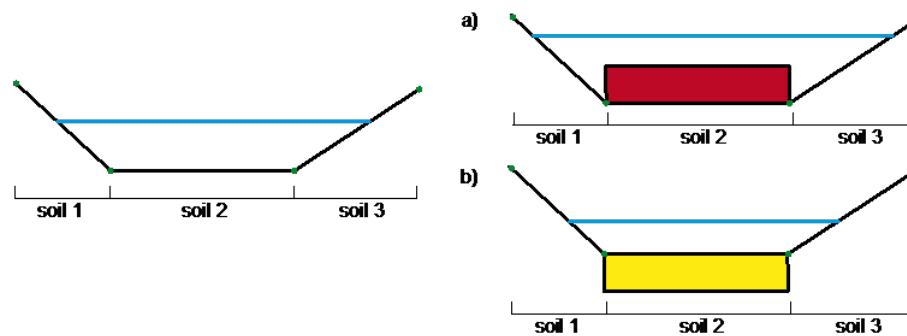
Figure 2.48 *Deposition a) and erosion b) due to suspended load with a wetting fraction of 0.1.*

which fraction of the elevation amplitude of the soil has to be below the water level by a parameter called *wetting_fraction*. In the example in Figure 2.45 the soil element of type 2 would be moved with a *wetting_fraction* of 0.3 but not with a *wetting_fraction* of 1. All points of a wetted soil element are affected by the same elevation change.

Figure 2.47, Figure 2.48, Figure 2.49, Figure 2.50, Figure 2.51 and Figure 2.52 give some simple examples of bedlevel changes to illustrate the mechanisms.
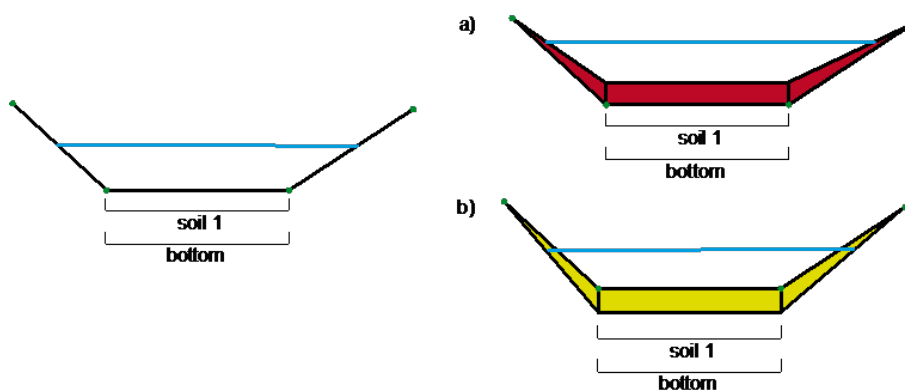
Figure 2.49 *Deposition a) and erosion b) due to bed load without cross section points on embankments.*
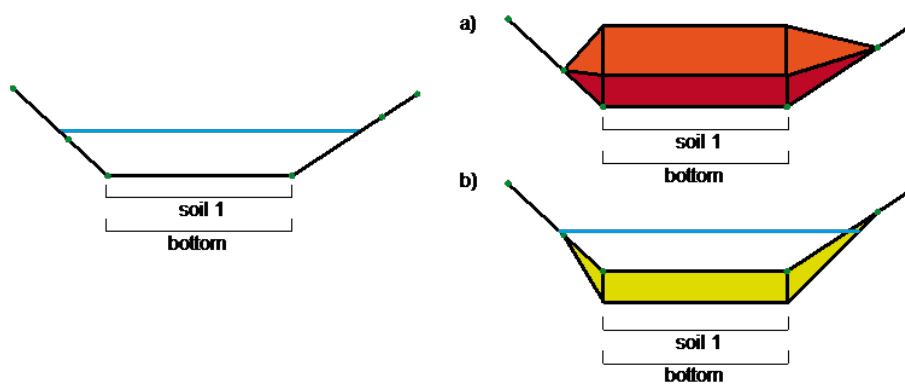
Figure 2.50 *Deposition a) and erosion b) due to bed load with cross section points on embankments.*
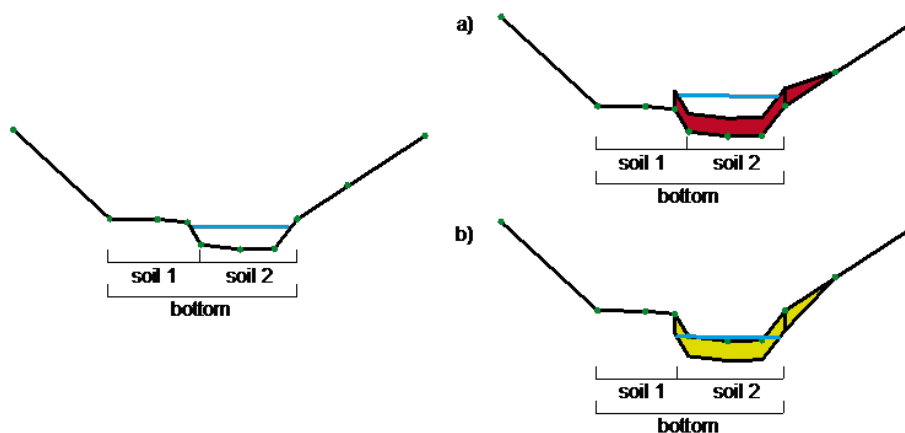
Figure 2.51 *Deposition a) and erosion b) due to bed load with 2 soils defined on the bottom.*
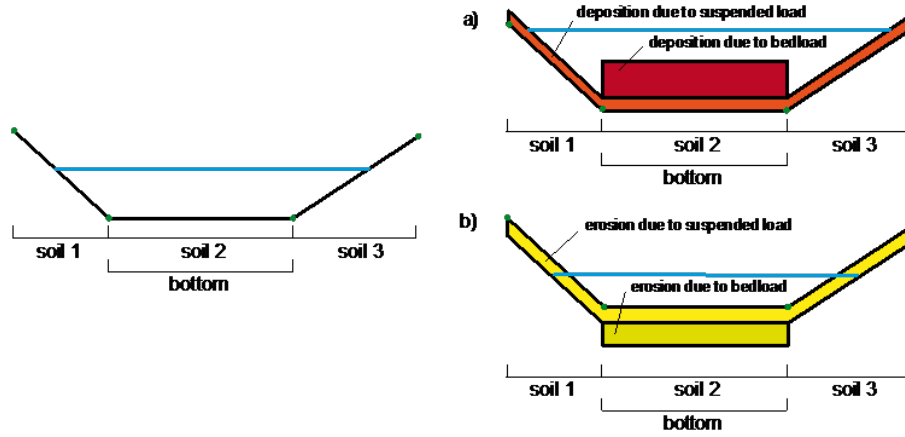
Figure 2.52 *Deposition a) and erosion b) due to bed load and suspended load.*

2.3.2.2   Discrete Form of Equations

2.3.2.2.1   Advection-Diffusion Equation

The one dimensional suspended sediment or pollutant transport in river channels is described by eq. 1.66 in the Mathematical Models section. This equation has to be solved for each grain class g in the same manner. For this reason in this section g is omitted and the equation becomes:

$$A\frac{\partial C}{\partial t} + Q\frac{\partial C}{\partial x} - \frac{\partial}{\partial x}\left(A\Gamma\frac{\partial C}{\partial x}\right) - S = 0 \tag{2.114}$$

C is the concentration of transported particles averaged over the cross-section.

For the moment the sources S, which vary for different types of transport, will be set to 0.

The eq. 2.114 is integrated over the element (see Figure 2.7)

$$\int\limits_{x_{iL}}^{x_{iR}}\left(A\frac{\partial C}{\partial t} + Q\frac{\partial C}{\partial x} - \frac{\partial}{\partial x}\left(A\Gamma\frac{\partial C}{\partial x}\right)\right)\mathrm{d}x = 0 \tag{2.115}$$

and the different parts are calculated as follows:

$$\int\limits_{x_{iL}}^{x_{iR}} A\frac{\partial C_i}{\partial t}\;\mathrm{d}x = A_i\int\limits_{x_{iL}}^{x_{iR}}\frac{\partial C}{\partial t}\;\mathrm{d}x \approx A_i\frac{\partial C_i}{\partial t}\Delta x_i \approx A_i\frac{C_i^{n+1} - C_i^n}{\Delta t}\Delta x_i \tag{2.116}$$

$$\int\limits_{x_{iL}}^{x_{iR}} Q\frac{\partial C}{\partial x}\;\mathrm{d}x = Q_i\int\limits_{x_{iL}}^{x_{iR}}\frac{\partial C}{\partial x}\;\mathrm{d}x = Q_i(C(x_{iR}) - C(x_{iL})) = (\Phi_{a,iR} - \Phi_{a,iL}) \tag{2.117}$$

$$\int\limits_{x_{iL}}^{x_{iR}} \frac{\partial}{\partial x}\left(A\Gamma\frac{\partial C}{\partial x}\right)\mathrm{d}x = \left(A\Gamma\left.\frac{\partial C}{\partial x}\right|_{x_{iR}} - A\Gamma\left.\frac{\partial C}{\partial x}\right|_{x_{iL}}\right) = \Phi_{d,iR} - \Phi_{d,iL} \tag{2.118}$$

The concentration at the new time is:

$$C_i^{n+1} = C_i^n - \frac{\Delta t}{\Delta x_i A_i} (\Phi_{a,iR} - \Phi_{a,iL} - \Phi_{d,iR} + \Phi_{d,iL}) \tag{2.119}$$

2.3.2.2.2   Computation of Diffusive Flux

The diffusive flux is calculated by finite differences.

$$\Phi_{d,iR} = A_{iR}\Gamma\frac{C_{i+1} - C_i}{x_{i+1} - x_i} \tag{2.120}$$

For the interpolation on the edge of the wetted area, known only in the cross sections, the geometric mean is used:

$$A_{iR} = \sqrt{A_{i+1}A_i} \tag{2.121}$$

If Γ is not given by the user it is calculated as follows:

$$\Gamma = \frac{\sqrt{\nu_L \nu_R}}{\sigma} \tag{2.122}$$

σ is generally assumed to be 0.5 (Celik and Rodi 1984).

The eddy viscosity averaged over the depth can be calculated by (Fäh, 1997):

$$\nu = uh\kappa/6 \tag{2.123}$$

2.3.2.2.3   Computation of Advective Flux

A general problem of the computation of the advective flux is that it leads to strong numerical diffusion. Several schemes can be used, four of them are implemented.

The first possibility to compute he advective flux over the edge (element boundary) is to interpolate the concentration values from the neighbouring elements, considering the flow direction, and multiply it with the water discharge over the edge.

a) QUICK-Scheme

For a positive flow from left to right the quick scheme determines the concentration due to advection at the upstream edge of element i by:

$$\begin{aligned} C_{a,iL} = {} & \frac{(x_{iL} - x_{i-1})(x_{iL} - x_{i-2})}{(x_i - x_{i-1})(x_i - x_{i-2})}C_i + \frac{(x_{iL} - x_{i-2})(x_{iL} - x_i)}{(x_{i-1} - x_{i-2})(x_{i-1} - x_i)}C_{i-1} \\ & + \frac{(x_{iR} - x_{i-1})(x_{iR} - x_i)}{(x_{i-2} - x_{i-1})(x_{i-2} - x_i)}C_{i-2} \end{aligned} \tag{2.124}$$

More in general the concentration is:

$$C_{a,iL} = \begin{cases} C_i + g_1(C_{i+1} - C_i) + g_2(C_i - C_{i-1}) \rightarrow u > 0 \\ C_{i+1} + g_3(C_i - C_{i+1}) + g_4(C_{i+1} - C_{i+2}) \rightarrow u < 0 \end{cases} \tag{2.125}$$

Between the velocities u_i and u_{i+1} the one with the larger absolute value is determinant.

$$g_1 = \frac{(x_{iR} - x_i)(x_{iR} - x_{i-1})}{(x_{i+1} - x_i)(x_{i+1} - x_{i-1})} \tag{2.126}$$

$$g_2 = \frac{(x_{iR} - x_i)(x_{i+1} - x_{iR})}{(x_i - x_{i-1})(x_{i+1} - x_{i-1})} \tag{2.127}$$

$$g_3 = \frac{(x_{iR} - x_{i+1})(x_{iR} - x_{i+2})}{(x_i - x_{i+1})(x_i - x_{i+2})} \tag{2.128}$$

$$g_4 = \frac{(x_{iR} - x_{i+1})(x_i - x_{iR})}{(x_{i+1} - x_{i+2})(x_i - x_{i+2})} \tag{2.129}$$

The QUICK scheme tends to get instable, especially for the pure advection-equation with explicit solution (Chen and Falconer, 1992). For this reason the more stable QUICKEST scheme (Leonard, 1979) is often used:

b) QUICKEST-Scheme

$$C_{iR,QUICKEST} = C_{iR,QUICK} - \frac{1}{2}Cr_{iR}(C_{i+1} - C_i) + \frac{1}{8}Cr_{iR}(C_{i+1} - 2C_i + C_{i-1}) \tag{2.130}$$

with

$$Cr_{iR} = \frac{u_{i+1} + u_i}{2}\frac{\Delta t}{\Delta x} \tag{2.131}$$

c) Holly-Preissmann

The QUICKEST-scheme still leads to an important diffusion. For this reason the HollyPreissmann scheme (Holly and Preissmann, 1977), which gives better results, is also implemented. This scheme is based on the properties of characteristics and can not be applied directly for the present Discretisation.

To find $C(x_{iR})$ of eq. 2.117 the properties of characteristics or finite differences are used, placing the edges on a new grid so that $C(x_{iR})$ becomes C_j and $C(x_{iL}) = C_{j-1}$. Considering only the advection part of the eq. 2.114 and dividing by the cross section area A:

$$\frac{\partial C}{\partial t} + \frac{u\partial C}{\partial x} = 0 \tag{2.132}$$

and

$$\frac{C_j^{n+1} - C_j^n}{\Delta t} = -u\frac{C_j^n - C_{j-1}^n}{x_j - x_{j-1}} \tag{2.133}$$

Thus the new concentration is

$$C_j^{n+1} = \left(1 - u\frac{\Delta t}{x_j - x_{j-1}}\right)C_j^n + u\frac{\Delta t}{x_j - x_{j-1}}C_{j-1}^n \qquad (2.134)$$

for a courant number $CFL = u(dt/dx) = 1 : C_j^{n+1} = C_{j-1}^n$. This means that the solute travels from one side of the cell to the other during the time step.

The Holly-Preissmann scheme calculates the values in function of CFL and the upstream value.

$$Y(Cr) = ACr^3 + BCr^2 + DCr + E \qquad (2.135)$$

$Y(0) = C_j^n$; $Y(1) = C_{j-1}^n$

$\dot{Y}(0) = \dfrac{\partial C_j^n}{\partial x}$; $\dot{Y}(1) = \dfrac{\partial C_{j-1}^n}{\partial x}$

$$C_j^{n+1} = a_1 C_{j-1}^n + a_2 C_j^n + a_3 \frac{\partial C_{j-1}^n}{\partial x} + a_4 \frac{\partial C_j^n}{\partial x} \qquad (2.136)$$

$$a_1 = Cr^2(3 - 2Cr) \qquad (2.137)$$

$$a_2 = 1 - a_1 \qquad (2.138)$$

$$a_3 = Cr^2(1 - Cr)\Delta x \qquad (2.139)$$

$$a_4 = -Cr(1 - Cr)^2 \qquad (2.140)$$

and

$$\frac{\partial C_j^{n+1}}{\partial x} = b_1 C_{j-1}^n + b_2 C_j^n + b_3 \frac{\partial C_{j-1}^n}{\partial x} + b_4 \frac{\partial C_j^n}{\partial x} \qquad (2.141)$$

$$b_1 = 6Cr(Cr - 1)/\Delta x \qquad (2.142)$$

$$b_2 = -b_1 \qquad (2.143)$$

$$b_3 = Cr(3Cr - 2) \qquad (2.144)$$

$$b_4 = (Cr - 1)(3Cr - 1) \qquad (2.145)$$

However this form is only valid for a constant velocity u. If u is not constant the velocities in the different cells and at different times have to be considered. The velocity u_* is determined by interpolation of u_{j-1}^n , u_j^n , u_j^{n+1} .

$$\bar{u}_j = \frac{1}{2}(u_j^n + u_j^{n+1}) \qquad (2.146)$$

$$\bar{u}^n = (u_{j-1}^n \theta) + (1 - \theta)u_{j+1}^n \tag{2.147}$$

with

$$\theta = u_i^n \frac{\Delta t}{x_{j-1} - x_j} \tag{2.148}$$

$$\hat{u} = \frac{1}{2}(\bar{u}_j + \bar{u}^n) \tag{2.149}$$

$$Cr = \hat{u}\frac{\Delta t}{\Delta x} = \frac{u_j^{n+1} + u_j^n}{2\frac{x_{j-1} - j}{\Delta t} - u_{j-1}^n + u_j^n} \tag{2.150}$$

The Holly-Preissmann scheme gives good results for the pure advection-diffusion equation. But if a sediment exchange with the bed takes place, because of the shifted grid, it does not react to the influence of the source term.

For the last 3 schemes the advective flux is computed multiplying the concentration on the edge with the discharge over the edge:

$$\Phi_{a,iR} = Q_{iR}C_{iR}$$

d) Modified Discontinuous Profile Method (MDPM)

The MDPM method presented by Badrot-Nico et al. (2007) is like a transposing of the Holly-Preissmann scheme from a finite difference to a finite volume context and thus much more adapted for the use within BASEMENT.

In this method the advective flux is calculated directly as a sediment discharge:

$$\Phi_{iR} = \frac{1}{\Delta t}A_i \int_{t^n}^{t^{n+1}} u(t)C(x_{iR}, t)dt \tag{2.151}$$

Using the invariance property along a characterstic line (Figure 2.53) this equation can be transformed to

$$\Phi_{iR} = \frac{1}{\Delta t}A_i \int_{x_{iL}}^{x_{iR}} C(x, t^n) \, dx \tag{2.152}$$

The function $C_i^n(x)$ is reconstructed from the mean concentration in the cell C_i^n and the concentration values on the edges C_{iL}^n and C_{iR}^n (Figure 2.54) by satisfying mass conservation in the cell.

$$\tilde{C}_i^n(x) = \begin{cases} U_{iL}^n & if \quad x \le x_{i-1/2} + \alpha_i \Delta x_i \\ U_{iR}^n & if \quad x > x_{i-1/2} + \alpha_i \Delta x_i \end{cases} \tag{2.153}$$

with
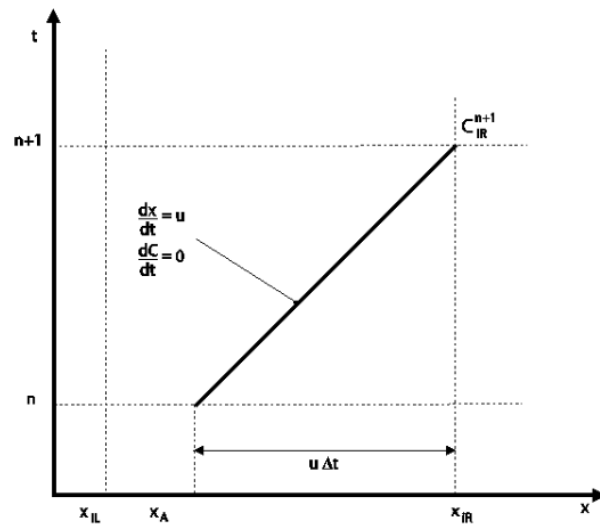
Figure 2.53 *Invariance of C along the characteristic line y*
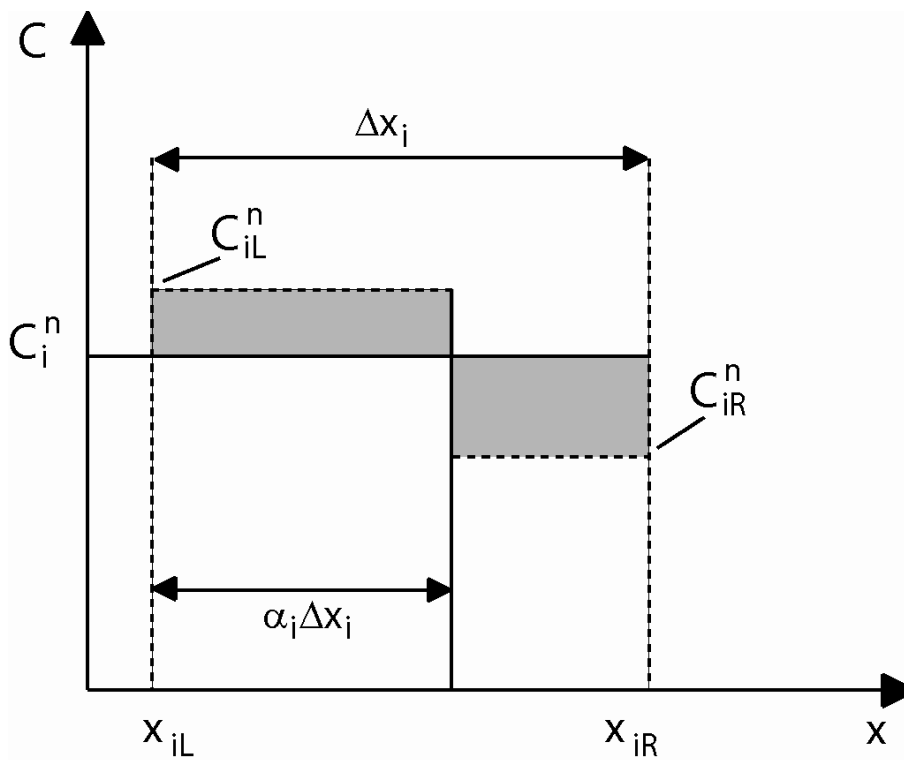
Figure 2.54 *Function $\tilde{C}_i^n(x)$*

$$\alpha_i = \frac{C_i^n - C_{iR}^n}{C_{iL}^n - C_{ir}^n} \tag{2.154}$$

and $\alpha_i \in [0,1]$.

If $C_{iL}^n = C_{iR}^n$ or $(C_{iL}^n - C_i^n)(C_i^n - C_{iR}^n) < 0$ the following values are set:

$$\left.\begin{array}{l} C_{iL}^n = C_i^n \\ C_{iR}^n = C_i^n \\ \alpha_i = \varepsilon \end{array}\right\}$$

ε is an arbitrary value between 0 and 1.

If the velocity is the flux of suspended load per unity of depth and width can now be determined as follows:

$$h_i = C_{iL}^n \, max(x_{i+1/2} - x - (1-\alpha_i)\Delta x_i, 0) + C_{iR}^n \, min(x_{i+1/2} - x, (1-\alpha_i)\Delta x_i) \tag{2.155}$$

If the velocity is negative respectively:

$$g_i = -C_{iL}^n min(x - x_{i+1/2}, \alpha_{i+1}\Delta x_{i+1}) - C_{iR}^n max(x - x_{i+1/2} - \alpha_{i+1}\Delta x_{i+1}, 0) \tag{2.156}$$

The abscissa of the foot of the characteristic is given by $x_A = x_{i+1/2} - u_{i+1/2}^n(t)\Delta t$.

Finally the advective flux is:

$$\Phi_{i+1/2} = \begin{cases} \dfrac{1}{\Delta t} h_i(x_A) & if \quad u_{x_{i+1/2}}^n > 0 \\[3mm] \dfrac{1}{\Delta t} g_{i+1}(x_A) & if \quad u_{x_{i+1/2}}^n < 0 \end{cases} \tag{2.157}$$

Furthermore the new concentrations on the edges have to be prepared for the computations of the next time step:

$$C_{i+1/2}^{n+1} = \begin{cases} C_{iL}^n & if \quad u_{x_{i+1/2}}^n(t) > 0 \quad and \quad Cr_x \geq 1-\alpha_i \\ C_{iR}^n & if \quad u_{x_{i+1/2}}^n(t) > 0 \quad and \quad Cr_x < 1-\alpha_i \\ C_{i+1L}^n & if \quad u_{x_{i+1/2}}^n(t) < 0 \quad and \quad Cr_x \geq -\alpha_i \\ C_{i+1R}^n & if \quad u_{x_{i+1/2}}^n(t) < 0 \quad and \quad Cr_x < -\alpha_i \end{cases} \tag{2.158}$$

The required Courant number is

$$Cr_x = \begin{cases} u_x(t)\dfrac{\Delta t}{\Delta x_i} & u_x(t) > 0 \\[4mm] u_x(t)\dfrac{\Delta t}{\Delta x_{i+1}} & u_x(t) < 0 \end{cases} \tag{2.159}$$

2.3.2.2.4   Global Bed Material Conservation Equation

As the y-direction is not considered, in the one dimensional case the mass conservation equation (Exner-equation) becomes:

$$(1-p)\frac{\partial z_B}{\partial t} + \left(\sum_{k+1}^{ng}\frac{\partial q_B}{\partial x} + s_g - sl_{B_g}\right) = 0 \tag{2.160}$$

q_B is the sediment flux per unit channel width. Integrating eq. 2.160 over the channel width, hence multiplying everything by the channel width, the following equation is obtained:

$$(1-p)\frac{\partial A_{Sed}}{\partial t} + \left(\sum_{k=1}^{ng}\frac{\partial Q_B}{\partial x} + S_g - Sl_{B_g}\right) = 0 \tag{2.161}$$

The discretisation is effected exactly in the same way as for the hydraulic mass conservation. The eq. 2.161 is integrated over the element (Figure 2.7):

$$\int_{x_{iL}}^{x_{iR}} \left((1-p)\frac{\partial A_{Sed}}{\partial t} + \left(\sum_{k=1}^{ng}\frac{\partial Q_B}{\partial x} + S_g - Sl_{B_g}\right)\right) dx = 0 \tag{2.162}$$

The parts of the eq. 2.162 are discretized as follows:

$$(1-p)\int_{x_{iL}}^{x_{iR}}\frac{\partial A_{Sed,i}}{\partial t}\ dx = (1-p)\frac{A_{Sed,i}^{n+1} - A_{Sed,i}^{n}}{\Delta t}\Delta x \tag{2.163}$$

$$\int_{x_{iL}}^{x_{iR}}\frac{\sum_{k=1}^{ng}Q_{B,i}}{\partial x}\ dx = \sum_{k=1}^{ng}Q_B(x_{iR}) - \sum_{k=1}^{ng}Q_B(x_{iL}) = \Phi_{B,iR} - \Phi_{B,iL} \tag{2.164}$$

$$\int_{x_{iL}}^{x_{iR}}\sum_{k=1}^{ng}(S_g - Sl_{B_g})\ dx = \sum_{k=1}^{ng}S_g - \sum_{k=1}^{ng}Sl_{B_g} \tag{2.165}$$

$\Phi_{B,iL}$ and $\Phi_{B,iR}$ are the bed load fluxes through the west and east side of the cell. Their determination will be discussed later (Section 2.3.2.3.1).

The change of the sediment area is thus calculated by:

$$\Delta A_{Sed} = A_{Sedi}^{n+1} - A_{Sedi}^{n} = \frac{\Delta t}{\Delta x_i}(\Phi_{B,iR} - \Phi_{B,iL}) - \frac{\Delta t}{\Delta x_i}\left(\sum_{k=1}^{ng}S_g - \sum_{k=1}^{ng}Sl_{B_g}\right) \tag{2.166}$$

As the result of the sediment balance is an area, the deposition or erosion height Δz_b has yet to be determined. The erosion or deposition is distributed over the wetted part of the cross section. If a bed bottom is defined the deposition height is equal and constant for all wetted slices. Only in the exterior slices the bed level difference is 0 where the cross section becomes dry. The repartition of the soil level change is illustrated in Figure 2.55.

The change of the bed level is calculated as follows:
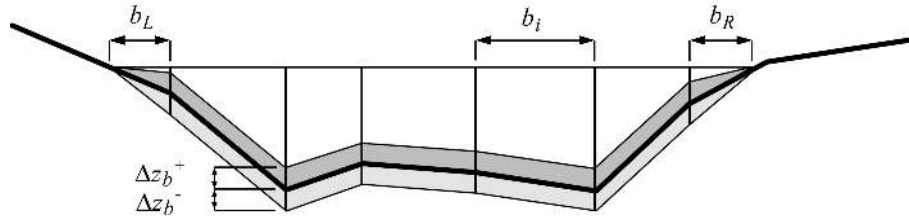
Figure 2.55 *Distribution of sediment area change over the cross section*

$$\Delta z_B = \frac{\Delta A_{Sed}}{\dfrac{b_r}{2} + \dfrac{b_l}{2} + \sum b_i} \tag{2.167}$$

As the sediments transported by bed load can be deposed only on the bed bottom, whilst the sediments transported by suspended load can be deposed on the whole wetted section, two separate values of ΔA_{Sed} and Δz_B are calculated for the two processes by splitting eq. 2.166 in two parts.

The change of bed level due to bed load is:

$$\Delta A_{Sed,bl} = \frac{\Delta t}{\Delta x_i}(\Phi_{B,iR} - \Phi_{B,iL}) - \frac{\Delta t}{\Delta x_i}\left(\sum_{k=1}^{ng} Sl_{B_g}\right) \tag{2.168}$$

The change of bed level due to suspended load accordingly:

$$\Delta A_{Sed,bl} = -\frac{\Delta t}{\Delta x_i}\left(\sum_{k=1}^{ng} S_g - \sum_{k=1}^{ng} Sl_g\right) \tag{2.169}$$

2.3.2.2.5   Bed material sorting equation

For the 1-D computation the bed material sorting equation computation is:

$$(1-p)\frac{\partial}{\partial t}(\beta_g h_B) + \frac{\partial q_{B_g}}{\partial x} + s_g - sf_g - sl_{B_g} = 0 \tag{2.170}$$

Considering the whole width of the cross section and introducing an active layer area A_B, eq. 2.170 becomes:

$$(1-p)\frac{\partial}{\partial t}(\beta_g A_B) + \frac{\partial Q_{B_g}}{\partial x} + S_g - Sf_g - Sl_{B_g} = 0 \tag{2.171}$$

The eq. 2.171 is integrated over the length of the control volume:

$$\int_{x_{iL}}^{x_{iR}} \left((1-p)\frac{\partial}{\partial t}(\beta_g A_B) + \frac{\partial Q_{B_g}}{\partial x} + S_g - Sf_g - Sl_{B_g}\right) dx = 0 \tag{2.172}$$

The different parts are discretized as follows:

$$(1-p)\int_{x_{iL}}^{x_{iR}} \frac{\partial}{\partial t}(\beta_g A_B)\, \mathrm{d}x = (1-p)\frac{\beta_g^{n+1}A_B^{n+1} - \beta_g^n A_B^n}{\Delta t}\Delta x \tag{2.173}$$

$$\int_{x_{iL}}^{x_{iR}} \frac{\partial Q_{B_g}}{\partial x}\, \mathrm{d}x = Q_B(x_{iR}) - Q_B(x_{iL}) = \Phi_{g,iR} - \Phi_{g,iL} \tag{2.174}$$

Then the new β_g at time $n+1$ can be calculated for every element i by:

$$\beta_{g_i}^{n+1} = \left((1-p)\beta_{gi}^n A_{B_i}^n - \frac{\Delta t}{\Delta x_i}(\Phi_{iR}^k - \Phi_{iL}^k) - (S_g - Sf_g - Sl_{B_g})\frac{\Delta t}{\Delta x_i}\right)/((1-p)A_{B_i}^{n+1}) \tag{2.175}$$

As for the global bed material conservation equation the bed material sorting equation is also solved twice: once for the bed load and once for the suspended load.

$$\beta_{g_i,bl}^{n+1} = \left((1-p)\beta_{gi}^n A_{B_i}^n - \frac{\Delta t}{\Delta x_i}(\Phi_{iR}^k - \Phi_{iL}^k) - (-Sf_g - Sl_{B_g})\frac{\Delta t}{\Delta x_i}\right)/((1-p)A_{B_i}^{n+1}) \tag{2.176}$$

and

$$\beta_{g_i,susp}^{n+1} = \left((1-p)\beta_{gi}^n A_{B_i}^n - (-Sf_g - Sl_g)\frac{\Delta t}{\Delta x_i}\right)/((1-p)A_{B_i}^{n+1}) \tag{2.177}$$

2.3.2.2.6 Interpolation

To solve the eq. 2.164 and eq. 2.174 the total bed load fluxes over the edges ($\Phi_{B,iL}$, $\Phi_{B,iR}$) and the fluxes for the single grain classes ($Q_{B_g,iL}$, $Q_{B_g,iR}$) are needed, but the data for the computation of bed load are available only in the cross sections. For this reason the values on the edges are interpolated from the values calculated for the cross sections, depending on a weight choice of the user (θ):

$$\Phi_{B,iL} = (\theta)Q_{B,i-1} + (1-\theta)Q_{B,i} \tag{2.178}$$

If all values for the computation of Q_B by a bed load formula are taken from the cross section, the results of sediment transport tend to generate jags, as some effects of discretisation accumulate instead of being counterbalanced. For this reason it has been preferred not to take all values from the same location. The local discharge Q is substituted by a mean discharge for the edge, computed with the discharges in the upstream and downstream elements of the edge. This means that the bed load in a cross section will be calculated twice with different values of Q.

2.3.2.3 Discretisation of Source Terms

2.3.2.3.1 External Sediment Sources and Sinks

The discretisation of external sediment sources and sinks is analogous to BASEplane. Please see Section 2.3.3.3.1

2.3.2.3.2   Sediment Flux through Bottom of Bed Load Control Volume

The discretisation of the sediment flux through the bottom of the bed load control volume is analogous to BASEplane. Please see Section 2.3.3.3.3

2.3.2.3.3   Source Term for Sediment Exchange between Water and Bottom

The source term S_g from eq. 1.72 is computed in different ways depending on the scheme used for the solution of the advection equation.

a) With MDPM scheme

The source term S_g is calculated for the concentration value on the left and the concentration value on the right according to eq. 1.72 :

$$S_{g,L} = f(C_{iL}^n)$$
$$S_{g,R} = f(C_{iR}^n)$$

The volume exchanged with the bottom during Δt is given by the sum of the exchange on the left and the exchange on the right:

$$E_{L1} = \alpha S_{g,L} B \Delta x \Delta t \tag{2.179}$$

$$E_{L2} = \alpha_2 S_{g,L} B \Delta x \Delta t / 2 \tag{2.180}$$

$$E_L = E_{L1} + E_{L2} \tag{2.181}$$

$$E_{R1} = \alpha_2 S_{g,R} B \Delta x \Delta t / 2 \tag{2.182}$$

$$E_{R2} = \alpha_3 S_{g,R} B \Delta x \Delta t \tag{2.183}$$

$$E_R = E_{R1} + E_{R2} \tag{2.184}$$

Where α is defined like in eq. 2.154, $\alpha_2 = min(Cr_x, 1-\alpha)$ and $\alpha_3 = max(1-\alpha-Cr_x, 0)$
The final mean source term is then:

$$S_g = \frac{(E_L + E_R)}{B \Delta t \Delta x} \tag{2.185}$$

The exchange values on the right and left side are used to adjust the concentration values on the edges. The new concentrations on the edges after deposition or erosion in the left and right part of the cell are calculated by:

$$C_{iL}^{n+1} = \frac{C_{iL}^n A_i \alpha \Delta x + E_{L1}}{A_i \alpha \Delta x} \tag{2.186}$$

$$C_{iR}^{n+1} = \frac{C_{iR}^n A_i \alpha_3 \Delta x + E_{R2}}{A_i \alpha_3 \Delta x} \tag{2.187}$$

b) With QUICK and QUICKEST scheme

   S_g is calculated for each cross section according to eq. 1.72 .

c) With Holly-Preissmann scheme

   The Holly-Preissmann scheme should not be used with material erosion and deposition.

2.3.2.3.4   Splitting of Bed Load and Suspended Load Transport

The same size of particles can be transported by bed load as well as by suspended load. van Rijn (1984b) found a parameter which describes the relation between the two transport modes depending on the shear velocity u_* and the sink velocity w_k determined in eq. 1.60 - eq. 1.62.

$$
\begin{aligned}
\varphi_k &= 0 & if & & \left(\frac{u_*}{w_k}\right) &< 0.4 \\
\varphi_k &= 0.25 + 0.325 ln\left(\frac{u_*}{w_k}\right) & if & \quad 0.4 \le & \left(\frac{u_*}{w_k}\right) &\le 10 \\
\varphi_k &= 1 & if & & \left(\frac{u_*}{w_k}\right) &> 10
\end{aligned}
\tag{2.188}
$$

The computation of bed load flux and the computation of the exchange flux between suspended load and bed (eq. 1.71) have to be modified as follows:

$$
Q_{B_g} = (1 - \varphi_g)Q_{B_g}
\tag{2.189}
$$

$$
S_g = w_g(\varphi_g \beta_g C_{e_g} - C_{d_g})
\tag{2.190}
$$

2.3.2.3.5   Abrasion

As BASEMENT always works with volumes, the abrasion after Sternberg (eq. 1.117 in the Mathematical Models section) is applied as follows:

$$
V(x) = V_0 e^{-c(x-x_0)}
\tag{2.191}
$$

For the sediment balance in the element, the incoming sediment flux over the upstream edge is reduced by the factor:

$$
f = e^{-c(x-x_0)}
$$

where the x is the position of the present element and x_0 the position of the upstream element. The factor f is constant for an edge and is computed at the beginning of the computation.

2.3.2.4   Solution Procedure

The solution procedure for the one dimensional sediment transport is described in chapter Section 2.2.2.5.
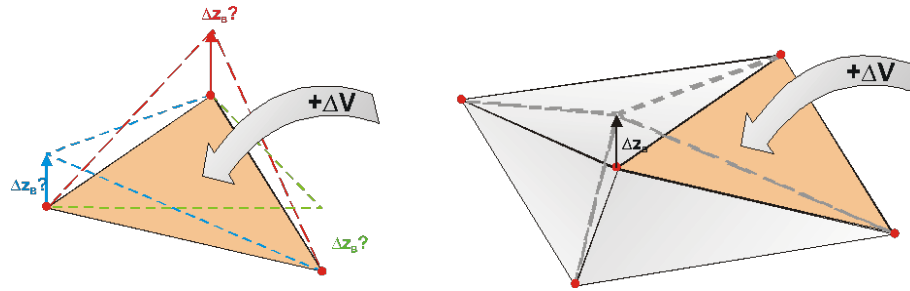
Figure 2.56 *Schematic illustration of problems related to the update of bed elevations*

2.3.3 Two Dimensional Sediment Transport

2.3.3.1 Spatial Discretisation

The finite volume method is applied to discretise the morphodynamic equations, slightly different from that of the hydrodynamic section. In the hydrodynamic discretisation a cell-centered approach is applied (see Figure 2.2). Thereby, the bed elevations are defined in the cell vertexes (nodes) of each cell. This arrangement, with bed elevations defined in the nodes, enables a more accurate representation of the topography compared to an approach with bed elevations defined in the cell centres. A further advantage of the chosen approach is that the slope within each cell is clearly defined by its nodal bed elevations.

But applying the same Discretisation approach for the sediment transport leads to several problematic aspects, which can be summarized briefly as follows.

- The change of a cell's sediment volume ΔV would have to be distributed on all nodes of the cell, where the bed elevations are defined. But it is not obvious by which criteria the sediment volume must be divided upon these nodes (see left part of Figure 2.56).

- If a nodal elevation would be changed due to a sediment inflow into a cell, this change in bed elevation would not only affect the sediment volume of this cell, but also the sediment volumes of all neighbouring cells (see right part of Figure 2.56). This situation is problematic regarding the conservation properties of the numerical scheme and it induces numerical fluxes into the neighbouring cells which cause undesired numerical diffusion.

To circumvent these problematic aspects and to ensure a fully conservative numerical scheme, a separate mesh is used for the spatial discretisation for the sediment transport. Because the hydraulic and sediment simulations are performed on different meshes, this approach is called "dual mesh morphodynamics" (DMMD) from here on. Both meshes, with its cells and edges, are illustrated in Figure 2.57 .

The cells of the sediment mesh are constructed around the nodes of the hydraulic mesh by connecting the midpoints of the edges and the centres of the hydraulic cells. This procedure results in the generation of median dual cells. Following this dual mesh approach all conservative variables of the sediment transport (z_b, β_g, C_g) are defined within the centres of the sediment cells, thus forming a standard finite volume approach regarding the sediment transport. Changes in bed elevation of a node do not influence the neighbouring sediment elements, as it is illustrated in Figure 2.58. Therefore this Discretisation approach is conservative and no diffusive fluxes into the neighbouring cells do occur.
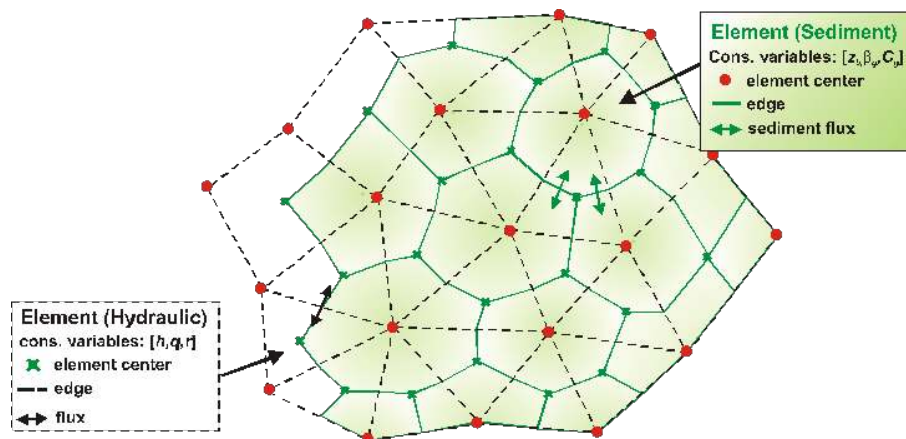
Figure 2.57 *Dual mesh approach with separate meshes for hydrodynamics (black) and sediment transport (green). Sediment cells have the bed elevations defined in their cell centers.*
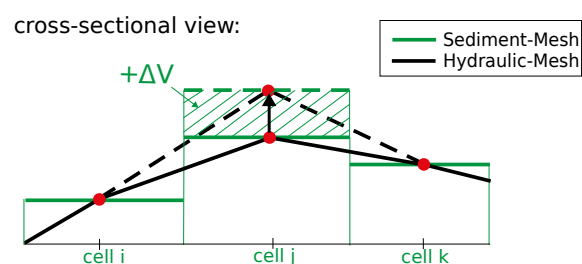
Figure 2.58 *Cross sectional view of dual mesh approach. Changes in sediment volume ΔV do not affect the neighboring cells' sediment volumes.*

The sediment mesh is generated automatically during the program start from the hydraulic mesh without the need of any additional information.

2.3.3.2   Discrete Form of Equations

2.3.3.2.1   Global Bed material Conservation Equation - Exner Equation

Considering the Exner equation (eq. 1.86) and applying the FVM, it can be written in the following integral form

$$(1-p)\int_{\Omega} \frac{\partial z_B}{\partial t}\,\mathrm{d}\Omega + \int_{\Omega}\sum_{g=1}^{ng}\left(\frac{\partial q_{B_g,x}}{\partial x} + \frac{\partial q_{B_g,y}}{\partial y}\right)\mathrm{d}\Omega = \int_{\Omega}\sum_{g=1}^{ng}(sl_g - s_{Bg})\,\mathrm{d}\Omega \qquad (2.192)$$

In which Ω is the same computational area as defined in hydrodynamic model (Figure 2.22). Using the Gauss' theory and assuming $\partial z_B/\partial t$ is constant over the element, one obtains

$$(1-p)\frac{z_B^{n+1} - z_B^n}{\Delta t} + \frac{1}{\Omega}\sum_{g=1}^{ng}\int_{\partial\Omega}\left(q_{B_g,x}n_x + q_{B_g,y}n_y\right)\mathrm{d}l = \frac{1}{\Omega}\sum_{g=1}^{ng}(Sl_g - S_{Bg}) \qquad (2.193)$$

Where n_x and n_y are components of the unit normal vector of the edge in x and y direction respectively.

2.3.3.2.2   Computation of Bed Load Fluxes

Direction of bed load flux

The direction of the bed load flux equals the direction of the velocity in near bed region and is a 2-D vector in a 2-D simulation. It is assumed here that this direction equals the direction of the depth-averaged velocity, although this assumption may become invalid in particular in curved channels with significant secondary flow motions. A correction of this flux direction is performed on sloped bed surfaces due to the gravitational induced lateral bed load flux component. The lateral transport component is perpendicular to the direction of flow velocity and therefore the resulting bed load flux vector is determined as

$$\overrightarrow{q}_{B_{res}} = \overrightarrow{q}_B + \overrightarrow{q}_{B_{lateral}} = q_B\begin{pmatrix}\cos\theta \\ \sin\theta\end{pmatrix} + q_{B_{lateral}}\begin{pmatrix}\sin\theta \\ -\cos\theta\end{pmatrix} \qquad (2.194)$$

where θ is the angle between the velocity vector and the x-axis.

Computation of bed load flux

The bed load transport capacity $\overrightarrow{q}_B$ and the lateral transport $\overrightarrow{q}_{B_{lateral}}$ are calculated using the transport formulas outlined in the Mathematical Models section. Different empirical transport formulas can be used and also fractional transport for multiple grain classes can be considered. These formulas require the flow variables and the soil compositions as data input.

As a consequence of the discretisation of the sediment elements as median dual cells, each sediment edge lies completely within a hydraulic element (see Figure 2.59, where sediment edges are indicated in green color). Therefore an obvious approach is to determine the
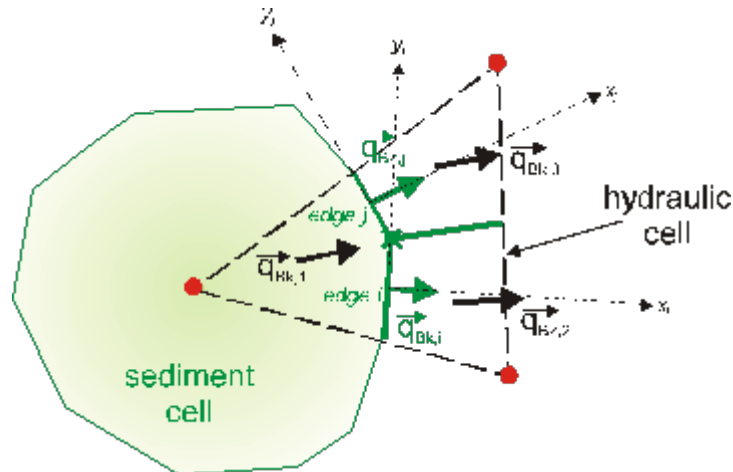
Figure 2.59 *Determination of bed load flux over the sediment cell edges i,j*

bed load fluxes over a sediment edge with the flow variables and the bottom shear stress defined in this hydraulic element. This eases the computations since no interpolations of hydraulic variables onto the sediment edges are necessary. Furthermore, it can be made use of the clearly defined bed slope within this hydraulic element, derived from its nodal elevations.

Following this approach the transport capacity is calculated with the flow variables defined in the centre of the hydraulic element. But since the transport capacity calculations also depend on the bed materials and grain compositions, this computation is repeated for every sediment element which partially overlaps the hydraulic element. Thus, one obtains multiple transport rates within the hydraulic element, as illustrated in Figure 2.59. (In case of single grain simulations, the bed material is the same over the hydraulic element and therefore the transport calculation must only be done once.) From these multiple transport rates an averaged transport rate over the sediment cell is determined by areal weighting.

Finally, the flux over the sediment cell edge is determined from the calculated transport capacities to its left and right sediment elements as

$$q_{B,edge} = [(\theta_{up})\overrightarrow{q}_{B,res,L} + (1 - \theta_{up})\overrightarrow{q}_{B,res,R}]\,\overrightarrow{n}_{edge} \qquad (2.195)$$

where θ_{up} is the upwind factor and $\overrightarrow{n}_{edge}$ is the normal vector of the edge.

Treatment of partially wetted elements

Per default no sediment transport is calculated within partially wetted hydraulic elements. This behaviour seems favourable in most situations. For example, in some cases it prevents upper parts of a river bank, which are not over flown, from automatically being eroded by sediment erosion which takes place only at the toe of the bank. But this default behaviour can be changed for special situations.

2.3.3.2.3   Flux Correction

When bed load fluxes are summed up over an element k, there may occur situations in which more sediment mass leaves the element than is actually available. Such situations

are observed for example when the bed level reaches a fixed bed elevation or bed armour where no further erosion can take place.

To guarantee sediment mass conservation over the whole domain a correction of the bed load fluxes which leave the element is applied in such situations. All the outgoing computed bed load fluxes of such an overdraft element k are reduced proportionally by a factor $\omega_{k,g}$. This factor is determined in a way that limits the overall outflow to the available sediment mass $V_{sed,k,g}$.

$$\omega_{k,g} = \begin{cases} \dfrac{\displaystyle\sum_{j=1}^{ng} Flux_{out,j,g} - V_{sed,k,g}}{\displaystyle\sum_{j=1}^{ng} Flux_{out,j,g}} & if \quad V_{sed,k,g} < \displaystyle\sum_{j=1}^{ng} Flux_{out,j,g} \\[4mm] 0 & if \quad V_{sed,k,g} \geq \displaystyle\sum_{j=1}^{ng} Flux_{out,j,g} \end{cases} \tag{2.196}$$

$$Flux_{out,k,g}^{corr} = (1 - \omega_{k,g}) Flux_{out,k,g}$$

2.3.3.2.4   Bed Material Sorting Equation

The global bed material conservation eq. 2.193 has to be solved first, because its results are needed to solve the bed material sorting equation. The bed material sorting equation (eq. 1.78) is also discretized using FVM. The discretized form is

$$(1-p)\frac{(h_m\beta_g)^{n+1} - (h_m\beta_g)^n}{\Delta t} + \frac{1}{\Omega}\int_{\partial\Omega} \left(q_{B_g,x}n_x + q_{B_g,y}n_y\right) dl + \frac{1}{\Omega}S_g - \frac{1}{\Omega}Sl_{B_g}^n - sf_g^* = 0 \tag{2.197}$$

The bed load control volume thickness h_m has to be determined before the new values of fractions are calculated through the eq. 2.197. The determination of h_m is detailed in Section 2.3.1.2. The solution of the bed material sorting equation finally yields the grain factions β_g^{n+1} at the new time level.

2.3.3.2.5   Advection-Diffusion Equation

The two dimensional suspended sediment or pollutant transport in river channels is described by eq. 1.68 in the Mathematical Models section. This equation has to be solved for each grain class g in the same manner. For this reason g is omitted in this section as well as the source terms. This yields to the following equation:

$$\frac{\partial}{\partial t}Ch + \frac{\partial}{\partial x}\left(Cq - h\Gamma\frac{\partial C}{\partial x}\right) + \frac{\partial}{\partial y}\left(Cr - h\Gamma\frac{\partial C}{\partial y}\right) = 0 \tag{2.198}$$

C is the concentration of transported suspended material averaged over flow depth.

The eq. 2.114 also can be written as follows.

$$(Ch)_t + \nabla(Cq, Cr) - \nabla(h\Gamma C_x, h\Gamma C_y) = 0 \tag{2.199}$$

It is integrated over the area of the hydraulic element Ω :

$$\int_{\Omega} \left((Ch)_t + \nabla(Cq, Cr) - \nabla(h\Gamma C_x, h\Gamma C_y) \right) \mathrm{d}\Omega = 0 \tag{2.200}$$

Applying the Gauss theorem, eq. 2.200 becomes:

$$\int_{\Omega} (Ch)_t \mathrm{d}\Omega + \oint_{\partial\Omega} (Cq, Cr)\overrightarrow{n}\,\mathrm{d}l - \oint_{\partial\Omega} (h\Gamma C_x, h\Gamma C_y)\overrightarrow{n}\,\mathrm{d}l = 0 \tag{2.201}$$

the different parts are calculated as follows:

$$\int_{\Omega} (C_i h_i)_t \mathrm{d}\Omega = h_i \int_{\Omega} \frac{\partial C_i}{\partial t}\mathrm{d}\Omega \approx h_i \frac{\partial C_i}{\partial t}\Omega \approx h_i \frac{C_i^{n+1} - C_i^n}{\Delta t}\Omega \tag{2.202}$$

the advective flux is:

$$\oint_{\partial\Omega} (Cq, Cr)\overrightarrow{n}\,\mathrm{d}l = \sum_j C_j(q_j, r_j)\overrightarrow{n_j}l_j = \sum_j \Phi_{a,j} \tag{2.203}$$

and the diffusive flux:

$$\oint_{\partial\Omega} (h\Gamma C_x, h\Gamma C_y)\overrightarrow{n}\,\mathrm{d}l = \sum_j h_j\Gamma(C_x, C_y)\overrightarrow{n_j}l_j = \sum_j \Phi_{d,j} \tag{2.204}$$

The concentration at the new time $n+1$ is:

$$C_i^{n+1} = C_i^n - \frac{\Delta t}{\Omega_i h_i}\sum_j \left(\Phi_{a,j} - \Phi_{d,j}\right) \tag{2.205}$$

2.3.3.2.6   Computation of Diffusive Flux

The diffusive flux over the edge is computed by the derivatives of the concentration over the edge multiplied with the vector normal to the edge, the edge length, the water depth on the edge and the diffusivity:

$$\Phi_{d,j} = h_j\Gamma \begin{pmatrix} \dfrac{\partial C_j}{\partial x} \\[2mm] \dfrac{\partial C_j}{\partial y} \end{pmatrix} \overrightarrow{n}_i l_j$$

The derivatives of the concentration are given by the mean value of the derivatives in the left and the right element of the edge:

$$\begin{pmatrix} \dfrac{\partial C_j}{\partial x} \\[2mm] \dfrac{\partial C_j}{\partial y} \end{pmatrix} = \frac{1}{2}\left[\begin{pmatrix} \dfrac{\partial C_L}{\partial x} \\[2mm] \dfrac{\partial C_L}{\partial y} \end{pmatrix} + \begin{pmatrix} \dfrac{\partial C_R}{\partial x} \\[2mm] \dfrac{\partial C_R}{\partial y} \end{pmatrix} \right] \tag{2.206}$$

The mean value of the derivatives of the concentration in an element can be transformed on a sum over the edges by the Gauss theorem:

$$\left(\frac{\partial C}{\partial x}\right)_i = \frac{1}{\Omega}\int_\Omega \frac{\partial C}{\partial x}\,\mathrm{d}\Omega \approx \frac{1}{\Omega}\sum_j C_j \Delta y_j \tag{2.207}$$

$$\left(\frac{\partial C}{\partial y}\right)_i = \frac{1}{\Omega}\int_\Omega \frac{\partial C}{\partial y}\,\mathrm{d}\Omega \approx \frac{1}{\Omega}\sum_j C_j \Delta x_j \tag{2.208}$$

2.3.3.2.7   Computation of Advective Flux using the Upwind Scheme

The Upwind Scheme is the simplest possible method to calculate the advective flux in eq. 2.203. The hydraulic discharges q and r over the edge are known from the hydraulic computation (eq. 2.53). The only challenge is the choice of the edge concentration C.

For the Finite Volume Method, the concentration is regarded as constant over every computation element. Instead of averaging element concentrations in order to get an edge concentration, the upwind scheme simply uses the concentration from the upwind element. The upwind element for an edge is the one element from which the hydraulic discharge originates.

Upwind schemes are computationally not expensive. However, their side effect is an increased numerical diffusion which flattens strong gradients within the concentration. If it was important to detect a sharp front in the concentration, a more accurate and more time-expensive scheme like the MDP-method should be used.

2.3.3.2.8   Computation of Advective Flux using the MDPM scheme

The computation of the advective flux with the MDP-method (Badrot-Nico et al., 2007), is described for the one dimensional case in Section 2.3.2.2.3 d). Because of the unstructured grid used in BASEMENT the MDP-method is not used separately in x and y-direction but directly in the direction of the local velocity.

The advective flux is calculated directly as a sediment discharge:

$$\Phi_{a,j} = C_j \begin{pmatrix} q_j \\ r_j \end{pmatrix} \overrightarrow{n}_j l_j$$

The discharge of water over the edge $\begin{pmatrix} q_j \\ r_j \end{pmatrix} \overrightarrow{n}_j$ is known from the hydraulic computation (eq. 2.53). As the concentration on the edge in the MDPM scheme is not constant in time, the flux is integrated over the time step.

The flux from the element i over the edge j can be described as:

$$\Phi_j = \frac{1}{\Delta t} h_j l_j \int_{t^n}^{t^{n+1}} u_i(t) C_j(t)\,\mathrm{d}t \tag{2.209}$$
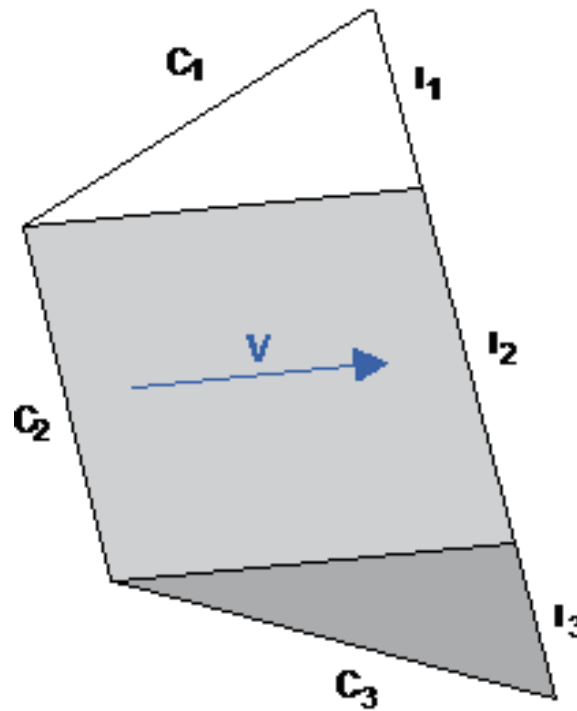
Figure 2.60 *Division of element and edge j in segments which receive the water from different upstream edges.*

The velocity u_i in the element is constant and its multiplication with the depth h_i on the edge gives the specific discharge q_i over the edge, which is known from the hydraulics. Hence the flux is:

$$\Phi_j = \frac{1}{\Delta t} q_j l_j \int\limits_{t^n}^{t^{n+1}} C_j(t)\,\mathrm{d}t \tag{2.210}$$

or:

$$\Phi_j = q_j l_j C_j(t) \tag{2.211}$$

Now the concentration $C_i(t)$ hast to be determined on the edge. It changes not only with time. If the edge is not perpendicular to the velocity, it is not constant over the whole length of the edge at one moment. It depends on the concentration on the edge from which the water comes. For an unstructured grid with triangular or quadrilateral elements these can be up to 3 different edges. For this reason the edge j is divided in k segments for which the fluxes Φ_k are computed separately and then summed up:

$$\Phi_j = \sum_1^k \Phi_k \tag{2.212}$$

This procedure is illustrated in figure Figure 2.60.

From now on only a segment concerned by one upstream edge will be considered. The concentration on the upstream edge is C_L and the one on the downstream edge C_R. The
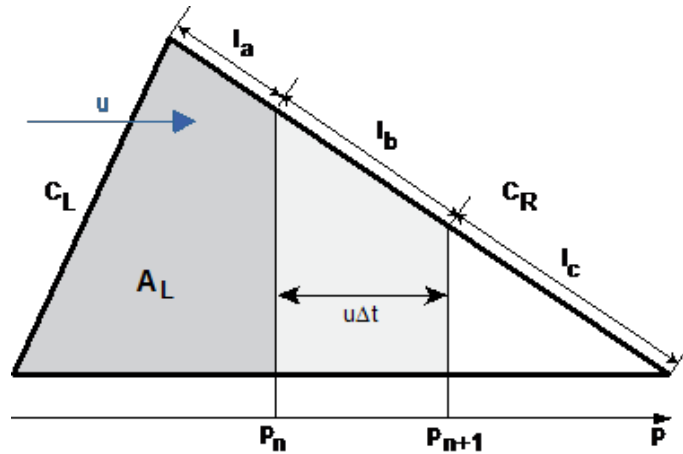
Figure 2.61 *Position of the front between the upstream and the downstream concentration at the beginning and at the end of the time step.*

part of the segment which is over flown with concentration C_L or C_R changes while the concentration front between the two concentrations advances.

The position of the front is determined in an analogous way to 1D with eq. 2.154. But instead of representing the fraction of the distance behind the front, α now represents the fraction of the area. The area which is behind the front therefore becomes:

$$A_L = \alpha_i A_i \tag{2.213}$$

The position of the front at the beginning of the time step p_n, can now be determined from the area A_L .

The position at the end of time step p_{n+1} is obtained by adding the distance covered during the time step $u\Delta t$. l_a indicates concentration C_L during the whole time step and l_c with concentration C_R. For l_b the fractions with C_L and C_R have to be integrated over time. It holdes:

$$\Phi_k = (C_L l_a + C_R l_c + (0.5C_L + 0.5C_R)l_b)q \tag{2.214}$$

Finally the concentrations on the edge j have to be replaced:

The value used as C_L, when the edge lies upstream, is replaced with C_L if the edge is partially behind the concentration front at the end of the time step. The value used as C_R, when the edge lies downstream, is replaced with C_L if the edge at the end of the time step lies completely behind the concentration front.

2.3.3.3   Discretisation of Source Terms

2.3.3.3.1   External Sediment Sources and Sinks

The source term Sl_B can be used, for example, to describe a local input or removal of sediment masses into a river or mass inflow due to slope failures. This source term can be added directly to the equations. It is specified as mass inflow with a defined grain mixture.

2.3.3.3.2   Dredge sources and sediment continuity at inner boundaries

A special subset of external sediment sinks is the so-called 'dredge source'. This source type allows for the definition of a constant dredge-level. Using the 'dredge source', sediment is removed from a cell if its bed elevation exceeds the specified dredge-level due to sediment deposition. The exceeding sediment is then removed by dredging. Additional parameters, like the maximum dredge rate, may be used to adjust the model to realistic scenarios.

This type of 'dredge source' can also be applied to achieve sediment continuity at inner boundary conditions, as e.g. an inner weir structure. It is possible to set dredge sources at the upstream region of the inner boundary, to prevent large sediment deposits and to hold a constant bed elevation. To achieve sediment continuity, one can add the removed sediment (due to dredging) to other elements, situated downstream of the inner structure. Thereby the sediments can pass the inner boundary.

2.3.3.3.3   Sediment Flux through Bottom of Bed Load Control Volume

The source term sf_g describes the change in volume of material of grain class g which enters or leaves the bed load control volume. The term sf_g is a time dependent source term and a function of grain fractions and the bed load control volume thickness. Therefore it has been handled in a special form in order to consider the time variation of the parameters. A two step method is applied, where in the first step the fractions are updated without the sf_g source term. Then, in the second step, this source term is computed with the advanced grain fractions values. The first step can be written as

$$\beta_g^{n+1/2} = \frac{1}{h_m^{n+1}} \left((h_m \beta_g)^n - \frac{\Delta t}{(1-p)\Omega} \int_{\partial\Omega} (q_{B_g,x} n_x + q_{B_g,y} n_y)^n \, \mathrm{d}l + \frac{\Delta t}{(1-p)\Omega} Sl_{B_g}^n \right)$$

(2.215)

After this predictor step, the advanced value $\beta_g^{n+1/2}$ is used for the calculation of the fractions of the new time level β_g^{n+1} by adding the $sf_g(\beta_g^{n+1/2})$ source term as

$$\beta_g^{n+1} = \beta_g^{n+1/2} + \frac{\Delta t}{h_m^{n+1}(1-p)} sf_g$$

(2.216)

And the sf_g source term, which describes the material which enters or leaves the bedload control volume, is finally discretized as given below.

$$sf_g = -(1-p)\frac{\beta_g^{n+1/2}(z_F^{n+1} - z_{sub}) - \beta_g^n(z_F^n - z_{sub})}{\Delta t}$$

(2.217)

In the calculation of this expression for the sf_g term, cases of erosion and deposition must be considered separately.

Erosion

In case of erosion the bottom of the bed load control volume z_F moves down and the fractions of the underneath layer enter the control volume. The fractions of this underneath layer are constant over time, i.e. $\beta_g = \beta_{g_{sub}}^n = \beta_{g_{sub}}^{n+1/2}$, and the source term therefore calculates as:
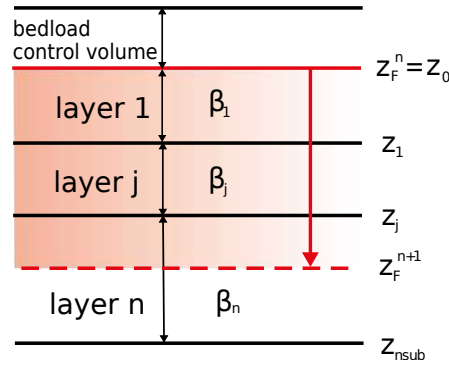
Figure 2.62 *Definition of mixed composition $\beta_{g_{mix}}$ for the eroded volume (red)*

$$sf_g = -(1-p)\beta_g \frac{(z_F^{n+1} - z_F^n)}{\Delta t} \tag{2.218}$$

Implementing this source term it must be paid attention to situations where the eroded bed volume comprises more than the first underneath sub layer. In such a situation the exchanged sediment does not exactly have the composition $\beta_{g_{sub}}^n$ of this first underneath layer, but a mixture $\beta_{g_{mix}}$ of the different compositions of all affected underneath sublayers (see Figure 2.62). In this implementation the number of the affected layers n_{sub} can be arbitrary.

The mixed grain composition $\beta_{g_{mix}}$ is determined by weighting the grain fractions with the layer thicknesses as

$$\beta_{g_{mix}} = \frac{1}{z_F^n - z_F^{n+1}} \sum_{j=1}^{nsub} \left[\beta_{g_j}(z_{j-1} - max(z_j, z_F^{n+1})) \right] \tag{2.219}$$

Deposition

In case of deposition, the bottom of the bed load control volume z_F moves up and material with the updated composition $\beta_g^{n+1/2}$ leaves the bed load control volume and enters the underneath layer. The source term therefore calculates as:

$$sf_g = -(1-p)\beta_g^{n+1/2} \frac{(z_F^{n+1} - z_F^n)}{\Delta t} \tag{2.220}$$

And a likewise term is used to update the grain compositions of the first underneath layer.

2.3.3.3.4   Sediment Exchange between Water and Bottom

The source term S_g describes the exchange between the suspended load in the water column and the sediment surface.

2.3.3.4   Gravitational Transport

2.3.3.4.1   Basic concepts

The erosion and deepening of stream leads to a steepening of the stream banks which finally can result in discontinuous mass collapses from time to time into the stream. The slope failures thereby are a main mechanism for the lateral widening of the stream. The occurrence of slope failures depends strongly on the soil characteristics and pore pressures present in the bank material. Furthermore, this gravitational induced transport type also plays a significant role in modelling dike or dam breaches due to overtopping.

The main idea of this geometrical approach is to assume that a slope failure takes place if the slope becomes steeper than a critical slope. If the critical value is exceeded, sediment material moves from the upper regions in direction of the slope and finally deposits in the lower region. This corresponds to a rotation of the cell in a way that its slope is flattened until the critical angle is reached. In order to be able to better represent the complex geotechnical aspects, it is distinguished here between three different critical failure angles in this approach:

1. A critical failure angle γ_{dry} , for partially saturated material at the bank which is not over flown. This angle may largely exceed the material's angle of repose γ_{rep} in small-grained materials due to stabilizing effects of negative pore pressures.

2. A critical failure angle γ_{wet} , for fully saturated material below the water surface. This angle can be assumed to be in the range of the material's angle of repose.

3. A deposition angle, γ_{dep} , for the deposited and not compacted material resulting from the slope collapse. This angle determines the sliding of the collapsed masses into the stream after the failure. It should be smaller than γ_{wet} and is supposed here to be in the range of about half the material's angle of repose.

These different critical angles thereby should fulfil the criteria $\gamma_{dry} > \gamma_{wet} > \gamma_{dep}$.

The idea of using different critical failure angles above and below the water surface already was successfully applied by previous numerical models for dike breaches (e.g. Faeh (2007)). In addition, recent laboratory tests of Soares-Frazão et al. (2007) clearly showed a formation of different side wall angles above and below the water surface in their experimental flume.

Algorithm of geometrical slope failure modelling

The geometrical approach is applied on the original mesh which is used for the hydraulic calculations (see Figure 2.63). This is advantageous because the slopes of the hydraulic cells are clearly defined by the elevations of their nodes. A similar approach of a 2D bank-failure operator applied on unstructured meshes was recently presented by Swartenbroekx et al. (2010). But due to the use of the dual-mesh approach the computation here differs significantly from their method. The computational algorithm consists of five successive computational steps:

(1) In a first step the steepness of a hydraulic cell's slope is used as an indicator if a slope failure has to be assumed. The appropriate critical failure angle is selected depending upon the water elevation. It is checked if the cell is wetted or dry and if the present sediment in the control volume was previously deposited or not.

(2) Then for each sediment edge i a volume V_i is calculated which must flow over the sediment edge in order to flatten the slope of the cell in a way that it no longer exceeds the critical value. Using Median-Dual cells for the sediment transport, this is easily possible since each sediment edge is situated completely within a hydraulic
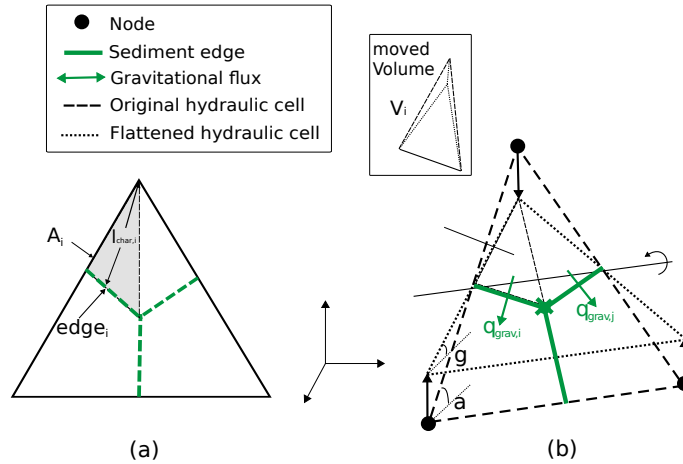
Figure 2.63 *Gravitational transport for an element with a slope angle larger than the critical slope angle (dashed = original situation, dotted = flattened slope after collapse)*

cell. The size of the volume V_i depends on the difference between the present slope in the cell and the critical slope which shall be set. The present slope and the critical slope in the cell are projected on the normal vector of this sediment edge i and the pyramidal volume is then determined as:

$$V_i = \frac{1}{3} A_i l_{char,i} (\boldsymbol{n}_i \boldsymbol{S}_i - \boldsymbol{n}_i \boldsymbol{S}_{crit,i})$$

with A_i = area above the sediment edge, $l_{char,I}$ = characteristic length, $\boldsymbol{S}_i$ = slope vector of the cell, $\boldsymbol{S}_{crit,i}$ = critical slope vector, $\boldsymbol{n}_i$ = normal vector of sediment edge and h_i = height of pyramidal volume. If γ_{dep} is set, then V_i is limited to the deposited material present in the cell.

(3) Finally the gravitational flux $q_{B_g,ngrav}$ over the edge is obtained by dividing this volume V_i by the time step size Δt. This flux leads to mass transport from the upper to the lower sediment cell which results in a flattening of the cell's slope.

$$q_{B_g,ngrav} = \frac{V_i}{\Delta t}$$

(4) The balancing of the gravitational fluxes and the determination of the new soil elevations z_B is achieved by applying and solving the Exner equation using the same numerical approaches as outlined for the bed load transport. This procedure ensures that fixed bed elevations or surface armouring layers are taken into account and the mass continuity is fulfilled.

(5) The modification of the slope of a cell in turn influences the slopes of all adjacent cells. For this reason the algorithm can be applied in an iterative manner also for the affected adjacent cells until finally no more slope angle is found which exceeds the critical value.

2.3.3.4.2   Fractional transport

The presented algorithm can be applied in this form for single grain computations only. For fractional transport additional aspects must be taken into account. In fractional transport
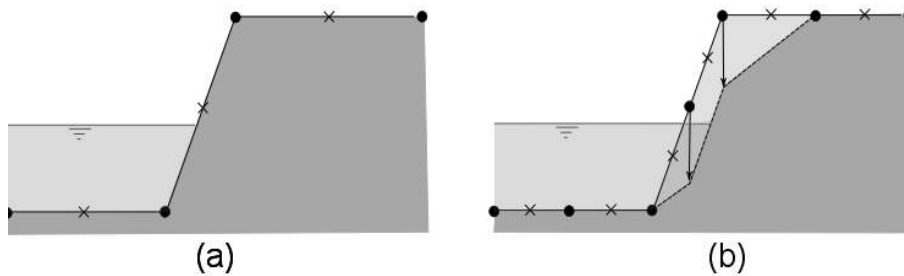
Figure 2.64 *Influence of grid resolution on the slope failures. (a) No slope failure with a coarse mesh at the indicated water level. (b) Slope failure at the same water level using a fine mesh.*

simulations, the failed slope may be composed of multiple soil layers with different grain compositions. If the slope fails, collapsed material from different soil layers with different grain compositions will be mixed and finally deposit at the toe of the slope. A detailed modelling of these mixing processes would require tracking the particles' individual motions and interactions, which cannot be achieved using the geometrical approach. Instead, a simple procedure is chosen to cope with this situation by applying the sorting equations to consider some mixing and the continuity of the failed masses. The moved material volume V_i is now limited to the available material within the bed load control volume. And because this control volume typically is rather small, with a height in the order of few grain diameters, the moved material in one computational step will be usually not sufficient to establish the critical slope. Therefore, the algorithm needs large number of iterations to establish the critical failure slopes.

2.3.3.4.3   Influence of grid resolution

Using the geometrical approach the side walls of a channel typically will collapse as soon as the water level wets at least one of the steep cells. If the cell becomes wetted the critical angle γ_{wet} is applied on this cell resulting in a collapse of the side wall. But in case of coarse discretization, the water level must be rather high until the critical angle γ_{wet} is applied. The slope failure therefore takes place slowly and may be underestimated. In contrast, in case of a finer grid resolution, the slope failure will take place earlier at a lower water level. Hence, the accuracy of the geometrical approach depends strongly on the grid resolution and a rather fine discretization should be applied at the areas of interest.

2.3.3.5   Management of Soil Layers

In case of sediment erosion, the bottom of the bed load control volume can sink below the bottom of the underneath soil layer. In such a situation the soil layer is completely eroded and consequently the data structure is removed. If the eroded layer was the last soil layer, than fixed bed conditions are set.

In case of sediment deposition, the uppermost soil layer grows in its thickness. And such an increase in layer thickness can continue up to large values during prolonged aggradation conditions. But very thick soil layers can be problematic, because the newly deposited sediments are completely mixed with the sediment materials over the whole layer thickness. Therefore a dynamic creation of new soil layers in multiple grain simulations should be enabled, which allows the formation of new soil layers.

Finding suitable general criteria for the creation of new soil layers is a difficult task. Here, two main conditions are identified and implemented:

- The thickness h_{sl} of the existing soil layer must exceed a given maximum layer thickness h_{max} before a new layer is created. Generation of a new soil layer: $h_{sl} > h_{max}$

- The grain composition of the depositing material must be different from the present grain composition of the soil layer. It is assumed here in a simple approach that this condition is fulfilled if the mean diameters of the grain compositions differ more than a given percentage P .

$$\left| \frac{d_{m,deposition} - d_{layer}}{d_{layer}} \right| > P/100$$

2.3.3.6   Solution Procedure

The manner - uncoupled, semi coupled or fully coupled - to solve eq. 1.34, eq. 1.77 and eq. 1.84 from the section Mathematical Models or appropriate derivatives with minor corrections has been often discussed over the last decade. A good overview is given by Kassem and Chaudhry (1998) or Cao et al. (2002). Uncoupled models are often blamed for their lacking of physical and numerical considerations. Vice versa coupled models are said to be very inefficient in computational effort and accordingly inapplicable for practical use. Kassem and Chaudhry (1998) showed that the difference of results calculated by coupled and semi coupled models is negligible. In addition Belleudy (2000) found that uncoupled solutions have nearly identical performance to coupled solutions even near critical flow conditions. Furthermore, the increasing difficulties and stability problems of coupled models that have to be expected when applying complicated sediment transport formulas or simulating multiple grain size classes are to be mentioned.

According to the preliminary state of this project the model with uncoupled solution of the water and sediment conservation equations has been chosen. This requires the assumption that changes in bed elevation and grain size distributions at the bed surface during one computational time step have to be slow compared to changes of the fluid variables, which dictates an upper limit on the computational time step. Consequently, the asynchronous solution procedure as depicted in Figure 2.65 and described in Figure 2.40 is justified: flow and sediment transport equations are solved uncoupled throughout the entire simulation period, i.e. with calculation of the flow field at the beginning of a given time step based on current bed topography and ensuing multiple sediment transport calculations based on the same flow field until the end of the respective time step. Thus the given duration of a calculation cycle Δt_{seq} (overall time step) consists of one hydraulic time step Δt_h and a resulting number of time steps Δt_s for sediment transport (see Figure 2.66).

By default, the mobile bed equations for sediment and suspension are solved using the current hydraulic time step with one cycle. For BASEplane, in case of quasi-stationary conditions where the changes in the hydraulic are small, the number of cycle step can be increased. The shallow water equations are solved using the hydraulic time step. The resulting water levels and velocities are then used to solve the mobile bed equations until they have been calculated for the number of cycle steps. This leads to a considerable speedup, reducing the calculation time as the hydraulic equations are solved only occasionally. The
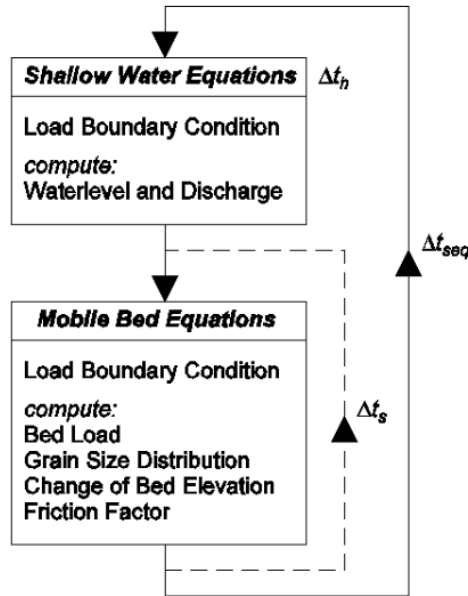
Figure 2.65 *Uncoupled asynchronous solution procedure consisting of sequential steps*
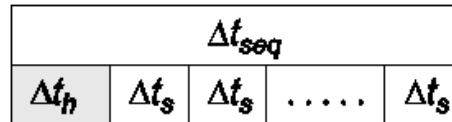
Figure 2.66 *Composition of the given overall calculation time step Δt_{seq}*

determination of the time step for the mobile bed equations depends on the model being used. For suspension transport, a time step size Δt_s is calculated to satisfy the numerical stability.

2.4 Time Discretisation and Stability Issues

2.4.1 Explicit Schemes

2.4.1.1 Euler First Order

The explicit time discretisation method implemented in BASEMENT is based on Euler first order method. According the method the full discretized equations are

$$\boldsymbol{U}_i^{n+1} = \boldsymbol{U}_i^n + RES(\boldsymbol{U})$$

$$z_B^{n+1} = z_B^n + RES(\boldsymbol{U}, d)$$

$$\beta_g^{n+1} = \beta_g^n + RES(\boldsymbol{U}, d_g, hm)$$

Where the $RES(\dots)$ is the summation of fluxes and source terms.

2.4.2 Determination of Time Step Size

2.4.2.1 Hydrodynamic

For explicit schemes, the hydraulic time step is determined according the restriction based on the Courant number. In case of the 2-D model the Courant number is defined as follows:

$$CFL = \frac{(\sqrt{u^2 + \nu^2} + c)\Delta t}{L} \leq 1 \tag{2.221}$$

Where L is the length of an edge with corresponding velocities of the element u, ν and $c = \sqrt{gh}$. In general, the CFL number has to be smaller than unity.

2.4.2.2 Bedload Transport

For the sediment transport another condition $c >> c_3$ holds true, which states that the wave speed of water c is much larger than the expansion velocity c_3 of a bottom discontinuity (eg. de Vries (1966)). Since the value of c_3 depends on multiple processes like bed load, lateral and gravity induced transport, its definitive determination is not obvious. Therefore the global time steps have been adopted based on the hydrodynamic condition.

2.4.2.3 Suspension Transport

For the suspension transport, the time step Δt_s is calculated similar to the hydraulic time step. However, the wave velocity $c = \sqrt{gh}$ is not taken into account, leading to slightly higher time step sizes. This time step is only active if a cycle step larger than 1 has been defined. By default, the cycle step is set to 1 and all mobile bed equations use the hydraulic time step.

2.4.3 Implicit Scheme

2.4.3.1 Introduction

In addition to the explicit scheme, BASEchain supports implicit calculations. To evaluate the evolution of the geometry of a channel as an effect of sediment transport, often long term computations are necessary. Additionally the calibration of a model with sediment transport is particularly laborious and needs many simulations. With the explicit solution of the hydraulics the needed simulation time becomes very large. This is because the explicit method uses a small time step, limited by the CFL-Number. The implicit method is needed to avoid this problem, allowing much larger time steps.

This chapter describes the implicit solution of the hydrodynamics in detail. The system of equations to solve is formed by eq. 1.3 and eq. 1.9 and applied to each cross section.

2.4.3.2 Time Discretisation

For the time discretisation of the differential equation $\partial u/\partial t = f(u)$ the θ-method is used:

$$\frac{u^{n+1} - u^n}{\Delta t} = \theta f(u^{n+1}) + (1 - \theta) f(u^n)$$

with $0 \leq \theta \leq 1$.

If $\theta = 0$ the explicit Euler method results: the values at time n+1 are computed solely from the old values at time n with limitation of the time step by the CFL-number. For $\theta = 0.5$ the scheme is second-order accuracy in time.

On the contrary, if $\theta = 1$ the solution is fully implicit. The equations are solved with the values at the new time n+1. As they are not known, initial values are assumed and the solution is approached to the exact solution by iteration. BASEchain uses the Newton-Raphson method, which has the quality to converge rapidly. However this is only the case if the initial values are sufficiently close to the exact solution. Here the values of the last time step are used as initial values for the iterations. This means that the more distant the new time from the old one and the bigger the change of the hydraulic state, the higher is the possibility that the solution cannot be found.

For the present implementation, the recommended value of θ is between 0.5 and 1.

2.4.3.2.1   Continuity Equation

The integration of the continuity equation

$$\int_{x_{iw}}^{x_{ie}} \left(\frac{\partial A_{si}}{\partial t} + \frac{\partial Q}{\partial x} - q \right) dx = F_{\hat{\imath}}^n + F_{\hat{\imath}}^\Phi + F_{\hat{\imath}}^q = 0$$

gives the following integral terms.

$$F_{\hat{\imath}}^n = \int_{x_{iw}}^{x_{ie}} \frac{\partial A_{si}}{\partial t} \, dx$$

$$F_{\hat{\imath}}^\Phi = \int_{x_{iw}}^{x_{ie}} \frac{\partial Q}{\partial x} \, dx$$

$$F_{\hat{\imath}}^q = - \int_{x_{iw}}^{x_{ie}} q \, dx$$

Applying the θ -method:

$$F_{\hat{\imath}}^{n+1} = F_{\hat{\imath}}^n + \theta (F_{\hat{\imath}}^\Phi + F_{\hat{\imath}}^q) + (1 - \theta)(F_{\hat{\imath}}^{\Phi 0} + F_{\hat{\imath}}^{q0}) \qquad (2.222)$$

2.4.3.2.2   Momentum Equation

The integration of the momentum equation:

$$\int\limits_{x_{iw}}^{x_{ie}} \left(\frac{\partial Q}{\partial t} + \frac{\partial}{\partial x} \left(\beta \frac{Q^2}{A_{red}} \right) + gA_{red}\frac{\partial z}{\partial x} + gA_{red}S_f \right) dx = G_i^n + G_i^\Phi + G_i^{sz} + G_i^{sf} = 0$$

leads to the following integral terms:

$$G_i^n = \int\limits_{x_{iw}}^{x_{ie}} \frac{\partial Q}{\partial t} \, dx$$

$$G_i^\Phi = \int\limits_{x_{iw}}^{x_{ie}} \frac{\partial}{\partial x} \left(\beta \frac{Q^2}{A_{red}} \right) dx$$

$$G_i^{sz} = \int\limits_{x_{iw}}^{x_{ie}} gA_{red}\frac{\partial z}{\partial x} \, dx$$

$$G_i^{sf} = \int\limits_{x_{iw}}^{x_{ie}} gA_{red}S_f \, dx$$

Applying the θ-method:

$$G_i^{n+1} = G_i^n + \theta(G_i^\Phi + G_i^{sz} + G_i^{sf}) + (1 - \theta)(G_i^{\Phi 0} + G_i^{sz0} + G_i^{sf0}) \qquad (2.223)$$

2.4.3.3   Solution

The equation system is

$$F(x) = 0$$

with

$$F = (F_1, G_1, F_2, G_2, \ldots, F_{n-1}, G_{n-1}, F_n, G_n)$$

and

$$x = (A_1, Q_1, A_2, Q_2, \ldots, A_{n-1}, Q_{n-1}, A_n, Q_n)$$

The system is solved by the Newton-Raphson method. Starting from an approximated solution x^k, the corresponding improved solution x^{k+1} is determined by the linear equation system.

$$Ax^{k+1} - c = 0 \qquad (2.224)$$

with

$$c = Ax^k - F$$

and

$A =$

$$
\begin{bmatrix}
\frac{\partial F_1}{\partial A_{red,1}} & \frac{\partial F_1}{\partial Q_1} & \frac{\partial F_1}{\partial A_{red,2}} & \frac{\partial F_1}{\partial Q_2} & & & & & & & & \\
\frac{\partial G_1}{\partial A_{red,1}} & \frac{\partial G_1}{\partial Q_1} & \frac{\partial G_1}{\partial A_{red,2}} & \frac{\partial G_1}{\partial Q_2} & & & & & & & & \\
\frac{\partial F_2}{\partial A_{red,1}} & \frac{\partial F_2}{\partial Q_1} & \frac{\partial F_2}{\partial A_{red,2}} & \frac{\partial F_2}{\partial Q_2} & \frac{\partial F_2}{\partial A_{red,3}} & \frac{\partial F_2}{\partial Q_3} & & & & & & \\
\frac{\partial G_2}{\partial A_{red,1}} & \frac{\partial G_2}{\partial Q_1} & \frac{\partial G_2}{\partial A_{red,2}} & \frac{\partial G_2}{\partial Q_2} & \frac{\partial G_2}{\partial A_{red,3}} & \frac{\partial G_2}{\partial Q_3} & & & & & & \\
& & \frac{\partial F_3}{\partial A_{red,2}} & \frac{\partial F_3}{\partial Q_2} & \frac{\partial F_3}{\partial A_{red,3}} & \frac{\partial F_3}{\partial Q_3} & \frac{\partial F_3}{\partial A_{red,4}} & \frac{\partial F_3}{\partial Q_4} & & & & \\
& & \frac{\partial G_3}{\partial A_{red,2}} & \frac{\partial G_3}{\partial Q_2} & \frac{\partial G_3}{\partial A_{red,3}} & \frac{\partial G_3}{\partial Q_3} & \frac{\partial G_3}{\partial A_{red,4}} & \frac{\partial G_3}{\partial Q_4} & & & & \\
& & & & & \ddots & & & \ddots & & & \\
& & & & & & \frac{\partial F_{n-2}}{\partial A_{red,n-3}} & \frac{\partial F_{n-2}}{\partial Q_{n-3}} & \frac{\partial F_{n-2}}{\partial A_{red,n-2}} & \frac{\partial F_{n-2}}{\partial Q_{n-2}} & \frac{\partial F_{n-2}}{\partial A_{red,n-1}} & \frac{\partial F_{n-2}}{\partial Q_{n-1}} \\
& & & & & & \frac{\partial G_{n-2}}{\partial A_{red,n.3}} & \frac{\partial G_{n-2}}{\partial Q_{n-3}} & \frac{\partial G_{n-2}}{\partial A_{red,n-2}} & \frac{\partial G_{n-2}}{\partial Q_{n-2}} & \frac{\partial G_{n-2}}{\partial A_{red,n-1}} & \frac{\partial G_{n-2}}{\partial Q_{n-1}} \\
& & & & & & & & \frac{\partial F_{n-1}}{\partial A_{red,n-2}} & \frac{\partial F_{n-1}}{\partial Q_{n-2}} & \frac{\partial F_{n-1}}{\partial A_{red,n-1}} & \frac{\partial F_{n-1}}{\partial Q_{n-1}} & \frac{\partial F_{n-1}}{\partial A_{red,n}} & \frac{\partial F_{n-1}}{\partial Q_n} \\
& & & & & & & & \frac{\partial G_{n-1}}{\partial A_{red,n-2}} & \frac{\partial G_{n-1}}{\partial Q_{n-2}} & \frac{\partial G_{n-1}}{\partial A_{red,n-1}} & \frac{\partial G_{n-1}}{\partial Q_{n-1}} & \frac{\partial G_{n-1}}{\partial A_{red,n}} & \frac{\partial G_{n-1}}{\partial Q_n} \\
& & & & & & & & & & \frac{\partial F_n}{\partial A_{red,n-1}} & \frac{\partial F_n}{\partial Q_{n-1}} & \frac{\partial F_n}{\partial A_{red,n}} & \frac{\partial F_n}{\partial Q_n} \\
& & & & & & & & & & \frac{\partial G_n}{\partial A_{red,n-1}} & \frac{\partial G_n}{\partial Q_{n-1}} & \frac{\partial G_n}{\partial A_{red,n}} & \frac{\partial G_n}{\partial Q_n}
\end{bmatrix}
$$

2.4.3.4 Integral Terms

2.4.3.4.1 Continuity Equation

The integral terms of the continuity equation are approximated as follows. For a general cross section i:

$$F_i^{n+1} = F_i^n + F_i^{\Phi} + F_i^q$$

$$F_i^n = \frac{A_{si} - A_{si}^0}{\Delta t} \Delta x_i$$

$$F_i^{\Phi} = \Phi_{ie}^c - \Phi_{iw}^c$$

$$F_i^q = -q_i^- \Delta x_i^- - q_i^+ \Delta x_i^+$$

For the first cross section i = 1:

$$F_1^n = \frac{A_1 - A_1^0}{\Delta t} \Delta x_1$$

$$F_1^{\Phi} = \Phi_{1e}^c - Q_{in}$$

$$F_1^q = -q_1^+ \Delta x_1$$

For the last cross section i = N

$$F_n^n = \frac{A_n - A_n^0}{\Delta t} \Delta x_n$$

$$F_n^{\Phi} = Q_{out} - \Phi_{nw}^c$$

$$F_n^q = -q_n^- \Delta x_n$$

2.4.3.4.2   Momentum Equation

The integral terms of the continuity equation are approximated as follows. For a general cross section i:

$$G_i^n = \frac{Q_i - Q_i^0}{\Delta t} \Delta x_i$$

$$G_i^{\Phi} = \Phi_{ie}^m - \Phi_{iw}^m$$

$$G_i^{sz} = gA_{red,i} \frac{z_{i+1} - z_{i-1}}{2}$$

$$G_i^{sf} = gA_{red,i}S_{fi}\Delta x_i \qquad S_{fi} = \frac{Q_i|Q_i|}{K_i^2}$$

For the first cross section i = 1:

$$G_1^n = \frac{Q_{in} - Q_{in}^0}{\Delta t} \Delta x_1$$

$$G_1^{\Phi} = \Phi_{1e}^m - \frac{\beta_1 Q_{in}^2}{A_{red,in}}$$

$$G_1^{sz} = gA_{red,1} \frac{z_2 - z_1}{2}$$

$$G_1^{sf} = gA_{red,1}S_{f1}\Delta x_1 \qquad S_{f1} = \frac{Q_1|Q_1|}{K_1^2}$$

For the last cross section i = n

$$G_n^n = \frac{Q_{out} - Q_{out}^0}{\Delta t} \Delta x_n$$

$$G_n^\Phi = \frac{\beta_n Q_{out}^2}{A_{red,out}} - \Phi_{nw}^m$$

$$G_n^{sz} = g A_n \frac{z_n - z_{n-1}}{2}$$

$$G_n^{sf} = g A_n S_{fn} \Delta x_n \qquad S_{fn} = \frac{Q_n |Q_n|}{K_n^2}$$

2.4.3.5   General Description of the Derivatives for the Matrix A

2.4.3.5.1   Derivatives of the Continuity Equation

For a general cross section i:

$$\frac{\partial F_i}{\partial A_{red,i-1}} = \frac{\partial \Phi_{ie}^c}{\partial A_{red,i-1}} - \frac{\partial \Phi_{iw}^c}{\partial A_{red,i-1}}$$

$$\frac{\partial F_i}{\partial A_{red,i}} = \frac{\Delta x_i}{\Delta t} \frac{dA_i}{dA_{red,i}} + \frac{\partial \Phi_{ie}^c}{\partial A_{red,i}} - \frac{\partial \Phi_{iw}^c}{\partial A_{red,i}}$$

$$\frac{\partial F_i}{\partial A_{red,i+1}} = \frac{\partial \Phi_{ie}^c}{\partial A_{red,i+1}} - \frac{\partial \Phi_{iw}^c}{\partial A_{red,i+1}}$$

$$\frac{\partial F_i}{\partial Q_{i-1}} = \frac{\partial \Phi_{ie}^c}{\partial Q_{i-1}} - \frac{\partial \Phi_{iw}^c}{\partial Q_{i-1}}$$

$$\frac{\partial F_i}{\partial Q_i} = \frac{\partial \Phi_{ie}^c}{\partial Q_i} - \frac{\partial \Phi_{iw}^c}{\partial Q_i}$$

$$\frac{\partial F_i}{\partial Q_{i+1}} = \frac{\partial \Phi_{ie}^c}{\partial Q_{i+1}} - \frac{\partial \Phi_{iw}^c}{\partial Q_{i+1}}$$

For the first cross section i = 0:

$$\frac{\partial F_1}{\partial A_{red,1}} = \frac{\Delta x_1}{\Delta t} \frac{dA_1}{dA_{red,1}} + \frac{\partial \Phi_{1e}^c}{\partial A_{red,1}} - \frac{\partial Q_{in}}{\partial A_{red,1}}$$

$$\frac{\partial F_1}{\partial A_{red,2}} = \frac{\partial \Phi_{1e}^c}{\partial A_{red,2}} - \frac{\partial \Phi_{in}^c}{\partial A_{red,2}}$$

$$\frac{\partial F_1}{\partial Q_1} = \frac{\partial \Phi_{1e}^c}{\partial Q_{1=in}} - \frac{\partial Q_{in}}{\partial Q_{1=in}}$$

$$\frac{\partial F_1}{\partial Q_2} = \frac{\partial \Phi_{1e}^c}{\partial Q_2} - \frac{\partial Q_{in}}{\partial Q_2}$$

For the last cross section i = n:

$$\frac{\partial F_n}{\partial A_{red,n-1}} = \frac{\partial Q_{out}}{\partial A_{red,n-1}} - \frac{\partial \Phi_{nw}^c}{\partial A_{red,n-1}}$$

$$\frac{\partial F_n}{\partial A_{red,n}} = \frac{\Delta x_n}{\Delta t}\frac{dA_n}{dA_{red,n}} + \frac{\partial Q_{out}}{\partial A_{red,n}} - \frac{\partial \Phi_{nw}^c}{\partial A_{red,n}}$$

$$\frac{\partial F_n}{\partial Q_{n-1}} = \frac{\partial Q_{out}}{\partial Q_{n-1}} - \frac{\partial \Phi_{nw}^c}{\partial Q_{n-1}}$$

$$\frac{\partial F_n}{\partial Q_n} = \frac{\partial Q_{out}}{\partial Q_{n=out}} - \frac{\partial \Phi_{nw}^c}{\partial Q_{n=out}}$$

2.4.3.5.2   Derivatives of the Momentum Equation

For a general cross section i:

$$\frac{\partial G_i}{\partial A_{red,i-1}} = \frac{\partial \Phi_{ie}^m}{\partial A_{red,i-1}} - \frac{\partial \Phi_{iw}^m}{\partial A_{red,i-1}} + \frac{g A_{red,i}}{2}\frac{dz_{i-1}}{dA_{red,i-1}}$$

$$\frac{\partial G_i}{\partial A_{red,i}} = \frac{\partial \Phi_{ie}^m}{\partial A_{red,i}} - \frac{\partial \Phi_{iw}^m}{\partial A_{red,i}} + \frac{g}{2}(z_{i+1} - z_{i-1}) + g\Delta x_i S_{fi}\left(1 - 2\frac{A_{red,i}}{K_i}\frac{dK_i}{dA_i}\right)$$

$$\frac{\partial G_i}{\partial A_{red,i+1}} = \frac{\partial \Phi_{ie}^m}{\partial A_{red,i+1}} - \frac{\partial \Phi_{iw}^m}{\partial A_{red,i+1}} + g\frac{A_{red,i}}{2}\frac{dz_{i+1}}{dA_{red,i+1}}$$

$$\frac{\partial G_i}{\partial Q_{i-1}} = \frac{\partial \Phi_{ie}^m}{\partial Q_{i-1}} - \frac{\partial \Phi_{iw}^m}{\partial Q_{i-1}}$$

$$\frac{\partial G_i}{\partial Q_i} = \frac{\Delta x_i}{\Delta t} + \frac{\partial \Phi_{ie}^m}{\partial Q_i} - \frac{\partial \Phi_{iw}^m}{\partial Q_i} + 2g A_{red,i}\Delta x_i\frac{S f_i}{Q_i}$$

$$\frac{\partial G_i}{\partial Q_{i+1}} = \frac{\partial \Phi_{ie}^m}{\partial Q_{i+1}} - \frac{\partial \Phi_{iw}^m}{\partial Q_{i+1}}$$

For the first cross section i = 1:

$$\frac{\partial G_1}{\partial A_{red,1}} = \frac{\partial \Phi_{1e}^m}{\partial A_{red,1}} - \frac{\partial \Phi_{in}^m}{\partial A_{red,1}} + \frac{g}{2}(z_2 - z_1) + g\Delta x_1 S_{f1}\left(1 - 2\frac{A_1}{K_1}\frac{dK_1}{dA_{red,1}}\right)$$

$$\frac{\partial G_1}{\partial A_{red,2}} = \frac{\partial \Phi_{1e}^m}{\partial A_{red,2}} - \frac{\partial \Phi_{in}^m}{\partial A_{red,2}} + g\frac{A_1}{2}\frac{dz_2}{dA_2}$$

$$\frac{\partial G_1}{\partial Q_1} = \frac{\Delta x_1}{\Delta t} + \frac{\partial \Phi_{1e}^m}{\partial Q_{in}} - \frac{\partial \frac{\beta_1 Q_{in}^2}{A_{red,in}}}{\partial Q_{in}} + 2g A_{red,1}\Delta x_1\frac{S f_1}{Q_{in}}$$

$$\frac{\partial G_1}{\partial Q_2} = \frac{\partial \Phi_{1e}^m}{\partial Q_2} - \frac{\partial \dfrac{\beta_1 Q_{in}^2}{A_{red,in}}}{\partial Q_2}$$

For the last cross section i = n:

$$\frac{\partial G_n}{\partial A_{red,n-1}} = \frac{\partial \Phi_{out}^m}{\partial A_{red,n-1}} - \frac{\partial \Phi_{nw}^m}{\partial A_{red,n-1}} + g\frac{A_{red,n}}{2}\frac{dz_{n-1}}{dA_{red,n-1}}$$

$$\frac{\partial G_n}{\partial A_{red,n}} = \frac{\partial \Phi_{out}^m}{\partial A_{red,n}} - \frac{\partial \Phi_{nw}^m}{\partial A_{red,n}} + \frac{g}{2}(z_n - z_{n-1}) + g\Delta x_n S_{fn}\left(1 - 2\frac{A_n}{K_n}\frac{dK_n}{dA_{red,n}}\right)$$

$$\frac{\partial G_n}{\partial Q_{n-1}} = \frac{\partial \Phi_{out}^m}{\partial Q_{n-1}} - \frac{\partial \Phi_{nw}^m}{\partial Q_{n-1}}$$

$$\frac{\partial G_n}{\partial Q_n} = \frac{\Delta x_n}{\Delta t} + \frac{\partial \Phi_{out}^m}{\partial Q_n} - \frac{\partial \Phi_{nw}^m}{\partial Q_n} + 2gA_{red,n}\Delta x_n\frac{Sf_n}{Q_n}$$

2.4.3.6 Determination of the Derivatives with Upwind Flux Determination

With the upwind method the flux is defined as follows:

$$f(x_{ie}) = \Gamma_i^{ie} f_i + \Gamma_{i+1}^{ie} f_{i+1} \tag{2.225}$$

with

$$\begin{array}{llllll} \Gamma_i^{ie} = 1 & and & \Gamma_{i+1}^{ie} = 0 & if & Q_i + Q_{i+1} \geq 0 \\ \Gamma_i^{ie} = 0 & and & \Gamma_{i+1}^{ie} = 1 & if & Q_i + Q_{i+1} < 0 \end{array}$$

2.4.3.6.1 Derivatives of the Continuity Equation

For a general cross section:

$$\frac{\partial F_i}{\partial A_{red,i-1}} = 0$$

$$\frac{\partial F_i}{\partial A_{red,i}} = \frac{\Delta x_i}{\Delta t}\frac{\partial A_i}{\partial A_{red,i}}$$

$$\frac{\partial F_i}{\partial A_{red,i+1}} = 0$$

$$\frac{\partial F_i}{\partial Q_{i-1}} = -\theta\Gamma_{i-1}^{i-0.5}$$

$$\frac{\partial F_i}{\partial Q_i} = \theta(\Gamma_i^w - \Gamma_i^e)$$

$$\frac{\partial F_i}{\partial Q_{i+1}} = \theta \Gamma_{i+1}^e$$

For cross section i =1:

$$\frac{\partial F_1}{\partial A_{red,1}} = \frac{\Delta x_1}{\Delta t} \frac{\partial A_1}{\partial A_{red,1}}$$

$$\frac{\partial F_1}{\partial Q_1} = \theta(\Gamma_1^e - 1)$$

$$\frac{\partial F_1}{\partial Q_2} = \theta \Gamma_2^e$$

For cross section i =n:

$$\frac{\partial F_n}{\partial A_n} = \frac{\Delta x_n}{\Delta t} \frac{\partial A_n}{\partial A_{red,n}}$$

$$\frac{\partial F_n}{\partial Q_{n-1}} = -\theta \Gamma_{n-1}^w$$

$$\frac{\partial F_n}{\partial Q_n} = \theta(1 - \Gamma_n^w)$$

2.4.3.6.2   Derivatives of the Momentum Equation

For a general cross section i:

$$\frac{\partial G_i}{\partial A_{red,i-1}} = \theta \left(\Gamma_{i-1}^w \left[\beta \frac{Q^2}{A_{red}} \right] \left(\frac{1}{A_{red,i-1}} - \frac{1}{\beta_{i-1}} \frac{d\beta_{i-1}}{dA_{red,i-1}} \right) \right) - g \frac{A_{red,i}}{2} \frac{dz_{i-1}}{dA_{red,i-1}}$$

$$\frac{\partial G_i}{\partial A_{red,i}} = \theta \left((\Gamma_i^w - \Gamma_i^e) \left[\beta \frac{Q^2}{A_{red}} \right]_i \left(\frac{1}{A_{red,i}} - \frac{1}{\beta_i} \frac{d\beta_i}{dA_{red,i}} \right) + \frac{g}{2}(z_{i+1} - z_{i-1}) \right) +$$
$$\theta \left(g\Delta x_i S_{fi} \left(1 - 2\frac{A_{red,i}}{K_i} \frac{dK_i}{dA_{red,i}} \right) \right)$$

$$\frac{\partial G_i}{\partial A_{red,i+1}} = \theta \left(-\Gamma_{i+1}^e \left[\beta \frac{Q^2}{A_{red}} \right]_{i+1} \left(\frac{1}{A_{red,i+1}} - \frac{1}{\beta_{i+1}} \frac{d\beta_{i+1}}{dA_{red,i}} \right) + g \frac{A_{red,i}}{2} \frac{dz_{i+1}}{dA_{red,i+1}} \right)$$

$$\frac{\partial G_i}{\partial Q_{i-1}} = -2\theta \Gamma_{i-1}^w \frac{1}{Q_{i-1}} \left[\beta \frac{Q^2}{A_{red}} \right]_{i-1}$$

$$\frac{\partial G_i}{\partial Q_i} = \frac{\Delta x_i}{\Delta t} + 2\theta \left((\Gamma_i^e - \Gamma_i^w) \frac{1}{Q_i} \left[\beta \frac{Q^2}{A_{red}} \right]_i + 2gA_{red,i}\Delta x_i \frac{Sf_i}{Q_i} \right)$$

$$\frac{\partial G_i}{\partial Q_{i+1}} = 2\theta\Gamma^e_{i+1}\frac{1}{Q_{i+1}}\left[\beta\frac{Q^2}{A_{red}}\right]_{i+1}$$

For cross section i =1:

$$\frac{\partial G_1}{\partial A_{red,1}} = \theta\left((1-\Gamma^e_1)\left[\beta\frac{Q^2}{A_{red}}\right]_1\left(\frac{1}{A_{red,1}} - \frac{1}{\beta_1}\frac{d\beta_1}{dA_{red,1}}\right) + \frac{g}{2}\left(z_2 - z_1 - A_{red,1}\frac{dz_1}{dA_{red,1}}\right)\right) +$$
$$\theta\left(g\Delta x_1 S_{f1}\left(1 - 2\frac{A_{red,1}}{K_1}\frac{dK_1}{dA_{red,1}}\right)\right)$$

$$\frac{\partial G_1}{\partial A_{red,2}} = \theta\left(-\Gamma^e_2\left[\beta\frac{Q^2}{A_{red}}\right]_2\left(\frac{1}{A_{red,2}} - \frac{1}{\beta_2}\frac{d\beta_2}{dA_{red,2}}\right) + g\frac{A_{red,2}}{2}\frac{dz_2}{dA_{red,2}}\right)$$

$$\frac{\partial G_1}{\partial Q_1} = \frac{\Delta x_1}{\Delta t} + 2\theta\left((\Gamma^e_1 - 1)\frac{1}{Q_1}\left[\beta\frac{Q^2}{A_{red}}\right]_1 + 2gA_{red,1}\Delta x_1\frac{Sf_1}{Q_1}\right)$$

$$\frac{\partial G_1}{\partial Q_2} = 2\theta\Gamma^e_2\frac{1}{Q_2}\left[\beta\frac{Q^2}{A_{red}}\right]_2$$

For cross section i =n:

$$\frac{\partial G_n}{\partial A_{red,n-1}} = \theta\left(\Gamma^w_{n-1}\left[\beta\frac{Q^2}{A_{red}}\right]_{n-1}\left(\frac{1}{A_{n-1}} - \frac{1}{\beta_{n-1}}\frac{d\beta_{n-1}}{dA_{n-1}}\right)\right) - g\frac{A_n}{2}\frac{dz_{n-1}}{dA_{red,n-1}}$$

$$\frac{\partial G_n}{\partial A_{red,n}} = \theta\left((\Gamma^w_n - 1)\left[\beta\frac{Q^2}{A_{red}}\right]_n\left(\frac{1}{A_n} - \frac{1}{\beta_n}\frac{d\beta_n}{dA_n}\right) + \frac{g}{2}\left(z_n - z_{n-1} + A_n\frac{dz_n}{dA_n}\right)\right) +$$
$$\theta\left(g\Delta x_n S_{fn}\left(1 - 2\frac{A_n}{K_n}\frac{dK_n}{dA_n}\right)\right)$$

$$\frac{\partial G_n}{\partial Q_{n-1}} = -2\theta\Gamma^w_{n-1}\frac{1}{Q_{n-1}}\left[\beta\frac{Q^2}{A}\right]_{n-1}$$

$$\frac{\partial G_n}{\partial Q_n} = \frac{\Delta x_n}{\Delta t} + 2\theta\left((1 - \Gamma^w_n)\frac{1}{Q_n}\left[\beta\frac{Q^2}{A}\right]_n + gA_n\Delta x_n\frac{Sf_n}{Q_n}\right)$$

2.4.3.7   Determination of the Derivatives with Roe Flux Determination

Fluxes and derivatives of the fluxes of the continuity equation:

$$\Phi^c_{ie} = Q(ie) = \frac{1}{2}(Q_i + Q_{i+1}) - R_{cA}(A_{i+1} - A_i) - R_{cQ}(Q_{i+1} - Q_i) \tag{2.226}$$

$$\frac{\partial \Phi_{ie}^c}{\partial A_{i-1}} = 0 \quad \frac{\partial \Phi_{ie}^c}{\partial A_i} = R_{cA}^{ie} \qquad \frac{\partial \Phi_{ie}^c}{\partial A_{i+1}} = -R_{cA}^{ie}$$

$$\frac{\partial \Phi_{ie}^c}{\partial Q_{i-1}} = 0 \quad \frac{\partial \Phi_{ie}^c}{\partial Q_i} = 0.5 + R_{cQ}^{ie} \quad \frac{\partial \Phi_{ie}^c}{\partial Q_{i+1}} = 0.5 - R_{cQ}^{ie}$$

$$\Phi_{iw}^c = Q(iw) = 0.5(Q_{i-1} + Q_i) - R_{cA}(A_i - A_{i-1}) - R_{cQ}(Q_i - Q_{i-1}) \tag{2.227}$$

$$\frac{\partial \Phi_{iw}^c}{\partial A_{i-1}} = R_{cA}^{iw} \qquad \frac{\partial \Phi_{iw}^c}{\partial A_i} = -R_{cA}^{iw} \qquad \frac{\partial \Phi_{iw}^c}{\partial A_{i+1}} = 0$$

$$\frac{\partial \Phi_{iw}^c}{\partial Q_{i-1}} = 0.5 + R_{cQ}^{iw} \quad \frac{\partial \Phi_{iw}^c}{\partial Q_i} = 0.5 - R_{cQ}^{iw} \quad \frac{\partial \Phi_{iw}^c}{\partial Q_{i+1}} = 0$$

Fluxes and derivatives of the fluxes of the momentum equation:

$$\Phi_{ie}^m = \beta \left.\frac{Q^2}{A}\right|_{ie} = 0.5 \left(\beta_i \frac{Q_i^2}{A_i} + \beta_{i+1} \frac{Q_{i+1}^2}{A_{i+1}} \right) - R_{mA}(A_{i+1} - A_i) - R_{mQ}(Q_{i+1} - Q_i) \tag{2.228}$$

$$\frac{\partial \Phi_{ie}^m}{\partial A_{i-1}} = 0 \quad \frac{\partial \Phi_{ie}^m}{\partial A_i} = -0.5\beta_i \frac{Q_i^2}{A_i^2} + R_{mA}^{ie} \quad \frac{\partial \Phi_{ie}^m}{\partial A_{i+1}} = -0.5\beta_{i+1} \frac{Q_{i+1}^2}{A_{i+1}^2} - R_{mA}^{ie}$$

$$\frac{\partial \Phi_{ie}^m}{\partial Q_{i-1}} = 0 \quad \frac{\partial \Phi_{ie}^m}{\partial Q_i} = \beta_i \frac{Q_i^2}{A_i^2} + R_{mQ}^{ie} \qquad \frac{\partial \Phi_{ie}^m}{\partial Q_{i+1}} = \beta_{i+1} \frac{Q_{i+1}^2}{A_{i+1}^2} - R_{mQ}^{ie}$$

$$\Phi_{iw}^m = \beta \left.\frac{Q^2}{A}\right|_{iw} = 0.5 \left(\beta_i \frac{Q_{i-1}^2}{A_i - 1} + \beta_i \frac{Q_i^2}{A_i} \right) - R_{mA}(A_i - A_{i-1}) - R_{mQ}(Q_i - Q_{i-1}) \tag{2.229}$$

$$\frac{\partial \Phi_{iw}^m}{\partial A_{i-1}} = -0.5\beta_{i-1} \frac{Q_{i-1}^2}{A_{i-1}^2} + R_{mA}^{iw} \quad \frac{\partial \Phi_{iw}^m}{\partial A_i} = -0.5\beta_i \frac{Q_i^2}{A_i^2} - R_{mA}^{iw} \quad \frac{\partial \Phi_{ie}^m}{\partial A_{i+1}} = 0$$

$$\frac{\partial \Phi_{iw}^m}{\partial Q_{i-1}} = \beta_{i-1} \frac{Q_{i-1}}{A_{i-1}} + R_{mQ}^{iw} \qquad \frac{\partial \Phi_{iw}^m}{\partial Q_i} = \beta_i \frac{Q_i}{A_i} - R_{mQ}^{iw} \qquad \frac{\partial \Phi_{iw}^m}{\partial Q_{i+1}} = 0$$

2.4.3.7.1 Derivatives of the Continuity Equation:

For a general cross section i:

$$\frac{\partial F_i}{\partial A_{red,i-1}} = -R_{cA}^{iw}$$

$$\frac{\partial F_i}{\partial A_{red,i}} = \frac{\Delta x_i}{\Delta t} \frac{\partial A_i}{\partial A_{red,i}} + R_{cA}^{ie} + R_{cA}^{iw}$$

$$\frac{\partial F_i}{\partial A_{red,i+1}} = -R_{cA}^{ie}$$

$$\frac{\partial F_i}{\partial Q_{i-1}} = -\frac{1}{2} - R_{cQ}^{iw}$$

$$\frac{\partial F_i}{\partial Q_i} = R_{cQ}^{ie} - R_{cQ}^{iw}$$

$$\frac{\partial F_i}{\partial Q_{i+1}} = \frac{1}{2} - R_{cQ}^{ie}$$

For the first cross section i=1:

$$\frac{\partial F_1}{\partial A_{red,1}} = \frac{\Delta x_1}{\Delta t}\frac{\partial A_1}{\partial A_{red,1}} + R_{cA}^{ie} - \frac{\partial Q_{in}}{\partial A_{red,1}}$$

$$\frac{\partial F_1}{\partial A_{red,2}} = -R_{cA}^{ie}$$

$$\frac{\partial F_1}{\partial Q_{1=in}} = R_{cQ}^{ie} - 0.5$$

$$\frac{\partial F_1}{\partial Q_2} = \frac{1}{2} - R_{cQ}^{ie}$$

For the last cross section i=n:

$$\frac{\partial F_n}{\partial A_{n-1}} = -R_{cA}^{iw}$$

$$\frac{\partial F_n}{\partial A_n} = \frac{\Delta x_n}{\Delta t}\frac{\partial A_n}{\partial A_{red,n}} + R_{cA}^{iw} + \frac{\partial Q_{out}}{\partial A_{red,n}}$$

$$\frac{\partial F_n}{\partial Q_{n-1}} = -\frac{1}{2} - R_{cA}^{iw}$$

$$\frac{\partial F_n}{\partial Q_n} = 0.5 + R_{cQ}^{iw}$$

2.4.3.7.2 Derivatives of the Momentum Equation

For a general cross section i:

$$\frac{\partial G_i}{\partial A_{red,i-1}} = \frac{1}{2}\beta_{i-1}U_{i-1}^2 - R_{mA}^{iw} + g\frac{A_{red,i}}{2}\frac{dz_{i-1}}{dA_{red,i-1}}$$

$$\frac{\partial G_i}{\partial A_{red,i}} = -\frac{1}{2}\beta_i U_i^2 + R_{mA}^{ie} + \frac{1}{2}\beta_i U_i^2 + R_{mA}^{iw} + \frac{g}{2}(z_{i+1} - z_{i-1}) + g\Delta x_i S_{fi}\left(1 - 2\frac{A_{red,i}}{K_i}\frac{dK_i}{dA_{red,i}}\right)$$

$$\frac{\partial G_i}{\partial A_{red,i+1}} = -\frac{1}{2}\beta_{i+1}U_{i+1}^2 - R_{mA}^{ie} + g\frac{A_{red,i}}{2}\frac{dz_{i+1}}{dA_{red,i+1}}$$

$$\frac{\partial G_i}{\partial Q_{i-1}} = \beta_{i-1}U_{i-1} - R_{mQ}^{iw}$$

$$\frac{\partial G_i}{\partial Q_i} = \frac{\Delta x_i}{\Delta t} + R_{mQ}^{ie} + R_{mQ}^{ie} + 2gA_{red,i}\Delta x_i\frac{Sf_i}{Q_i}$$

$$\frac{\partial G_i}{\partial Q_{i+1}} = \beta_{i+1}U_{i+1} - R_{mQ}^{ie}$$

For the first cross section i=1:

$$\frac{\partial G_1}{\partial A_{red,1}} = -\frac{1}{2}\beta_1 U_1^2 + R_{mA}^{1e} - \frac{\partial \Phi_{in}^m}{\partial A_{red,1}} + \frac{g}{2}(z_2 - z_1) + g\Delta x_1 S_{f1}\left(1 - 2\frac{A_{red,1}}{K_1}\frac{dK_1}{dA_{red,1}}\right)$$

$$\frac{\partial G_1}{\partial A_{red,2}} = -\frac{1}{2}\beta_2 U_2^2 + R_{mA}^{1e} + g\frac{A_{red,1}}{2}\frac{dz_2}{dA_{red,2}}$$

$$\frac{\partial G_1}{\partial Q_1} = \frac{\Delta x_1}{\Delta t} + \beta_1 U_1 + R_{mQ}^{1e} - \frac{\partial \Phi_{in}^m}{\partial Q_{1=in}} + 2gA_1\Delta x_1\frac{Sf_1}{Q_1}$$

$$\frac{\partial G_1}{\partial Q_2} = \beta_2 U_2 - R_{mQ}^{1e}$$

For the last cross section i=n

$$\frac{\partial G_n}{\partial A_{red,n-1}} = \frac{1}{2}\beta_{n-1}U_{n-1}^2 - R_{mA}^{nw} + g\frac{A_{red,n}}{2}\frac{dz_{n-1}}{dA_{red,n-1}}$$

$$\frac{\partial G_n}{\partial A_{red,n}} = -\frac{1}{2}\beta_n U_n^2 + R_{mA}^{ne} + \frac{1}{2}\beta_n U_n^2 + R_{mA}^{nw} + \frac{g}{2}(z_n - z_{n-1}) + g\Delta x_n S_{fn}\left(1 - 2\frac{A_{red,n}}{K_n}\frac{dK_n}{dA_{red,n}}\right)$$

$$\frac{\partial G_n}{\partial Q_{n-1}} = \beta_{n-1}U_{n-1} - R_{mQ}^{nw}$$

$$\frac{\partial G_n}{\partial Q_n} = \frac{\Delta x_n}{\Delta t} + \beta_n U_n + R_{mQ}^{ne} - \beta_n U_n + R_{mQ}^{nw} + 2gA_n\Delta x_n\frac{Sf_n}{Q_n} = \frac{\Delta x_n}{\Delta t} + R_{mQ}^{ne} + R_{mQ}^{ne} + 2gA_n\Delta x_n\frac{Sf_n}{Q_n}$$

2.4.3.8   Derivatives of the fluxes for an inner Weir

w is the weir width and p the Poleni factor

2.4.3.8.1   Derivatives for the continuity flux

$$\frac{\partial \Phi_{ie}^c}{\partial A_i} = wp\sqrt{2g(z_i - z_{weir})}\frac{dz_i}{dA_i} \quad \frac{\partial \Phi_{iw}^c}{\partial A_i} = 0$$

$$\frac{\partial \Phi_{ie}^c}{\partial A_{i-1}} = 0 \qquad\qquad\qquad \frac{\partial \Phi_{iw}^c}{\partial A_{i-1}} = wp\sqrt{2g(z_{i-1} - z_{weir})}\frac{dz_{i-1}}{dA_{i-1}}$$

$$\frac{\partial \Phi_{ie}^c}{\partial A_{i+1}} = 0 \qquad\qquad\qquad \frac{\partial \Phi_{iw}^c}{\partial A_{i+1}} = 0$$

$$\frac{\partial \Phi_{ie}^c}{\partial Q_{i-1}} = 0 \quad \frac{\partial \Phi_{iw}^c}{\partial Q_{i-1}} = 1$$

$$\frac{\partial \Phi_{ie}^c}{\partial Q_i} = 1 \quad \frac{\partial \Phi_{iw}^c}{\partial Q_i} = 0$$

$$\frac{\partial \Phi_{ie}^c}{\partial Q_{i+1}} = 0 \quad \frac{\partial \Phi_{iw}^c}{\partial Q_{i+1}} = 0$$

2.4.3.8.2   Derivatives of the momentum flux

$$\frac{\partial \Phi_{ie}^m}{\partial A_{i-1}} = 0 \qquad\qquad \frac{\partial \Phi_{iw}^m}{\partial A_{i-1}} = \frac{8gp^2w^2(z_{i-1} - z_{weir})^2}{3A_{weir}}\frac{dz_{i-1}}{dA_{i-1}}$$

$$\frac{\partial \Phi_{ie}^m}{\partial A_i} = \frac{8gp^2w^2(z_{i-1} - z_{weir})^2}{3A_{weir}}\frac{dz_i}{dA_i} \quad \frac{\partial \Phi_{iw}^m}{\partial A_i} = 0$$

$$\frac{\partial \Phi_{ie}^m}{\partial A_{i+1}} = 0 \qquad\qquad \frac{\partial \Phi_{iw}^m}{\partial A_{i+1}} = 0$$

$$\frac{\partial \Phi_{ie}^m}{\partial Q_{i-1}} = 0 \qquad\qquad \frac{\partial \Phi_{iw}^m}{\partial Q_{i-1}} = \frac{2Q_{i-1}}{A_{i-1}}$$

$$\frac{\partial \Phi_{ie}^m}{\partial Q_i} = \frac{2Q_i}{A_i} \qquad\qquad \frac{\partial \Phi_{iw}^m}{\partial Q_i} = 0$$

$$\frac{\partial \Phi_{ie}^m}{\partial Q_{i+1}} = 0 \qquad\qquad \frac{\partial \Phi_{iw}^m}{\partial Q_{i+1}} = 0$$

2.5   Numerical Solution of Sub-surface Flow

2.5.1   Introduction

The numerical solution of the Richard's equation is a challenging task due to strong non-linearities introduced by the constitutive models. Additionally, at the interfaces between different soils in heterogeneous embankments, steep jumps and abrupt changes in the variables may occur. Many models were presented in the past solving the Richard's equation based on Finite-Difference or Finite-Element methods and showed good results. A novel application of the Lattice-Boltzmann method on the Richard's equation was recently presented by Ginzburg et al. (2004) and Ginzburg (2006), basing on a LBM approach for generic anisotropic advection-dispersion equations. The application of the LBM has some advantages which can make it an interesting alternative choice compared to classical continuum approaches. The method is simple and easy to implement and it allows for

Figure 2.67 *Comparison of different levels of description and conceptual approaches applied to determine fluid motion.*

the modeling of complex geometries using bounce-back boundaries. Also, the method is local and therefore suited well for parallelization. Ginzburg adapted solution strategies for advection-diffusion problems to different formulations of the Richard's equation, like the moisture θ formulation and mixed moisture-pressure head $\theta - h$ formulation. This method of Ginzburg is applied and adapted here to simulate the 3-D sub-surface flow in the saturated and partially-saturated zone.

2.5.2   Lattice-Boltzmann Method

The LBM is a mesoscopic modelling approach which is positioned in between microscopic, particle-based dynamics and macroscopic continuum approaches. The underlying theory bases on the Boltzmann equation from kinetic theory which was derived by the Austrian physicist and philosopher L. Boltzmann.

The Boltzmann equation is formulated for a probability distribution function $f(\overrightarrow{r}, t)$ of particles in the 6-D phase-space $\overrightarrow{r}(\overrightarrow{x}, \overrightarrow{v})$. This phase-space is formed by the three spatial coordinates and the three velocity components. The distribution function may be seen as a representation of particles at time t with locations and velocities in between $\overrightarrow{r}$ and $\overrightarrow{r} + \Delta \overrightarrow{r}$. The integration of the distribution function over the phase-space results in the macroscopic fluid density. Since the LBM is used here to solve the macroscopic Richard's equation, the function $f(\overrightarrow{r}, t)$ may be interpreted as a directional saturation density, whereas the integration over the phase-space results in the macroscopic water saturation. The Boltzmann equation for the scalar distribution function $f(\overrightarrow{r}, t)$ can be written in 1D as (Mohamad, 2011)

$$\frac{\partial f(\overrightarrow{r}, t)}{dt} + v \frac{\partial f(\overrightarrow{r}, t)}{dx} = \Omega \tag{2.230}$$

with Ω as the collision operator, which describes the mutual influences of distribution functions $f(\overrightarrow{r}, t)$ on each other. This partial differential equation has the simple form of a single linear transport equation, even in higher dimensions. The main problem for solving the Boltzmann equation, however, is the treatment of its complex collision operator. The single relaxation time BGK (Bhatnagar et al., 1954) approach is often applied and treats the collision as simple relaxation of the distribution function f towards its equilibrium state, characterized by the local equilibrium distribution function f^{eq}. The collision operator then results to $\Omega = \omega(f - f^{eq})$ with the relaxation parameter ω.
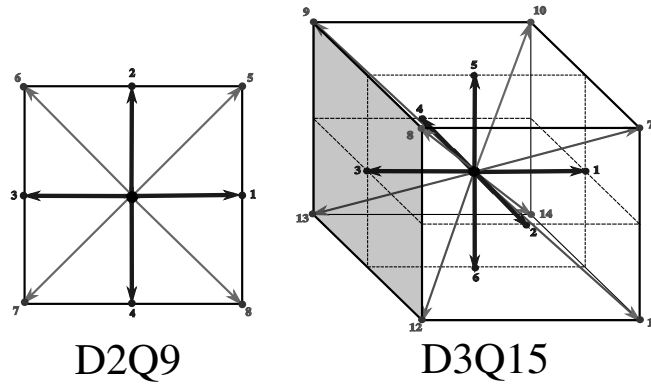
Figure 2.68 *2D lattice with 9 directions (left), 3D lattice with 15 directions (right).*

The LBM solves the Boltzmann equation in a discrete form on a uniform mesh. The discrete Boltzmann equation for the spatial mesh directions q reads

$$\underbrace{f_q(\overrightarrow{r} + \Delta t\,\overrightarrow{c}_q, t + \Delta t) = f_q(\overrightarrow{r}, t)}_{\text{advection step}} + \underbrace{\omega[f_q(\overrightarrow{r}, t) - f_q^{eq}(\overrightarrow{r}, t)]}_{\text{collision step}} + \underbrace{Q_q/c_m}_{\text{source}} \quad q = 1\ldots n_q$$

(2.231)

with a single time relaxation parameter ω being determined as a function of the diffusivity D as $\omega = -1.0/(D/c_s^2 + 0.5)$. The diffusivity is determined as

- $D = k_r(\theta)k_f \partial h/\partial \theta$ for the θ formulation, and
- $D = k_r(\theta)k_f$ for the mixed $\theta - h$ formulation.

The variable Q_q on the right hand side is an external source for modelling water infiltration into the embankment. The parameter c_s^2 is determined as c/ϑ with the free adjustable constant ϑ.

The uniform mesh is constructed with cubic cells using a set of q discrete velocities c_q which connect the grid cells with each other. Overall, $n_q =15$ different directions are used (3DQ15). The directions of the q discrete velocities (compare Figure 2.68) are set as

$$c_q = c_m \begin{cases} (0,0,0), (1,0,0), (0,1,0), (-1,0,0), (0,-1,0), (0,0,1), (0,0,-1) & q = 0\ldots 6 \\ (1,1,1), (-1,1,1), (-1,-1,1), (1,-1,1) & q = 7\ldots 10 \\ (1,1,-1), (-1,1,-1), (-1,-1,-1), (1,-1,-1) & q = 11\ldots 14 \end{cases}$$

(2.232)

2.5.3 Solution procedure

Mainly three explicit computational steps are applied to solve the discrete Boltzmann equation. The first two steps are hereby analogous to particle based approaches, whereas the third step reflects the handling with particle distribution functions instead of single particles.

2.5.3.1   Advection step

At the advection step, the distribution functions are just moved along the discrete lattice directions from each cell to its adjacent cells. At the lattice boundaries special treatments are required as described below.

2.5.3.2   Collision step

The collision operator is approximated using the BGK approach which assumes a simple relaxation of f towards its equilibrium distribution function f^{eq} , i.e. the solution approaches the equilibrium over time according to the single time relaxation parameter w.

The equilibrium distribution function hereby is the key element of the collision step where the main physics of the problem is included. The solution strategy in the LBM stays largely the same even for different physical problems, like e.g. fluid motion governed by the Navier-Stokes equation, whereas mainly the equilibrium distribution function has to be replaced. The equilibrium distribution function is here applied only in first order accuracy as provided in a general formulation by Ginzburg et al. (2004). It is outlined in the next section in detail.

2.5.3.3   Update of macroscopic variables

Using the relationships given above and the empirical constitutive model, the discrete Boltzmann equation is solved in each direction q by applying the propagation and collision steps mentioned above. The macroscopic variables of interest can finally be derived from the computed distribution functions f_q at the new time level.

The effective water saturation θ is simply obtained by summing up the distribution functions of all directions of a cell, which corresponds to an integration of f over phase-space. Afterwards, the pore-water pressure head h can be derived using the water retention curve. According to Ginzburg et al. (2004) one obtains:

$$\theta = \sum_{q=0}^{n} f_q$$

$h = f(\theta)$

$$\overrightarrow{v}_f = (\theta_s - \theta_r) \cdot \left[\sum_{q=0}^{n} \overrightarrow{c} f_q^{eq} + \begin{pmatrix} c_{x0} & \cdots & c_{xn} \\ c_{y0} & \cdots & c_{yn} \end{pmatrix} \cdot \begin{pmatrix} f_0^{eq} - f_0 \\ \vdots \\ f_n^{eq} - f_n \end{pmatrix} \right] \tag{2.233}$$

2.5.4   Equilibrium functions

The equilibrium distribution function for the θ formulation is given in first order accuracy in all mesh directions by

$$f_q^{eq} = \begin{cases} \left(1.0 - \dfrac{7}{3}c_s^2\right) \cdot \theta & q = 0 \\ t_q \theta c_s^2 & q = 1, 2, 3, 4 \\ t_q \cdot (\theta c_s^2 + \overrightarrow{I} \cdot \overrightarrow{c}) & q = 5, 7, 8, 9, 10 \\ t_q \cdot (\theta c_s^2 - \overrightarrow{I} \cdot \overrightarrow{c}) & q = 6, 11, 12, 13, 14 \end{cases} \tag{2.234}$$

where $c_s^2 = c/\vartheta$ with ϑ being a free constant. The method is local because the equilibrium function needs no information from neighbouring cells.

Accordingly, the equilibrium distribution function for the mixed $\theta - h$ formulation is obtained in all mesh directions as

$$f_q^{eq} = \begin{cases} \theta - 7/3c_s^2 \cdot h & q = 0 \\ t_q \cdot (hc_s^2) & q = 1, 2, 3, 4 \\ t_q \cdot (hc_s^2 + \overrightarrow{I} \cdot \overrightarrow{c}) & q = 5, 7, 8, 9, 10 \\ t_q \cdot (hc_s^2 - \overrightarrow{I} \cdot \overrightarrow{c}) & q = 6, 11, 12, 13, 14 \end{cases} \tag{2.235}$$

In contrast to the θ formulation, the mixed $\theta - h$ formulation is able to reproduce the continuous transition of the pressure head at the interface of different soils correctly. Therefore, the mixed $\theta - h$ formulation should be applied in cases of heterogeneous embankments with core and filter zones. The θ formulation, however, has advantageous stability conditions for imbibition problems (Ginzburg, 2006) and as such is recommended for use in case of homogeneous embankments. The weighting factors t_q for the lattice directions q can be derived for the chosen lattice configuration. The values for the 3DQ15 model are:

q	t_q (D3Q15)
1-6	1/3
7-14	1/24

To calculate the equilibrium functions given above, the advective, gravitational term $\overrightarrow{I}$, which acts in vertical downward direction, is needed and is evaluated as

$$\overrightarrow{I} = -k_r(\theta)k_s \overrightarrow{e}_z \tag{2.236}$$

2.5.5 Boundary and initial conditions

At the mesh boundaries the values of the distribution function f_q in the incoming directions are unknown and must be provided.

- For solid walls standard bounce-back boundaries are used. The unknown incoming distribution functions f_q thereby are set equal to the outgoing, anti-symmetric values to simulate wall reflection. Using this type of boundary condition allows incorporating even complex boundaries.

- A water column above the embankment is modelled using a pressure boundary, thereby presuming a hydrostatic pressure distribution. Equilibrium conditions are

      assumed at the boundary which allows for computing the values for the incoming directions ($f_q = f_q^{eq}$). For the mixed $\theta - h$ formulation, the water depth can be directly used as pore-water pressure head. For the θ formulation, the water saturation θ is needed instead and can be derived from the inverse water retention curve $\theta = f(h)$.

- The seepage flow out of the embankment is modelled with a combined approach. In the saturated zone ($q \geq 1.0$) a constant saturation of 1.0 is set at the boundary cells. In the unsaturated zone ($q < 1.0$) a bounce-back boundary is set.

The exact treatment of sloped or curved boundaries may become difficult, especially in 3D. Here, for simplicity, the sloped embankment faces are approximated using a series of steps using reflection angles of 0°, 45° or 90°. These simplifications can reduce the numerical accuracy in the vicinity of the embankment faces.

As initial conditions, the pore pressures or saturations in the domain can be given. The initial distribution functions are then set equal to the corresponding equilibrium values ($f_q = f_q^{eq}$) assuming equilibrium conditions.

3

References

Agoshkov, V.I., Quarteroni, A... and Saleri, F. (1994). Recent Developments in the Numerical-Simulation of Shallow-Water Equations. 1. Boundary-Conditions. *Applied Numerical Mathematics [Peer Reviewed Journal]*, 15(2): 175–200.

Ashida, K. and Michiue, M. (1971). An investigation over river bed degradation downstream of a dam. *Proceedings of the 14th congress of IAHR*, No. 3: 247–256. Paris, France.

Badrot-Nico, F., Brissaud, F. and Guinot, V. (2007). A finite volume upwind scheme for the solution of the linear advection-diffusion equation with sharp gradients in multiple dimensions. *Advances in Water Resources*, 30: 2002–2025.

Bechteler, W., Nujić, M. and Otto, A. (1993). Program Package "FLOODSIM" and its Application. *Proceedings of the international conference on hydroscience & engineering*, No. 1: 762–767. Washington DC, USA.

Beffa, C.J. (1994). Praktische Lösung der tiefengemittelten Flachwassergleichungen. *VAW-Mitteilung* 133, Versuchsanstalt für Wasserbau, Hydrologie und Glaziologie (VAW), ETH Zürich.

Begnudelli, L. and Sanders, B. (2006). Unstructured Grid Finite-Volume Algorithm or Shallow-Water Flow and Scalar Transport with Wetting and Drying. *Journal of Hydraulic Engineering*, 132(4): 371–384.

Belleudy, P. (2000). Modelling of deposition of sediment mixtures, part 1: analysis of a flume experiment. *Journal of Hydraulic Research, IAHR*, 38(6).

Bennett, J. and Nordin, C. (1977). Simulation of Sediment Transport and Armouring. (XXII, Ed.) *Hydrological Sciences Bulletin*, 22(4): 555–569.

Bertoldi, W., Siviglia, A., Tettamanti, S., Toffolon, M., Vetsch, D. and Francalanci, S. (2014). Modeling vegetation controls on fluvial morphological trajectories. *Geophysical Research Letters*, 41(20): 7167–7175.

Bezzola, G.R. (2002). Fliesswiederstand und Sohlenstabilität natürlicher Gerinne [PhD thesis]: Eidgenössische Technische Hochschule Zürich.

Bhatnagar, P., Gross, E. and Krook, M. (1954). A Model for Collision Processes in Gases. I. Small Amplitude Processes in Charged and Neutral One-Component Systems. *Physical review*, 94(3): 511–525.

Blanckaert, K. (2011). Hydrodynamic processes in sharp meander bends and their morphological implications. *Journal of Geophysical Research: Earth Surface*, 116(F1): F01003.

Bollrich, G. (2000). Technische Hydromechanik 1. *Beuth Verlag GmbH*, Berlin.

Borah, D.K., Alonso, C.V. and Prasad, S.N. (1982). Routing Graded Sediments in Streams - Formulations. *Journal of the Hydraulics Division-Asce*, 108: 1486–503.

Brooks, R.H. and Corey, A.T. (1964). Hydraulic Properties of Porous Media. *Hydrology papers* 3, Colorado State University; Colorado State University.

Brufau, P., Garcia-Navarro, P., Playán, E. and Zapata, N. (2002). Numerical Modeling of Basin Irrigation with an Upwind Scheme. *Journal of Irrigation and Drainage Engineering*, 128(4): 212–223.

Cao, Y., Williams, D. and Larsen, D.P. (2002). Comparison of Ecological Communities: The Problem of Sample Representativeness. *Ecological Monographs*, 72(1): 41–56.

Chanson, H. (1999). The Hydraulics of Open Channel Flow: An Introduction. *Edward Arnold*, London, UK.

Chaudhry, M.H. (1993). Open-Channel Flow. *Prentice Hall*, Englewood Cliffs, New Jersey.

Chen, Y. and Falconer, R.A. (1992). Advection-diffusion modelling using the modified QUICK scheme. *International Journal for Numerical Methods in Fluids*, 15: 1171–1196.

Chen, X., Ma, J. and Dey, S. (2010). Sediment transport on arbitrary slopes: Simplified model. *Journal of Hydraulic Engineering-ASCE*, 136(5): 311–317.

Cunge, J.A., Holly, F.M. and Verwey, A. (1980). Practical aspects of computational river hydraulics. *Pitman Advanced Pub*, Boston.

de Vries, M. (1966). Application of luminophores in sandtransport-studies. Meinema, TU Delft.

Delis, A., Skeels, C. and Ryrie, S. (2000). Evaluation of Some Approximate Riemann Solvers for transient Open Channel Flows. *Journal of Hydraulic Research.*, 38(3): 217–231.

Engelund, F. (1974). Flow and bed topography in channel bends. *Journal of the Hydraulics Division ASCE*, 100(11): 1631–1648.

Engelund, F. and Hansen, E. (1972). A monograph on sediment transport in alluvial streams. *Teknisk Forlag, Copenhagen*,

Faeh, R. (2007). Numerical Modeling of Breach Erosion of River Embankments. *Journal of Hydraulic Engineering*, 133(9): 1000–1009.

Fäh, R. (1997). Numerische Simulation der Strömung in offenen Gerinnen mit beweglicher Sohle. *VAW-Mitteilung* 153, Versuchsanstalt für Wasserbau, Hydrologie und Glaziologie(VAW), ETH Zürich.

Garcia-Navarro, P. and Vazquez-Cendon, M. (2000). On numerical treatment of the source terms in the shallow water equations. *Computers & Fluids*, 29(8): 951–979.

Ginzburg, I. (2006). Variably saturated flow described with the anisotropic Lattice Boltzmann methods. *Computers & Fluids*, 35: 831–848.

Ginzburg, I., Carlier, J. and Kao, C. (2004). Lattice Boltzmann approach to Richard's equation. *Proceedings of the CMWR XV, CT Miller*, 583–597. Chapel Hill, NC, USA.

Glaister, P. (1988). Approximate Riemann solutions of the shallow water equations. *Journal of Hydraulic Research*, 26: 293–306.

Godunov, S.K. (1959). A difference method for numerical calculation of discontinuous solutions of the equations of hydrodynamics. *Matematicheskii Sbornik*, 89(3): 271–306.

Harten, A., Lax, P. and van Leer, B. (1983). On Upstream Differencing and Godunov-Type Schemes for Hyperbolic Conservation Laws. *Siam Review*, 25(1): 35–61.

Hinton, E. and Owen, D.R.J. (1979). An introduction to finite element computations. *Pineridge Press Ltd*, Swansea, UK.

Holly, F.M. and Preissmann, A. (1977). Accurate calculation of transport in two dimensions. *Journal of Hydraulic Div. Am. Soc. Civ. Engineering*, 103(11): 1259–1277.

Hunziker, R.P. (1995). Fraktionsweiser Geschiebetransport [PhD thesis]: Eidgenössische Technische Hochschule Zürich.

Hunziker, R.P. and Jaeggi, M.N.R. (2002). Grain sorting processes. *Journal of Hydraulic Engineering-Asce*, 128(12): 1060–1068.

Ikeda, S. (1982). Lateral Bed-Load Transport on Side Slopes. *Journal of the Hydraulics Division-Asce*, 108(11): 1369–1373.

Jäggi, M. (1995). Vorlesung: Flussbau. ETH Abt. II, VIII, und XC. ETH Zürich.

Kassem, A.A. and Chaudhry, M.H. (1998). Comparison of Coupled and Semicoupled Numerical Models for Alluvial Channels. *Journal of Hydraulic Engineering-Asce*, 124(8): 794–802.

Knapp, F. (1960). Ausfluss, Überfall und Durchfluss im Wasserbau. *Verlag G. Braun*, Karlsruhe.

Komaei, S. and Bechteler, W. (2004). An improved, robust implicit solution for the two-dimensional shallow water equations on unstructured grids. *Proc., 2nd Int. Conf. on Fluvial Hydraulics*, Greco, M. ed., No. 2: 1065–1072. Balkema, Rotterdam, The Netherlands.

Lam, L., Fredlund, D.G. and Barbour, S.L. (1987). Transient seepage model for saturated-unsaturated soil systems: a geotechnical engineering approach. *Canadian Geotechnical Journal*, 24: 565–580.

Leonard, B.P. (1979). A Stable and Accurate Conservative Modelling Procedure Based on Quadratic Upstream Interpolation. *Comput. Methods Appl. Mech. Eng.*, 19(1): 59–98.

Li, S. and Millar, R. (2011). A two-dimensional morphodynamic model of gravel-bed river with floodplain vegetation. *Earth Surface Processes and Landforms*, 36(2): 190–202.

Lin, B. (1984). Current Study of Unsteady Transport of Sediment in China. *Proceedings of Japan-China Bilateral Seminar on River Hydraulics and Engineering Experiences, Proceedings of Japan-China Bilateral Seminar on River Hydraulics; Engineering Experiences, Tokyo-Kyoto –Sapporo.*, Tokyo-Kyoto-Sapporo.

Lu, G.Y. and Wong, D.W. (2008). An adaptive inverse-distance weighting spatial interpolation technique. *Computers & geosciences*, 34(9): 1044–1055.

Malcherek, A. (2001). Sedimenttransport und Morphodynamik. Vorlesungsskript der Universität der Bundeswehr München, München.

Marti, C. (2006). Morphologie von verzweigten Gerinnen: Ansätze zur Abfluss-, Geschiebetransport und Kolktiefenberechnung [PhD thesis]: ETH Zürich, Versuchsanstalt für Wasserbau, Hydrologie und Glaziologie -VAW-.

Meyer-Peter, E. and Müller, R. (1948). Formulas for Bed-Load Transport, 2nd Meeting IAHR, Stockholm, Sweden.

Minh Duc, B. (1998). Berechnung der Strömung und des Sedimenttransports in Flüssen mit einem tiefengemittelten numerischen Verfahren [PhD thesis]: Karlsruhe University, Germany.

Mohamad, A.A. (2011). Lattice Boltzmann Method – Fundamentals and Engineering Applications with Computer codes. *Springer-Verlag*, London.

Mohamadian, A., Le Roux, D.Y., Tajrishi, M. and Mazaheri, K. (2005). A mass conservative scheme for simulating shallow flows over variable topographies using unstructured grid. *Advances in Water Resources*, 28(5): 523–539.

Mualem, Y. (1976). A New Model for Predicting the Hydraulic Conductivity of Unsaturated Porous Media. *Water Resources Research*, 12(3): 513–552.

Nujić, M. (1998). Praktischer Einsatz eines hochgenauen Verfahrens für die Berechnung von tiefengemittelten Strömungen. Institut für Wasserwesen. Universität der Bundeswehr München. Institut für Wasserwesen. Universität der Bundeswehr München.

Osher, S. and Solomon, F. (1982). Upwind Difference-Schemes for Hyperbolic Systems of Conservation-Laws. *Mathematics of Computation*, 38: 339–374.

Parker, A.J., G.; Sutherland (1990). Fluvial armor. *Journal of Hydraulic Reasearch*,

Parker, G. (1990). Surface-based bedload transport relation for gravel rivers. *Journal of Hydraulic Reasearch*, 28(4): 417–436.

Parker, G. (2008). Transport of Gravel and Sediment Mixtures. *Sedimentation engineering: Processes, measurements, modeling and practice*, No. 110: 165–252. *ASCE*, Virginia.

Parker, G., Klingeman, P.C. and McLean, D.G. (1982). Bedload and Size Distribution in Paved Gravel-Bed Streams. *Journal of Hydraulic Division, ASCE*, 108(4): 544–571.

Rickenmann, D. (1990). Bedload transport capacity of slurry flows at steep slopes [PhD thesis]: Versuchsanstalt für Wasserbau, Hydrologie und Glaziologie (VAW), ETH Zürich.

Rickenmann, D. (1991). Hyperconcentrated Flow and Sediment Transport at Steep Slopes. *Journal of Hydraulic Engineering*, 117(11): 1419–1439.

Roe, P.L. (1981). Approximate Riemann Solvers, Parameter Vectors and Difference Schemes. *Journal of computational physics*, 43: 357–372.

Rozovskii, I.L. (1957). Flow of Water in Bends of Open Channels. *Academy of Science of the Ukrainian S.S.R, Institute of Hydrology; Hydraulic Engineering*,

Shields, A. (1936). Anwendungen der Ähnlichkeitsmechanik und der Turbulenzforschung auf die Geschiebebewegungen. Mitteilung der Preussischen Versuchsanstalt für Wasserbau und Schiffbau. Berlin, Deutschland.

Smart, G. and Jaeggi, M. (1983). Sediment Transport on Steep Slopes. *VAW-Mitteilung* 64, Versuchsanstalt für Wasserbau,Hydrologie und Glaziologie (VAW). Zürich, ETH Zürich.

Soares-Frazão, S., Le Grelle, N., Spinewine, B. and Zech, Y. (2007). Dam-break induced morphological changes in a channel with uniform sediments: measurements by a laser-sheet imaging technique. *Journal of Hydraulic Research*, 45: 87–95.

Sternberg, H. (1875). Untersuchungen über Längen- und Querprofil geschiebeführender Flüsse. (p. 4.-5. Zeitschrift für Bauwesen 25, Ed.) *Zeitschrift für Bauwesen, Wien*, 25: 483–506.

Sun, Z. and Donahue, J. (2000). Statistically Derived Bedload Formula for any Fraction of Nonuniform Sediment. *Journal of Hydraulic Engineering-Asce*, 126(2): 105–111.

Swartenbroekx, C., Soares-Frazão, S., Staquet, R. and Zech, Y. (2010). Two-dimensional operator for bank failures induced by water-level rise in dam-break flows. *Journal of Hydraulic Research*, 48(3): 302–314.

Talmon, A., Struiksma, N. and van Mierlo, M. (1995). Laboratory measurements of the direction of sediment transport on transverse alluvial-bed slopes. *Journal of Hydraulic Research*, 33(4): 495–517.

Toro, S., E. F. (1994). Restoration of the contact surface in the HLL-Riemann solver. *Shock Waves*, 4: 25–34.

Toro, E.F. (1997). Riemann Solvers and Numerical Methods for Fluid Dynamics. *Springer-Verlag*, Berlin.

Toro, E.F. (2001). Shock-Capturing Methods for Free-Surface Shallow Flows. *John Wiley*, Chichester, New York.

Valiani, A., Caleffi, V. and Zanni, A. (2002). Case Study: Malpasset Dam-Break Simulation using a Two-Dimensional Finite Volume Method. *Journal of Hydraulic Engineering*, 128(5): 460–472.

van Genuchten, M.T. (1980). A Closed-form Equation for Predicting the Hydraulic Conductivity of Unsaturated Soils. *Soil Science Society of America Journal*, 44(5): 892–898.

van Leer, B. (1982). Flux-Vector Splitting for the Euler Equations. *In Proceedings of the 8th International Conference on Numerical Methods in Fluid Dynamics*, No. 170: 507–512.

van Rijn, L.C. (1989). Handbook Sediment Transport by Current and Waves. *Delft Hydraulics Laboratory*, Delft, The Netherlands.

van Rijn, L.C. (1984a). Sediment Transport, Part I: Bed Load Transport. *Journal of Hydraulic Engineering, ASCE*, 110(10): 1431–1456.

van Rijn, L.C. (1984b). Sediment Transport, Part II: Suspended Load Transport. *Journal of Hydraulic Engineering, ASCE*, 110(11): 1613–1641.

Voigt, H. (1971). Abflussberechnung gleichzeitig über- und unterströmter Stauelemente. *Dissertation* TU Dresden.

Westrich, B. and Juraschek, M. (1985). Flow transport capacity for suspended sediment. *XXIth IAHR Congress*, No. 3: 590–594. Melbourne, Australia.

Wilcock, P.T.C.J.C. (2003). Surface-based transport model for mixed-size sediment. *Journal of Hydraulic Engineering*,

Wong, M. and Parker, G. (2006). Reanalysis and correction of bed-load relation of meyer-peter and mã¼ller using their own database. *Journal of Hydraulic Engineering*, 132(11): 1159–1168.

Wu, W. (2007). Computational River Dynamics. *Taylor & Francis Ltd*, London, UK.

Wu, W., Wang, S.S.Y. and Jia, Y. (2000). Non-Uniform Sediment Transport in Alluvial Rivers. *Journal of Hydraulic Research*, 38(6): 427–434.

Xu, Y. (1998). Numerical Modeling of Suspended Sediment Transport in Rivers [PhD thesis]: Mitteilung 98, Institut für Wasserbau, Universität Stuttgart.

Yalin, M. and Silva, A. da (2001). Fluvial processes. *International Association of Hydraulic Engineering; Research (IAHR), Delft, The Netherlands*,

Zen, S., Zolezzi, G., Toffolon, M. and Gurnell, A.M. (2016). Biomorphodynamic modelling of inner bank advance in migrating meander bends. *Advances in water resources*, 93: 166–181.

Zhang, R. (1961). River Dynamics. *Industry Press*, Beijing, China.

Zyserman, J. and Fredsøe, J. (1994). Data Analysis of Bed Concentration of Suspended Sediment. *Journal of Hydraulic Engineering, ASCE*, 120(9): 1021–1042.

TUTORIALS

of BASEMENT

Contents

1

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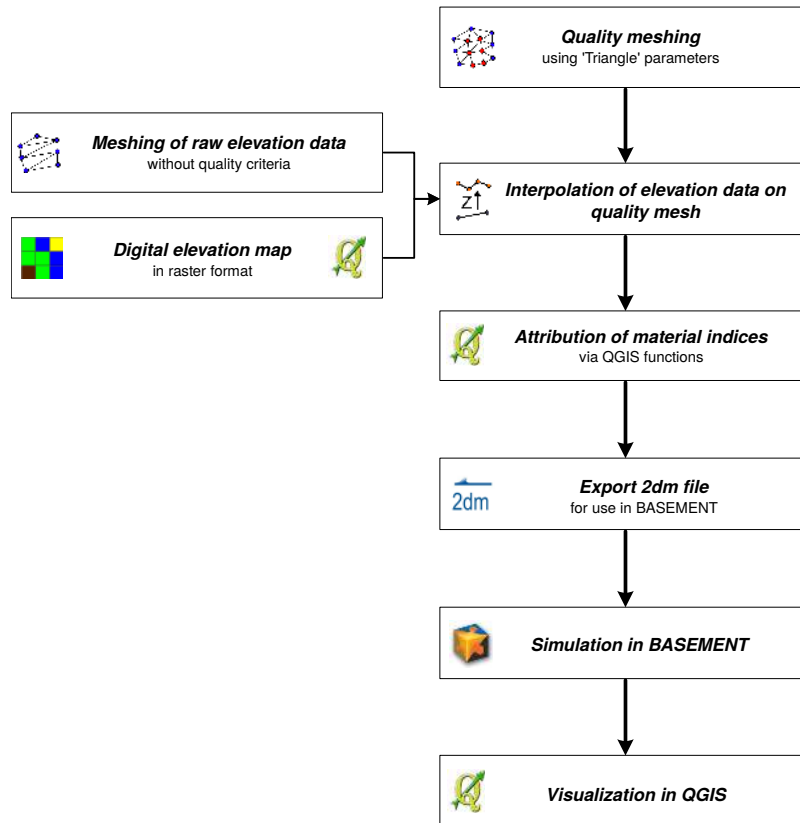
Figure 1.1 *General workflow for the creation of a computational mesh with BASEmesh.*

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.
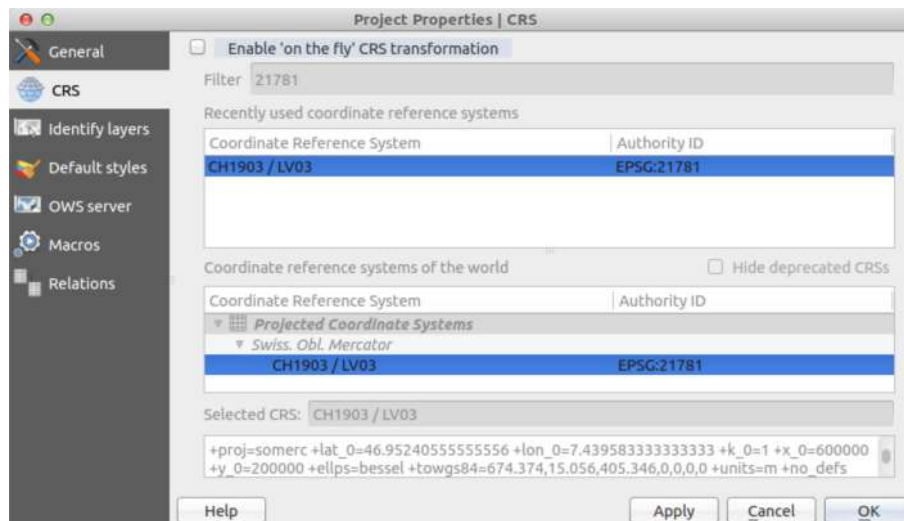
Figure 1.2 *Project Properties, CRS*

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 refered 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*.
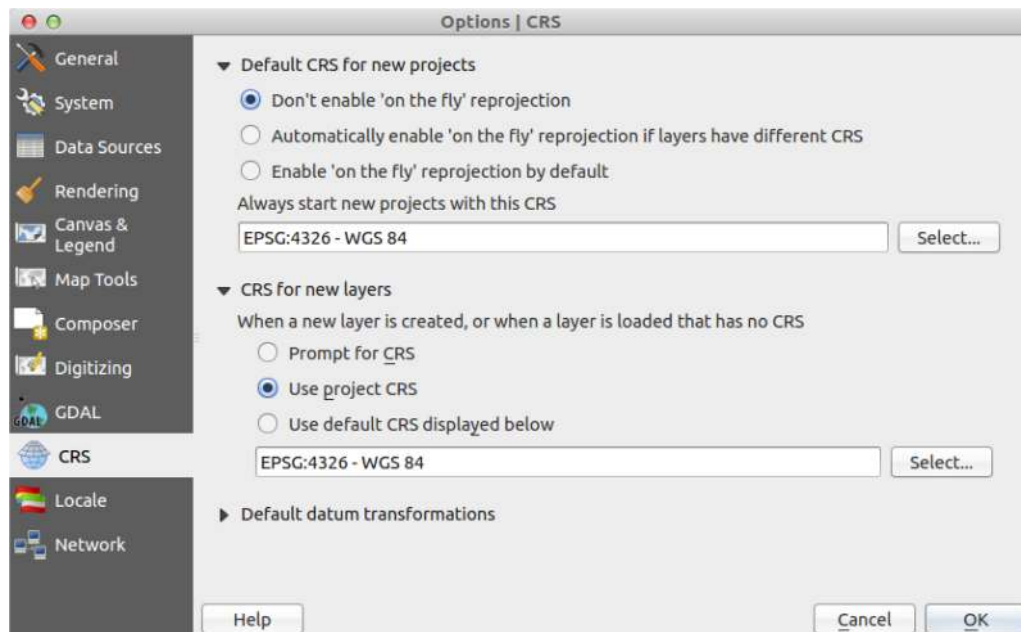
Figure 1.3 *Settings, CRS*

(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).
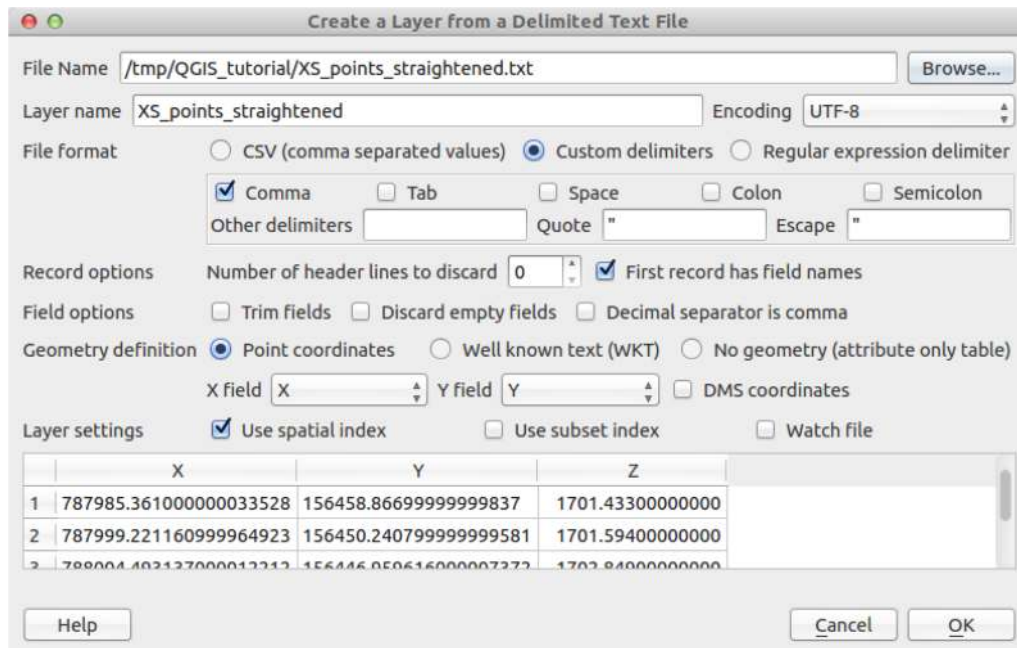
Figure 1.4 *Add delimeted text layer*

Figure 1.5 *Save vector layer*
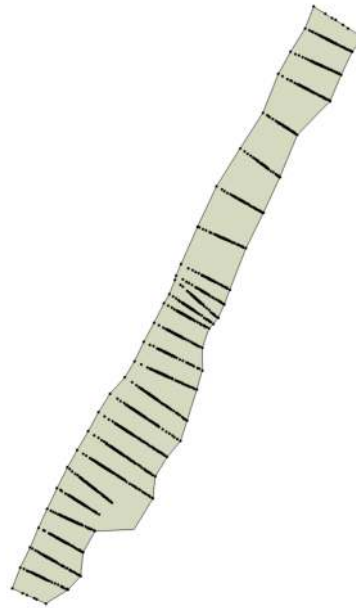
Figure 1.6 *Model Boundary*

(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.

(2) Zoom into the area of interest. You should see small red 'x' for each polygon vertex.
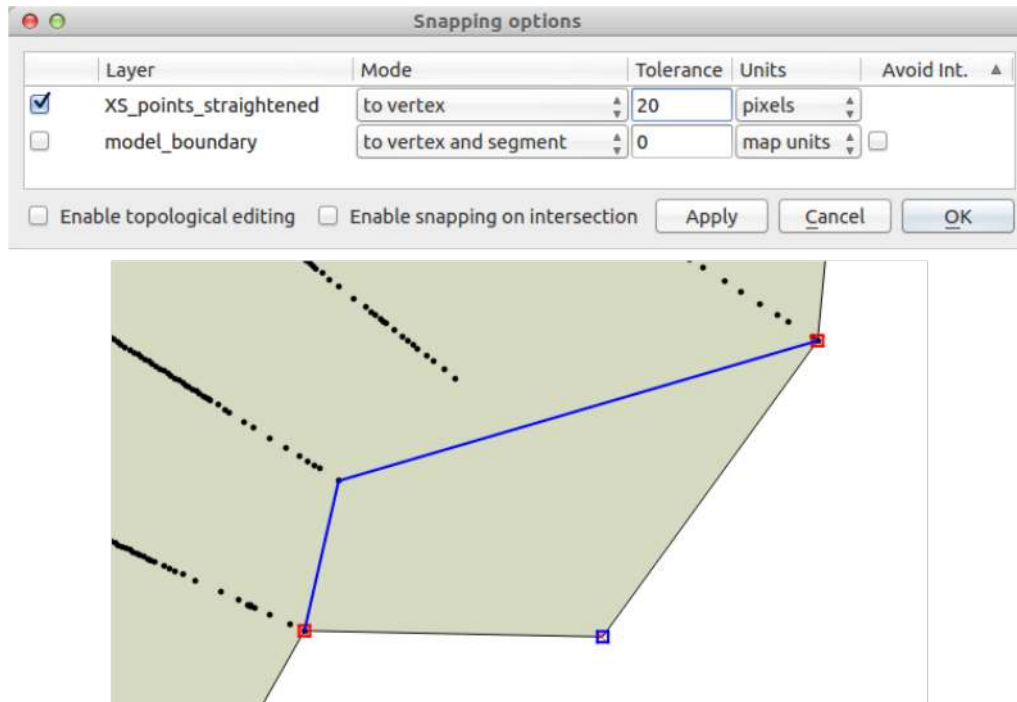
Figure 1.7 *Snapping Options*

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 ist 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'.
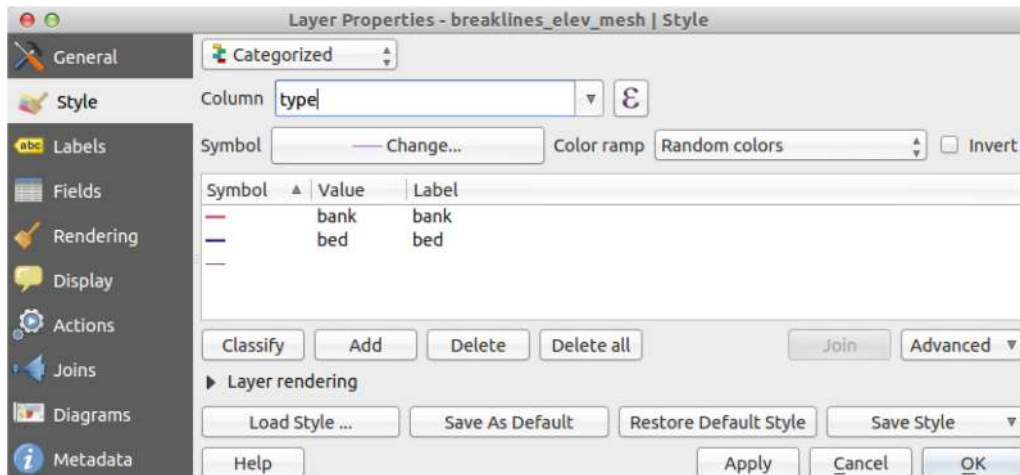
Figure 1.8 _Layer properties_

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
    ![toolbar icon] in the toolbar.

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 it's 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.

(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.
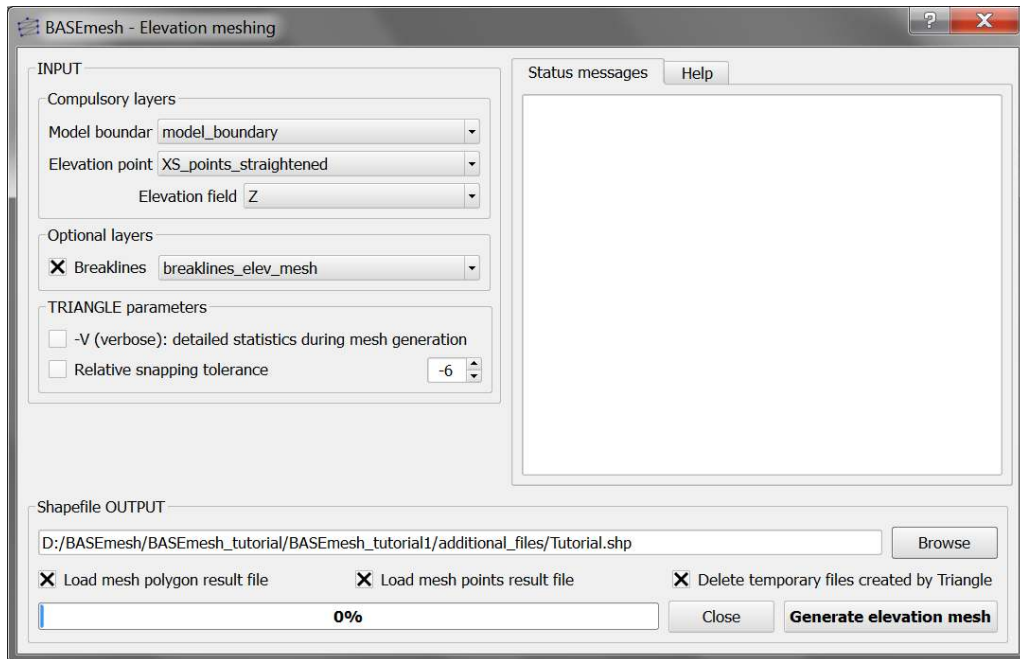
Figure 1.9 *BASEmesh Elevation Meshing*

1.2.9   Adaption of the Breaklines for Quality Meshing

For most tasks, quality meshing requires the same basic breaklines as elevation meshing does. Neverthless, 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).

(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.*
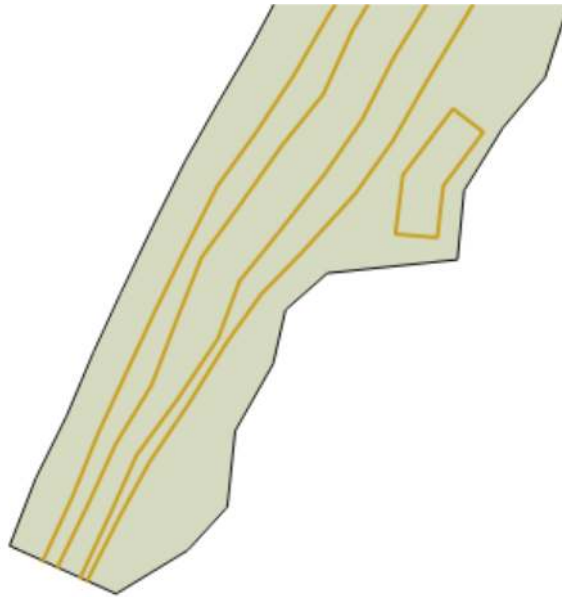
Figure 1.10 *Breaklines Quality Meshing*

(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 atrributes 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 acount 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.

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).
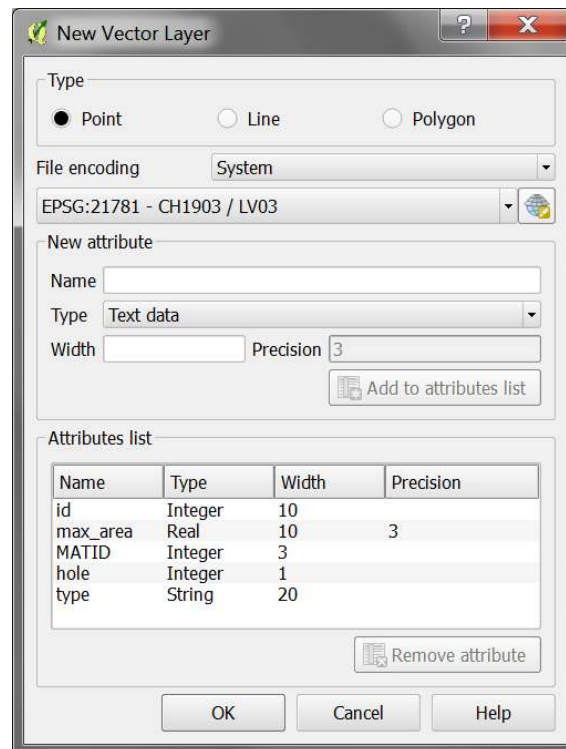
Figure 1.11 *New point layer 'region_points'*

(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

(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.
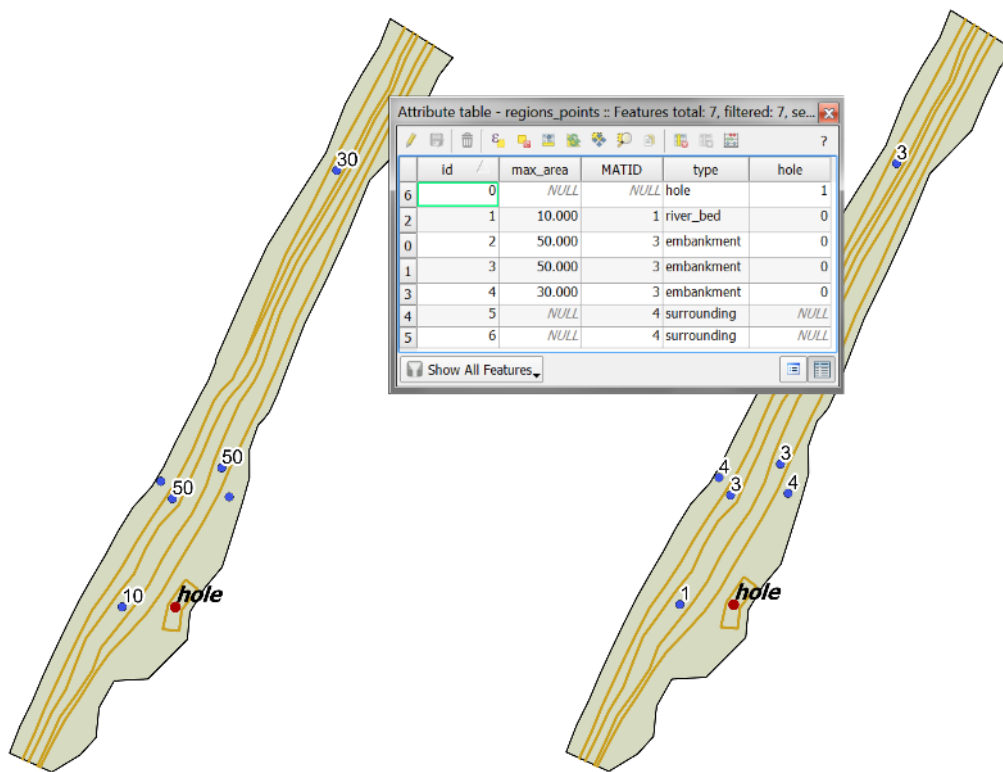
Figure 1.12 *Region marker points*

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 rights side status messages as well as Triangle's output messages are displayed during meshing. Tool-specific help can be found in the *Help*-tab.

(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.
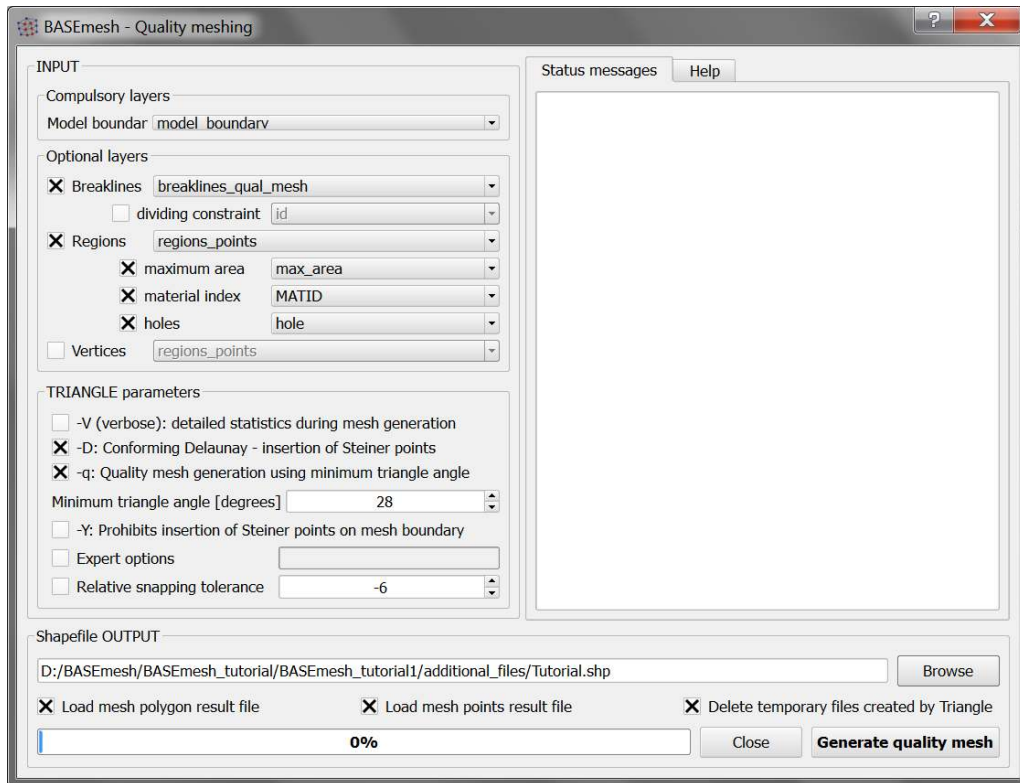
Figure 1.13 *BASEmesh Quality Meshing*

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 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).
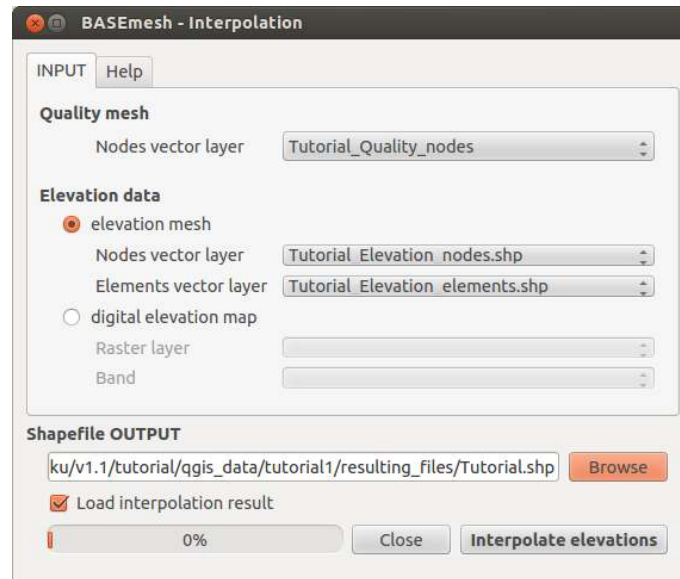
Figure 1.14 *BASEmesh Interpolation*

(1) Go to *Plugins → BASEmesh → Interplolation*, 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 (>10'000), 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.

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
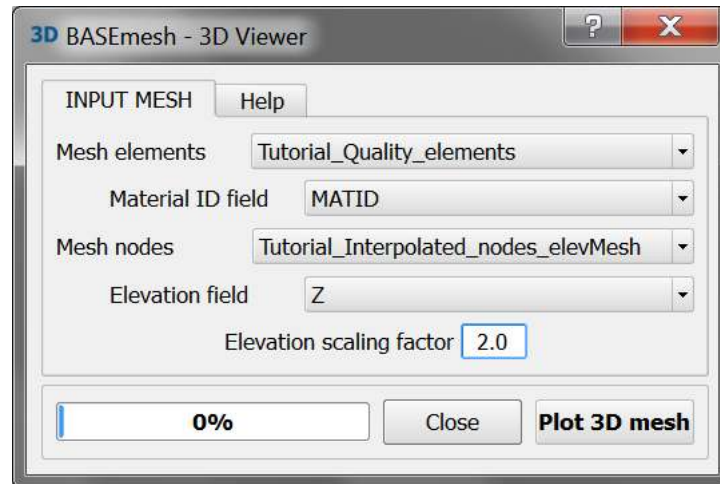
Figure 1.15 *BASEmesh 3D view*

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 colorized 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*).

(5) As input for the *Nodes* select the node layer with the interpolated elevation data 'Tutorial_interpolated_nodes_elevMesh.shp'.
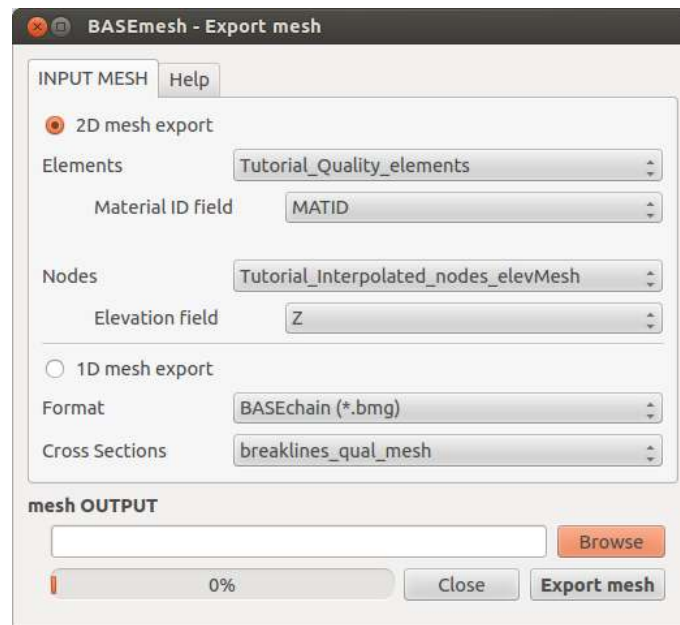
Figure 1.16 *BASEmesh Export Mesh*

(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.

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
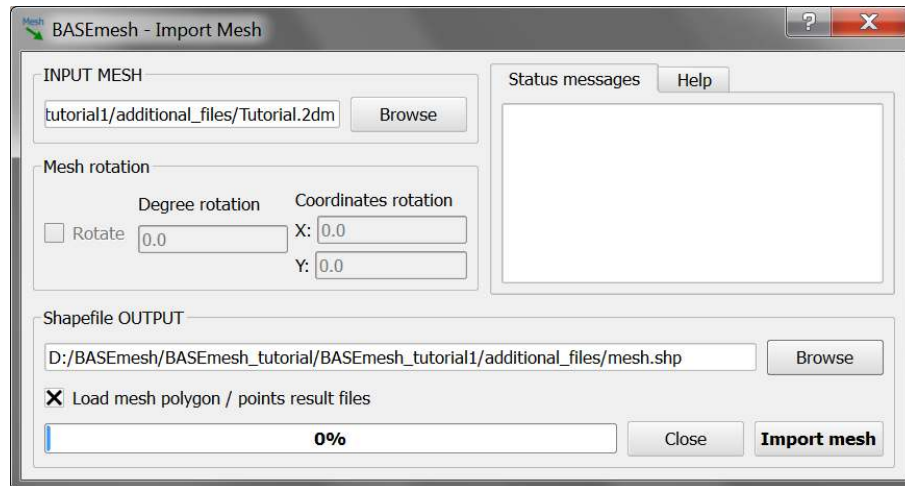
Figure 1.17 *BASEmesh Import Mesh*

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 [Mesh icon] 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 querys.

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

(1) Load the QGIS plugin manager by choosing *Manage and Install Plugins...* in the menue *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 sucessfully, 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 attribut 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.

(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.
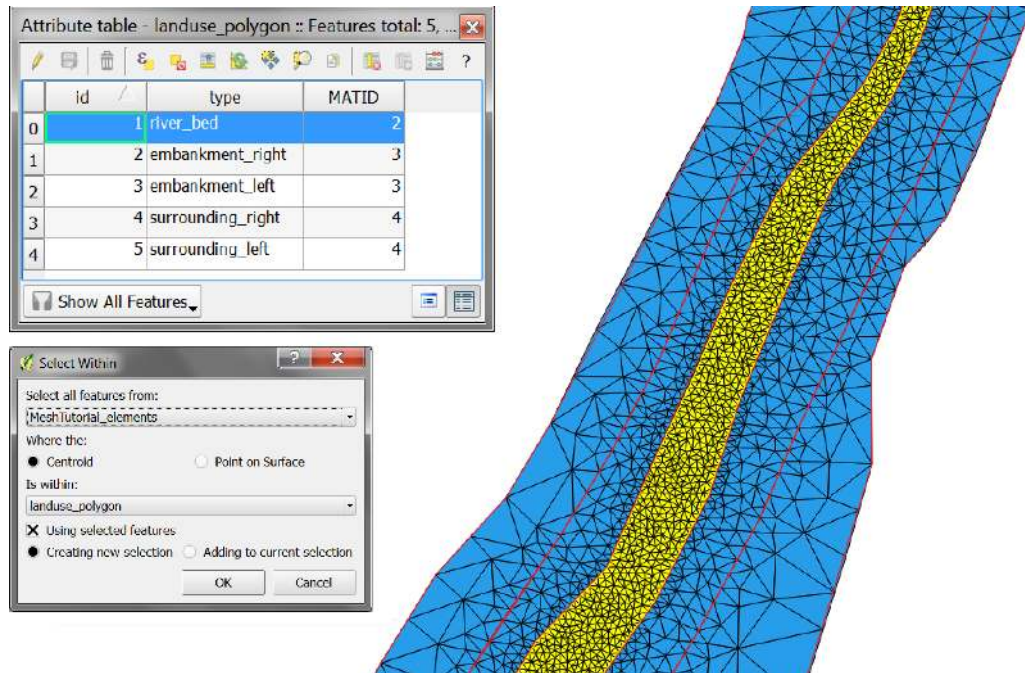
Figure 1.18 *Selected mesh elements according to their location*

(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 demonstated 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 everithing went well, your QGIS canvas should now look like in Figure 1.20.
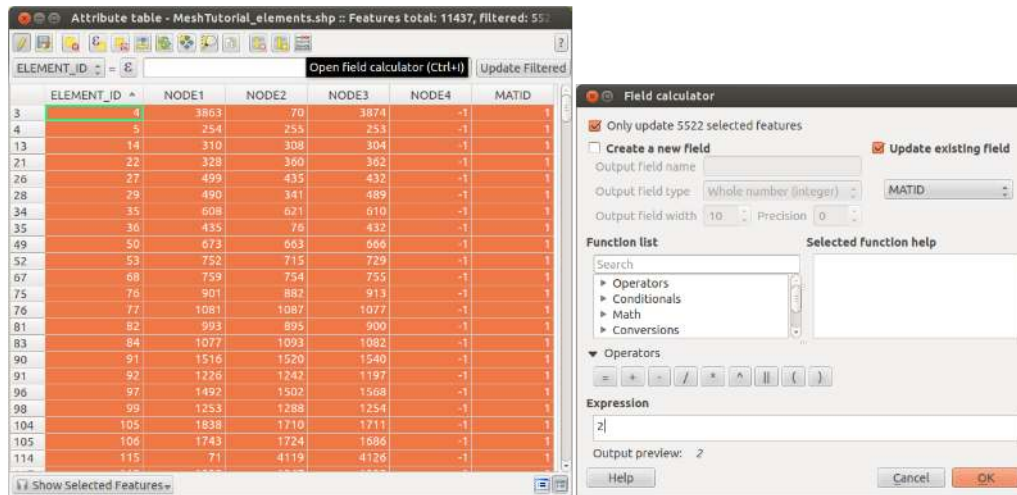
Figure 1.19 *Editing of selected mesh elements*
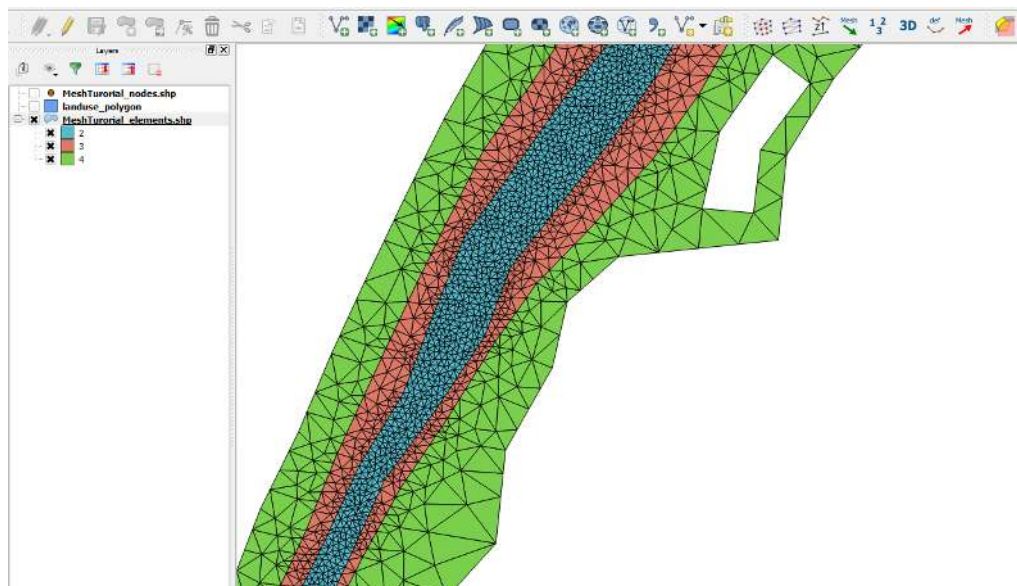
Figure 1.20 *Mesh material indices: river bed=2, embankment=3, surrounding=4*

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* ELEMENT*ID* 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 anologous 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 specofied.

Thumb rules for mesh editing:
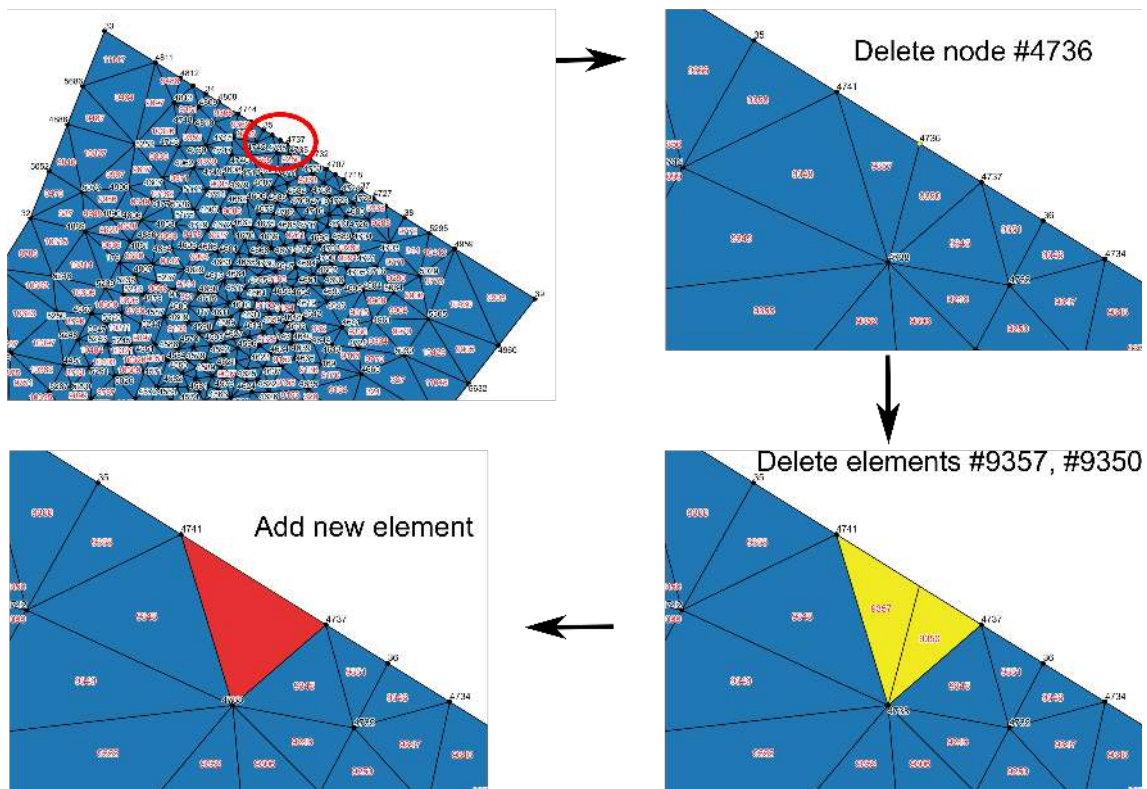
Figure 1.21 *Modify mesh elements*

- Element vertices and nodes must be at exactly the some 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 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.*
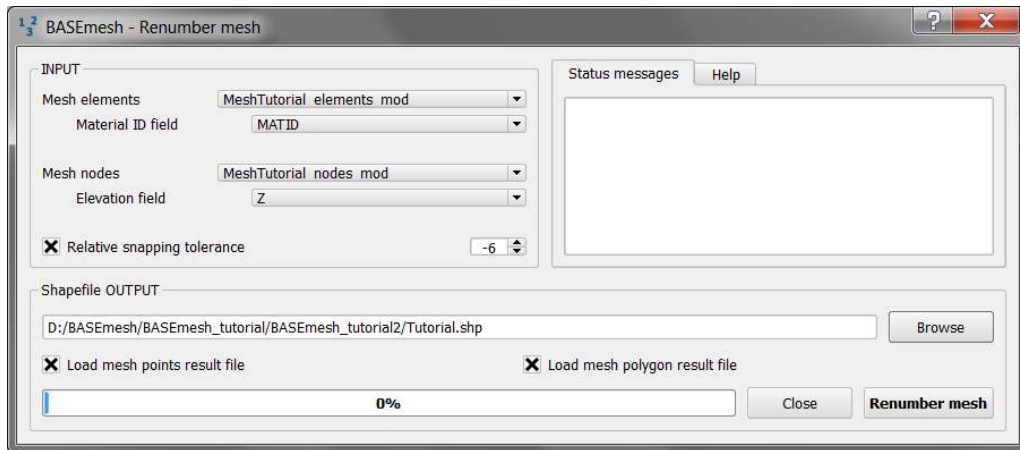
Figure 1.22 *Renumbering the mesh*

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
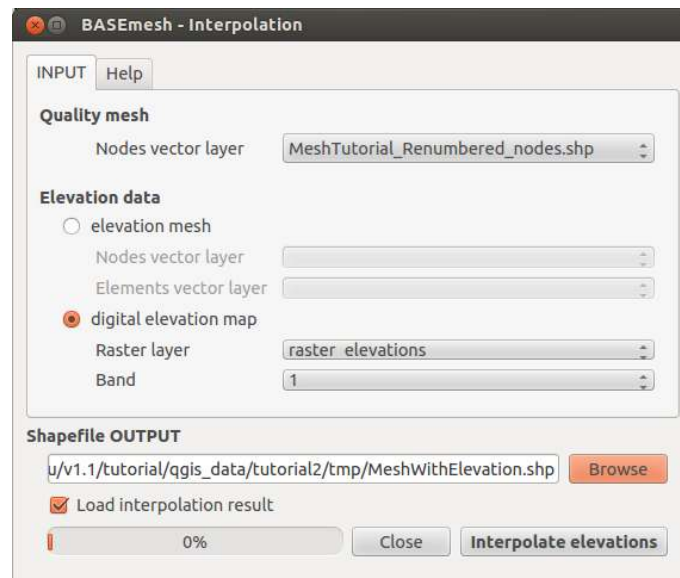
Figure 1.23 *Interpolation from raster data*

dividing costraints along cross sections in order to determine the number of mesh elements along a certain breakline during quality meshing. This is espescielly relevant for the use of Inner Boundary Conditions in BASEMENT, where an equal number of elments 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'
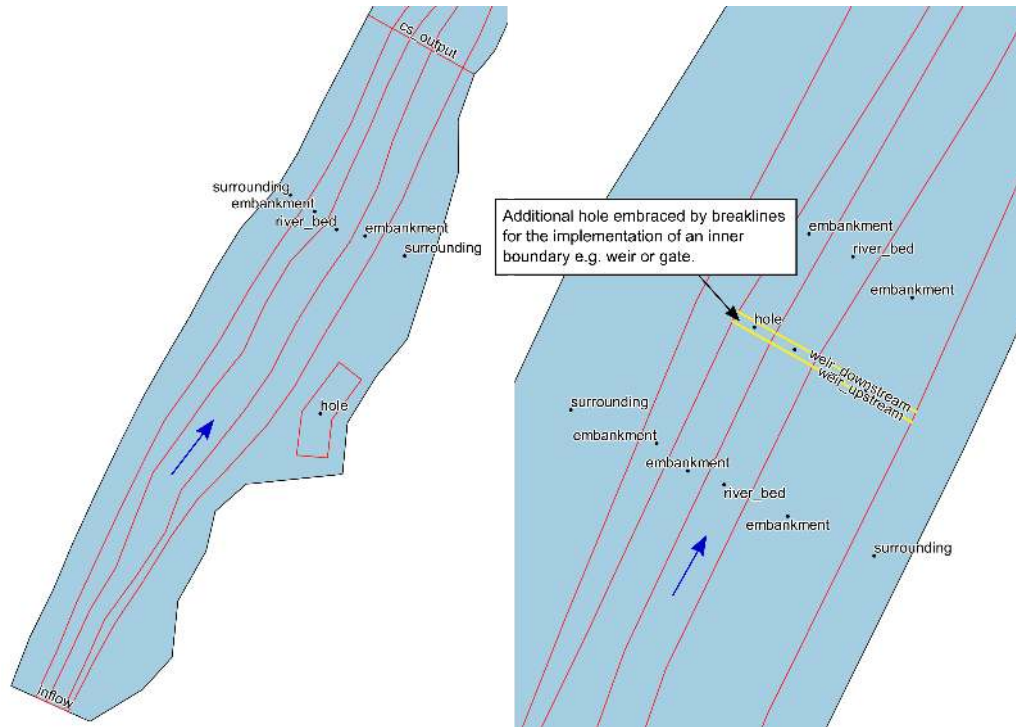
Figure 1.24 *Additional breaklines inflow boundary, cross section output and inner boundary*

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 demonstated in Tutorial 1 (Section 1.2.11), check the *dividing constraint* and select the attribute 'nCells' to determine the number of elements.
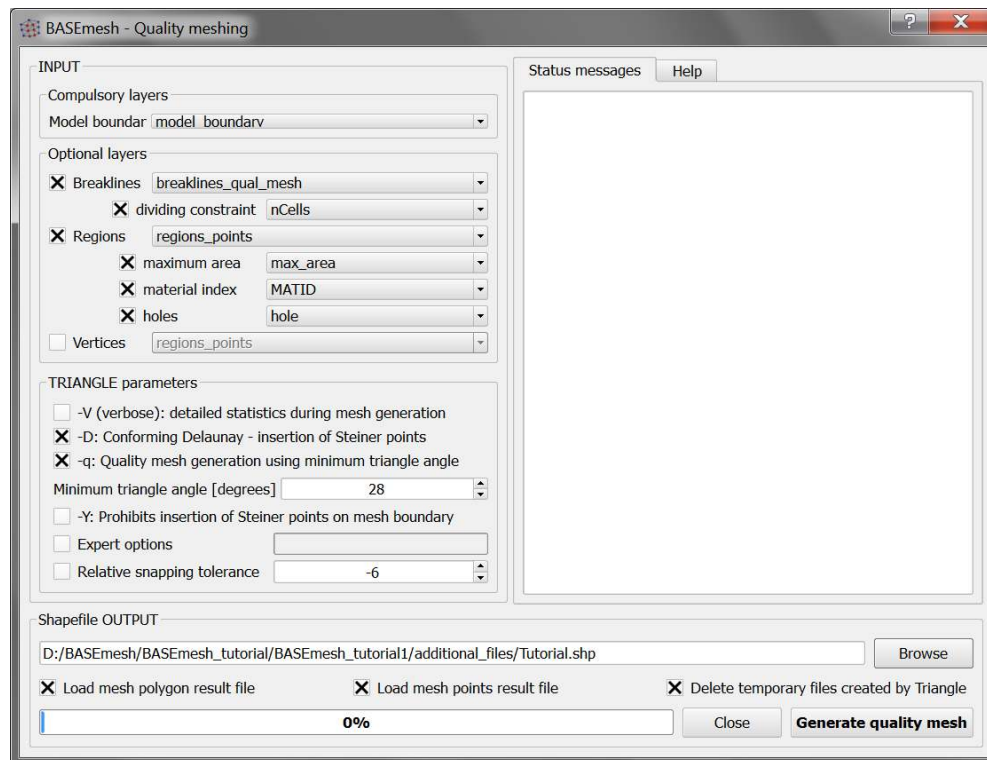
Figure 1.25 *BASEmesh Quality Meshing using dividing constraints*

(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, to 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 extrat the information from the nodes layer, which is generated when importing the mesh. The examplary usage of this tool is shown in the following steps:
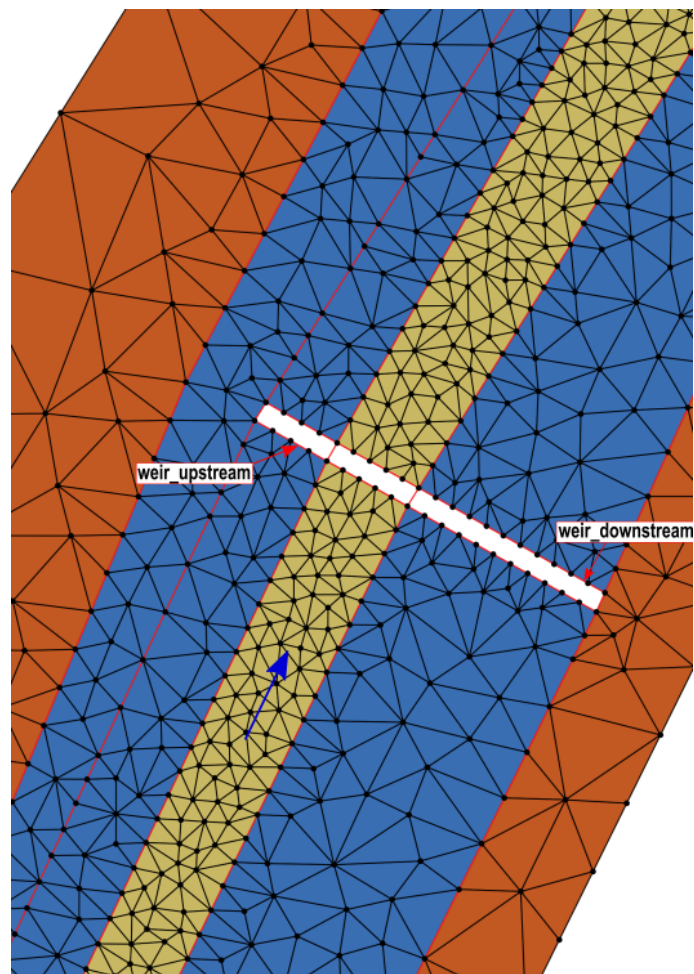
Figure 1.26 *Quality Mesh with dividing constraints along breakline*
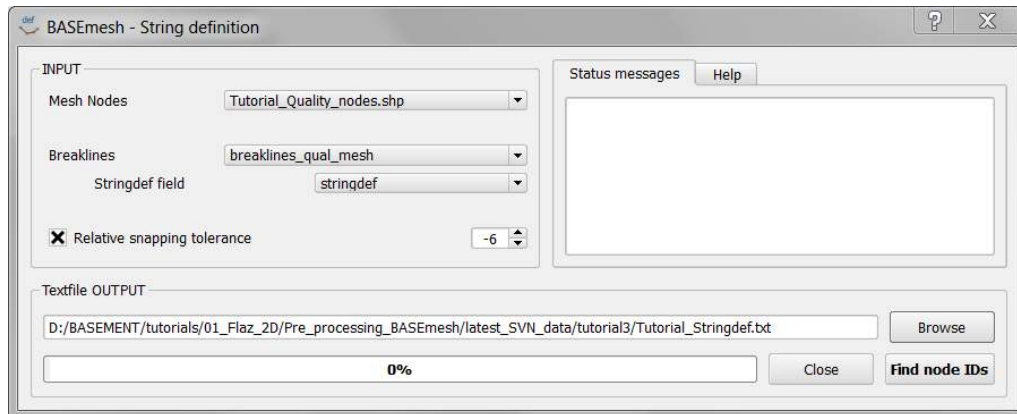
Figure 1.27 *BASEmesh extract Stringdef*

(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).
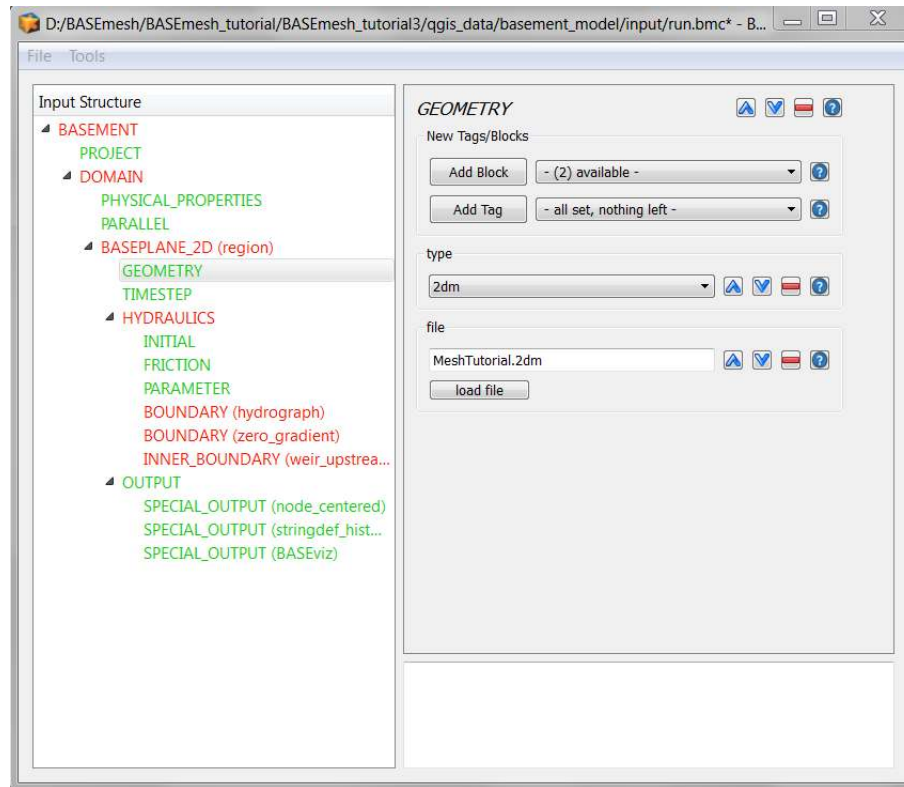
Figure 1.28 *BASEMENT command, GEOMETRY*

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.
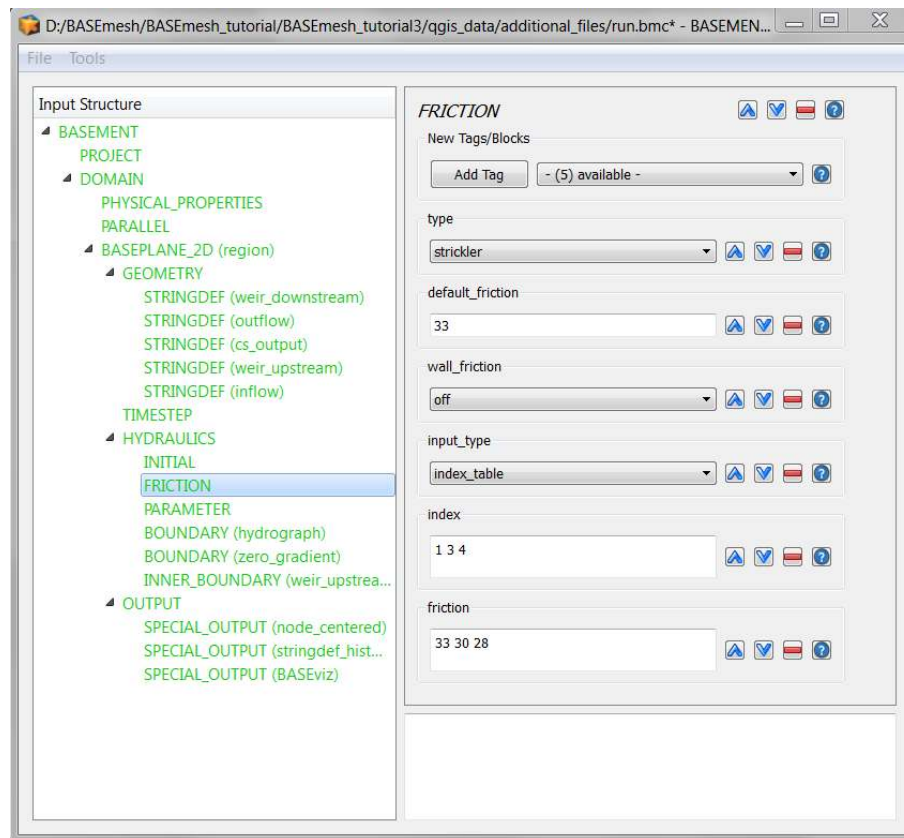
Figure 1.29 *BASEMENT command, FRICTION*

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!).
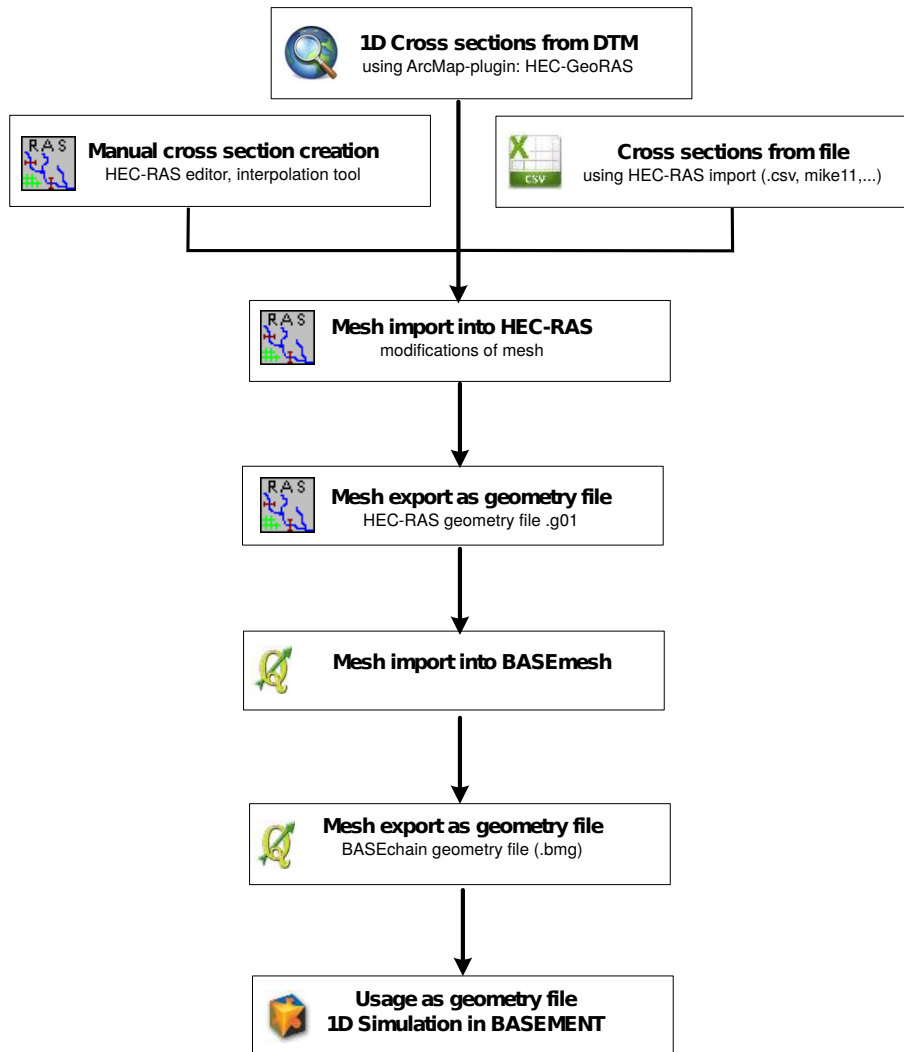
Figure 1.30 *Workflow HEC RAS*

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 strutures). 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.

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.
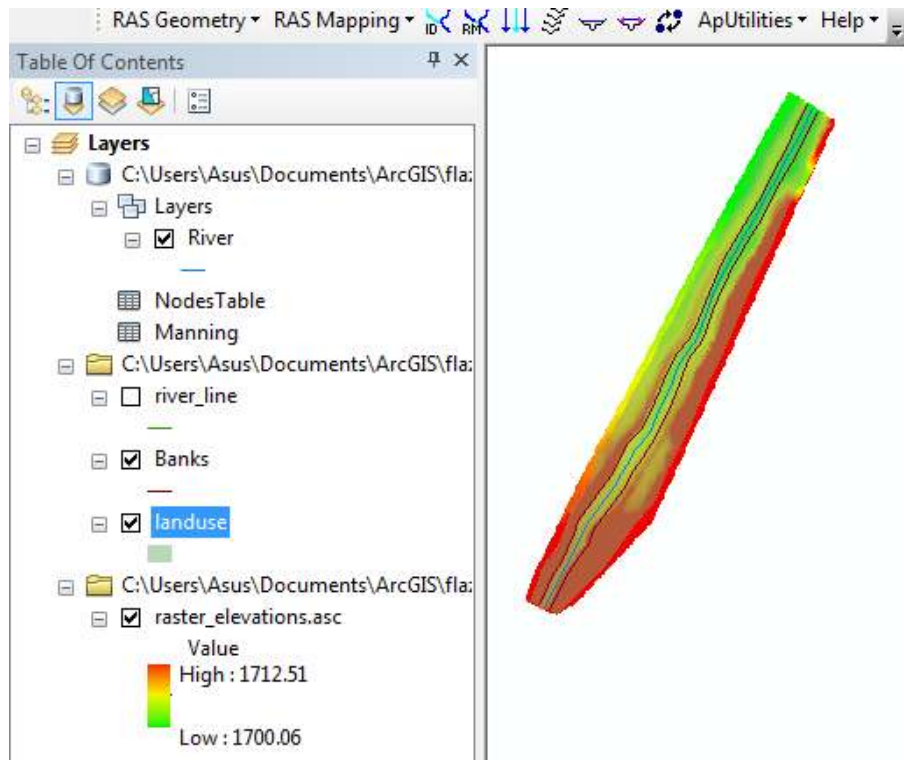
Figure 1.31 *ArcMap*

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 adaptions.

(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.
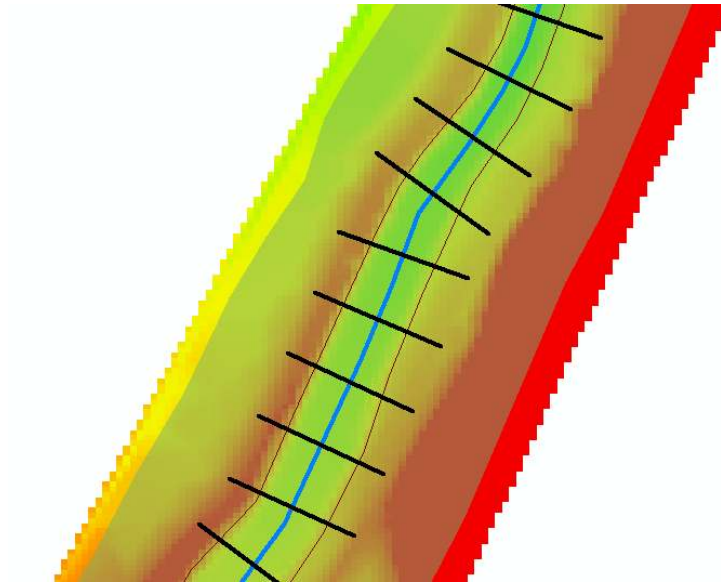
Figure 1.32 *XS Cutlines*

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 additon, 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.
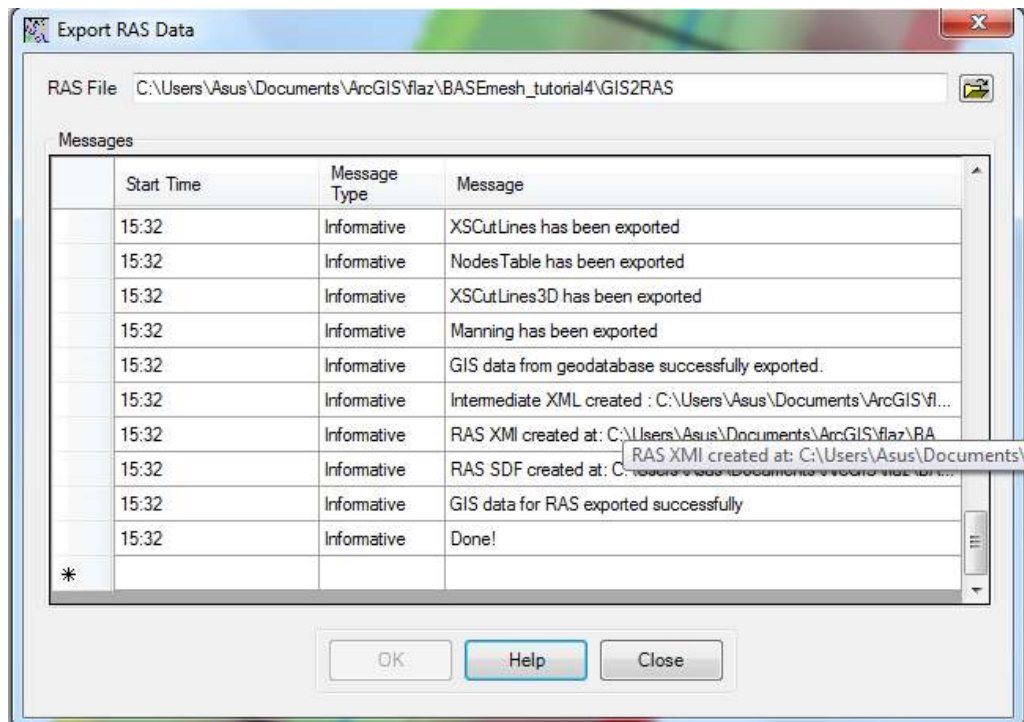
Figure 1.33 *Export RAS data*

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*.

(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
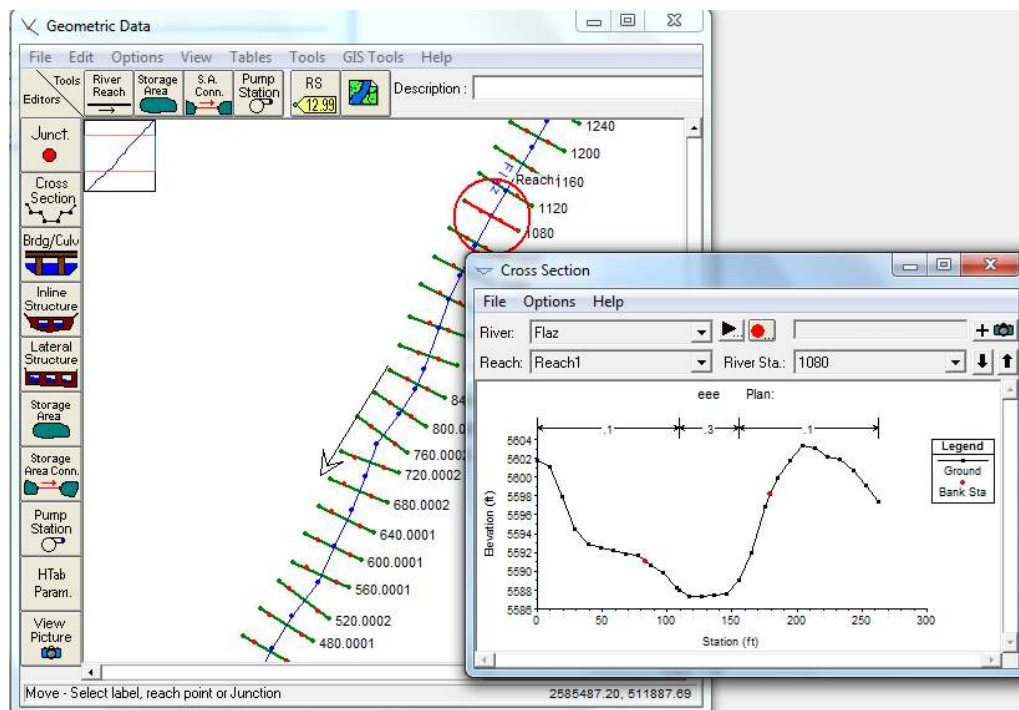
Figure 1.34 *HEC-RAS geometry data*

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.

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.
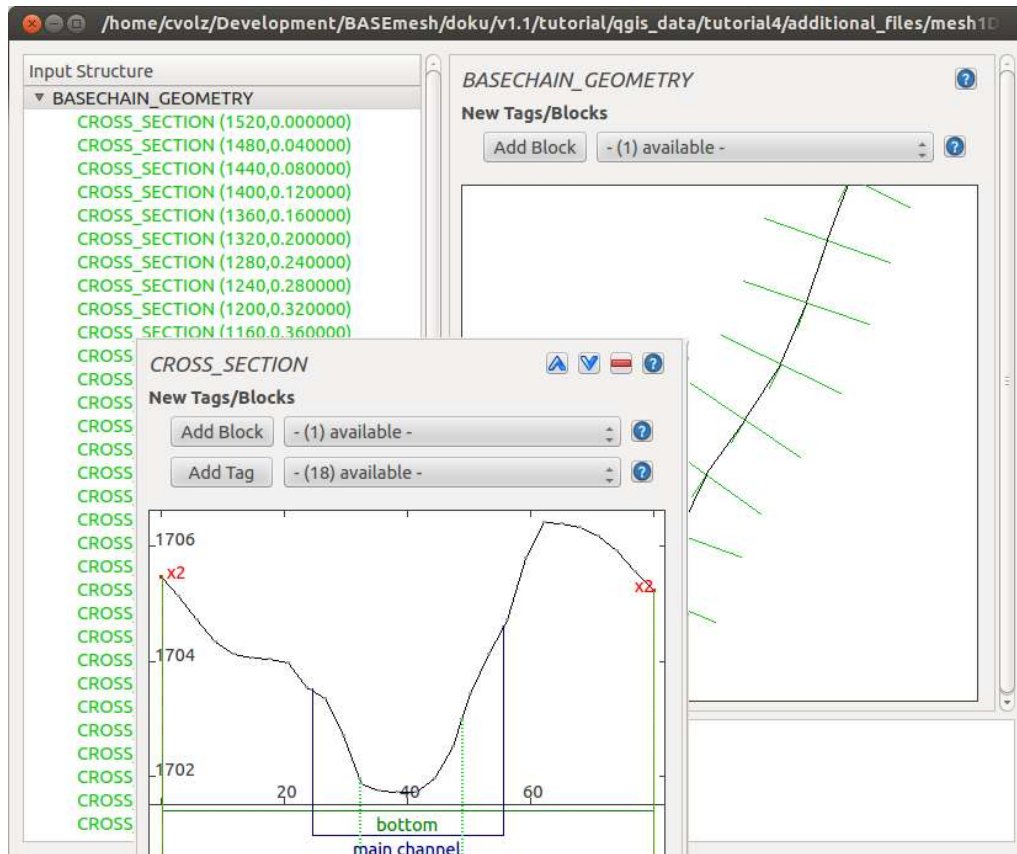
Figure 1.35 *BASEchain geometry data*

- 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 geomertry 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 successfull and all data was preserved.

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 worlwide 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 infinitively 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-clickg. 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* ᵈᵉᶠ. 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. Futhermore 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*
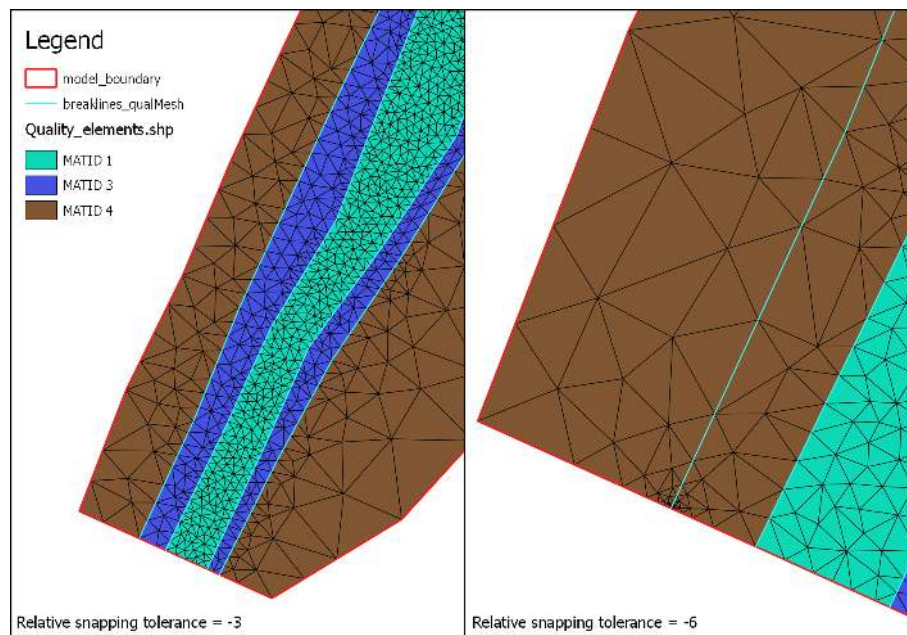
Figure 1.37 *Quality meshing using the Parameter 'Relative snapping tolerance'*

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_centred* results. Be aware that most of the data (e.g. depth, velocity, wse,...) is calculated on the elements during simulation. When chosing type *node_centered*, these results are interpolated to the nodes by BASEMENT and written to the solution files (*.sol).
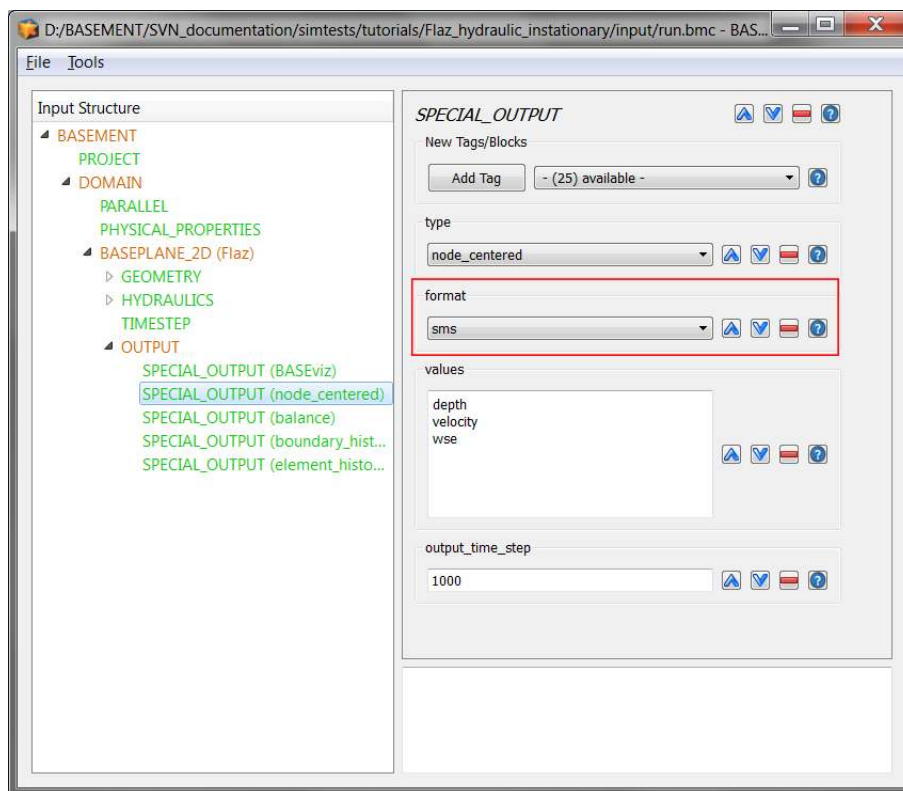
Figure 2.1 *Settings defined in the* SPECIAL_OUTPUT *Block of BASEMENT to generate time dependend result-files (*.sol) of a 2D simulation*

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

1. Quantity layer tree
2. Contour/vector overlay lock
3. Time control/slider
4. Contour settings
5. Vector properties
6. Disable/enable mesh
7. Active contour values
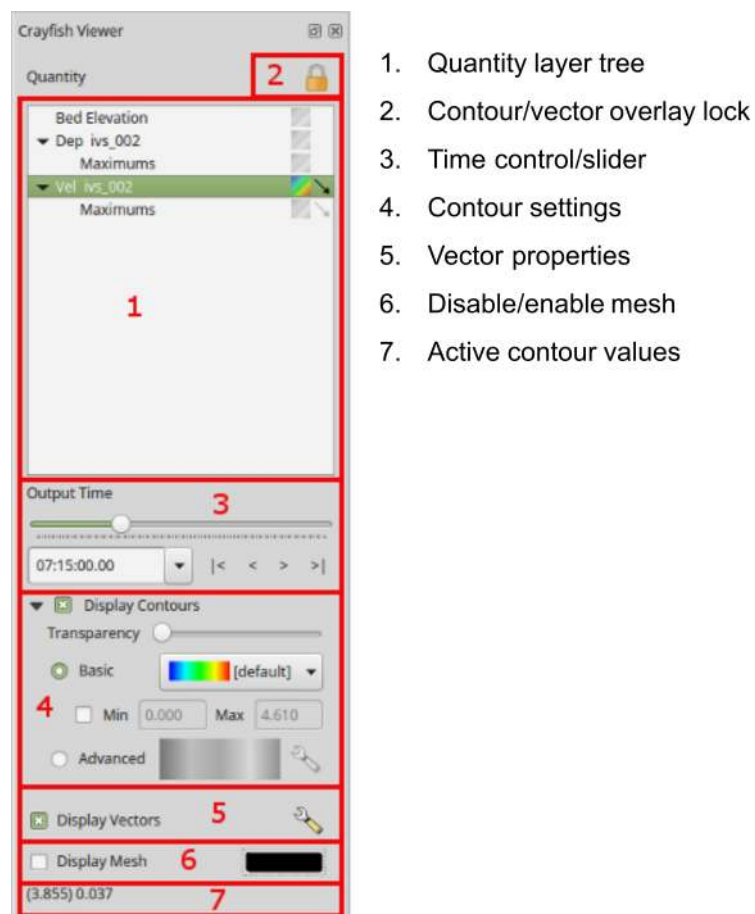
Figure 2.2 *Crayfish Viewer control panel, Crayfish version 2.1.3*

(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 the 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).
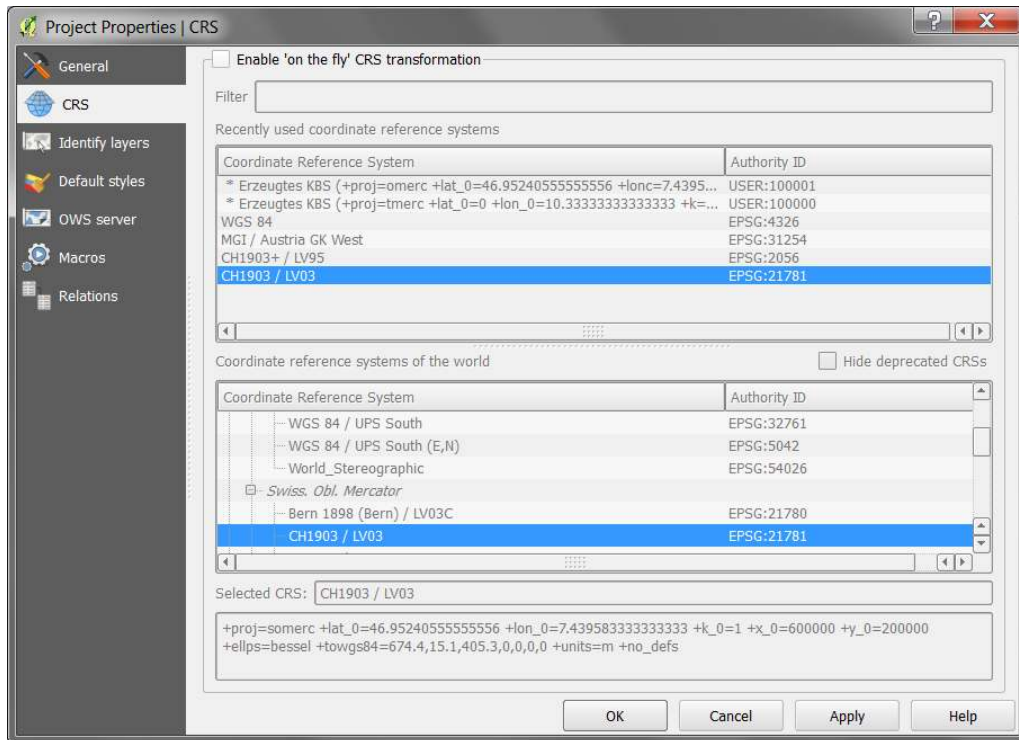
Figure 2.3 *Project Properties, CRS*

(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.
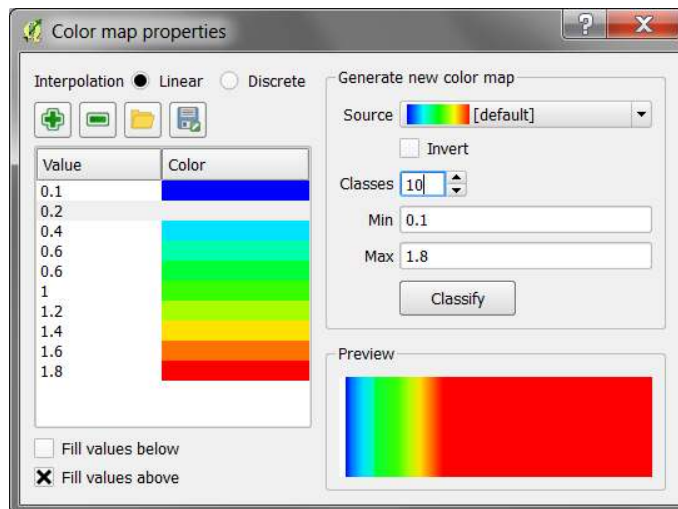
Figure 2.4 *Advanced Contour Options window of the Crayfish Viewer*
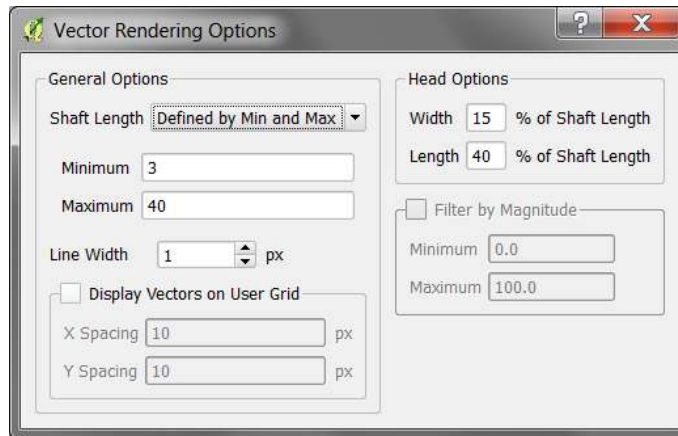
Figure 2.5 *Vector Options window of the Crayfish Viewer*
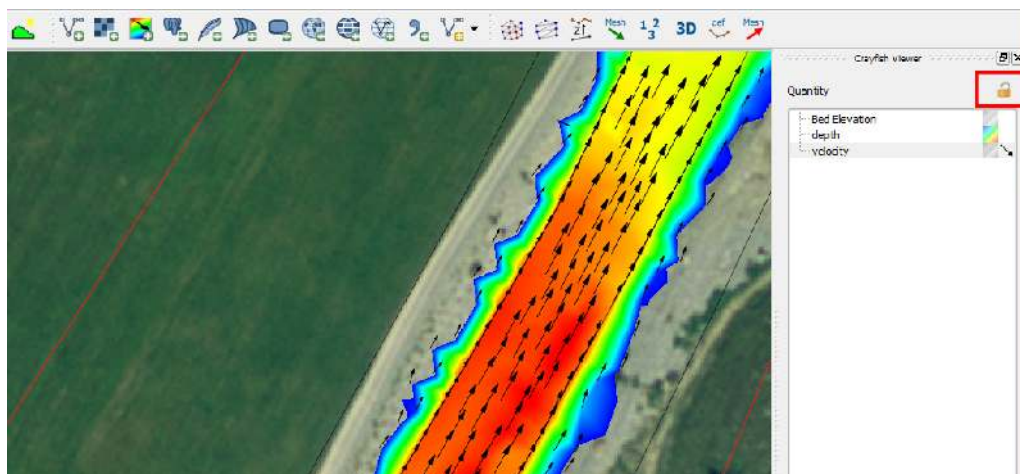
Figure 2.6 *Unlock Crayfish Viewer control panel, Orthophoto: © 2016 swisstopo (JD100041)*
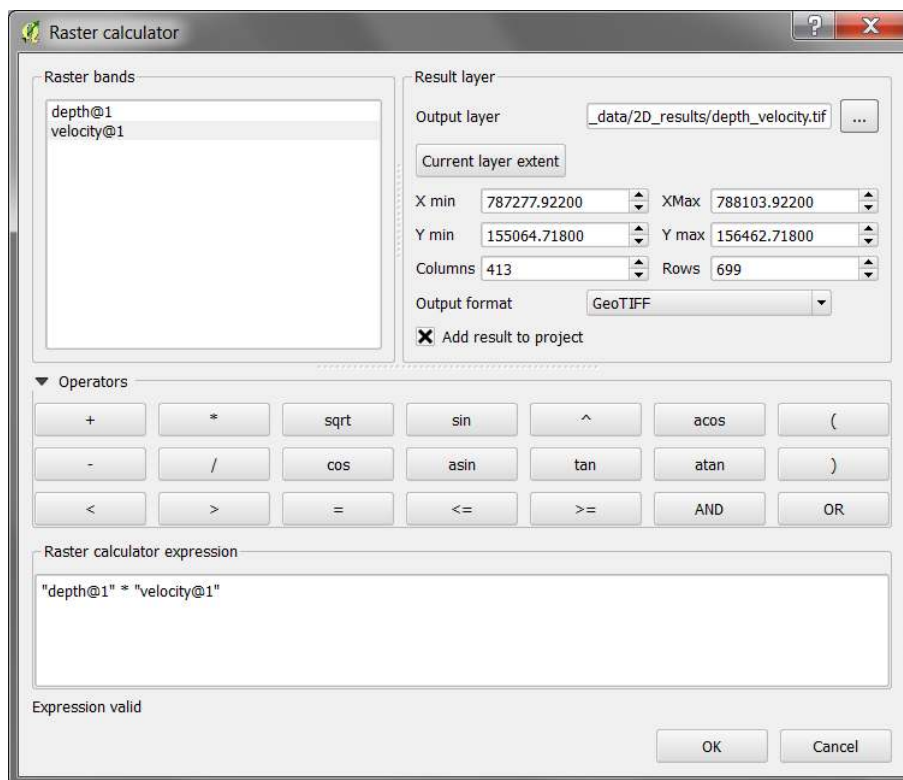
Figure 2.7 *Applying QGIS Raster Calculator*

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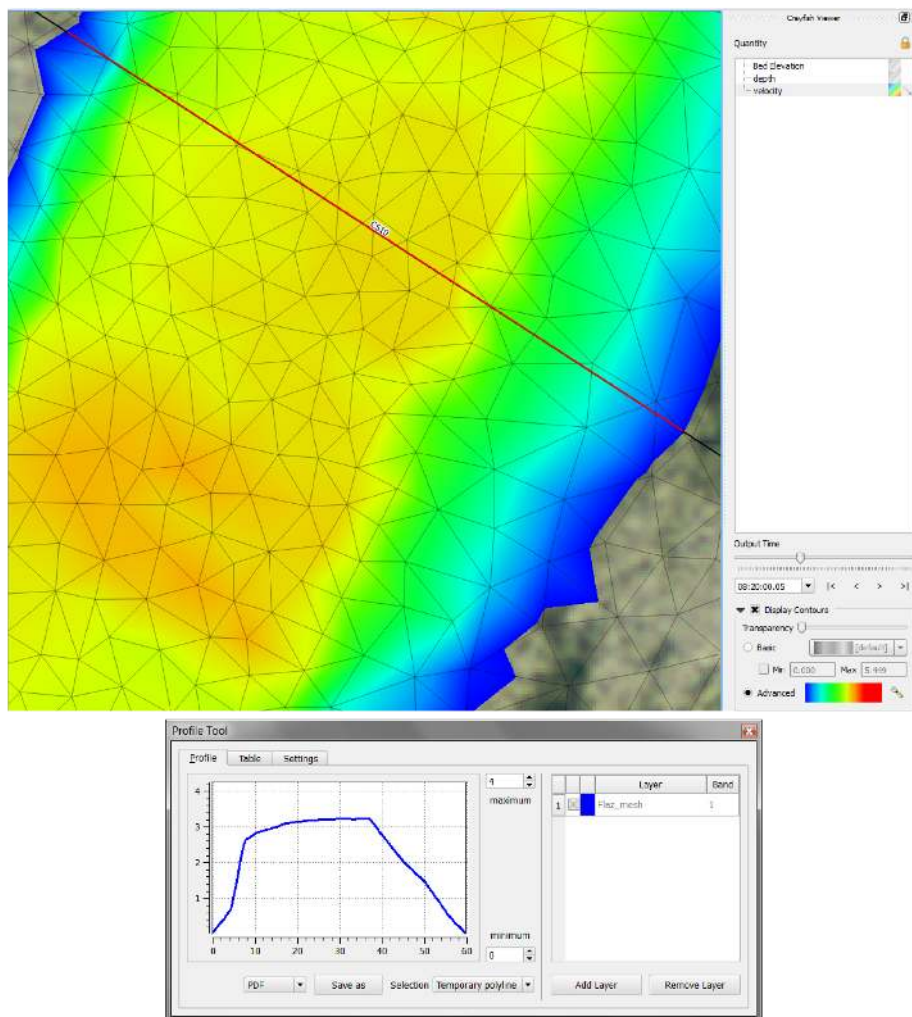
Figure 2.8 *Velocity profile along the red line drawn at the location of cross section 'CS10' generated with the 'Profile Tool' plugin*

(1) First make sure to *Deselect Features from All Layers* pressing the respective toolbar button [icon]

(2) Again click on the *Terrain profile* icon [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.
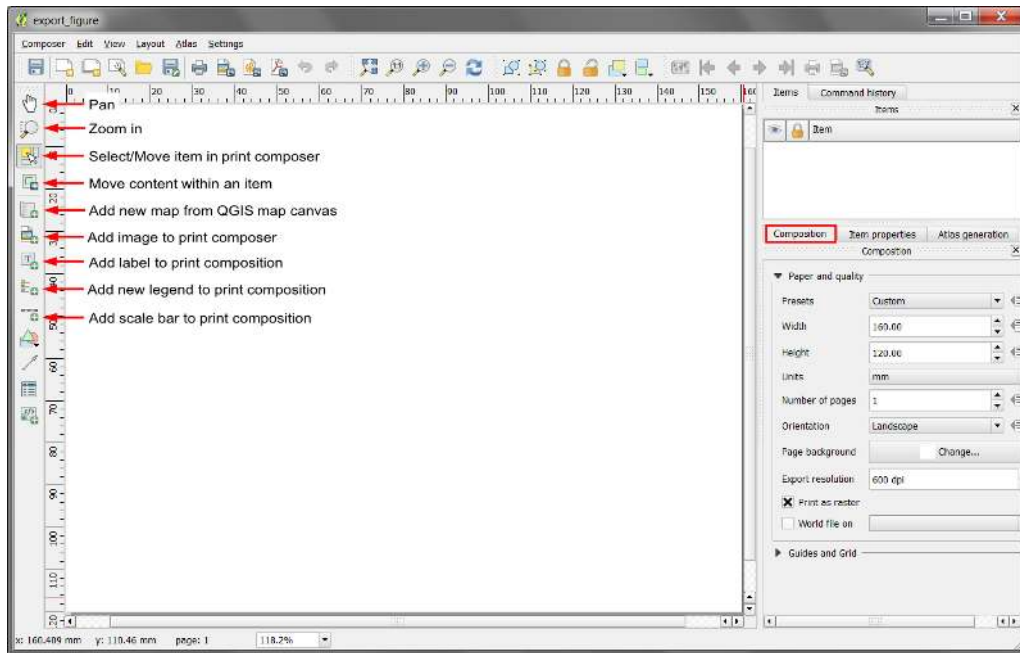
Figure 2.9 *QGIS Print Composer with some oft the basic control features*

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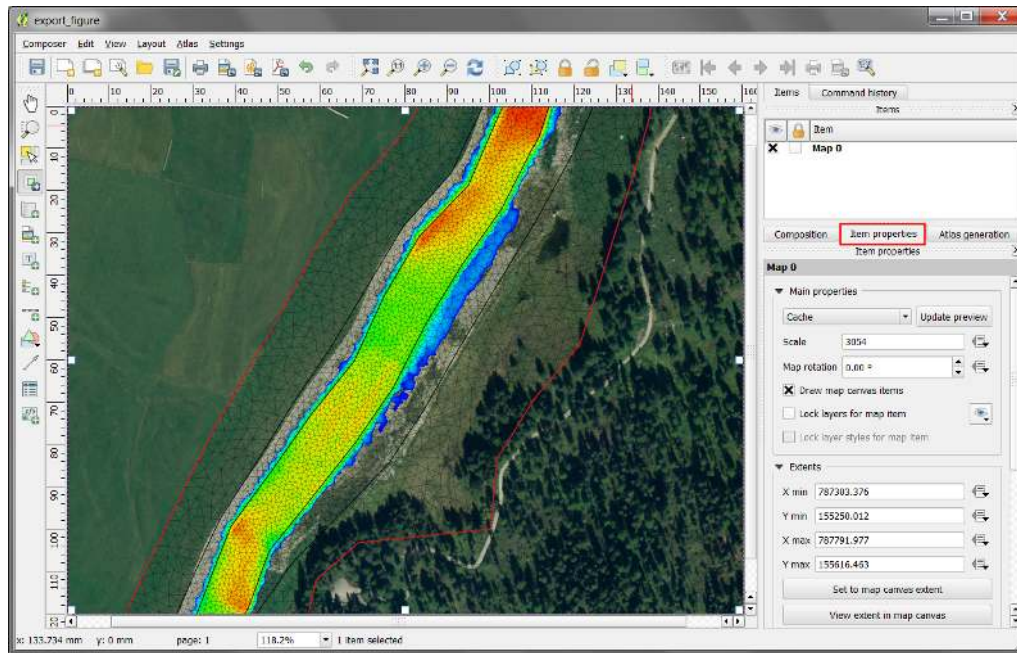
Figure 2.10 *QGIS Print Composer with displayed Crayfish layer 'depth'. Orthophoto: ©
2016 swisstopo (JD100041)*

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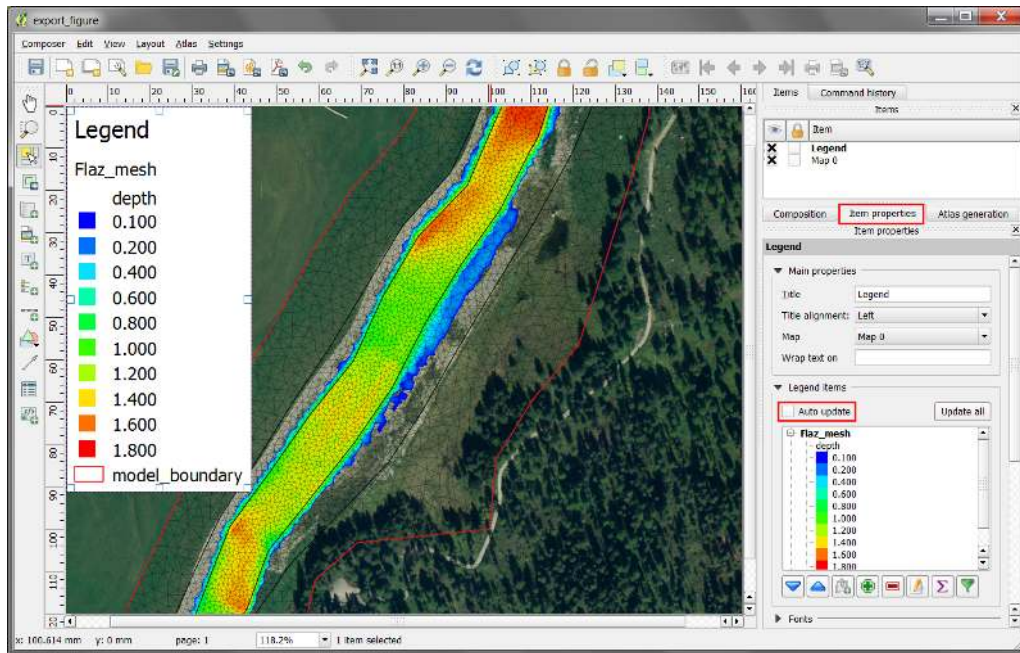
Figure 2.11 *QGIS print composer with displayed Crayfish layer 'depth' after inserting the legend. Orthophoto: © 2016 swisstopo (JD100041)*

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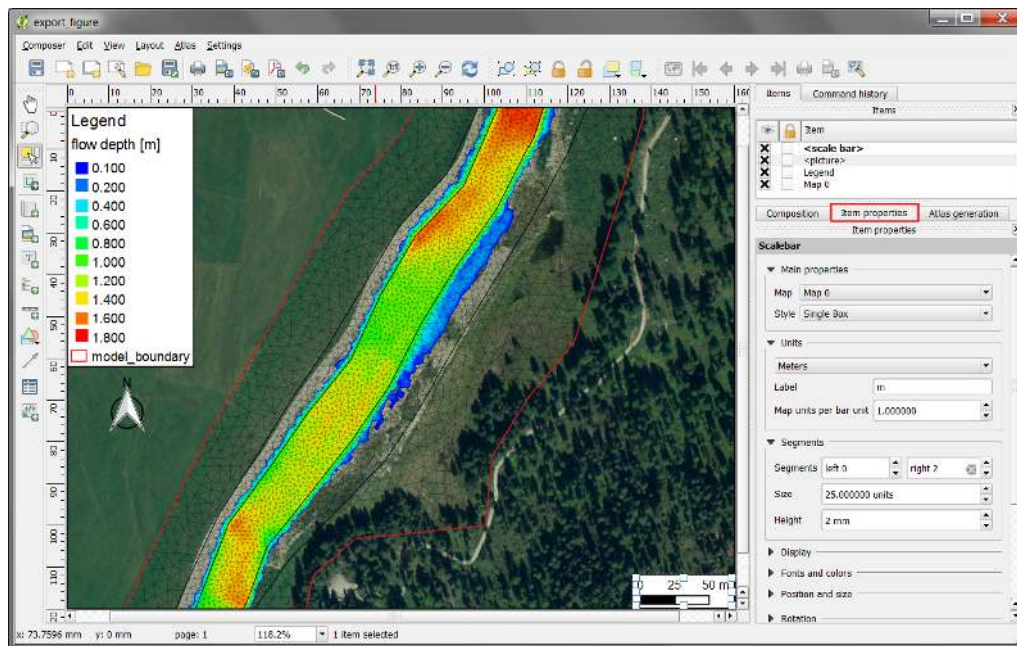
Figure 2.12 *QGIS print composer with displayed Crayfish layer 'depth' after inserting legend, scalebar and north arrow. Orthophoto: © 2016 swisstopo (JD100041)*

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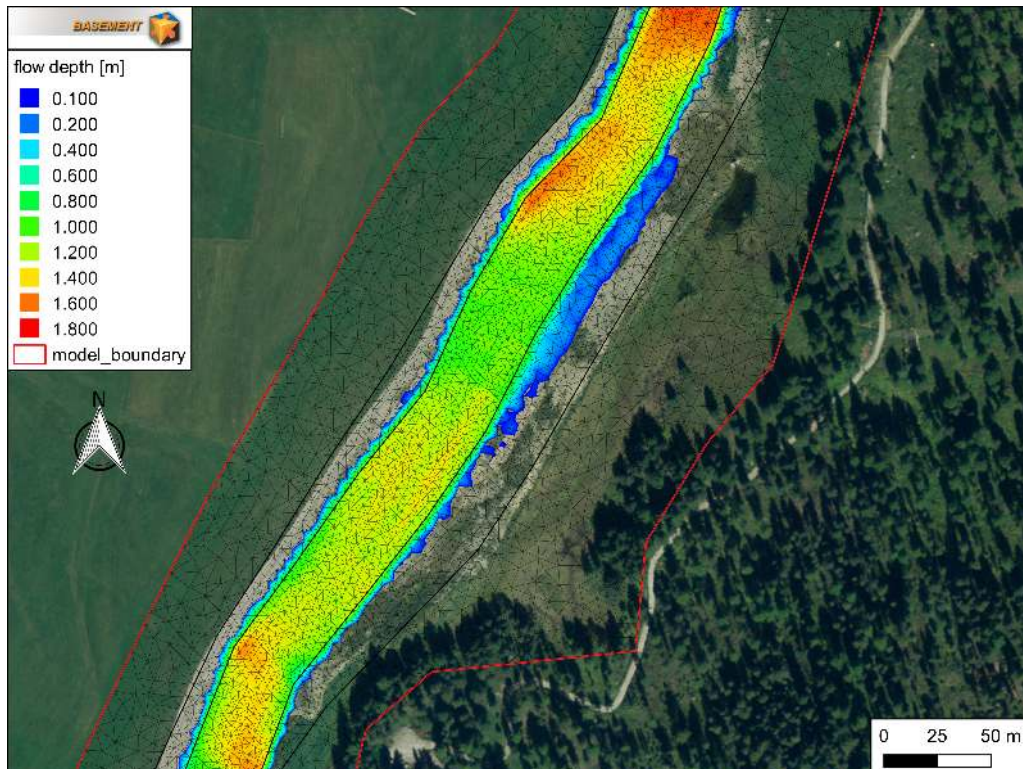
Figure 2.13 *Maximum flow depth at the river Flaz during the unsteady flow simulation. Orthophoto: © 2016 swisstopo (JD100041).*

2.2.5.2   Export animation

If your simulation results are time dependend 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
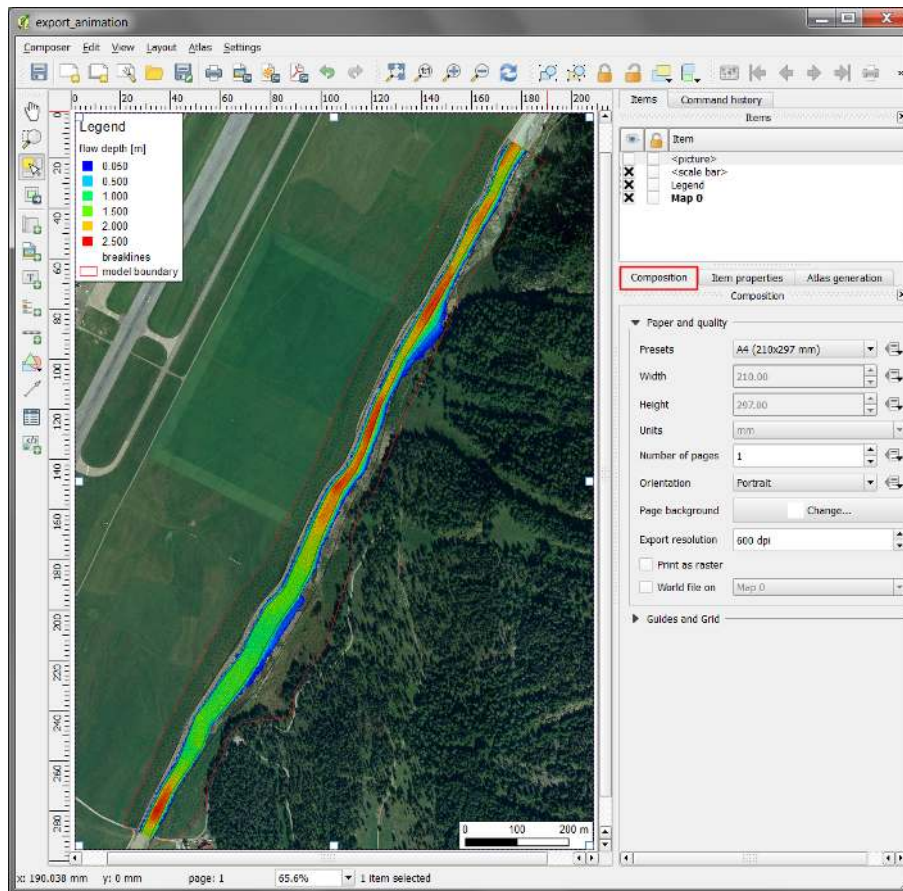
Figure 2.14 *QGIS print composer with displayed Crayfish layer 'depth' zoom to the full extent of the model for export as animation. Orthophoto: © 2016 swisstopo (JD100041)*
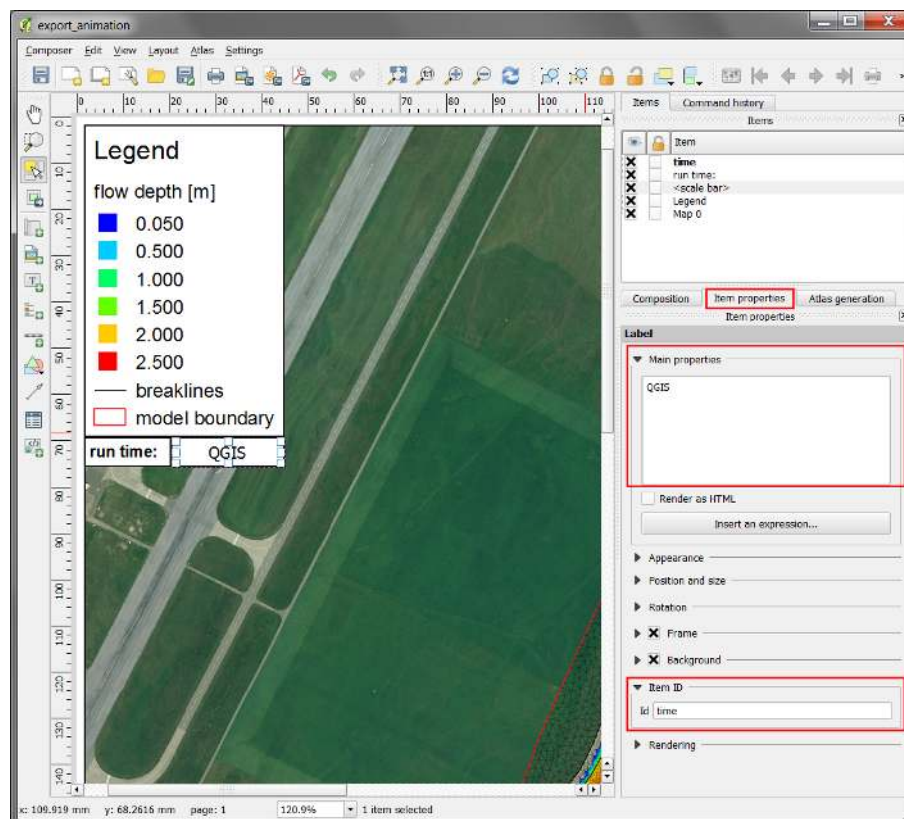
Figure 2.15 *Additional label to display the simulation run time during the animation. Orthophoto: © 2016 swisstopo (JD100041)*
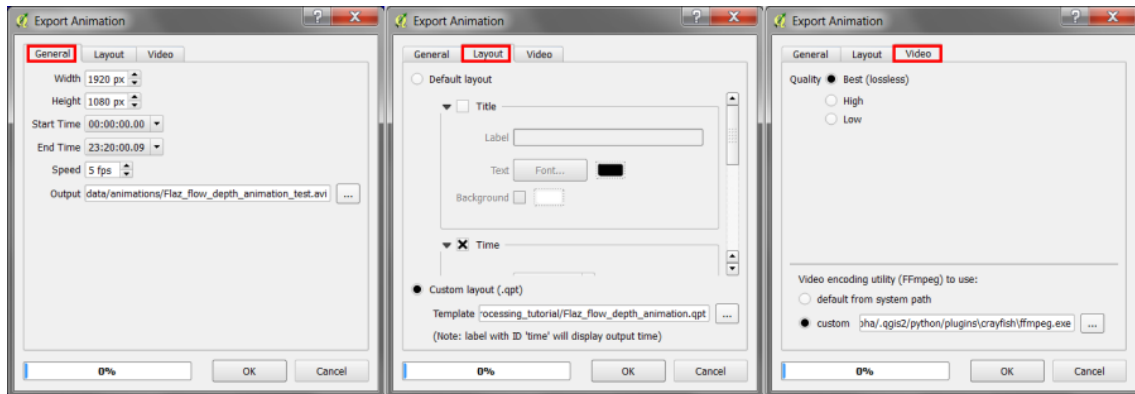
Figure 2.16 *QGIS Export Animation window*

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
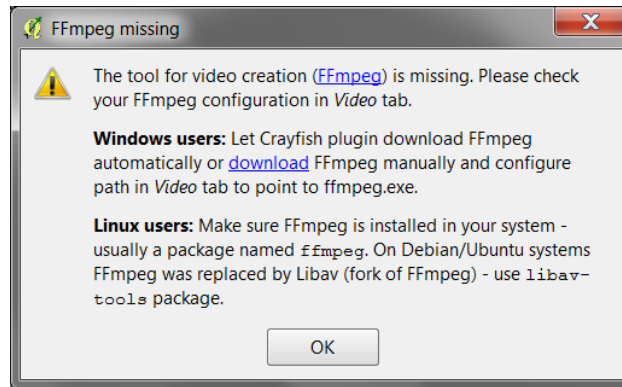
Figure 2.17 *QGIS Export Animation FFmpeg*

with ParaView version 4.3. ParaView is able to handle result-files of type *node_centered* as well as *element_centred*. 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 provieded for this tutorial. Note that the provided results where computed using a computational mesh with a higer 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.
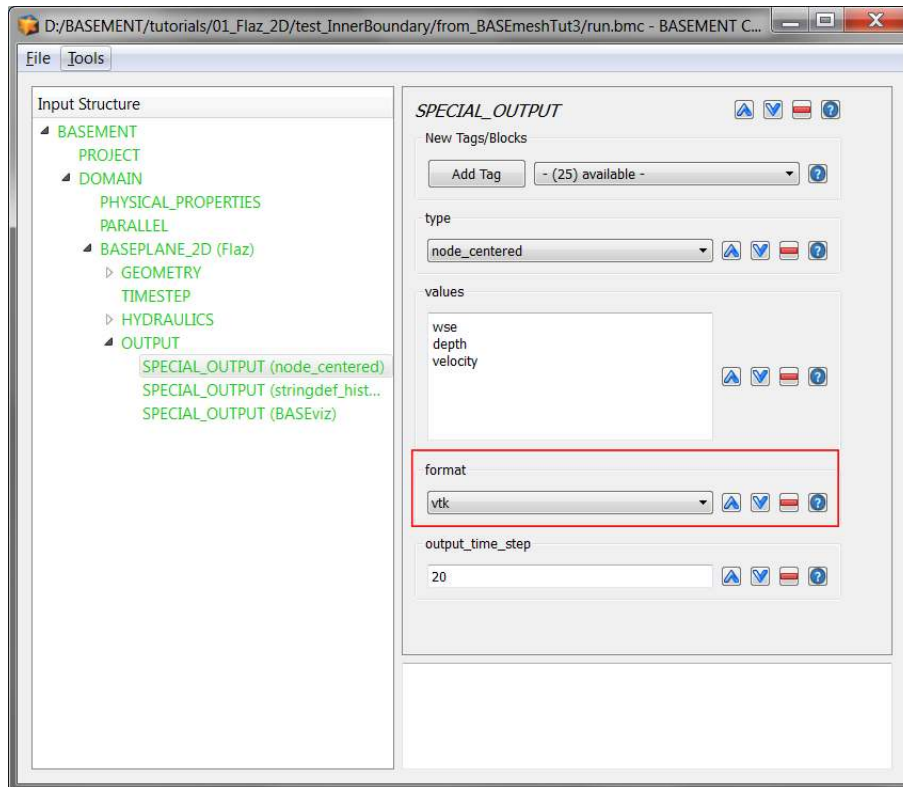
Figure 2.18 *Settings defined in the* SPECIAL_OUTPUT *Block of BASEMENT to generate time dependend result-files (*.vtk) of a 2D simulation*

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.
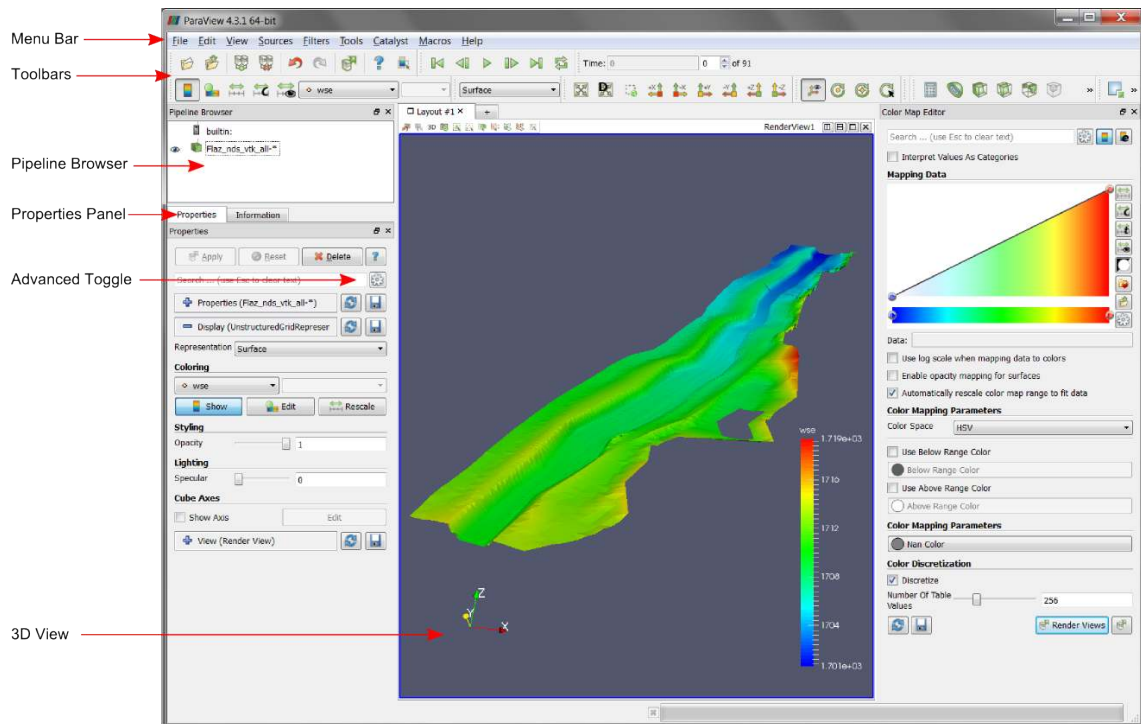
Figure 2.19 *ParaView GUI*

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 by modified under *Edge Styling*, which is enabled by toggling the advanced properties (see Figure 2.20).
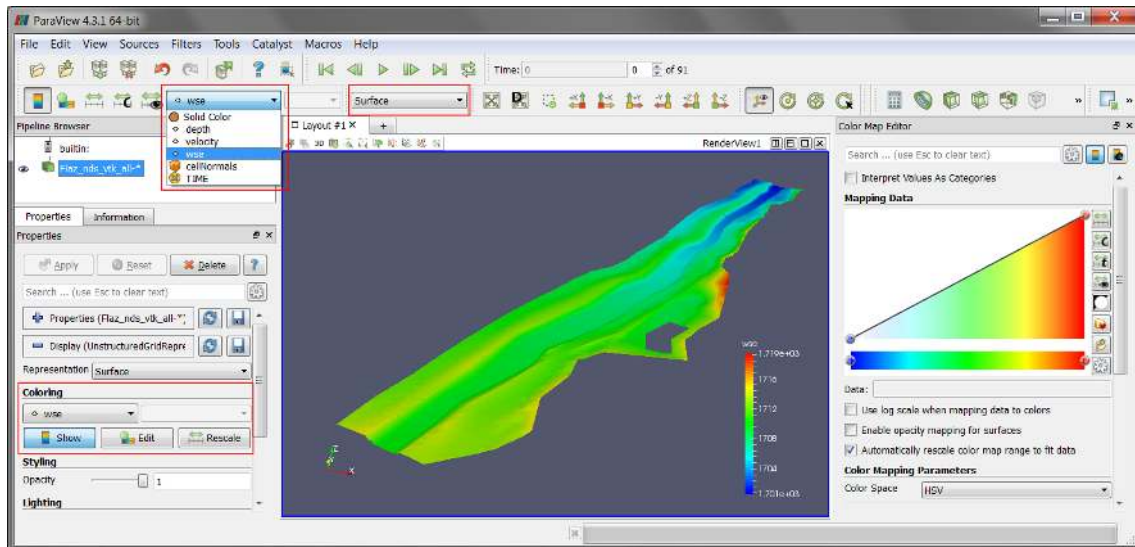
Figure 2.20 *ParaView GUI with 3D view of the domain Flaz*

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-definied expression on a per-point or cell basis.

Contour   Extracts the points, curves, or surfaces where a scalar field is equal to a user-definied 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.
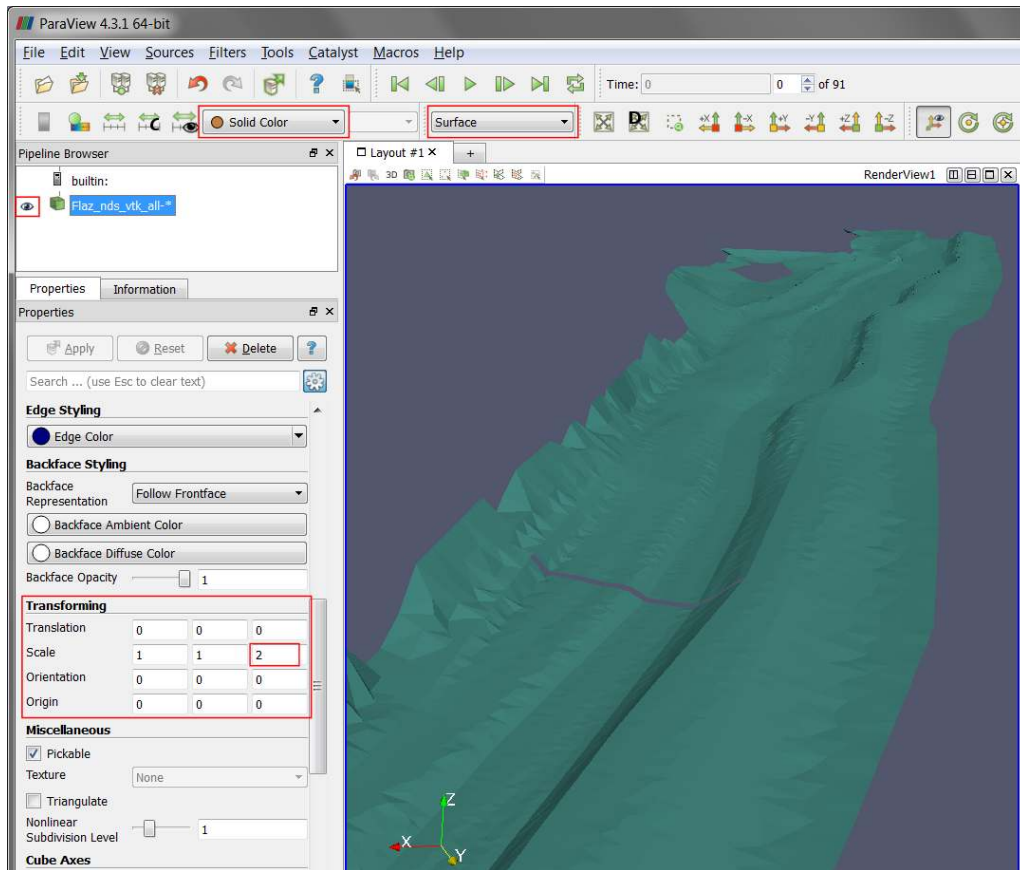
Figure 2.21 *Vertical exaggeration of solid mesh*

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 *Properies* 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*.
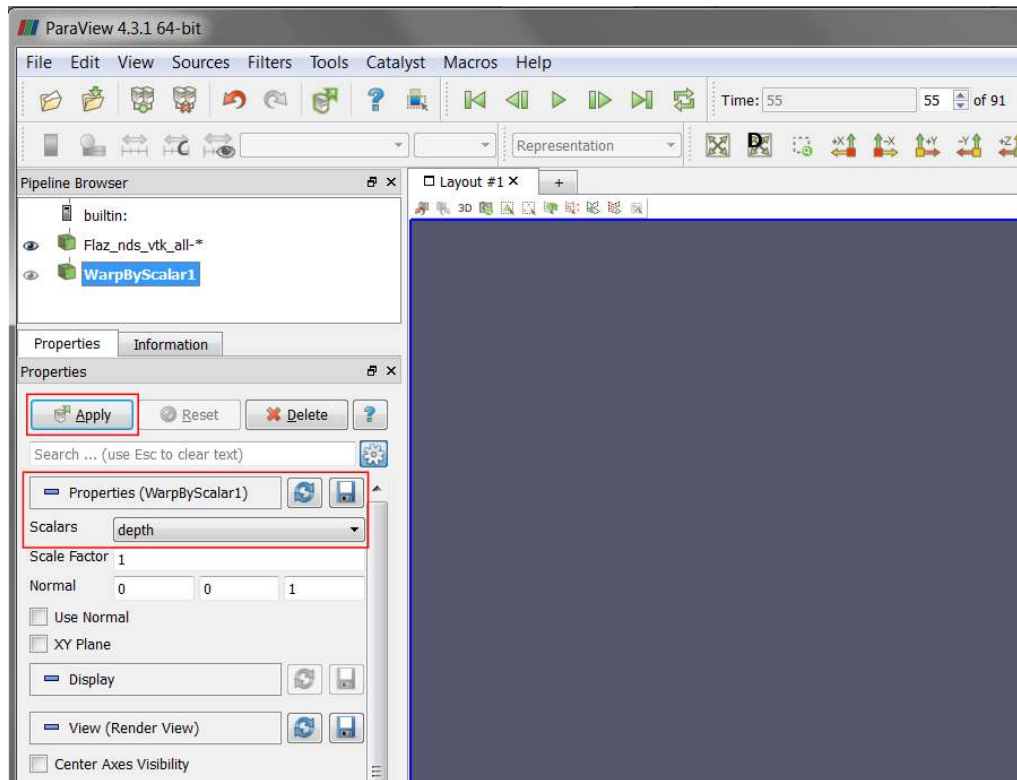
Figure 2.22 *Filter, Warp by Scalar*

(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 [image] 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.
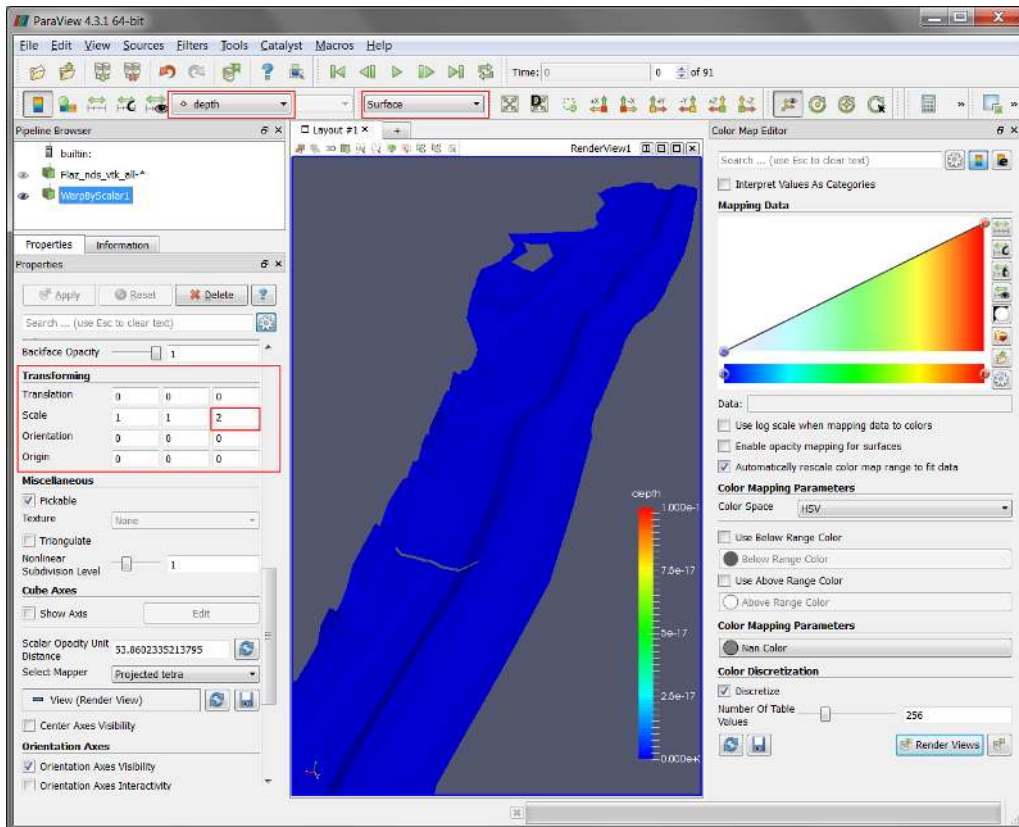
Figure 2.23 *Warp by Scalar, properties*
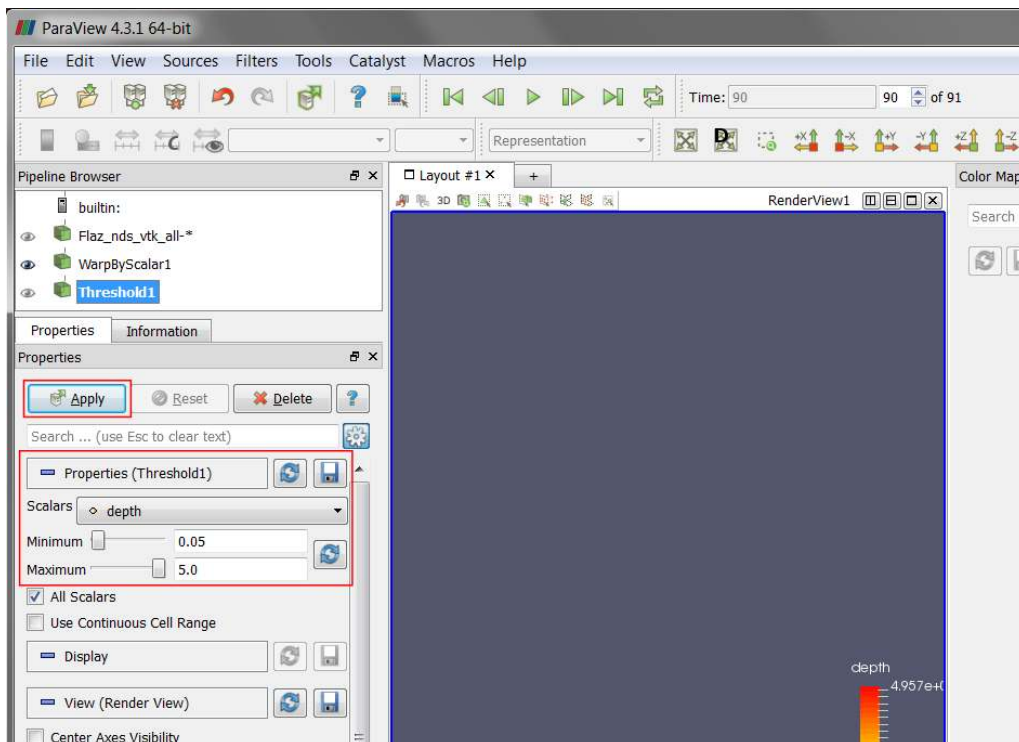
Figure 2.24 *Filter, Threshold*

(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 leeds 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 like 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.
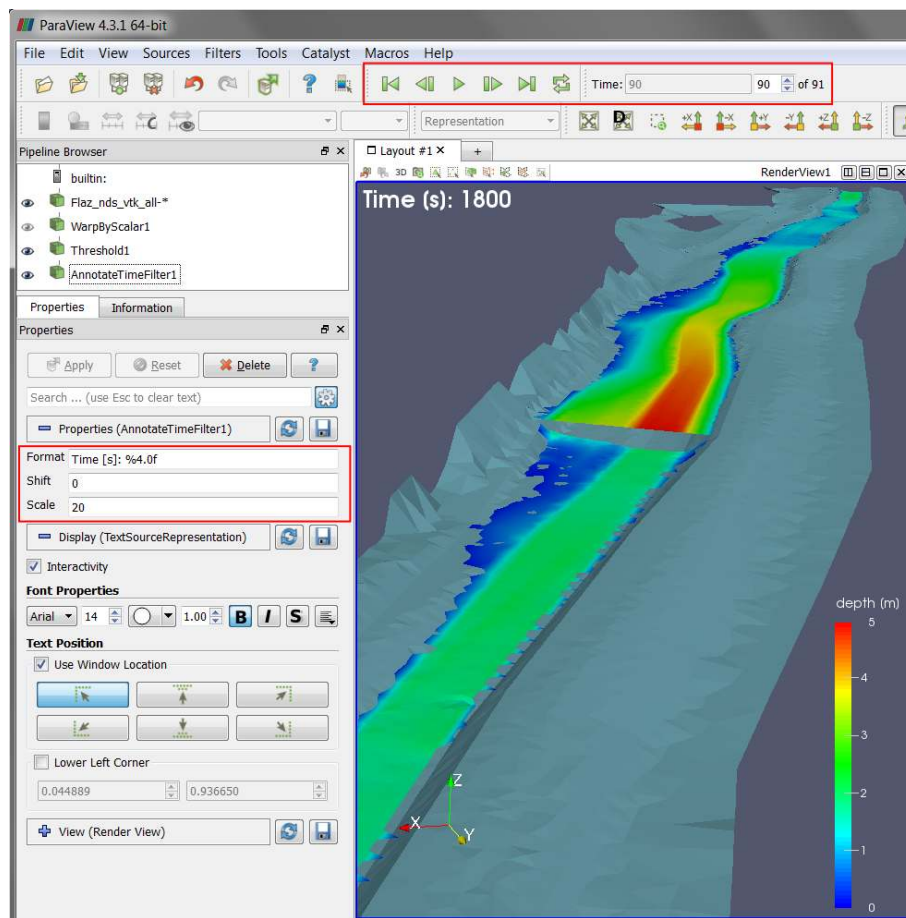
Figure 2.25 *Annotation Time Filter*
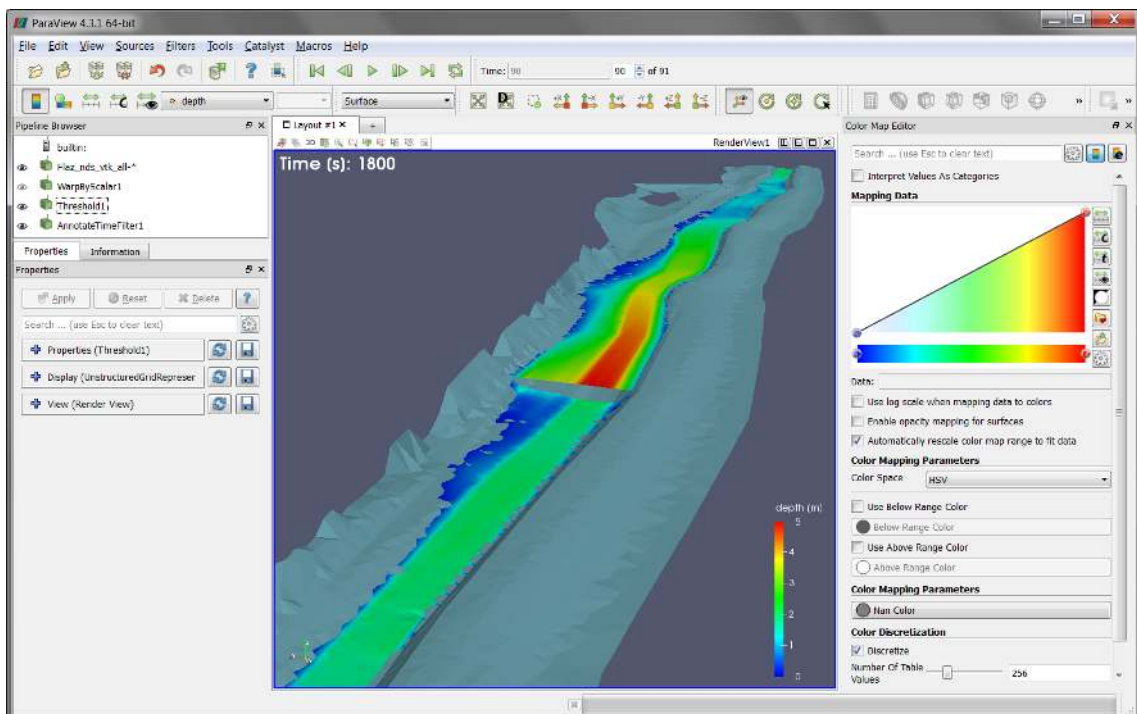
Figure 2.26 *Water depth at the steady state of the simulation*
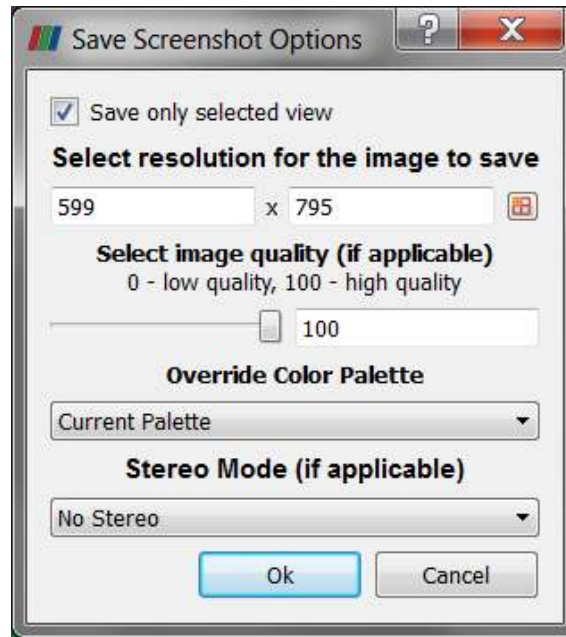
Figure 2.27 *Export Screenshot*

The save screenshot option saves the image as a raster graphic, which is natural for rendered images and very efficient for large data sets. Howerver, 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*.
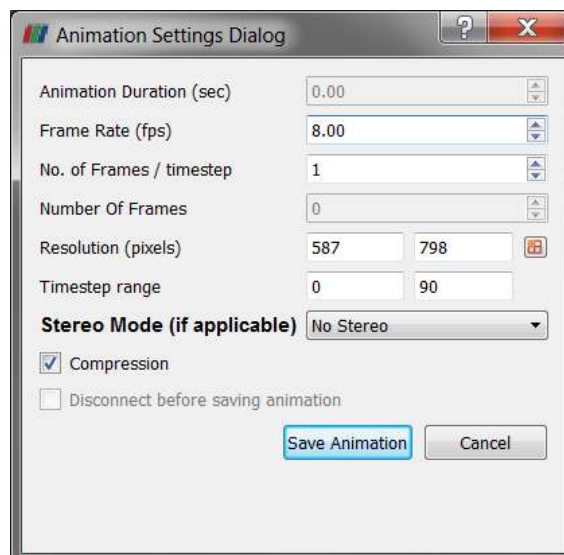
Figure 2.28 *Export Animation*

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
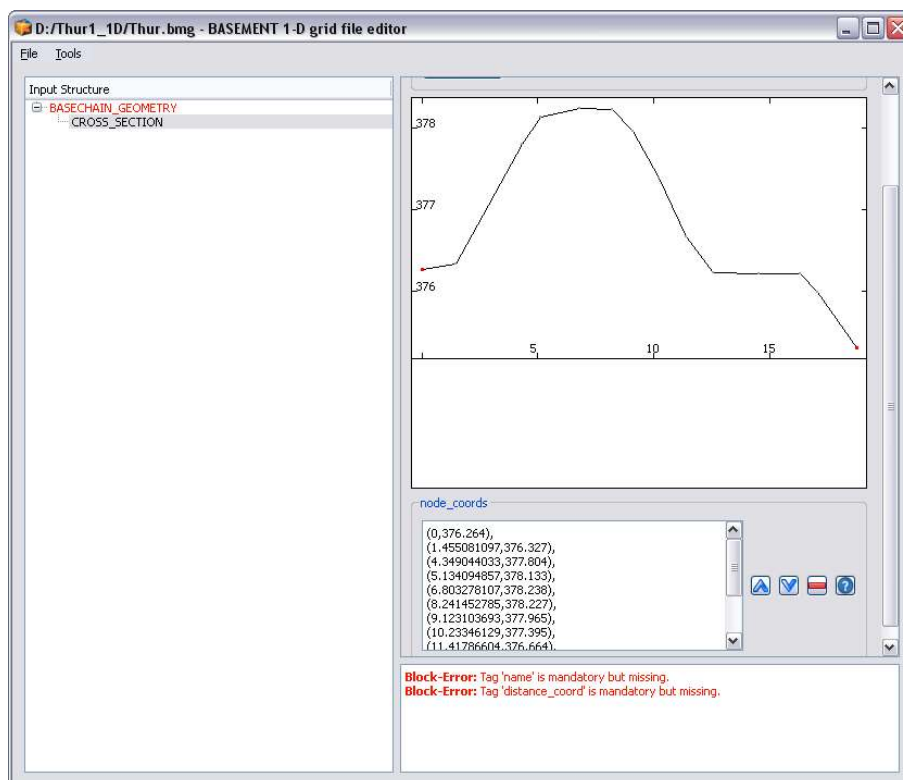
Figure 3.2 *Cross section points inserted in the topography editor*

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_{Str} values are used:

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

For the bed bottom the following transformation, based on the grain characteristics of the sediment is used ($d_{90} = 5$ cm):
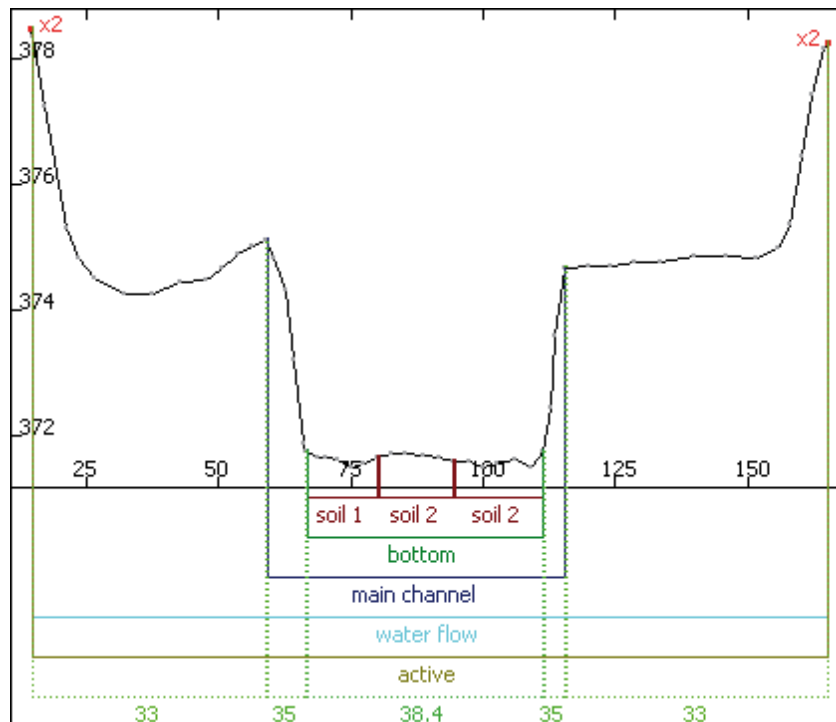
Figure 3.3 *Delimitation of cross section zones*

Figure 3.4 *Definition of cross section properties*

Figure 3.5 *Definition of table values*

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

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.
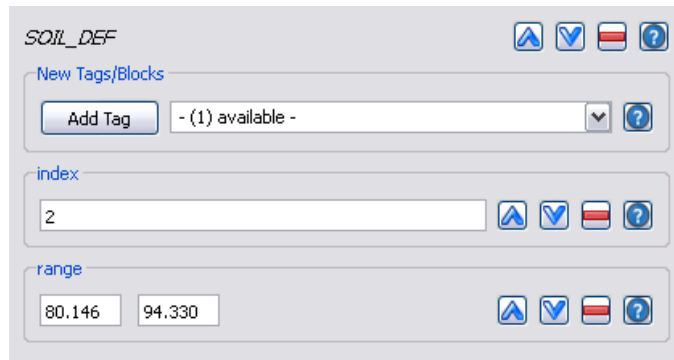
Figure 3.6 *Declaration of soil types*
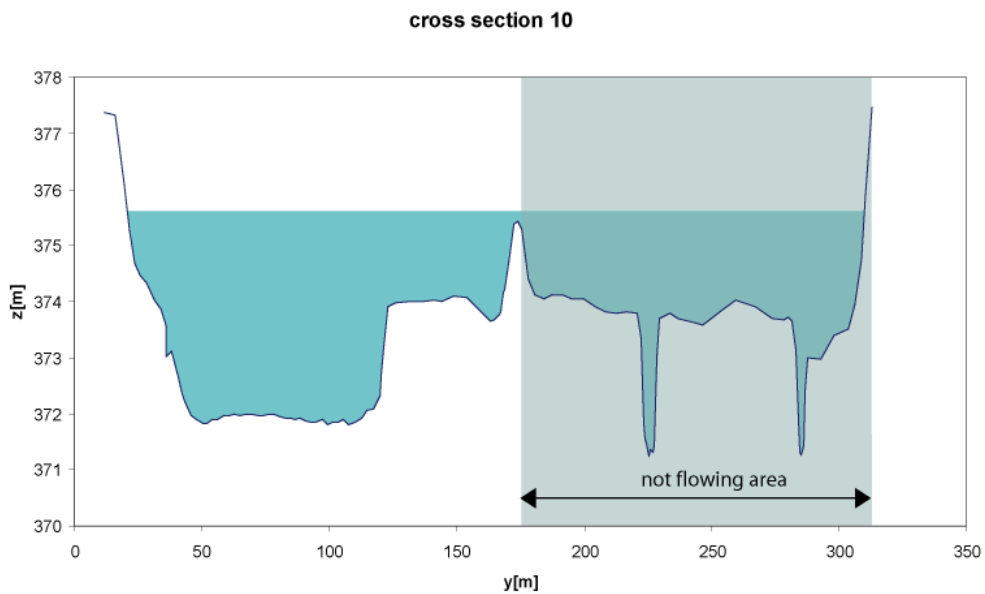
Figure 3.7 *Definition of non flowing areas*

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_[m^3/s]". If the discharges correspond to the steady inflow discharge of 30 m^3/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	...
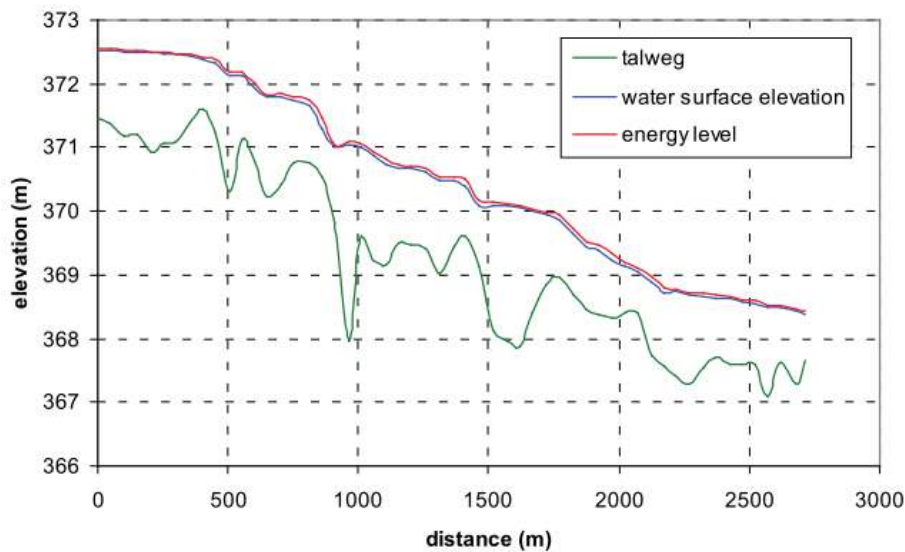
Figure 3.8 *Longitudinal profile*

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
```
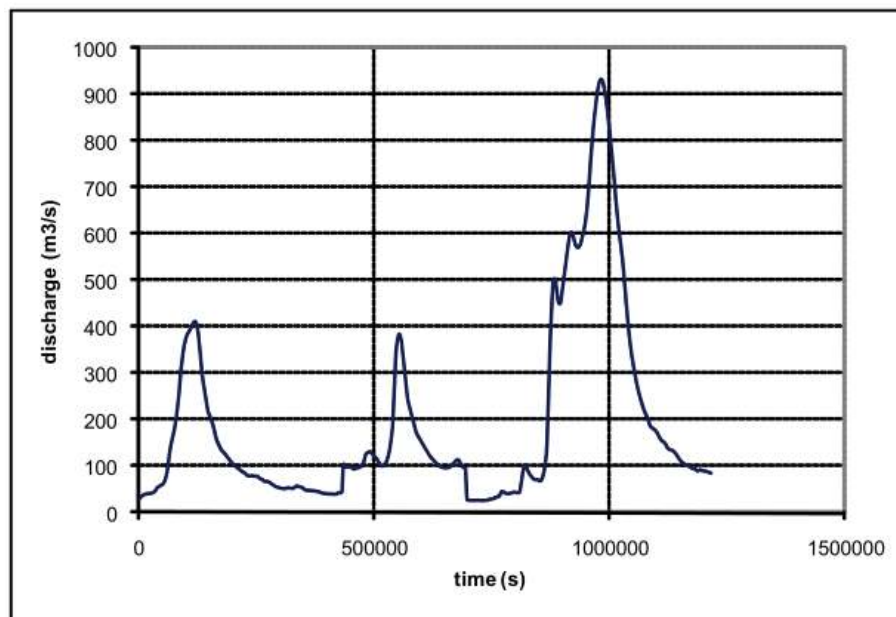
Figure 3.9 *Hydrograph of the flood event of 2005*

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 m^3/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.
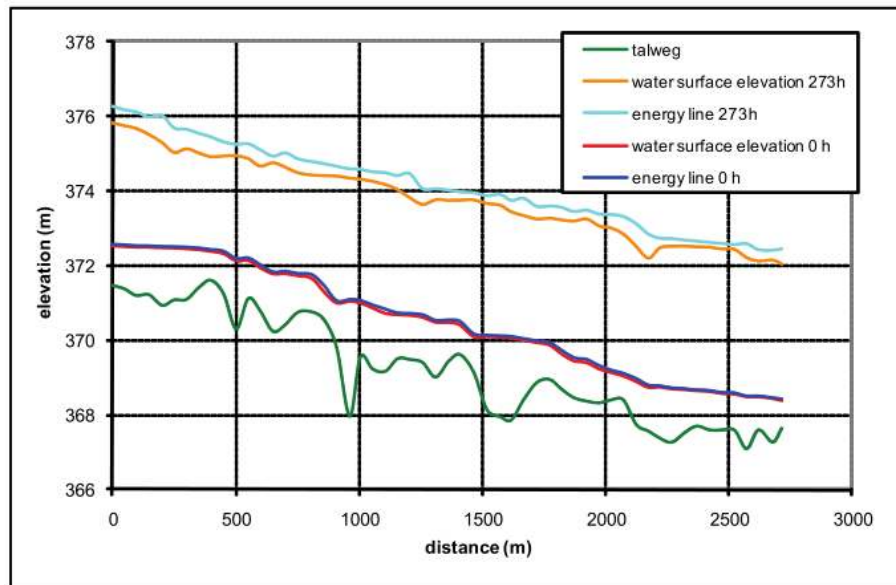
Figure 3.10 *Longitudinal profile for maximum discharge*
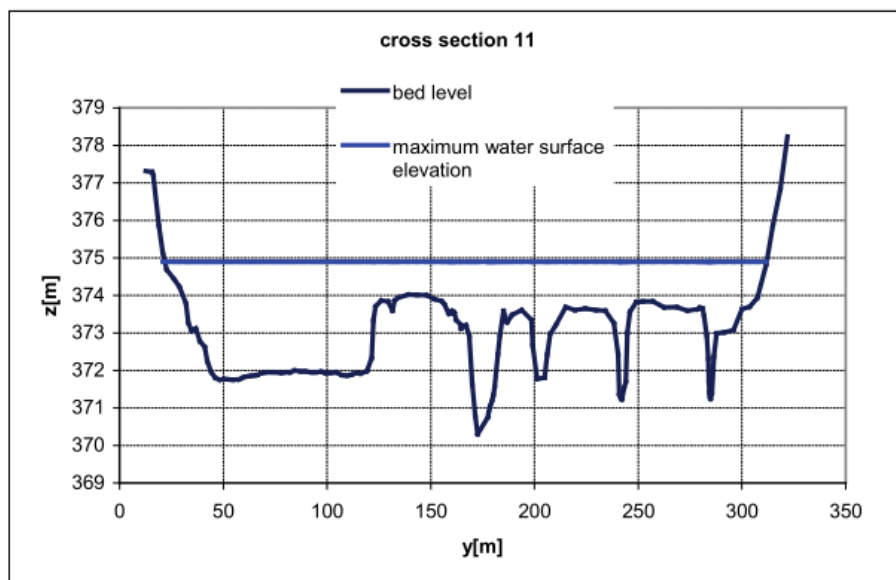
Figure 3.11 *Example of resulting water surface elevation*

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
    }
}
```
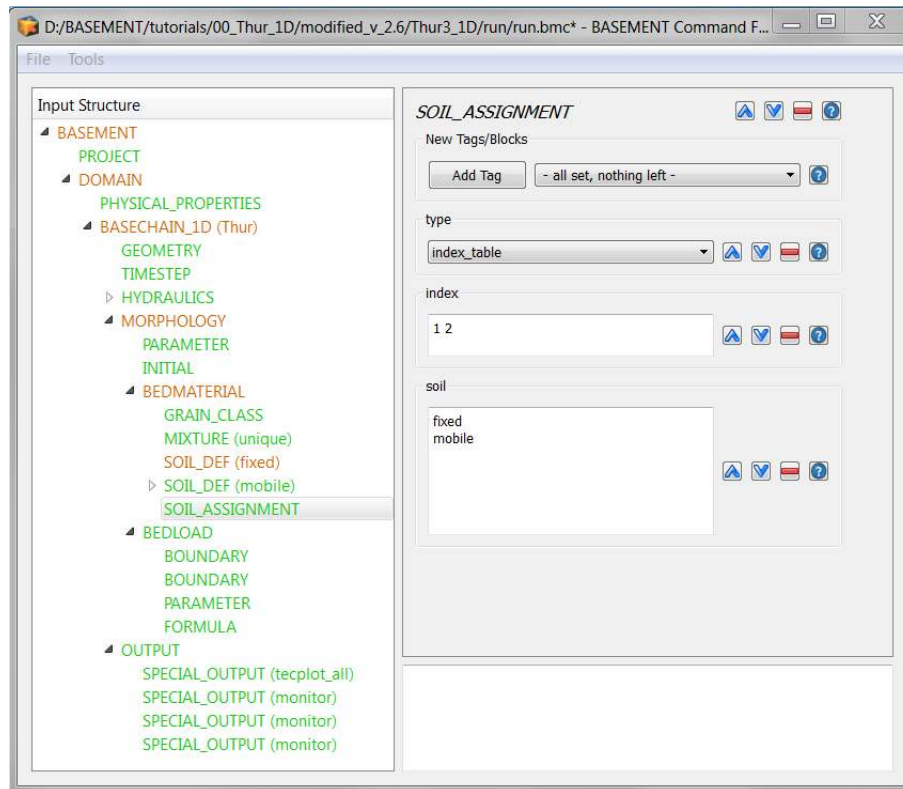
Figure 3.12 *Soil assignment in the user interface*

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 wich 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 choosen.

```
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
}
```
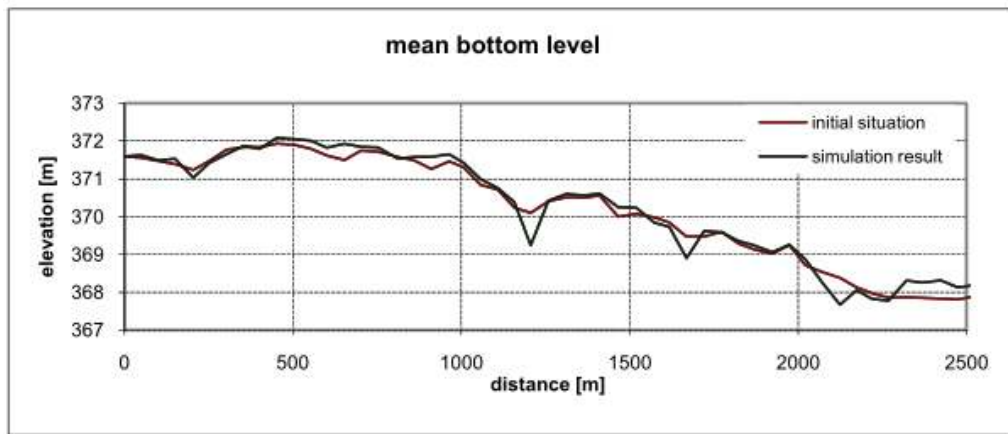
Figure 3.13 *Longitudinal profile of mean bottom level*

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.
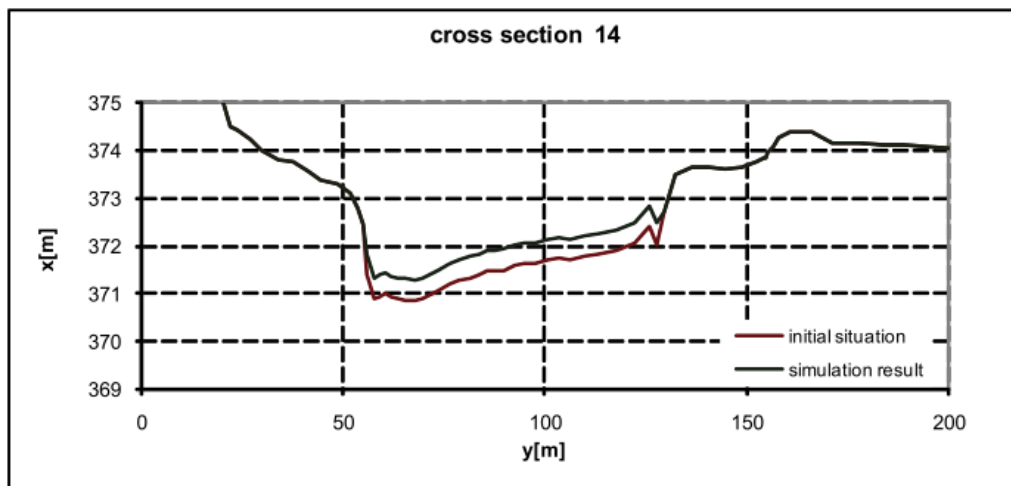
Figure 3.14 *Transversal cross section profile*

4

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

4.1 Introduction

In the 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
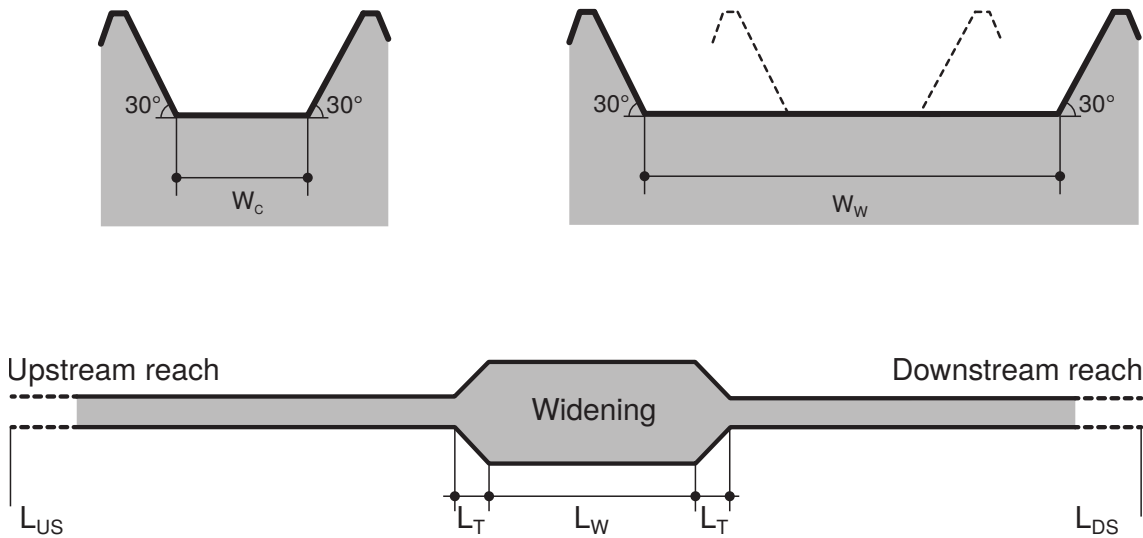
Figure 4.1 *Sketch of symmetric widening used for the numerical simulations in this tutorial.*
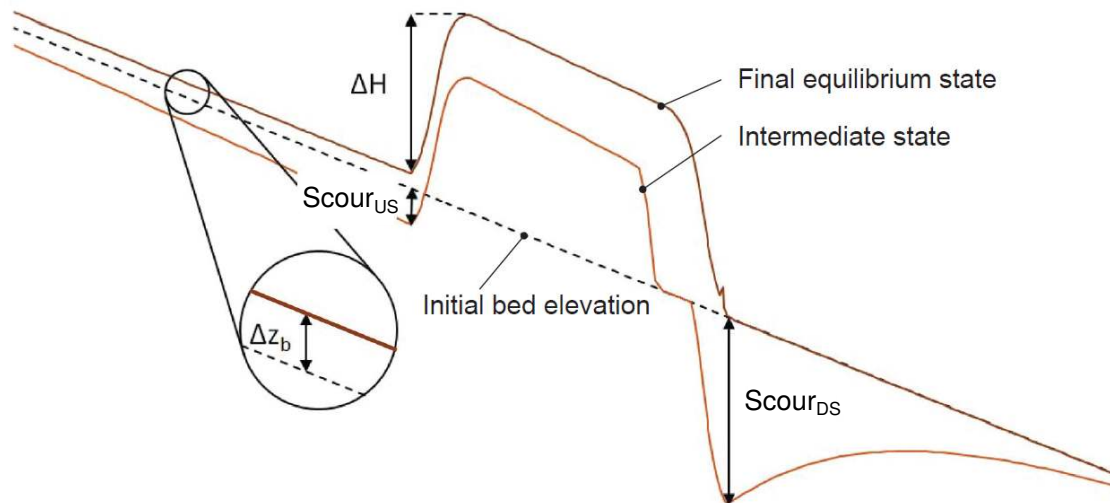
Figure 4.2 *Temporal evolution of the longitudinal bed profile due to a local river widening.*

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
max dz table	Change in bottom elevation to trigger recomputation of tables [tables1D]	0.1 [m]	0.01 [m]
upwind	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

Type	Symbol	Units	Description	Default value	Variation
	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	
	S_W	%	Initial slope widening	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
	$k_{St,C}$	m$^{1/3}$/s	Strickler roughness channel	34	
	$k_{St,W}$	m$^{1/3}$/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	$Q_{S,Q800,d40} = 0.067$
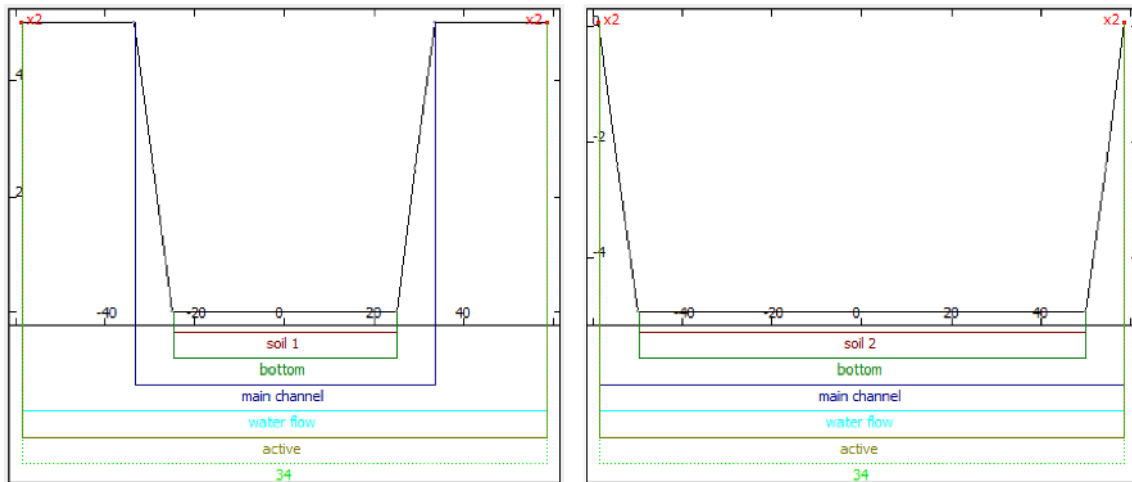
Figure 4.4 *Cross sections of the BASEMENT topography representing the channel (left) and the widening (right) to illustrate the geometry and the bottom/soil definition.*

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.

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
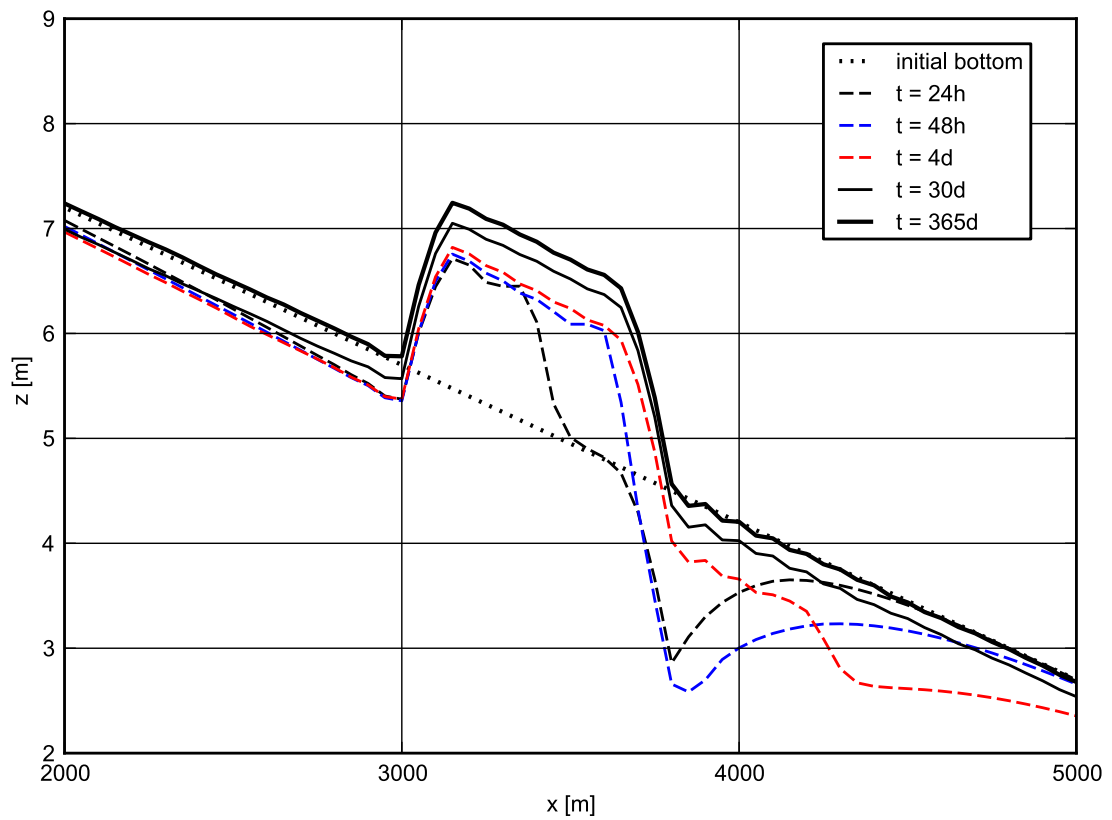
Figure 4.5 *Temporal evolution of the mean bottom elevation of the reference geometry with default configuration.*
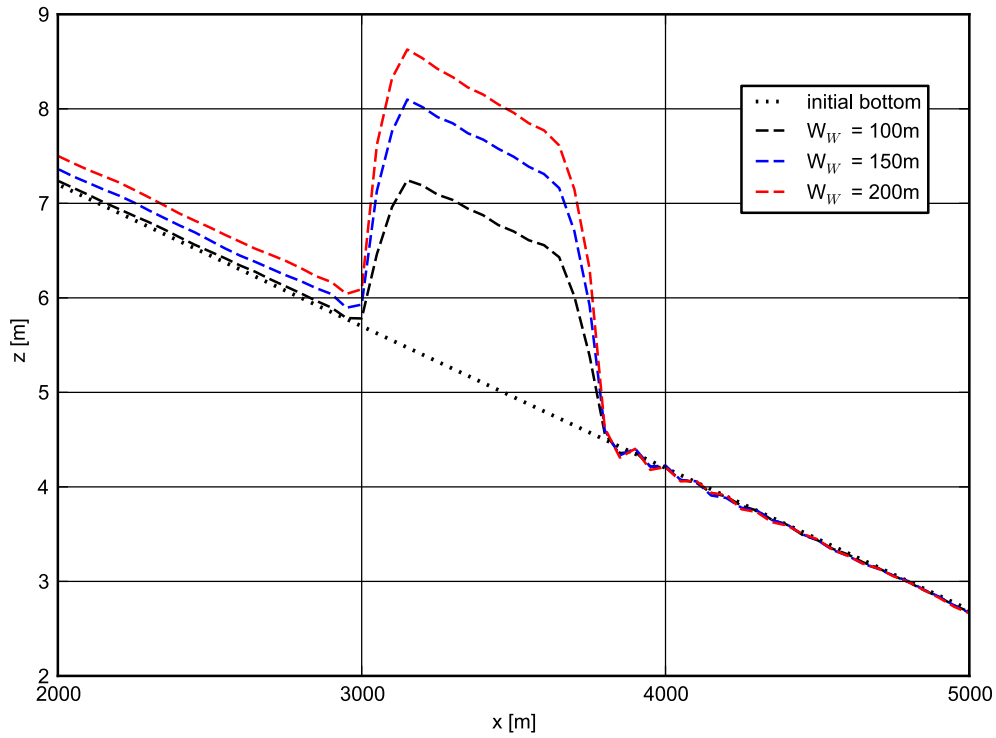
Figure 4.6 *Longitudinal bed profile of the initial and resulting steady state for different widening width W_W.*
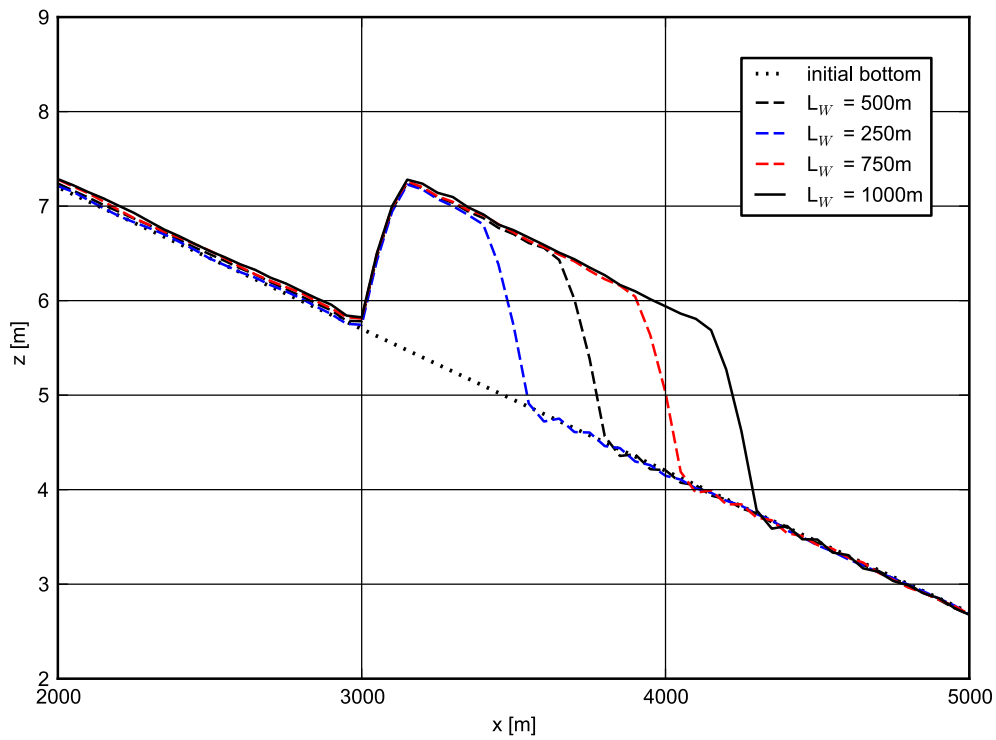
Figure 4.7 *Longitudinal bed profile of the initial and resulting steady state for different widening length L_W.*
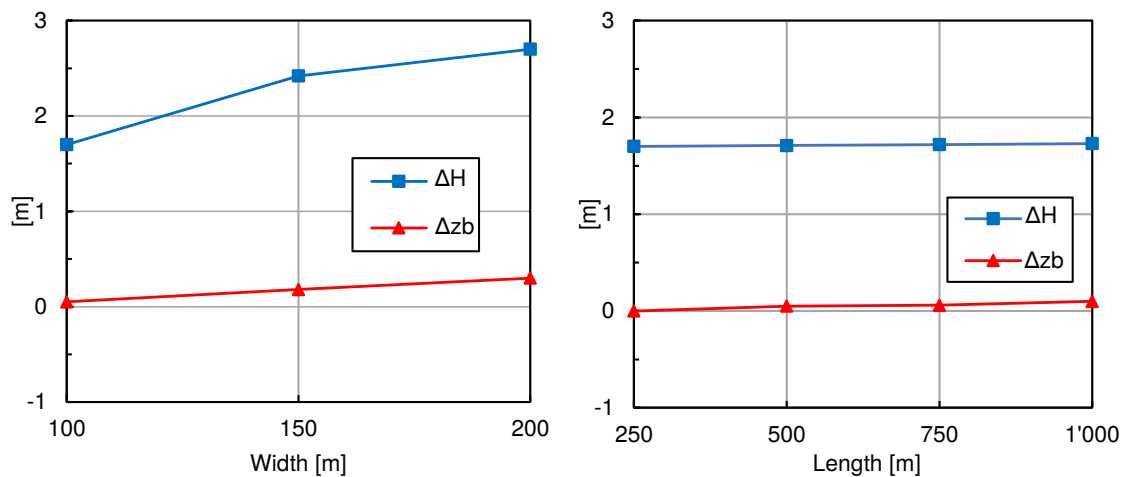
Figure 4.8 *Bottom offset ΔH and upstream displacement Δz_b for different W_W (left) and L_W (right), parameter variation according to table 4.2.*

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 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.
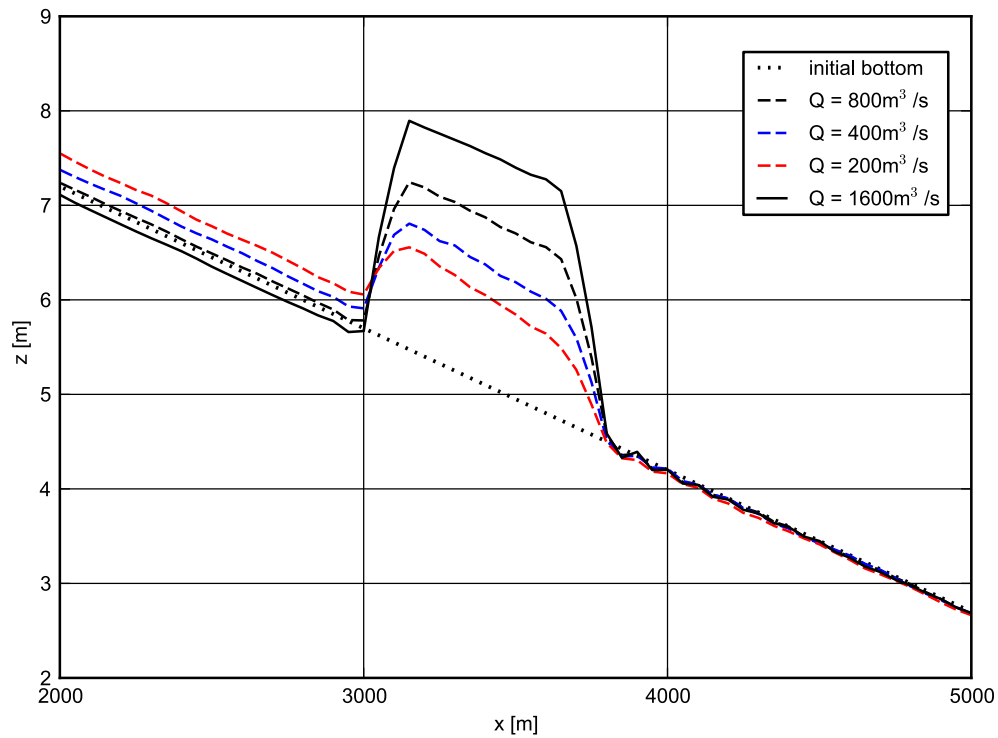
Figure 4.9 *Longitudinal bed profile of the initial and resulting steady state for different discharges Q.*
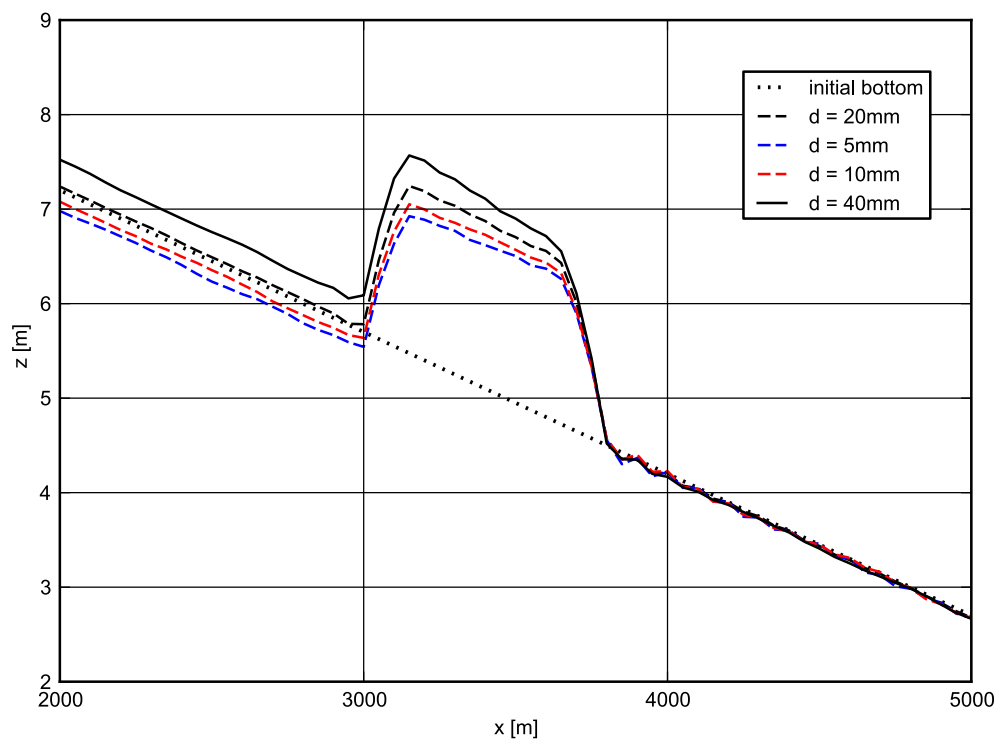
Figure 4.10 *Longitudinal bed profile of the initial and resulting steady state for different mean grain size diameters d.*
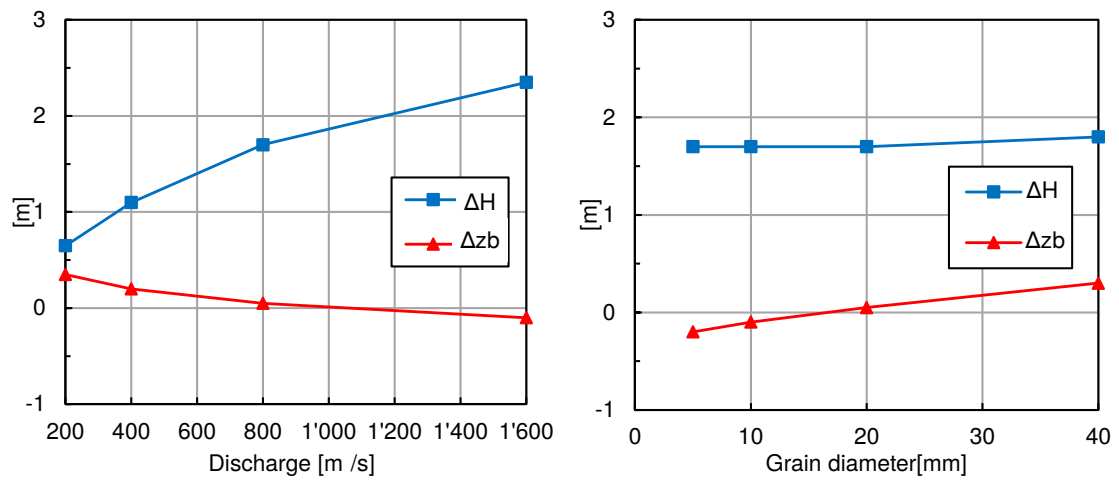
Figure 4.11 *Bottom offset ΔH and upstream displacement Δz_b for different discharges Q (left) and mean grain size diameter d (right) parameter variation according to table 4.2.*

5

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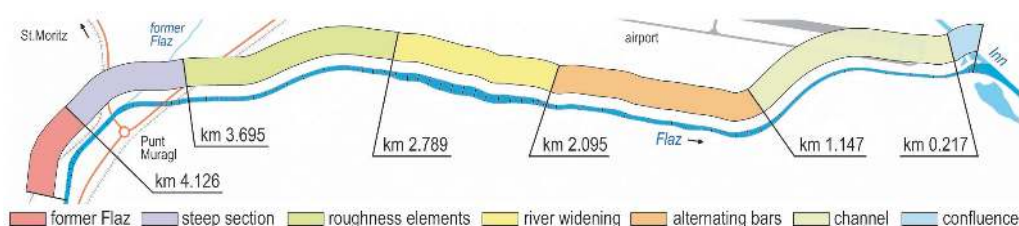
Figure 5.1 *Different morphological river subsections of the new section of the river Flaz*

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.
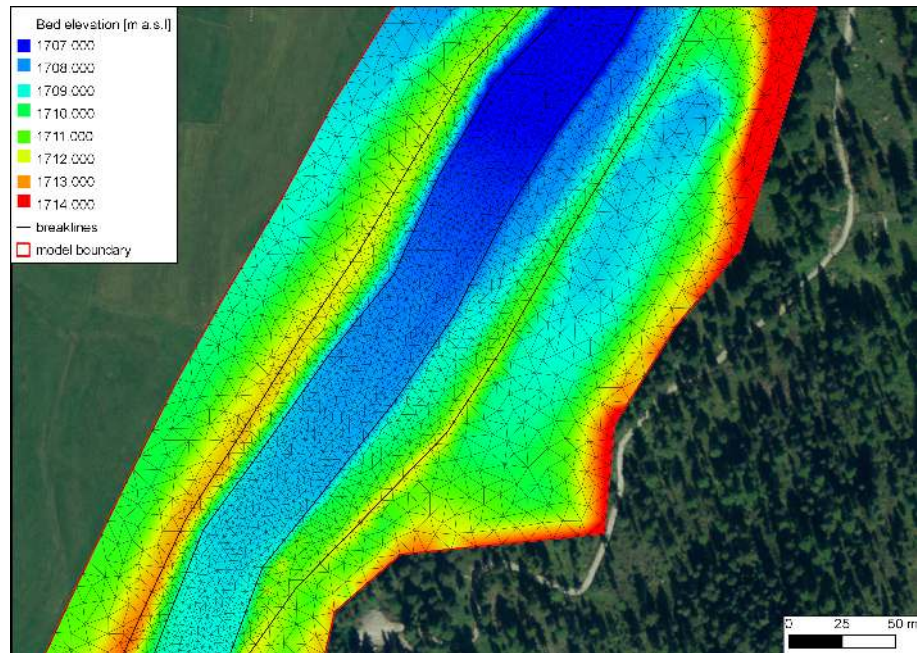
Figure 5.2 *Computational mesh generated with BASEmesh for a section in the widening part. The bold black lines are breaklines for the mesh.*
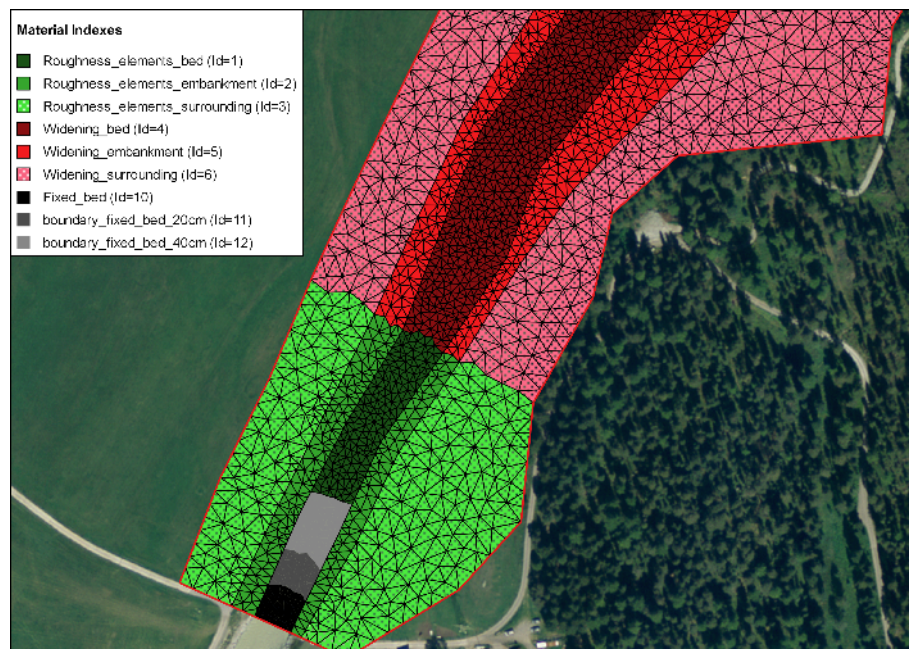
Figure 5.3 *Material indexes (Ids) used for assignment the friction factor and the soil properties.*

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
```

```
    {...}
    PHYISICAL_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
    {...}
}
```
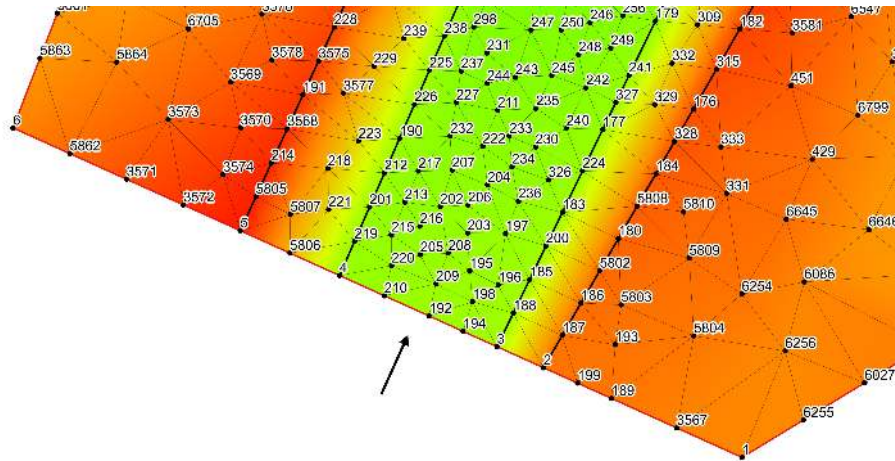
Figure 5.4 *Node id numbers for the definition in the STRING_DEF block for the inflow boundary.*

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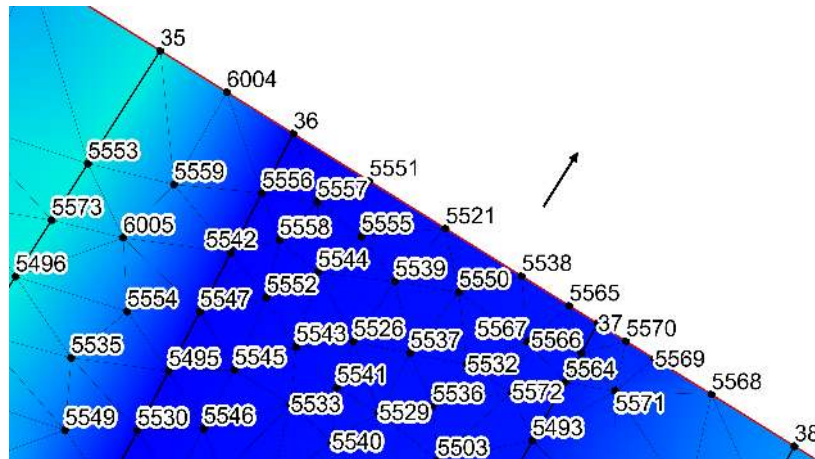
Figure 5.5 *Node id numbers for the definition in the STRING_DEF block for the outflow boundary.*

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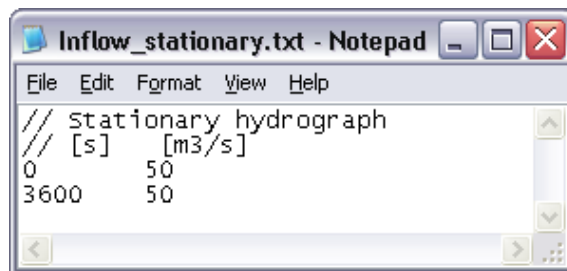
Figure 5.6 *Stationary hydrograph file saved as Inflow_stationary.txt.*

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 m³/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.

```
FRICTION
{
    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
}
}
```
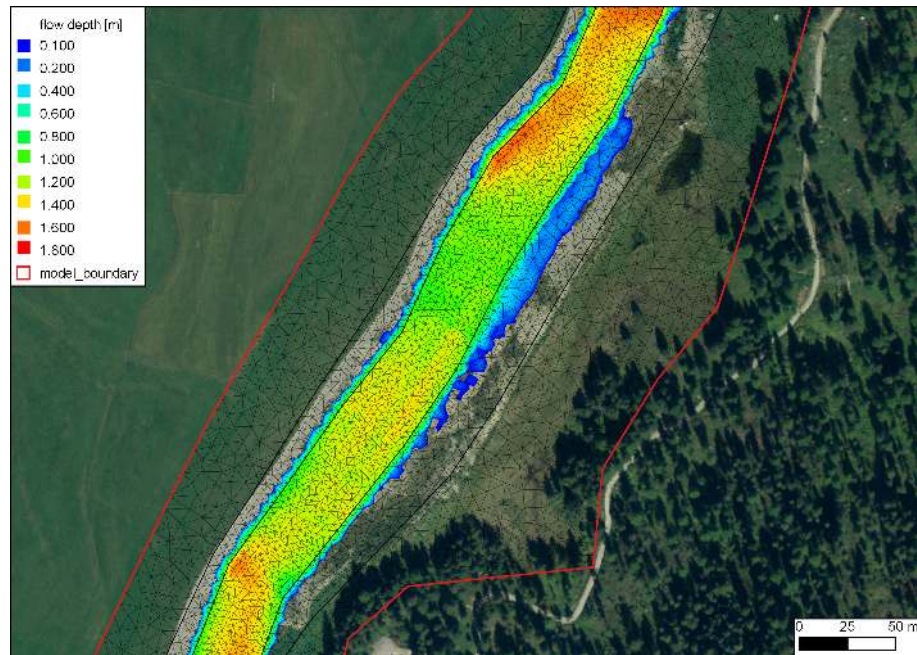
5.4 Perform hydraulic simulation

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
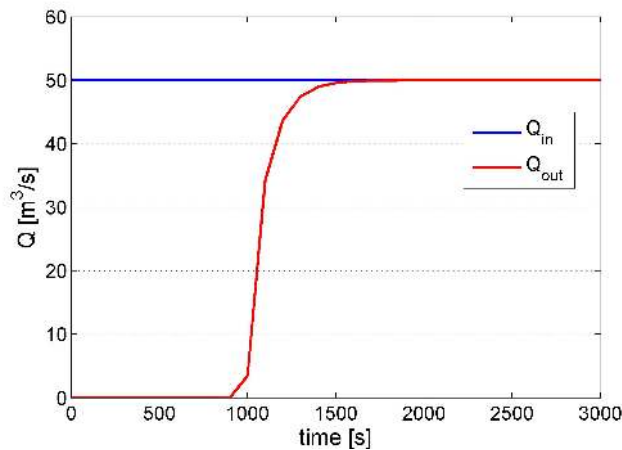
Figure 5.7 *Steady inflow hydrograph and outflow hydrograph.*

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
```
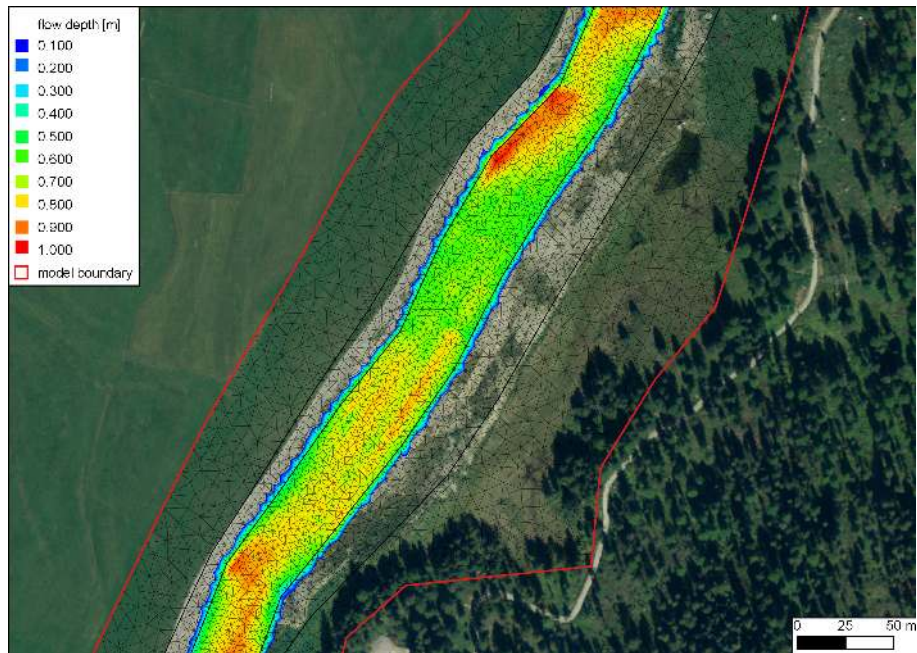
Figure 5.8 *Flow depth at the steady state of the model.*
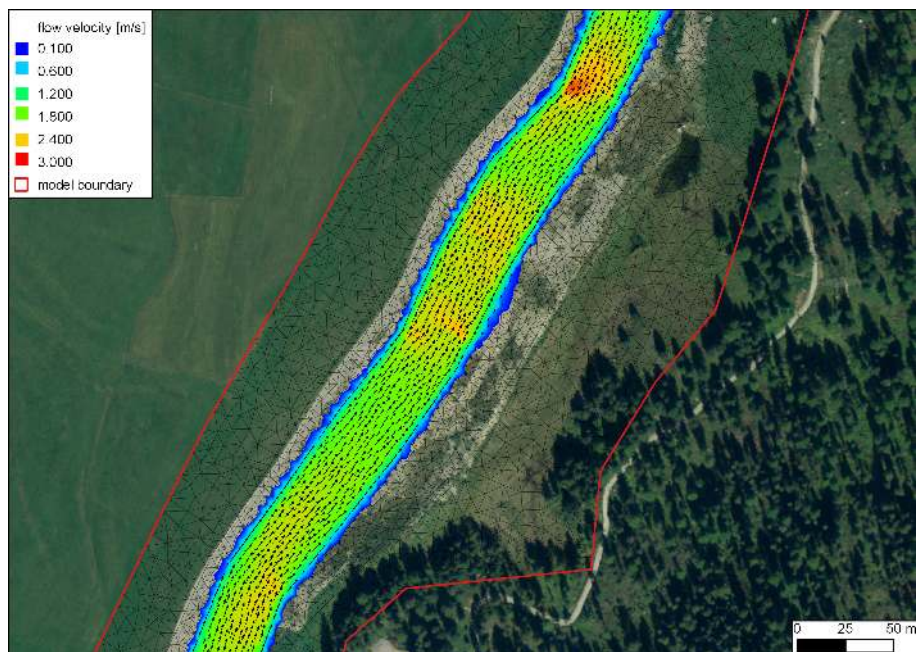
Figure 5.9 *Flow velocity and velocity vectors at the steady state of the model.*
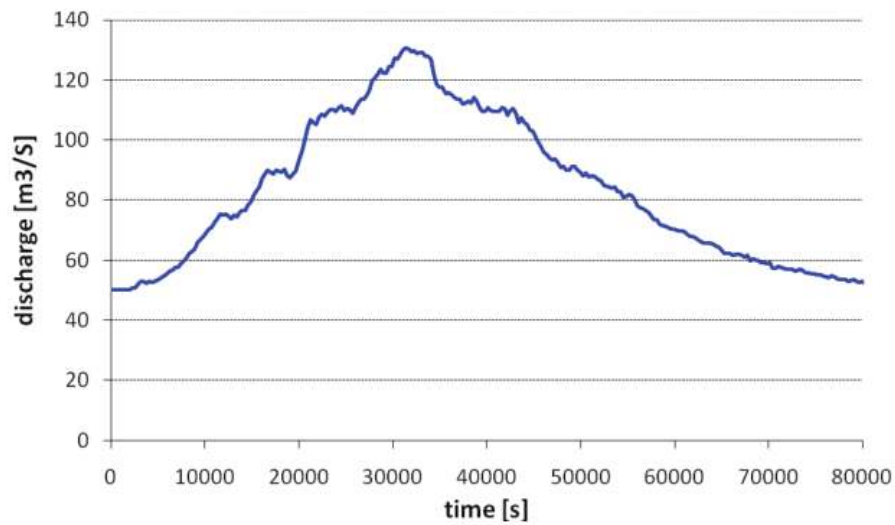
Figure 5.10 *Hydrograph of the flood event of July 2004.*

```
}

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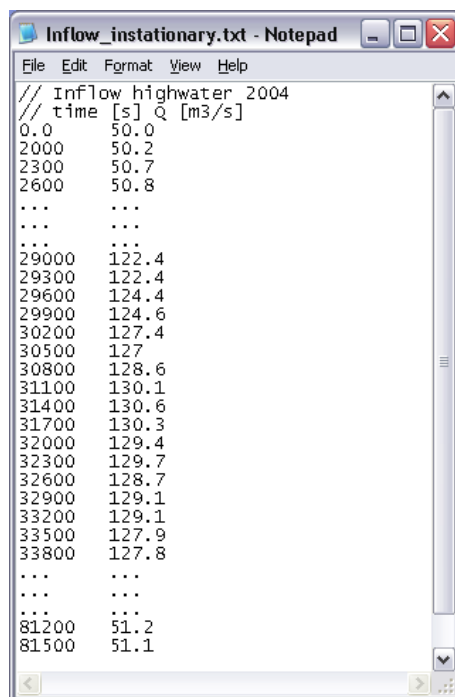
Figure 5.11 *Inflow hydrograph stored in the file Inflow_transient.txt. Note that the points (. . .) are just illustrative in order to show the first and last line of the file.*

```
...
}
```

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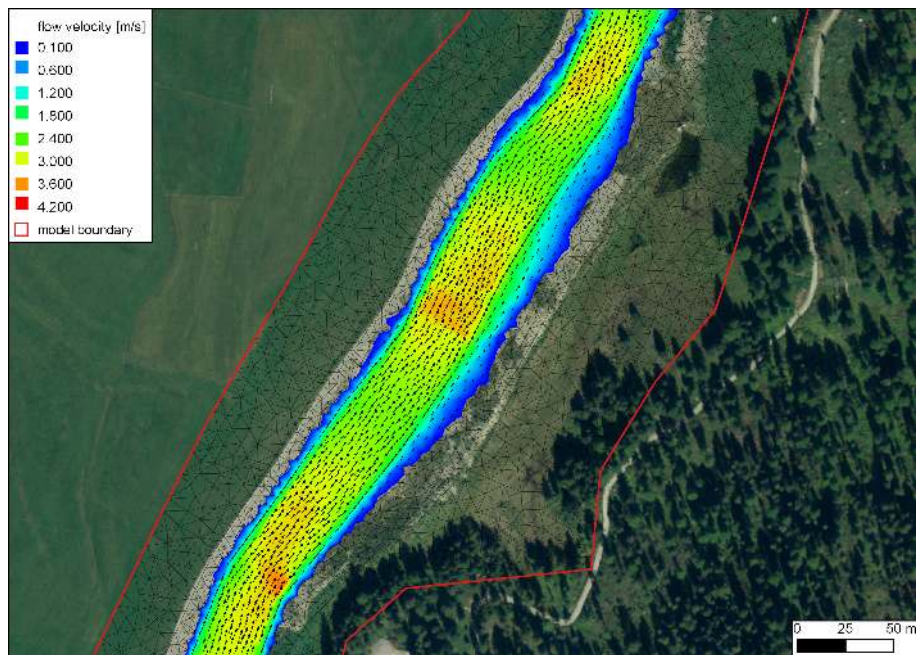
Figure 5.13 *Maximal flow velocity and velocity vectors of the unsteady flow simulation oberserved at the flood peak after a run time of about 9 hours.*

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)
}
```

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.
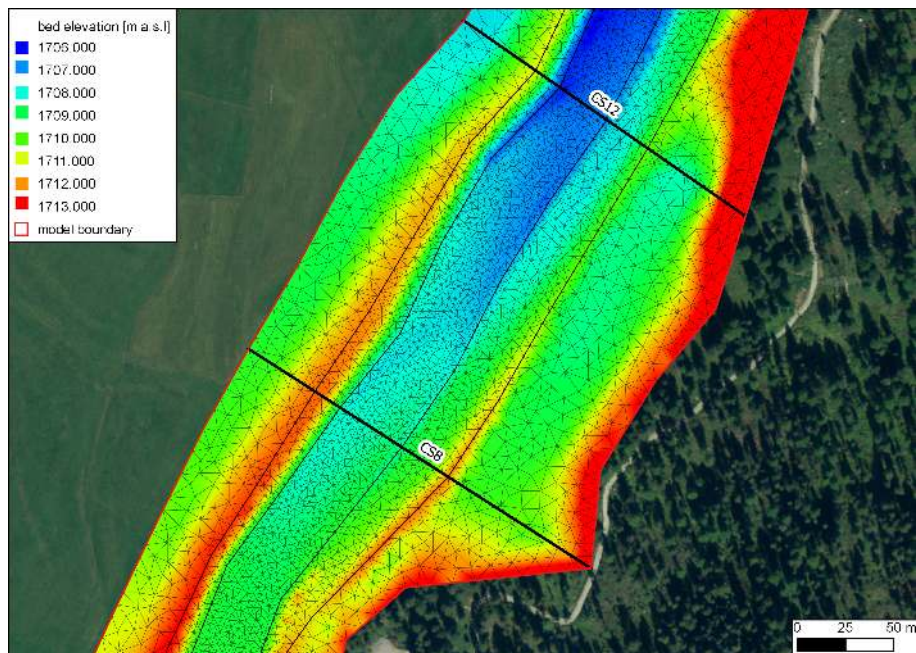
Figure 5.14 *Modeled bed elevation (z_bed) and two cross sections defined in the widening part.*
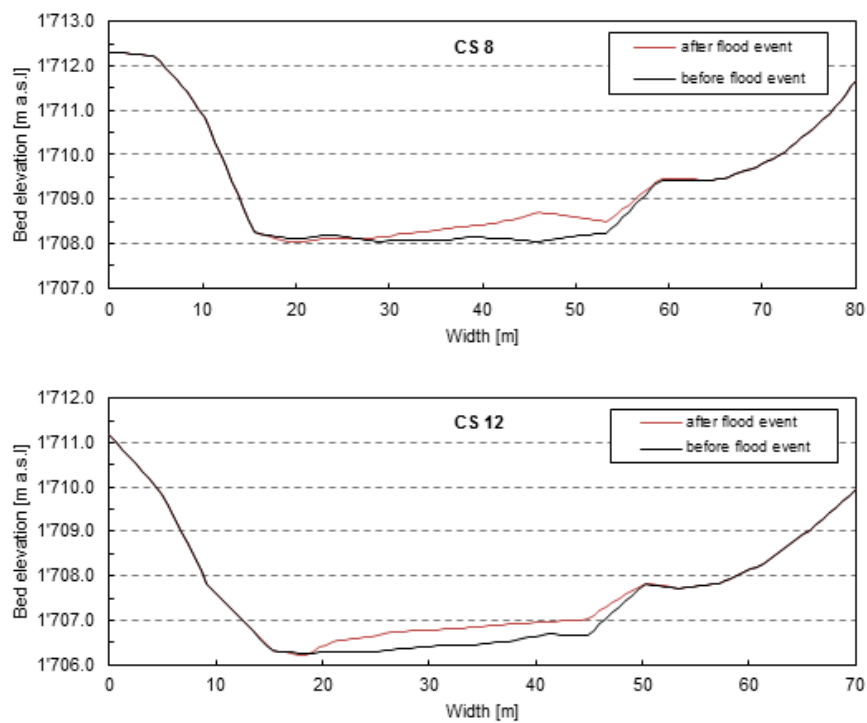
Figure 5.15 *Comparison of the river bed before and after the flood event in cross-section CS 8 and CS 12.*
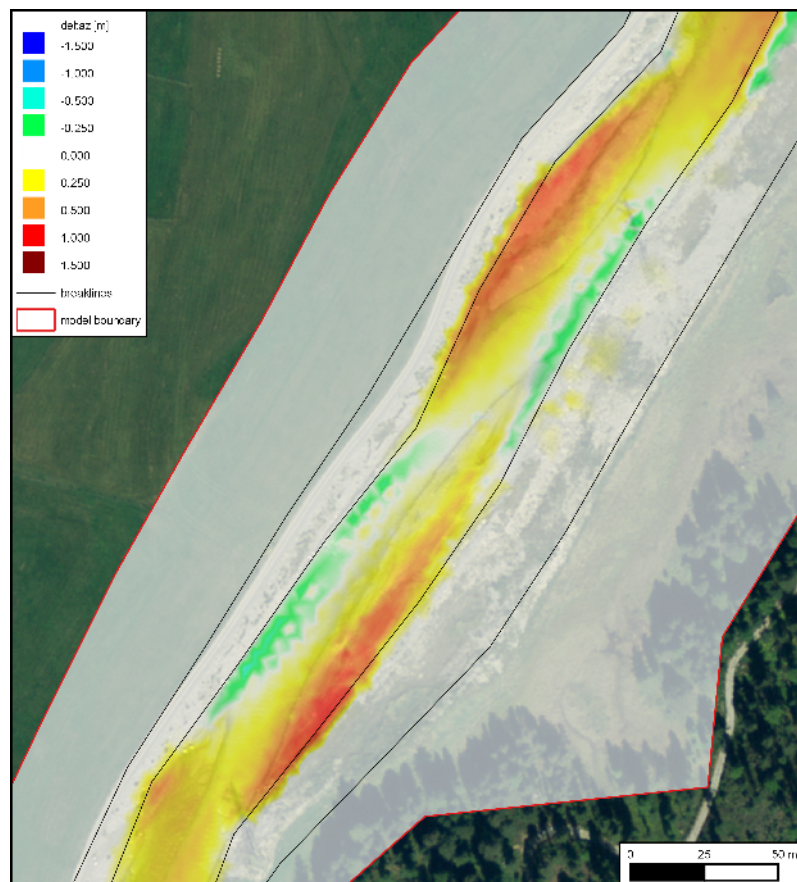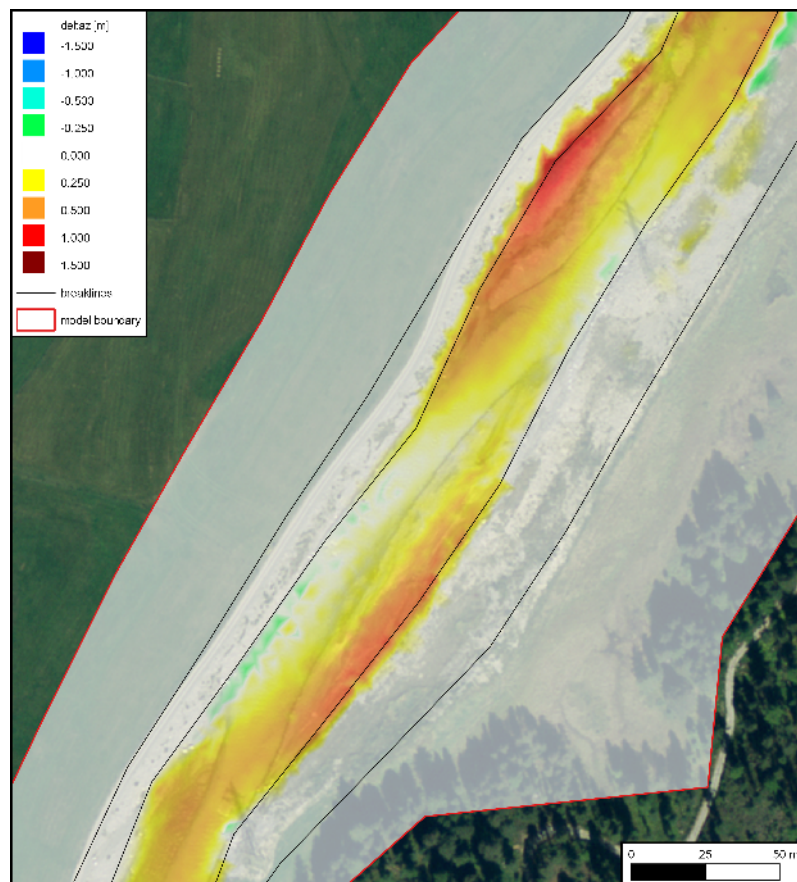
Figure 5.16 *Changes of the morphology (deltaz) due to the flood event with the single-grain model. The red colour range represents deposition and the green/blue colour range shows erosion.*

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*

6

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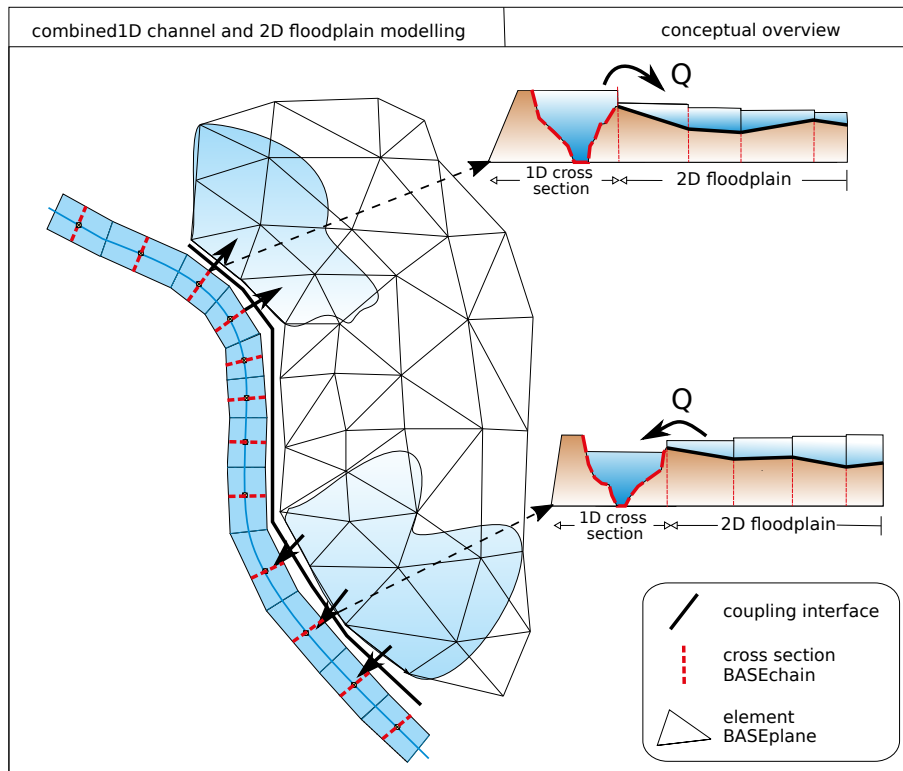
Figure 6.1 *Conceptual sketch of lateral coupling of 1D BASEchain (river channel) and 2D BASEplane (overland flow) sub-domains. Water is exchanged as weir overflow over the river dykes.*

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 tuturial 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.
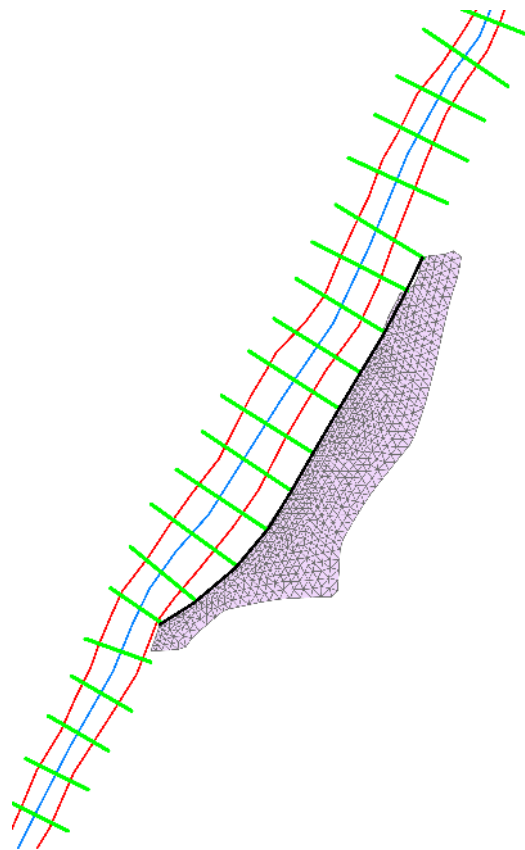
Figure 6.2 *1D cross-sections (green lines), 2D mesh (gray) and dyke break-line (black). Cross-sections and 2D mesh are connected along the river dyke, without gaps or overlapping areas. Flow direction is from bottom left to top right.*
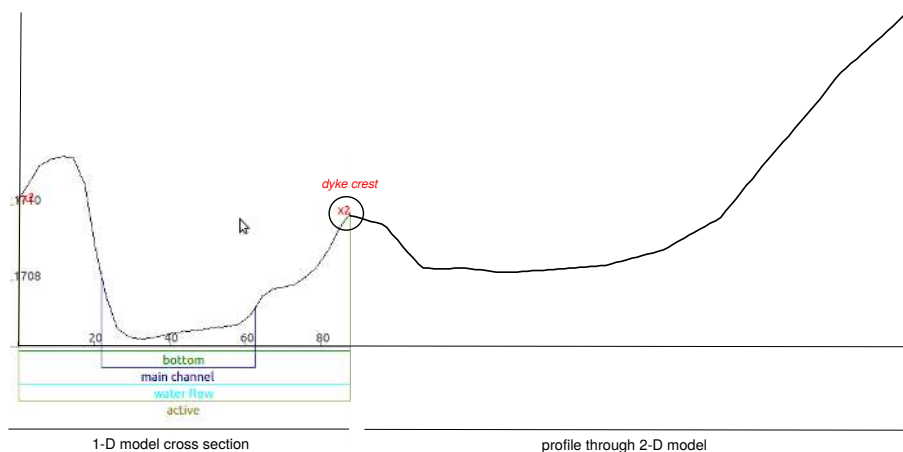
Figure 6.3 *Connection of 1D cross section (left) with 2D model (transversal profile, right) at the dyke location. Water overtopping takes place via the coupling connection.*

```
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 m^3/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.
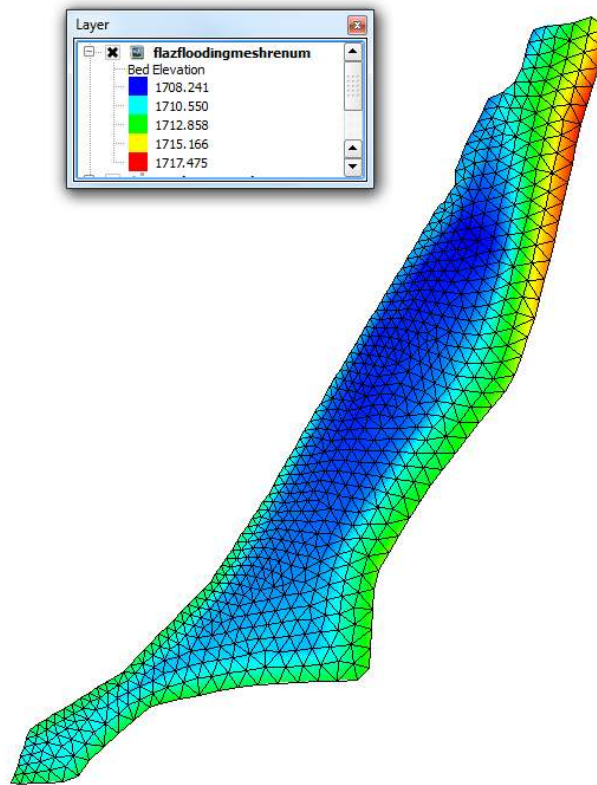
Figure 6.4 *Unstructured 2D mesh of the floodplain located to the right of the 1D river channel.*

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: *conections_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 -> 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-deliminated 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 → BASEmesh → Import mesh*),
3. Load the 2D floodplain mesh file, .2dm (*Plugins → BASEmesh → Import mesh*),
4. Load the generated shapefile with the connection data, *lateral_connection_stats.shp* (*Layer → 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
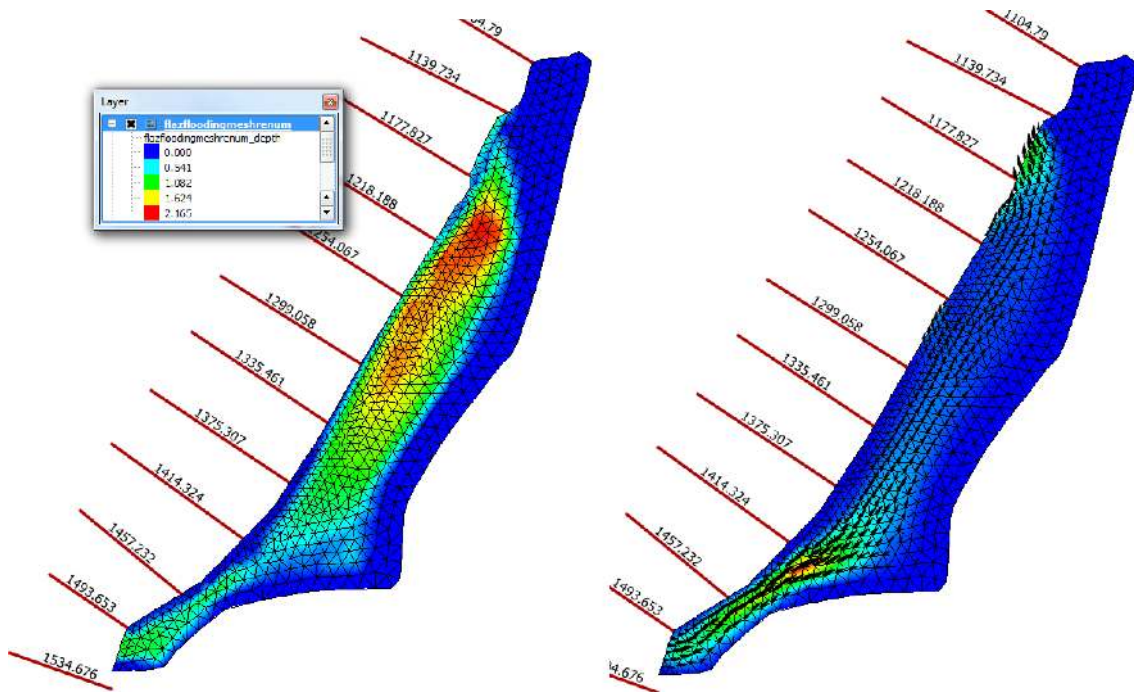
Figure 6.6 *Computed water depths (left) and velocity vectors (right) in the floodplain at steady-state conditions. Water enters the floodplain at the lower left and leaves it at the upper right.*

neighboured 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.

TEST CASES

of *BASEMENT*

Contents

Hydrodynamics

1.1 Introduction

The test catalogue is intended for the validation of the program. The catalogue consists of different test cases, their geometric and hydraulic fundamentals and the reference data for the comparison with the computed results. The test cases are built up on each other going from easy to more and more complex problems. The hydraulic solver is validated against analytical and experimental solutions using the test cases in Section 1.2. Common test cases are suitable for both, 1-D and 2-D simulations. Specific test cases are intended for either 1-D or 2-D simulations.

1.2 Common Test Cases

1.2.1 H_1: Dam break in a closed channel

1.2.1.1 Intention

The simplest test case check verifies the conservation property of the fluid phase and whether a stable condition can be reached for all control volumes after a finite time. Additionally, wet and dry problematic is tested.

1.2.1.2 Description

Consider a rectangular flume without sediment. The basin is initially filled in the half length with a certain water depth leaving the other side dry. After starting the simulation, the water flows in the other half of the system until it reaches a quiescent state (velocity = 0, half of initial water surface elevation (WSE) everywhere).
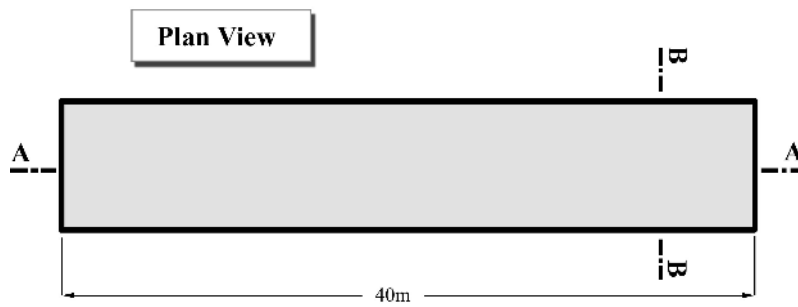
Figure 1.1 *H_1: Dam break in a closed channel, plan view*
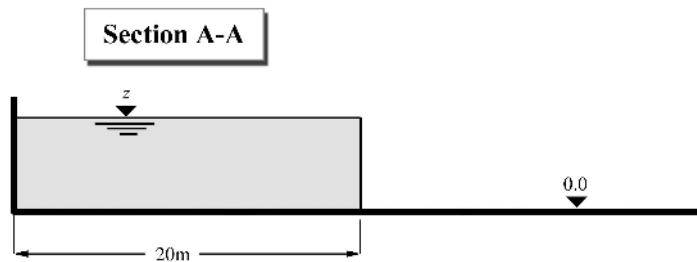
Figure 1.2 *H_1: Dam break in a closed channel, section A-A*

1.2.1.3 Geometry and Initial Conditions

Initial Condition: water at rest (all velocities zero), the water surface elevation is the same on the half of channel. This test uses a WSE of $z_S = 5.0$ m.

1.2.1.4 Boundary Conditions

No inflow and outflow. All of the boundaries are considered as a wall.

Friction: Manning's factor $n = 0.010$

1.2.2 H_2: One dimensional dam break on planar bed

1.2.2.1 Intention

This test case is similar to the previous one but this one compares the WSE a short time after release of the dam with an analytical solution for this frictionless problem. It is based on the verification of a 1-D transcritical flow model in channels by Tseng (1999).
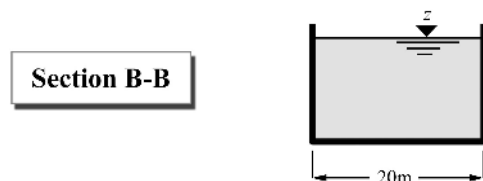
Figure 1.3 *H_1: Dam break in a closed channel, section B-B*
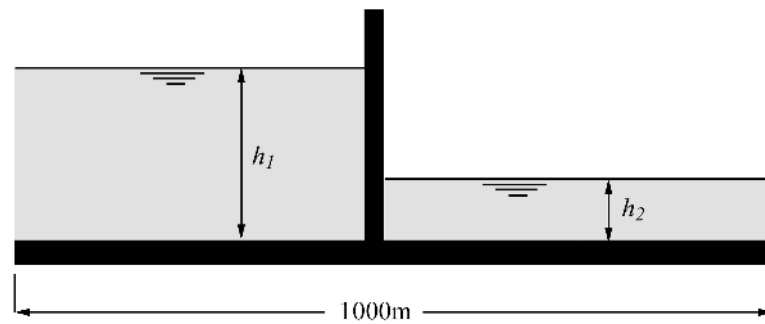
Figure 1.4 *H_2: One dimensional dam break on planar bed*

1.2.2.2   Description

A planar, frictionless rectangular flume has initially a dam in the middle. On one side, the water level is h_2 , on the other side, the water level is at h_1. At the beginning, the dam is removed and a wave is propagating towards the shallow water. The shape of the wave depends on the fraction of the water depths (h_2/h_1) and can be compared to an analytical, exact solution.

1.2.2.3   Geometry and Initial Conditions

Two cases are considered:

- $h_2/h_1 = 0.001$; $h_1 = 10.0$ m (this avoids problems with wet/dry areas)
- $h_2/h_1 = 0.000$; $h_1 = 10.0$ m (downstream is initially dry)

The dam is exactly in the middle of the channel.

Since it is a one dimensional problem, it should not depend on the width of the model, which can be chosen arbitrary.

1.2.2.4   Boundary Conditions

All cases are frictionless, Manning factor n=0.

1.2.3   H_3: One dimensional dam break on sloped bed

1.2.3.1   Intention

Different from the previous cases, this problem uses a sloped bed with friction. The aim is to reproduce an accurate wave front in time. It is based on the verification of a 1-D transcritical flow model in channels by Tseng (1999).
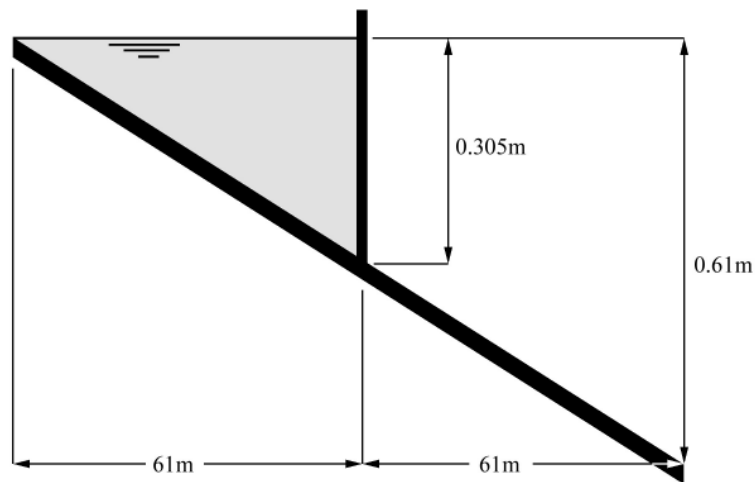
Figure 1.5 *H_3: One dimensional dam break on sloped bed*

1.2.3.2 Description

In the middle of a slightly sloped flume, an initial dam is removed at time zero (see Figure 1.5). The wave propagates downwards. Comparison data comes from an experimental work. Friction is accounted for with a Manning roughness factor estimated from the experiment.

1.2.3.3 Geometry and Initial Conditions

Length of flume:	$122\ m$
Width of flume:	$1.22\ m$
Slope:	$0.61\ m/122\ m$
Initial water level:	$0.035\ m$ (at the dam location)

1.2.3.4 Boundary Conditions

Manning friction factor: 0.009

No inflow/outflow. Boundaries are considered as walls.

1.2.4 H_4 : Parallel execution

1.2.4.1 Intention

To test the parallel performance of hydraulic simulations on a multi-core shared memory computer a test case is set up. The aim is to demonstrate the increase in performance (the speedup) when running BASEMENT in parallel with an increasing number of cores.

1.2.4.2 Description

A hydraulic simulation is repeated with varying number of threads on a multi-core shared memory system. The simulation times are measured and compared to the sequential

execution time in order to evaluate the speedups.

1.2.4.3   Geometry, initial conditions and boundary conditions

A simple rectangular channel with constant slope is considered. The channel is discretized with a large number of cross sections (2000) in BASEchain and a large number of elements (16000) in BASEplane. There is an inflow located at the upper boundary of the channel and an outflow at the lower boundary. The inflowing discharge is constant over the time and steady state conditions within the channel are reached.

Large sized scenarios and evenly distributed work loads (steady state conditions) are set up in order to check for the full potential of the parallel implementation.

1.2.5   H_5: Controlled Boundary Condition

1.2.5.1   Intention

To verify and test the correct action of a PID-controller, steering a boundary condition to keep the water level at a certain cross section or element on the predefined level.

1.2.5.2   Description

The calculation includes a simple channel ending with a weir. In 2-D, two parallel channels are given. One channel is terminated with a weir, the other with a gate. The weir height and gate level is controlled during the simulation by the PID controller and is automatically updated to maintain the predefined water level.

For the 1-D simulation, the discharge in steady state is 100 m^3/s. Over 5000 seconds it is increased up to 150 m^3/s. The water level is fixed at QP_A18 to 112.6 m. The calculation also includes a feed forward component, directly translating the deviation of the inflow from steady state into an adaption of the weir height.

In 2-D, the discharge is increased from 300 m^3/s to 450 m^3/s and then lowered to 180 m^3/s before it is again set to 300 m^3/s. For the 2-D simulation, the target level is not constant, but dependent on the current inflow, which is measured upstream.

Inflow (m^3/s)	Water level (m)
150	503.0
300	503.2
450	503.4
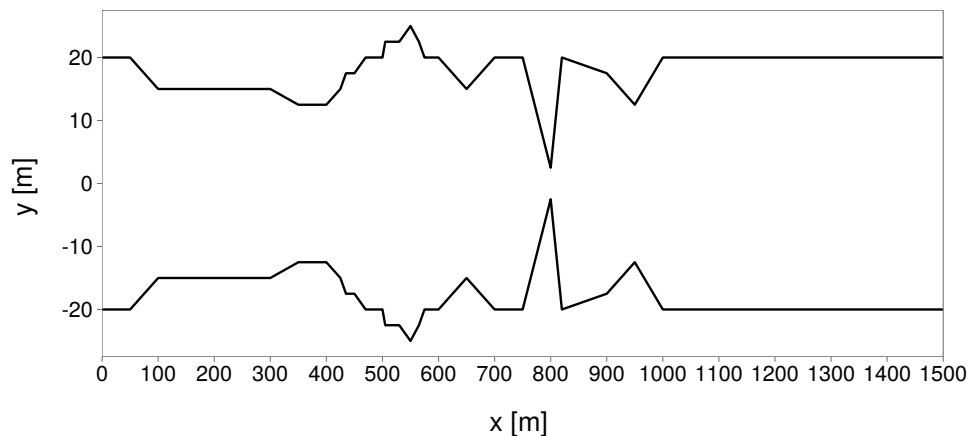
Figure 1.6 *H_BC_1: Top view: cross-sections*

1.3 BASEchain Specific Test Cases

1.3.1 H_BC_1: Fluid at rest in a closed channel with strongly varying geometry

1.3.1.1 Intention

This test case checks for the conservation of momentum. Numerical artefacts (mostly due to geometrical reasons) could generate impulse waves although there is no acting force.

1.3.1.2 Description

The computational area consists of a rectangular channel with varying bed topology and different cross-sections. The fluid is initially at rest with a constant water surface elevation and there is no acting force. There should be no changes of the WSE in time.

1.3.1.3 Geometry and Initial Conditions

The cross sections and bed topology were chosen as in Figure 1.6 and Figure 1.7. Notice the strong variations within the geometry. The water surface elevation is set to $z_S = 12$ m.

1.3.1.4 Boundary Conditions

All boundaries are considered as walls. There is no friction acting.

1.3.2 H_BC_2: bed load simulation with implicit hydraulic solution

1.3.2.1 Intention

The aim of this test case is to verify if the use of the implicit computation mode leads to the expected gain of computational time for a long sediment transport simulation.
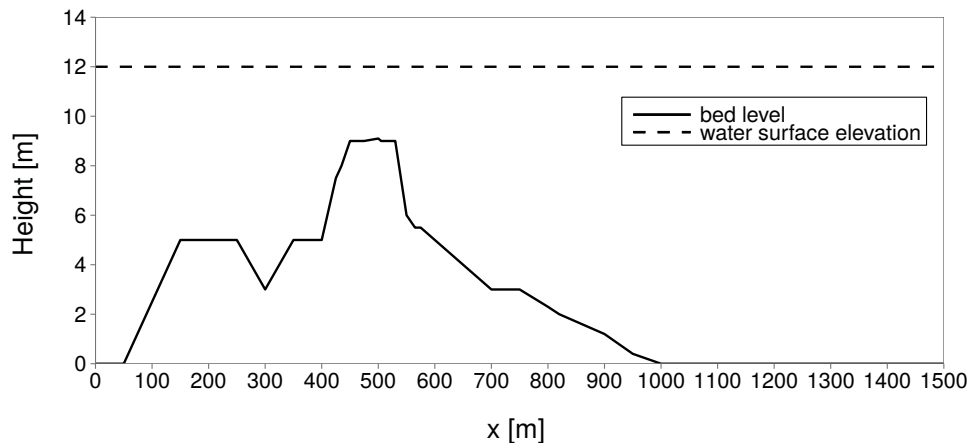
Figure 1.7 *H_BC_1: Bed topography and water surface elevation*

1.3.2.2   Description

The test case simulates a section of the river Thur near Altikon. At this place, there is a widening of which the morphological evolution should be evaluated.

1.3.2.3   Geometry and general data

The model is composed by 55 irregular cross sections, the mean grain size is 2.5 cm and the length of the transport generating hydrograph is 338 hours. The given time steps for the implicit simulation are 60, 120 and 180 seconds. The Program is allowed to reduce the time step if necessary, but this leads to time loss. The precisions of the results for which the iteration is interrupted are 0.1 m^2 for the wetted area and 0.1 m^3/s for the discharge.

1.4   BASEplane Specific Test Cases

1.4.1   H_BP_1: Rest Water in a closed area with strongly varying bottom

1.4.1.1   Intention

This test in two dimensions checks for the conservation of momentum. Numerical artefacts (mostly due to geometrical reasons) could generate impulse waves although there is no acting force.

1.4.1.2   Description

The computational area consists of a rectangular channel with varying bottom topography. The fluid is initially at rest with a constant water surface elevation and there is no acting force. There should be no changes of the WSE in time.
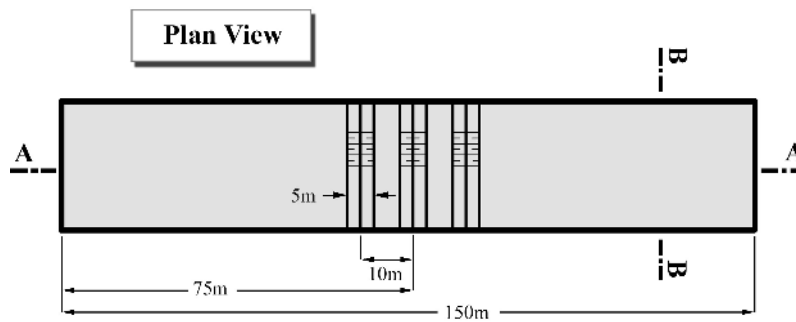
Figure 1.8 *H_BP_1: Rest Water in a closed area with strong varying bottom level, plan view*
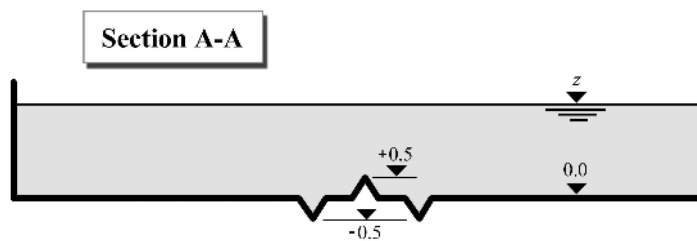
Figure 1.9 *H_BP_1: Rest Water in a closed area with strong varying bottom level, section A-A*

1.4.1.3   Geometry and Initial Conditions

Initial Condition: totally rest water, the WSE is the same and constant over the domain. The WSE can be varied e.g. from z_S (min) = 0.8 to z_S (max) 3.0 m .

1.4.1.4   Boundary Conditions

No inflow and outflow. All of the boundaries are considered as a wall.

Friction: frictionless test; Manning's factor $n = 0.0$.
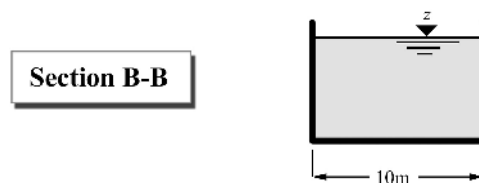
Figure 1.10 *H_BP_1: Rest Water in a closed area with strong varying bottom level, section B-B*
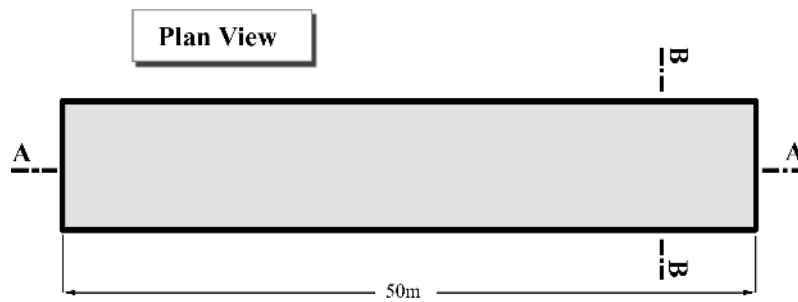
Figure 1.11 *H_BP_2: Rest Water in a closed area with partially wet elements, plan view*
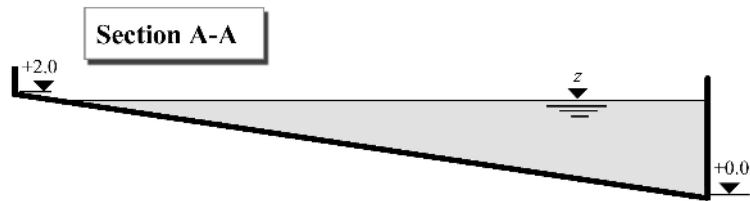
Figure 1.12 *H_BP_2: Rest Water in a closed area with partially wet elements, section A-A*

1.4.2   H_BP_2: Rest Water in a closed area with partially wet elements

1.4.2.1   Intention

This test case checks the behaviour of partially wet control volumes. The wave front respectively the border wet/dry is always the weak point in a hydraulic computation. There should be no momentum due the partially wet elements.

1.4.2.2   Description

In a sloped, rectangular channel, the WSE is chosen to be small enough to allow for partially wet elements. The water is at rest and no acting force is present.

1.4.2.3   Geometry and Initial Conditions

Initial Condition: totally rest water, the WSE is the same and constant over the whole domain. The WSE is $z_S = 1.9$ m.

1.4.2.4   Boundary Conditions

No inflow and outflow. All of the boundaries are considered as a wall.

Friction: frictionless test; Manning's factor $n = 0.0$.

The mesh (grid) can be considered as following pictures. In this form, the partially wet cells (elements) can be handled.
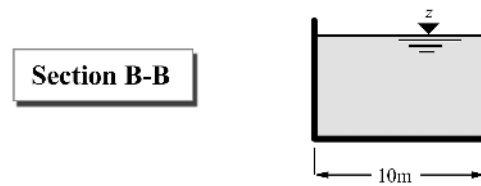
Figure 1.13 *H_BP_2: Rest Water in a closed area with partially wet elements, section B-B*
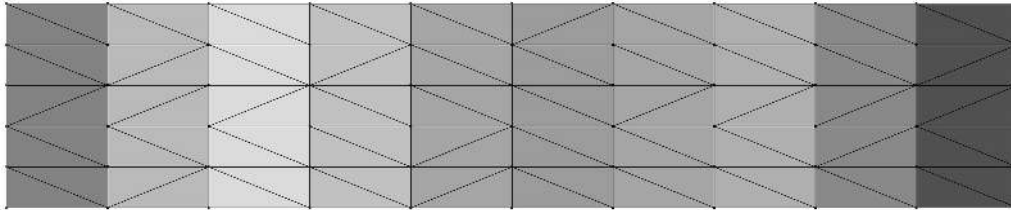
Figure 1.14 *H_BP_2: Computational grid*

1.4.3 H_BP_3: Dam break within strongly bended geometry

1.4.3.1 Intention

This test case challenges the 2-D code. The geometry permits a strongly two dimensional embossed flow. The results are compared against experimental measurements at certain control points.

1.4.3.2 Description

In a two-dimensional geometry with a jump in bed topology, a gate between reservoir and some outflow channel is removed at the beginning. The bended channel is initially dry and has a free outflow discharge as boundary condition at the downstream end.

1.4.3.3 Geometry and Initial Conditions

The geometry is described in Figure 1.16 and Figure 1.15, G1 to G6 are the measurement control points for the water elevation. Initial condition is a fluid at rest with a water surface elevation of 0.2 m above the channels ground. The channel itself is dry from water. At $z = 0$, the gate between reservoir and gate is removed.

1.4.3.4 Boundary Conditions

No inflow. At the downstream end of the channel, a free outflow condition is employed. All other boundaries are considered as walls.

The friction factor is set to 0.0095 (measured under stationary conditions) for the whole computational domain.
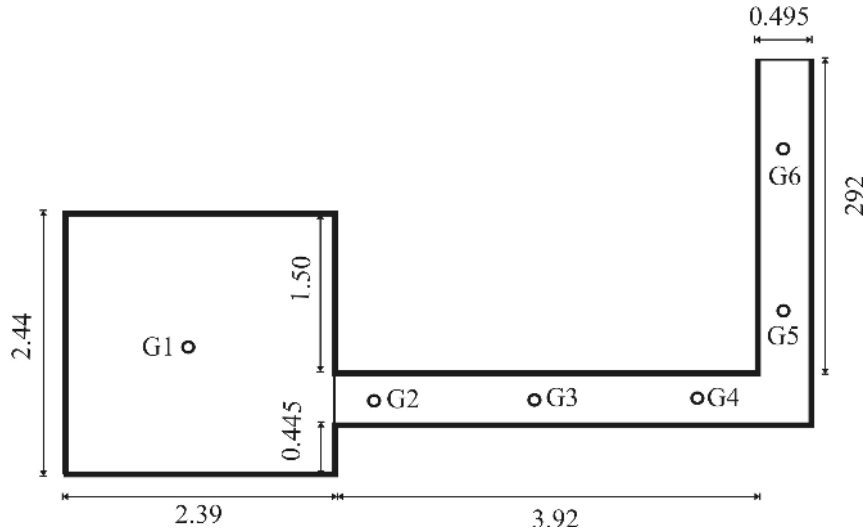
Figure 1.15 *H_BP_3: Strongly bended channel (plane view)*
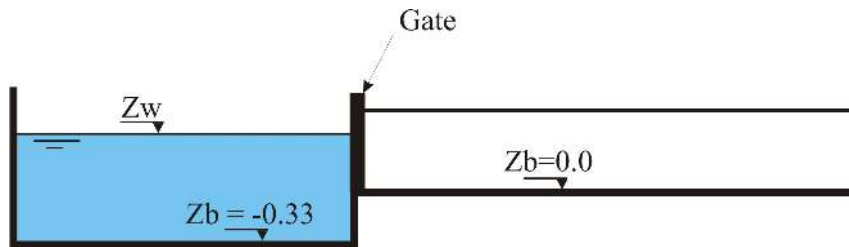
Figure 1.16 *H_BP_3: Strongly bended channel (side view)*

1.4.4   H_BP_4: Malpasset dam break

1.4.4.1   Intention

This final hydraulic test case is the well known real world data set from the Malpasset dam break in France. The complex geometry, high velocities, often and sudden wet-dry changes and the good documentation allow for a fundamental evaluation of the hydraulic code.

1.4.4.2   Description

The Malpasset dam was a doubly-curved equal angle arch type with variable radius. It breached on December 2nd, 1959 all of a sudden. The entire wall collapsed nearly completely what makes this event unique. The breach created a water flood wall 40 meters high and moving at 70 km/h. After 20 minutes, the flood reached the village Frejus and still had 3 m depth. The time of the breach and the flood wave can be exactly reconstructed, as the time is known, when the power of different stations switched off.

1.4.4.3   Geometry and Initial Conditions

Figure 1.18 shows the computational grid used for the simulation. The points represent "measurement" stations. The initial water surface elevation in the storage lake is set to +100.0 m.a.s.l. and in the downstream lake to 0.0 m.a.s.l. The area downstream of the wall is initially dry. At t=0, the dam is removed.
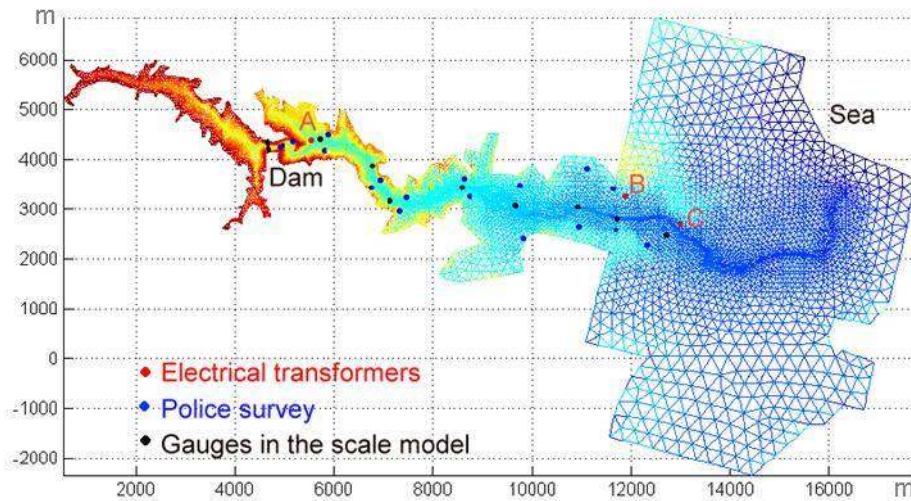
Figure 1.17 *H_BP_4: Computational area of the malpasset dam break*

1.4.4.4 Boundary Conditions

The computational grid was constructed to be large enough for the water to stay within the bounds. Friction is accounted for by a Manning factor of 0.033.

1.4.5 H_BP_5: Circular dam break wave

1.4.5.1 Intention

The circular dam break problem is a demanding two-dimensional test case with some distinct features. It is used to evaluate the capability to correctly model complex interactions of shock and rarefaction waves. The results are qualitatively compared to results obtained by Toro (2001), and other numerical studies, for this idealized dam break scenario. Also the results of the exact and approximate Riemann solvers are compared to each other.

1.4.5.2 Description

A virtual circular dam is located in the center of a computational domain. At the beginning, at time t = 0.0s, the dam is removed. The evolution of subsequent wave patterns is examined until about t = 5s after the dam break.

1.4.5.3 Geometry and Initial Conditions

The computational domain has a width and height of 40m and is modelled with a uniform rectangular grid, which consists of 160'000 quadratic elements of the size 0.1m x 0.1m. The grid is selected large enough to fully capture the outward propagating primary shock wave during the simulation time of about 5s. The small element sizes shall enable the modelling of the circular flow patterns with sufficient accuracy.
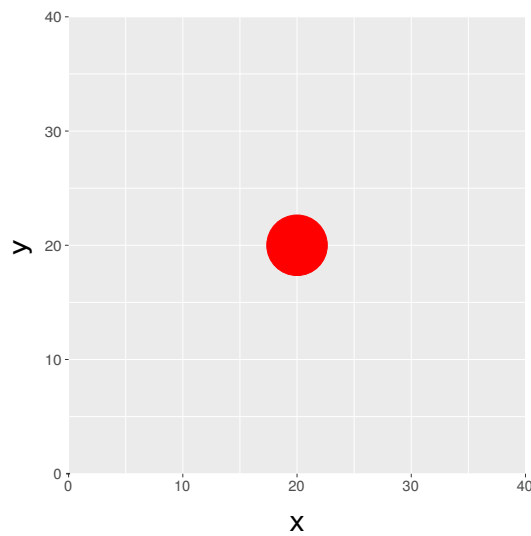
Figure 1.18 *H_BP_5: Computational area (rectangular grid) of circular dam break*

The initial height of the water column is 2.5m, whereas the surrounding initial water surface level is set to 0.5m. The water column behind the circular dam has a radius of 2.5m. The time step is chosen according to a CFL number of 0.9.

1.4.5.4 Boundary Conditions

Friction: frictionless test; Manning's factor $n = 0.0$.

1.5 Results

1.5.1 Common Test Cases

1.5.1.1 H_1: Dam break in a closed channel

The conservation of mass is validated. After $t = 1107$ s, the discharges still have a magnitude of 10^{-6} m^3/s with decreasing tendency. Both, BASEchain and BASEplane deliver similar results.

1.5.1.2 H_2: One dimensional dambreak on planar bed

1.5.1.2.1 Results obtained by BASEchain

A comparison of the 1-D test case using $h_2/h_1 = 0.000$ with the analytical solution shows an accurate behaviour of the fluid phase in time.

1.5.1.2.2 Results obtained by BASEplane

In two dimension, the dam break test case was computed using $h_2/h_1 = 0.0$ and $h_1 = 10$ m. The length of the channel was discretized using 1000 control volumes. The CFL number
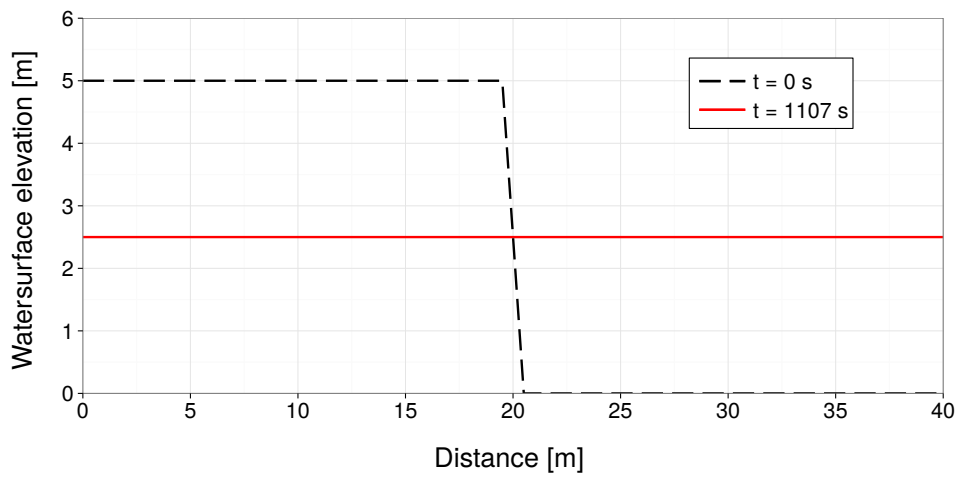
Figure 1.19 *H_1: Initial and final water surface elevation (BC & BP)*
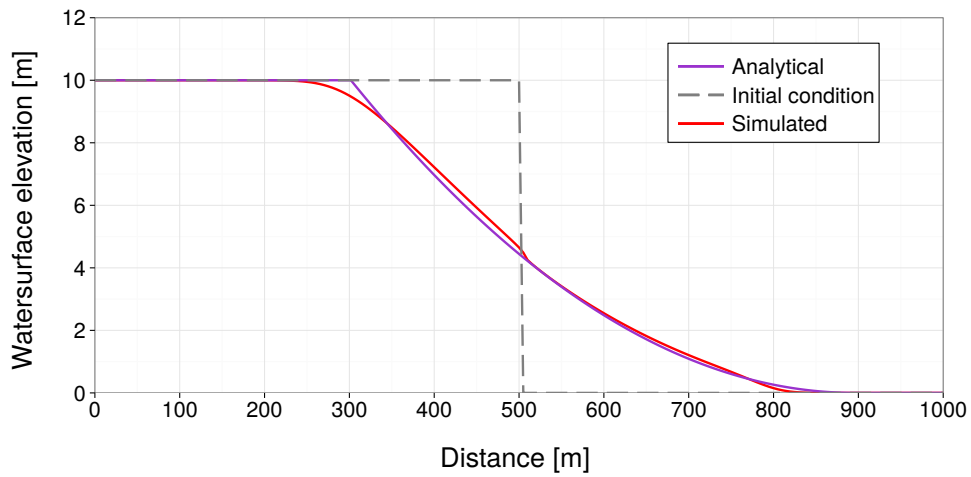
Figure 1.20 *H_2: Water surface elevation after 20 s (BC)*
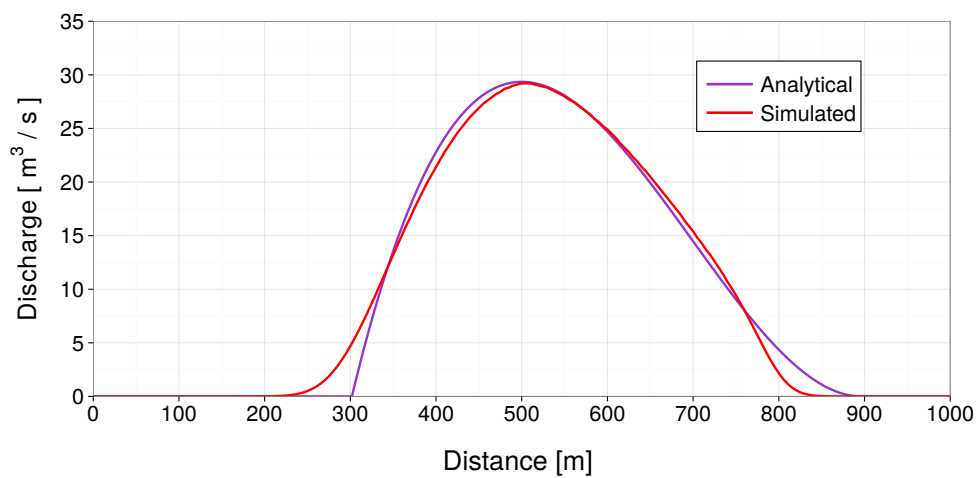
Figure 1.21 *H_2: Distribution of discharge after 20 s (BC)*
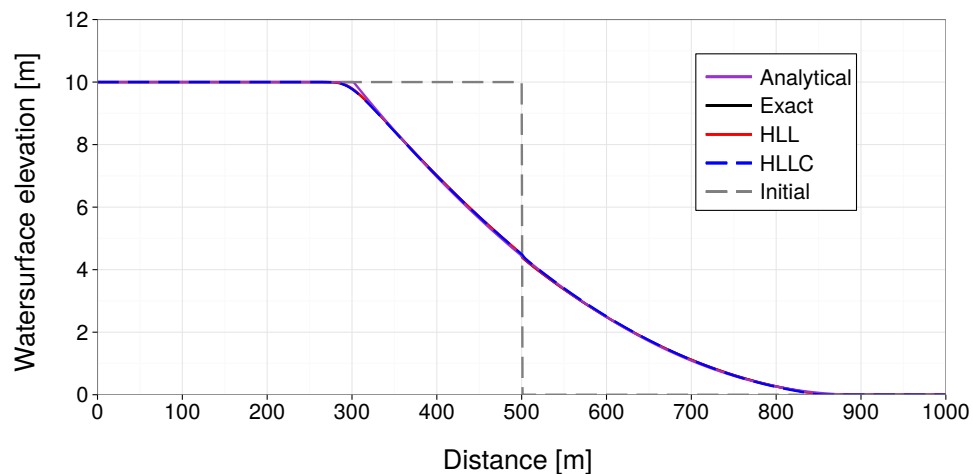
Figure 1.22 H_2: Flow depth 20 sec after dam break (BP)
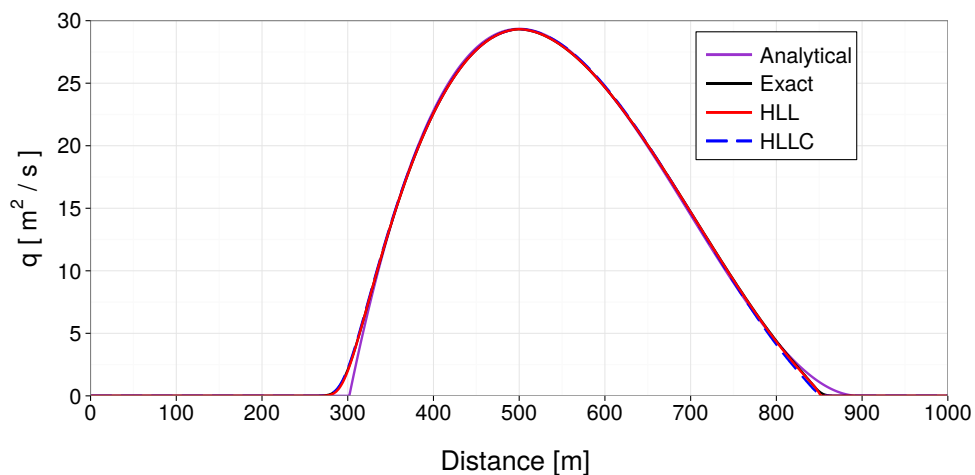
Figure 1.23 H_2: Discharge 20 sec after dam break (BP)

was set to 0.85. The results show a good agreement between simulation and analytical solution.

1.5.1.3   H_3: One dimensional dam break on sloped bed

1.5.1.3.1   Results obtained by BASEchain

The dam break in a sloped bed was compared against experimental results. The agreement is satisfying.

1.5.1.3.2   Results obtained by BASEplane

For two dimensions, the numerical results show a good agreement with the experimental measurements. Although in certain areas, the water depth is underestimated, the error is just at 7 % and only for a limited time. The comparison is still successful.
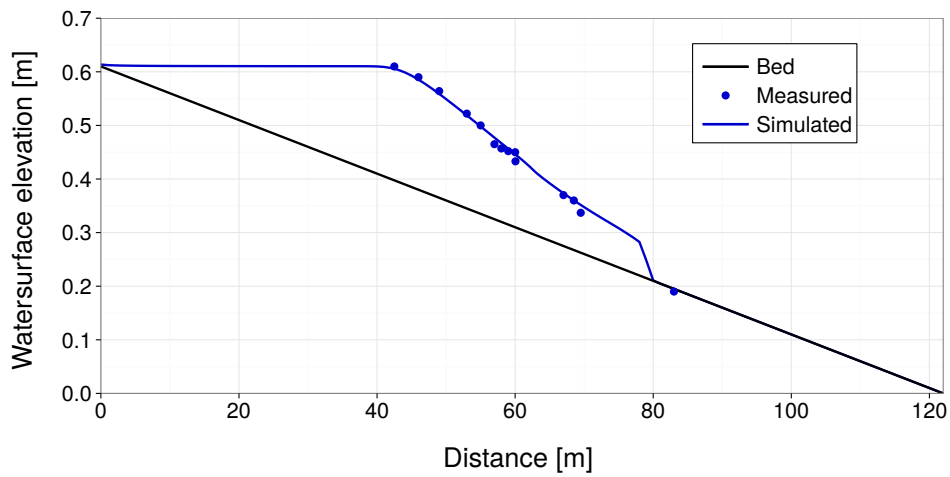
Figure 1.24 *H_3: Water surface elevation after 10 s (BC)*
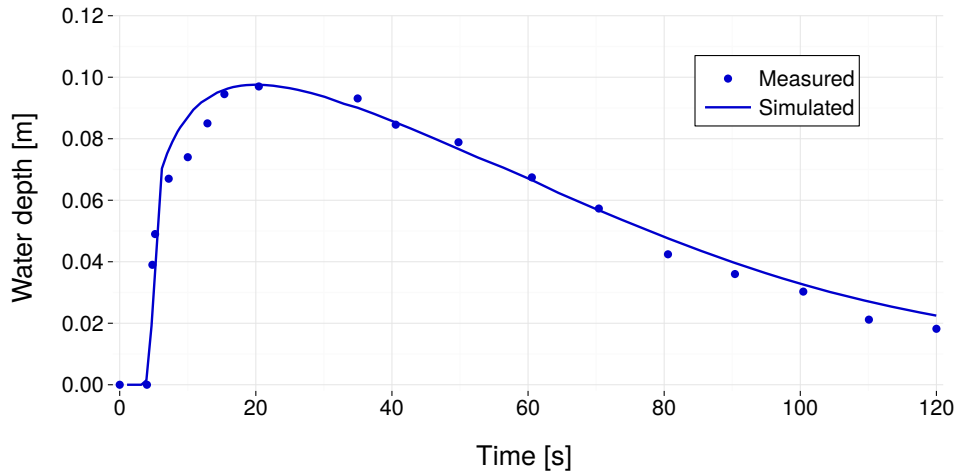
Figure 1.25 *H_3: Temporal behaviour of water depth at x = 70.1 m (BC)*
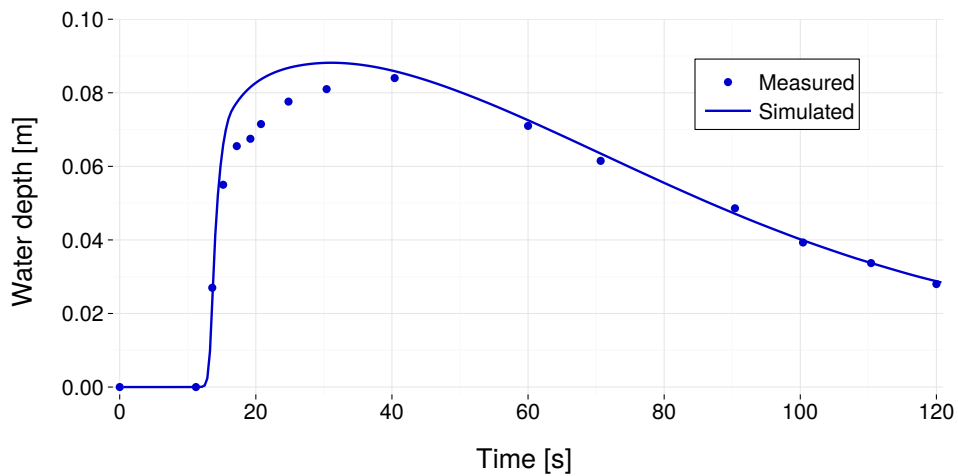
Figure 1.26 *H_3: Temporal behavior of water depth at x = 85.4 m (BC)*
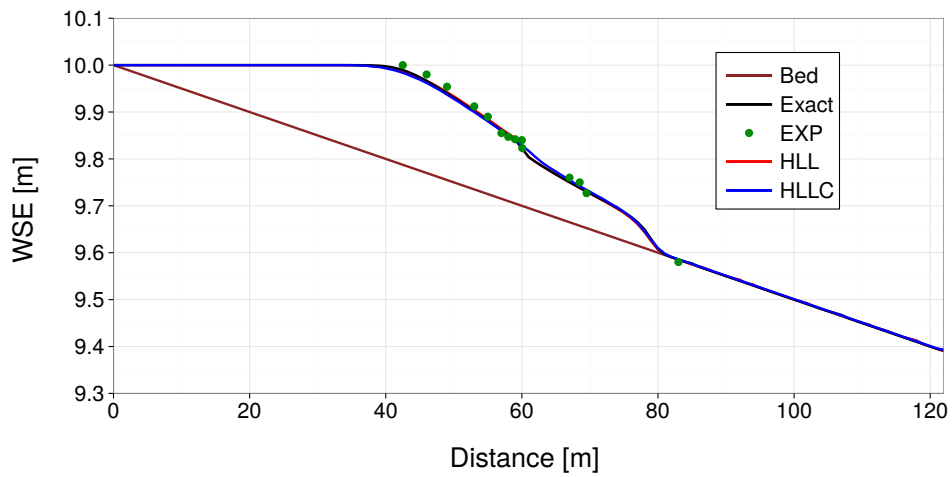
Figure 1.27 *H_3: Cross sectional water level after 10 s simulation time (BP)*
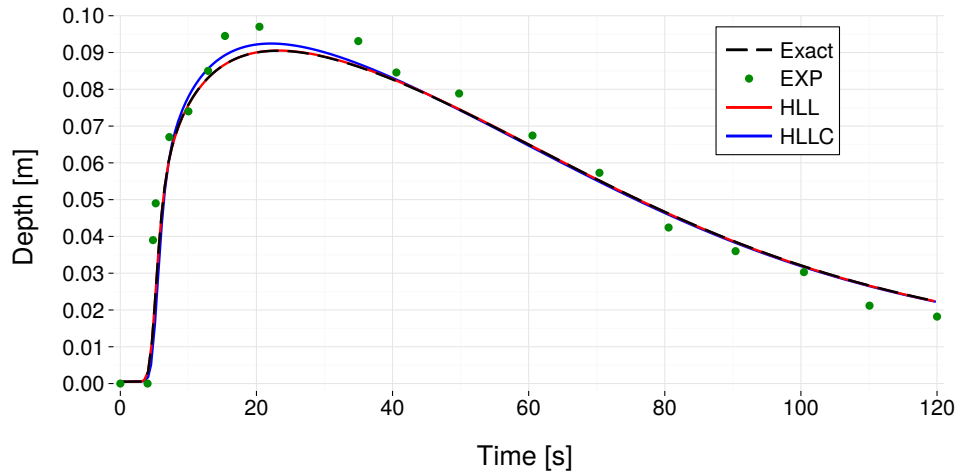
Figure 1.28 *H_3: Chronological sequence of water depth at point x = 70.1 m (BP)*
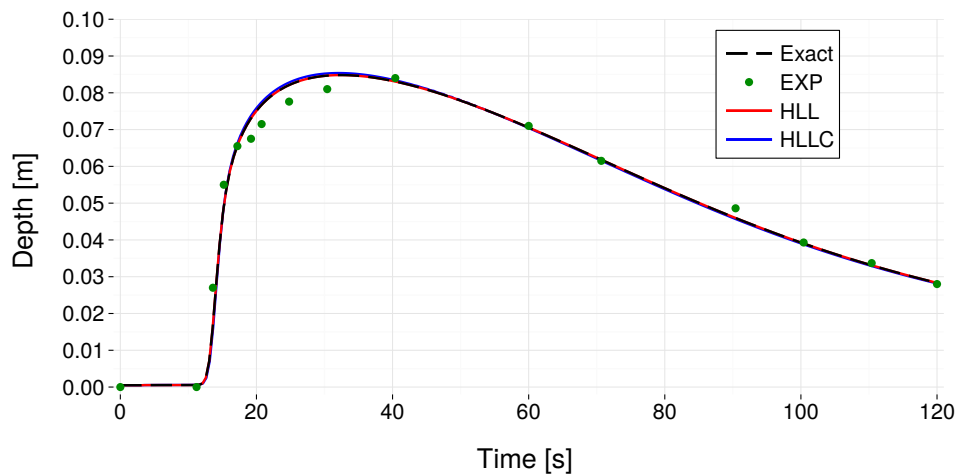
Figure 1.29 *H_3: Chronological sequence of water depth at point x = 85.4 m (BP)*
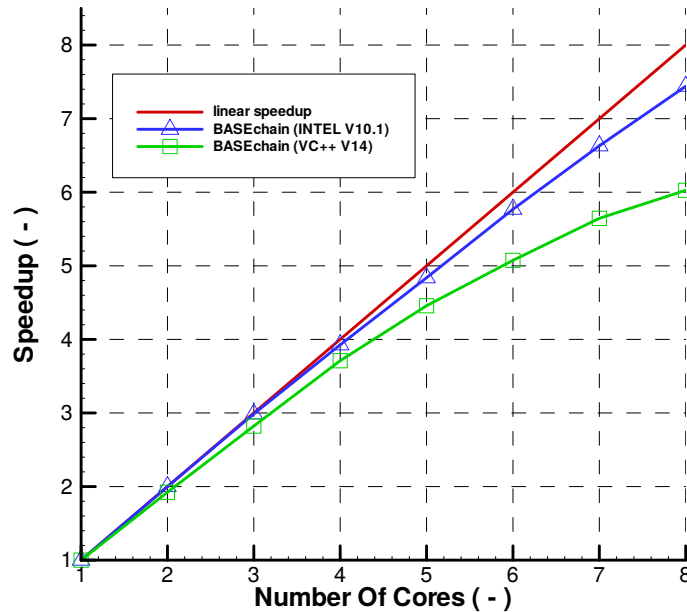
Figure 1.30 *H_4 (BASEchain) – Parallel speedup for 1-D test case*

1.5.1.4 H_4 : Parallel execution

1.5.1.4.1 General

Hydraulic simulations in a rectangular channel with steady state conditions are repeated multiple times and the execution times are measured. Thereby only the number of used threads is varied from one simulation to another. The selected model scenarios are suited well for parallel execution regarding size and load balancing in order to check the full potential of the parallel execution (in many practical model setups the observed speedups may be significantly lower).

1.5.1.4.2 System configuration

The simulations were performed an Intel multi-core shared memory system with 8 cores. The used operating system was WinXP 64.

The simulation procedure is repeated with two different versions of BASEMENT. One version (blue) is compiled with the Intel C++ compiler (V 10.1), whereas the other version (green) is compiled with the Microsoft VC++ compiler (V14).

1.5.1.4.3 Results obtained by BASEchain

The results indicate an excellent speedup and scalability. The speedup increases nearly linear with the number of used cores. The parallel speedups obtained by the Intel compiled binary are superior to those obtained by the VC++ compiled binary.

1.5.1.4.4 Results obtained by BASEplane

The results indicate a satisfactory speedup and scalability. The effect of increasing slope of the blue curve, when running this scenario with 8 cores, is probably due to caching effects. The parallel speedups obtained with the Intel compiled binary are superior to those obtained with the VC++ binary, if the number of cores exceeds 2.
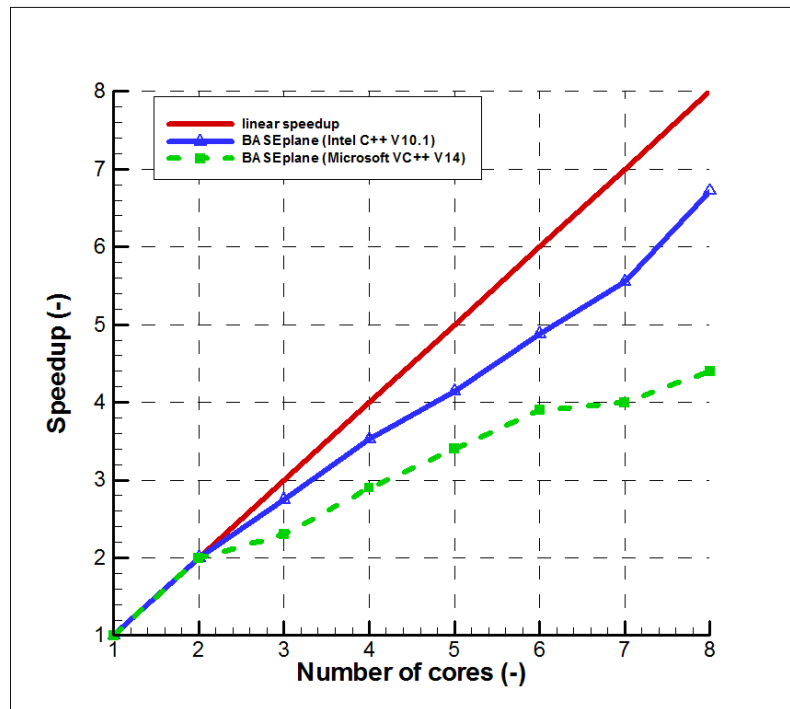
Figure 1.31 *H_4 (BASEplane) – Parallel speedup for 2-D test case*

1.5.1.5   H_5 : Controlled boundary conditions

1.5.1.5.1   Results for BASEchain

As can be seen in Figure 1.32, the controller is able to maintain the constant water level at 112.6 m precisely.

1.5.1.5.2   Results for BASEplane

Compared to the 1-D simulation, the 2-D test case is a more complicated problem as it involves discharge dependent targets and changes of discharge from 450 m^3/s down to 180 m^3/s within 5 minutes. This rapid change makes the strict maintaining of the water level difficult. In principle, this could be handled by a stronger reaction of the weir and gate, which would require also a stronger imposed upper limit of the time step to prevent numerical oscillations of the weir and gate height.

1.5.2   BASEchain Specific Test Cases

1.5.2.1   H_BC_1: Fluid at rest in a closed channel with strongly varying geometry

The test was successfully carried out. There was no movement at all – the water surface elevation remained constant over the whole area. No picture will be shown as there is nothing interesting to see.
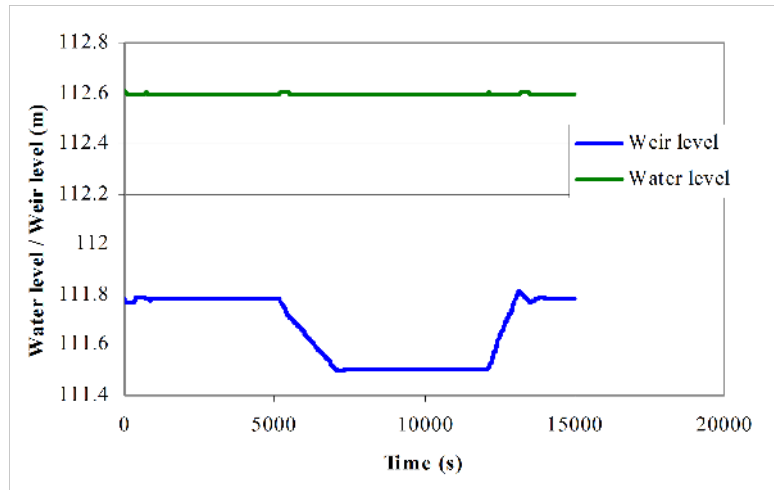
Figure 1.32 *H_5 (BASEchain) Controlled weir with changing inflow.*
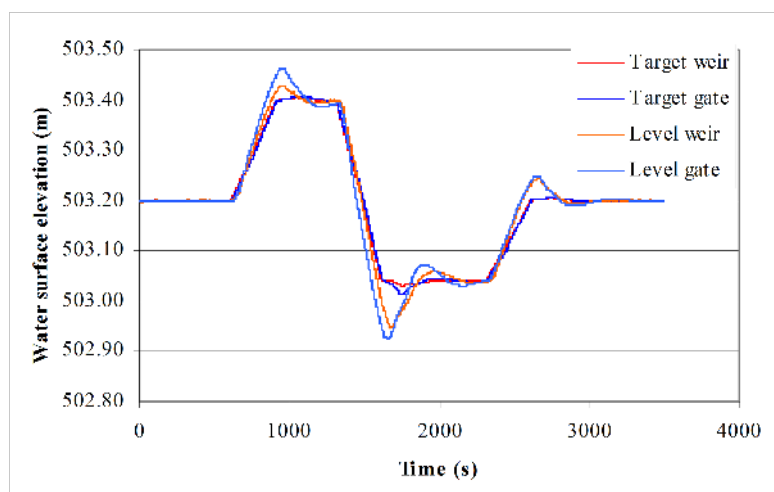
Figure 1.33 *H_5 (BASEplane) Controlled weir and gate. The target levels and the actual levels with changing inflow are given.*

1.5.2.2   H_BC_2: bed load simulation with implicit hydraulic solution

The time steps and simulation times for the explicit and implicit computations are listed in the table below. The reduction of simulation time is satisfying.

Table 1.2 *H_BC_2 : Performance of implicit calculations*

	Time step (seconds)	Simulation time (hours)	Speedup (times faster)
Explicit	2-5	13.94	
Implicit, base time step = 60	3-90	0.9	15
Implicit, base time step = 120	3-120	0.6	23
Implicit, base time step = 180	3-180	0.47	30

1.5.3   BASEplane Specific Test Cases

1.5.3.1   H_BP_1: Rest water in a closed area with strongly varying bottom

The test was successfully carried out. There was no movement at all – the water surface elevation remained constant over the whole area. No picture will be shown as there is nothing interesting to see.

1.5.3.2   H_BP_2: Rest Water in a closed area with partially wet elements

The test was successfully carried out. There was no movement at all – the water surface elevation remained constant over the whole area. No picture will be shown as there is nothing interesting to see.

1.5.3.3   H_BP_3: Dam break within strongly bended geometry

The computational area was discretized using a mesh with 1805 elements and 1039 vertices. The vertical jump in bed topography between the reservoir and the channel was modelled with a strongly inclined cell (a node can only have one elevation information). Figure 2.1 shows the water level contours and the velocity vectors. The following figures compare the time evolution of the measured water level with the simulated results at the control points G1, G3 and G5.

The results are showing an accurate behaviour expect at the control point G5, where some bigger differences can be observed. They are caused by the use of a coarse mesh around that area. Using a higher mesh density, the water level in the critical area after the strong bend is better resolved and delivers more accurate results. Figure 1.40 shows a comparison at point G5 for a coarse and a dense mesh.
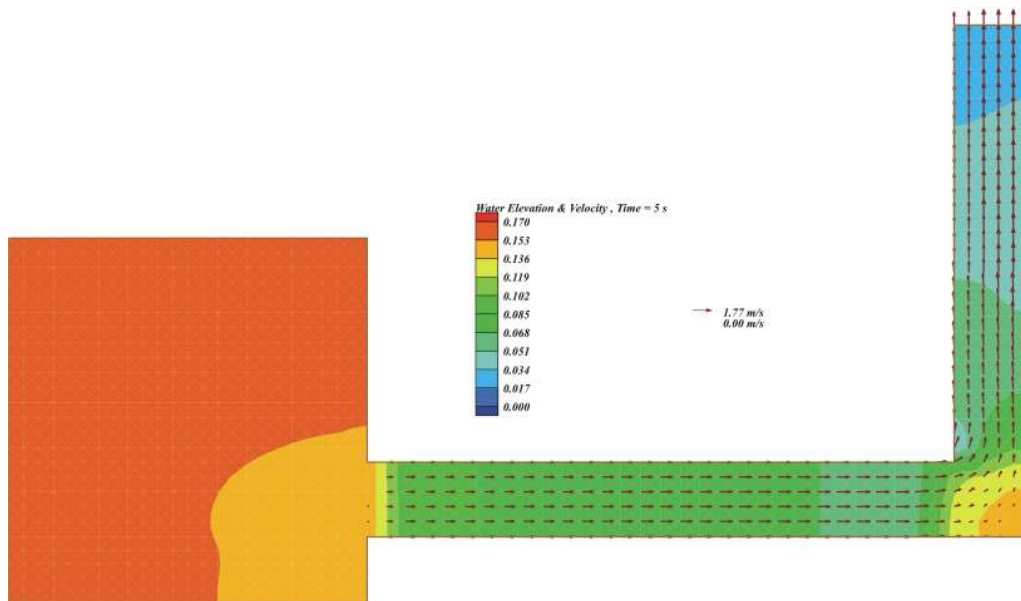
Figure 1.34 *H_BP_3: Water level and velocity vectors 5 seconds after the dam break*
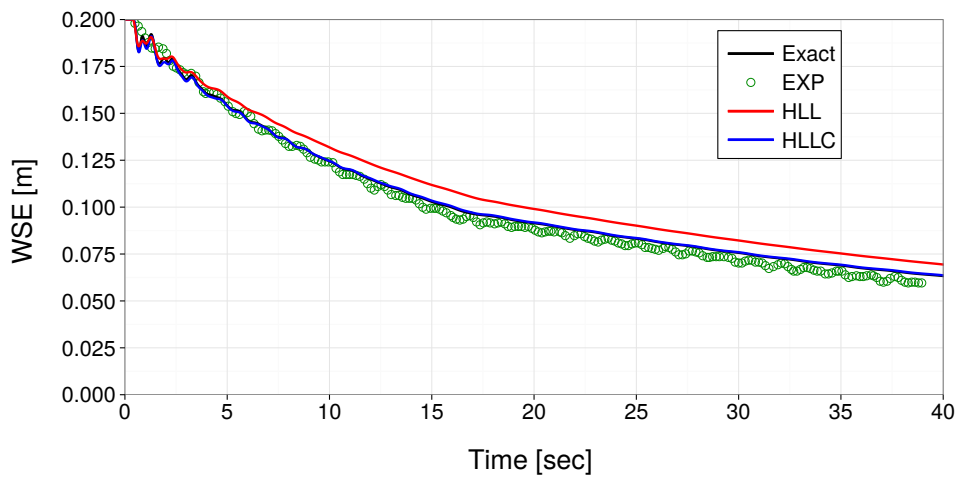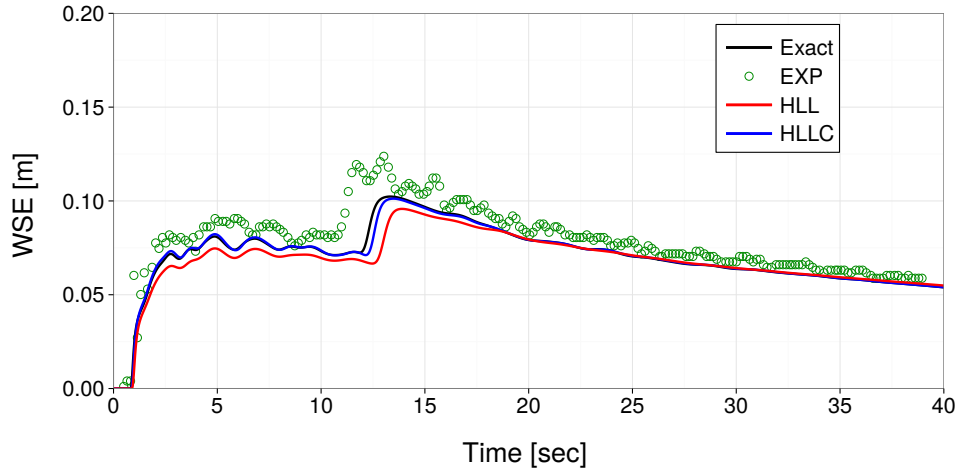

Figure 1.35 *H_BP_3: Water surface elevation at G1*

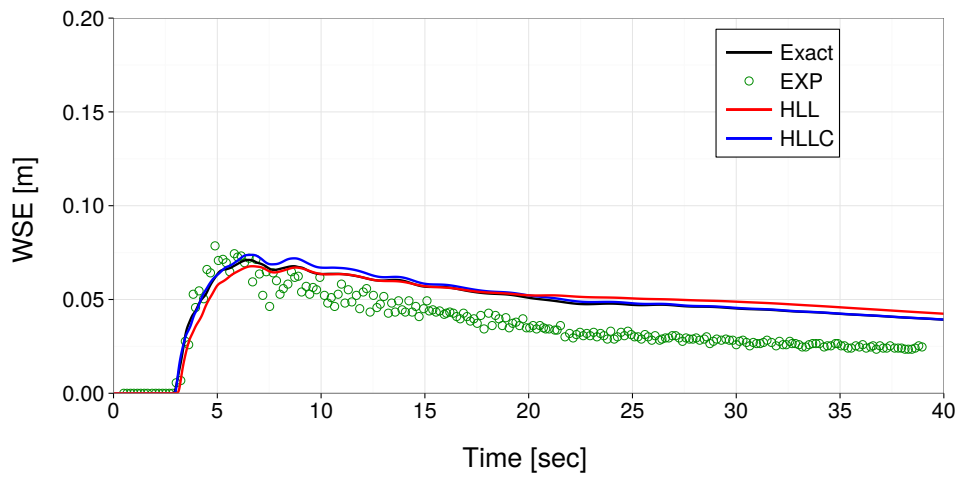***Figure 1.36*** *H_BP_3: Water surface elevation at G3*

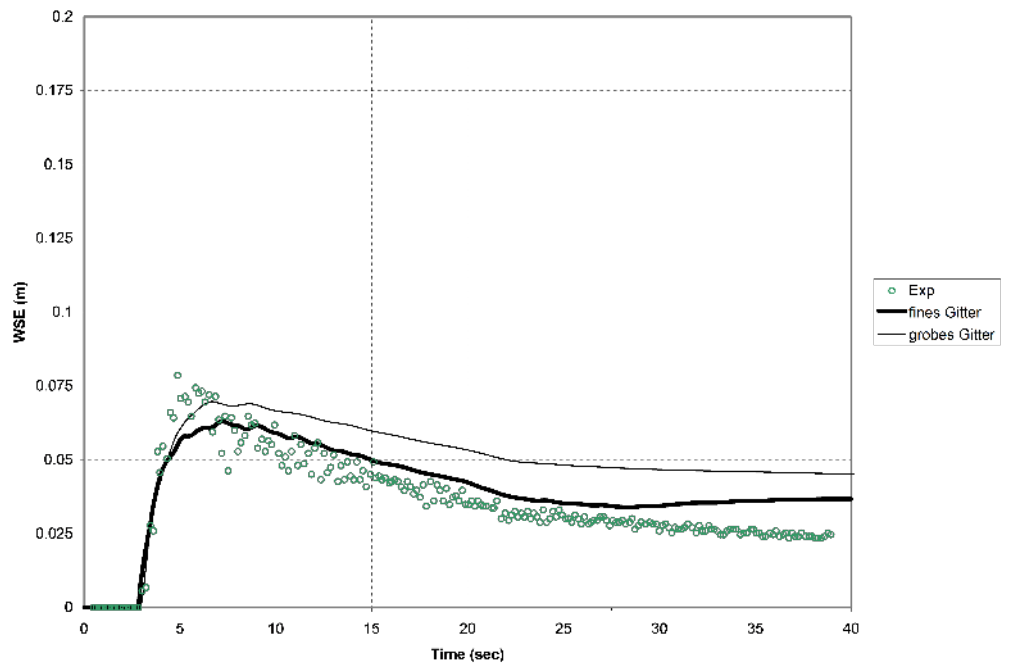***Figure 1.37*** *H_BP_3: Water surface elevation at G5*

Figure 1.38 *H_BP_5: Comparison of the water level at point G5 for a coarse and a dense grid.*

1.5.3.4 H_BP_4: Malpasset dam break

The computational grid consists of 26'000 triangulated control volumes and 13'541 vertices. The time step was chosen according to a CFL number of 0.85. Figure 1.39 shows the computed water depth and velocities 300 seconds after the dam break. Figure 1.40 compares several computed with the observed data on water level elevation at different stations.

BASEplane results show a good agreement with the observed values as also other simulation results (Valiani et al. (2002) and Yoon and Kang (2004)). The differences between computed and measured data are maximally around 10%.

1.5.3.5 H_BP_5: Circular dam break

The obtained results for the circular dam break test case are plotted in two different ways. A 3-D perspective view of the depth is shown from Figure 1.41 to Figure 1.46 to illustrate the overall flow and wave patterns. Additional water surface and velocity profiles are plotted from Figure 1.47 to Figure 1.48 for distinct times.

The following flow patterns of the 2-D circular dam break are reported by Toro and other numerical studies and are also observed here.

After the collapse of dam at t = 0.0 s an outward propagating, primary shock wave is created. A sharp depth gradient develops behind this shock wave. Also, a rarefaction wave is generated which propagates inwards in the direction of the center of the dam break. The rarefaction wave finally reaches the center and generates a very distinct dip of the surface elevation at the center (t = 0.4 s) This dip travels outwards resulting in a rapid drop of the surface elevation at the center, which falls even below the initial outer water level (t = 1.4 s) and finally nearly reaches the ground (t = 3.5 s). A second shock wave develops
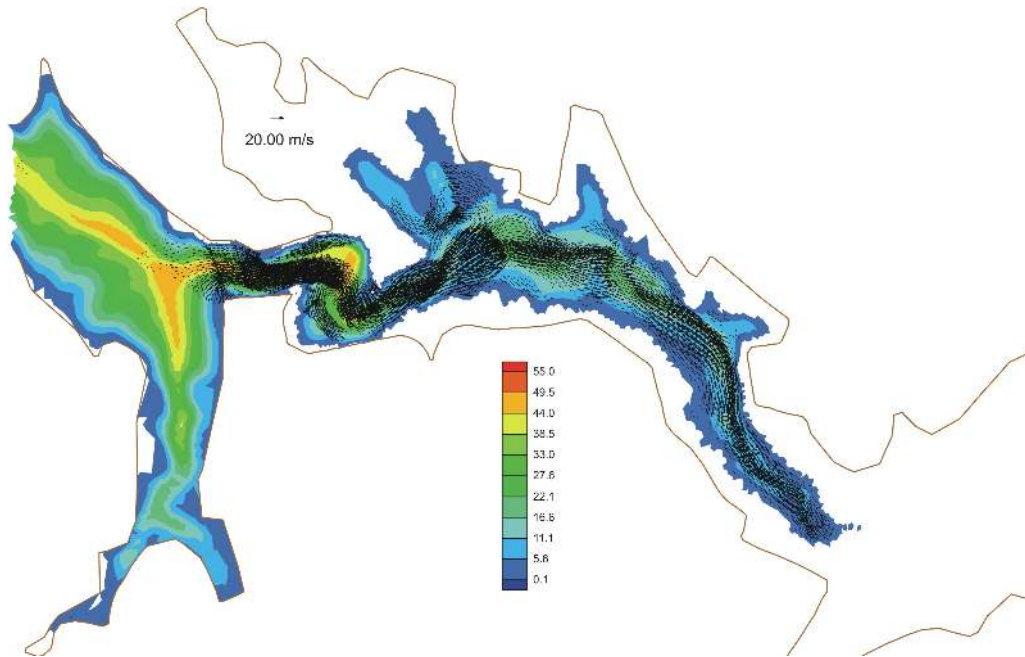
Figure 1.39 *H_BP_4: Computed water depth and velocities 300 seconds after the dam break.*
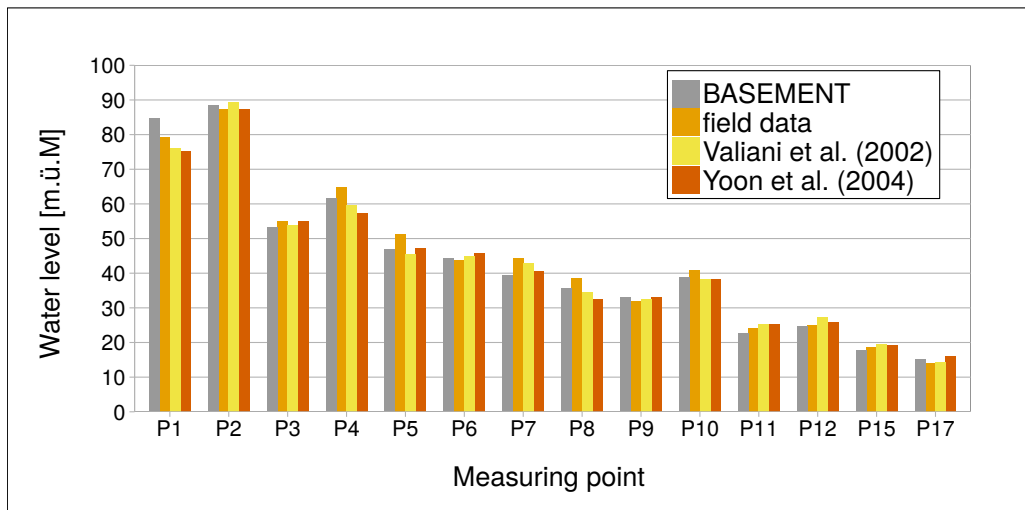

Figure 1.40 *H_BP_4: Computed and observed water surface elevation at different control points*
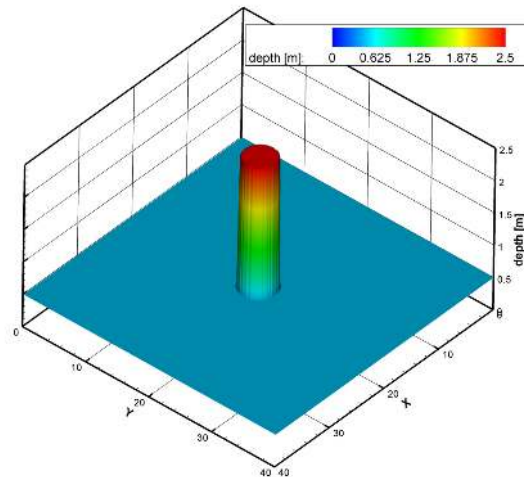
Figure 1.41 *H_BP_5: Perspective view of the circular dam break wave patterns, t=0.0s*

which propagates inwards while the primary shock wave continues to propagate outwards with decreasing strength. This secondary shock wave converges to the center and finally implodes. This generates a sharp jump in the surface water elevation at the center (t = 4.7 s).

As a result it can be stated that BASEplane is able to reproduce these distinct features of the flow and wave patterns of the 2-D circular dam break. Comparisons with the numerical results of Toro show qualitative agreement of the calculated water surface and velocities profiles. Generally, a more diffusive behaviour is observed compared to Toro's results. This may be attributed to the use of first order Godunov methods and to the use of a lower CFL number in the simulations. Despite the use of a rectangular grid, the cylindrical symmetry of the wave propagations is maintained well. Only at the first moments of the dam break, some water surface modulations can be seen at the crest of the primary shock wave which diminish with proceeding time.

The resulting water surface and velocity profiles of the approximate Riemann solvers match well the results of the exact Riemann solver. Both approximate Riemann solvers, HLL and HLLC, reproduce the flow and wave patterns.
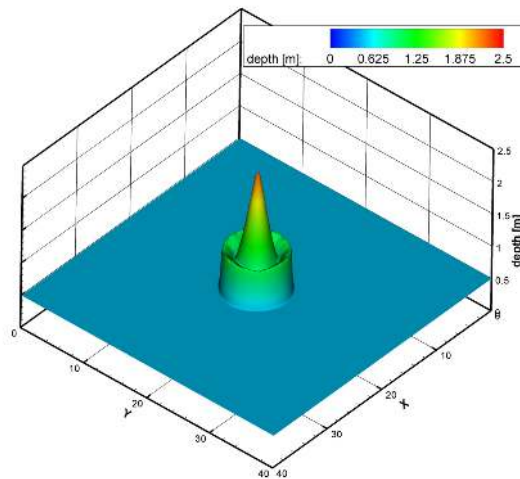
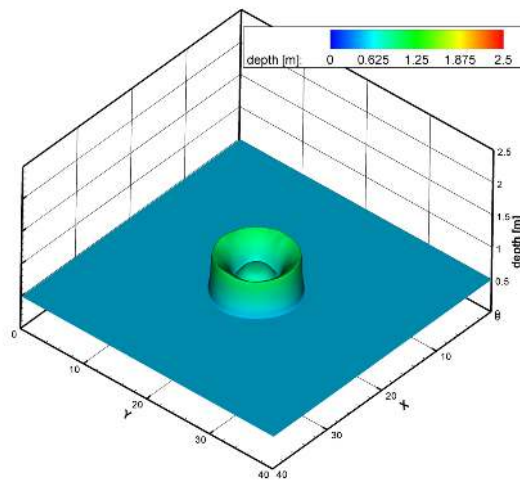**Figure 1.42** *H_BP_5: Perspective view of the circular dam break wave patterns, t=0.4s*

Figure 1.43 *H_BP_5: Perspective view of the circular dam break wave patterns, t=0.7s*
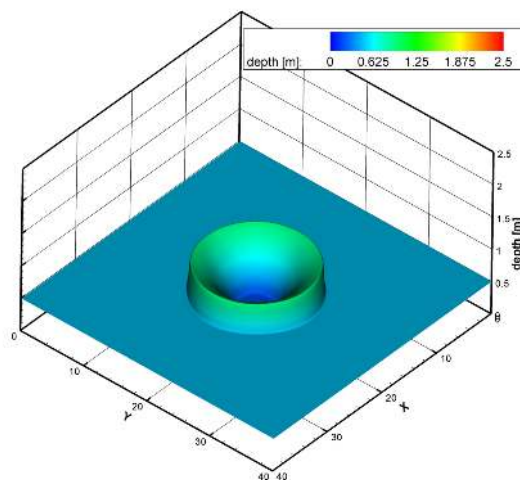

Figure 1.44 *H_BP_5: Perspective view of the circular dam break wave patterns, t=1.4s*
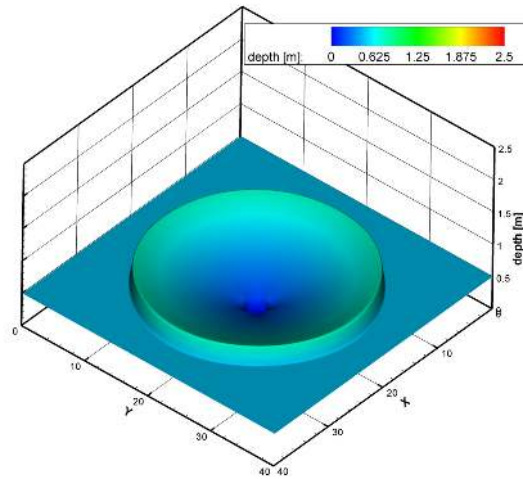
Figure 1.45 *H_BP_5: Perspective view of the circular dam break wave patterns, t=3.5s*
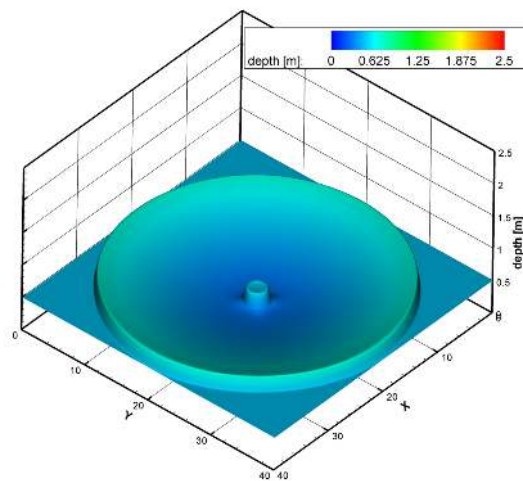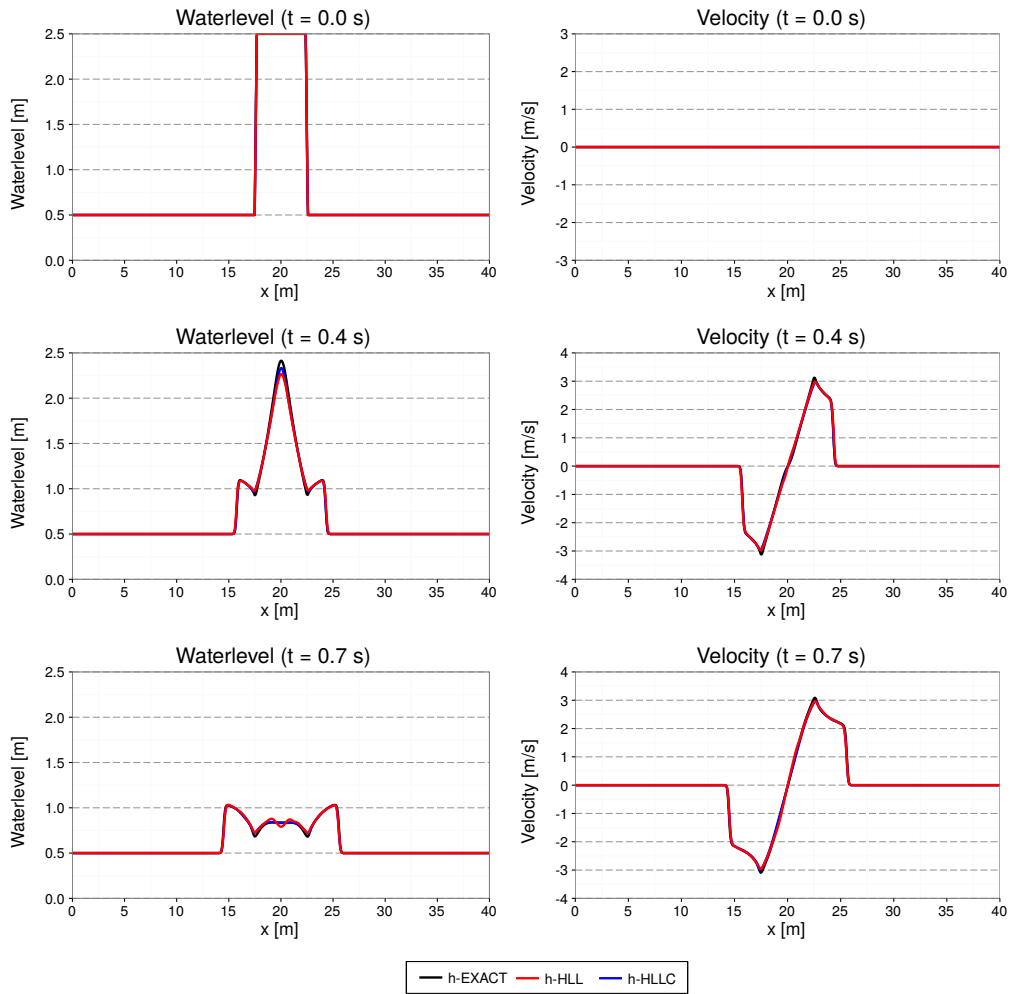

Figure 1.46 *H_BP_5: Perspective view of the circular dam break wave patterns, t=4.7s*

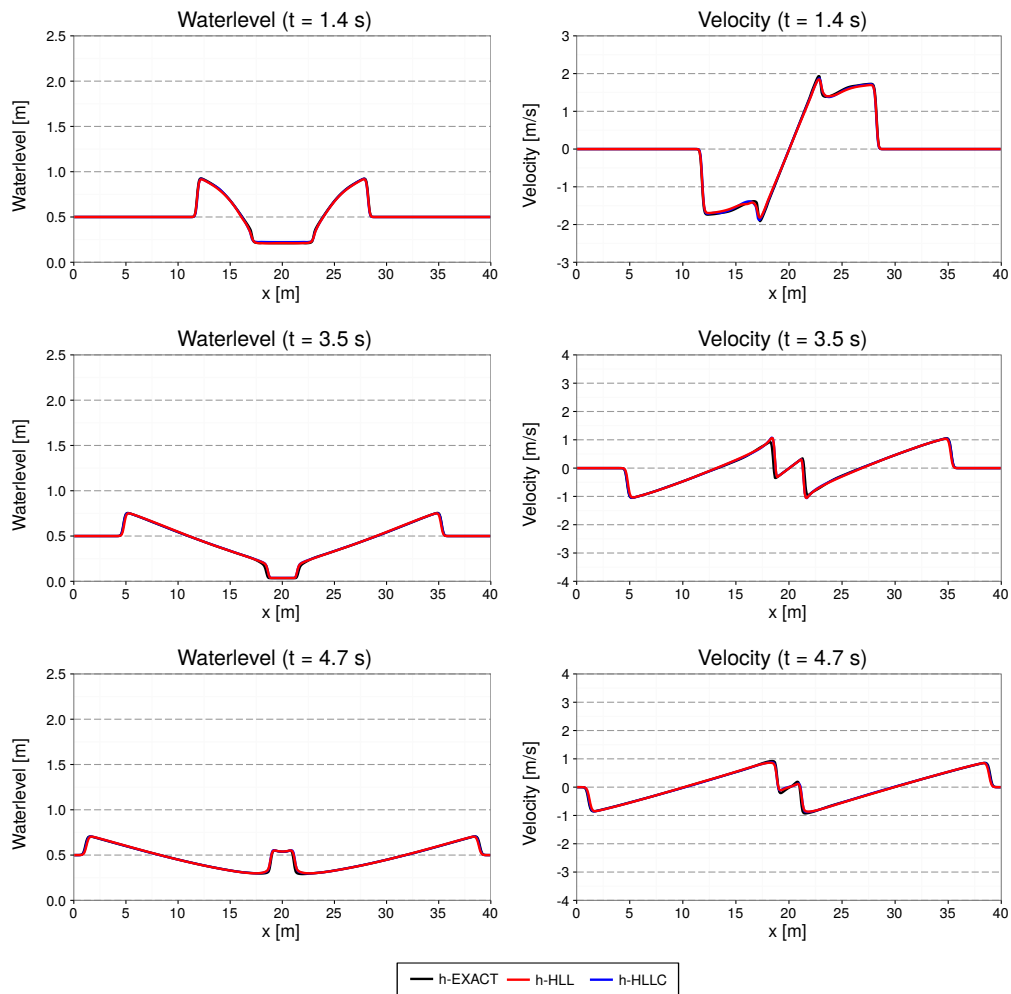**Figure 1.47** *H_BP_5: Water surface along y=20m*

Figure 1.48 *H_BP_5: Water surface along y=20m*

<div align="right">

2

</div>

Sediment Transport

2.1 Introduction

The test catalogue is intended for the validation of the program. The catalogue consists of different test cases, their geometric and hydraulic fundamentals and the reference data for the comparison with the computed results. The test cases are built up on each other going from easy to more and more complex problems. The sediment transport module is tested against some simple experimental test cases described in Section 2.2. Common test cases are suitable for both, 1-D and 2-D simulations. Specific test cases are intended for either 1-D or 2-D simulations.

2.2 Common Test Cases

2.2.1 ST_1: Soni et al: Aggradation due to overloading

2.2.1.1 Intention

This validation case is intended to reproduce equilibrium conditions for steady flow followed by simple aggradation due to sediment overloading at the upstream end. The test is suitable for 1-D and 2-D simulations and uses one single grain size.

2.2.1.2 Description

Soni et al. (1980) and Soni (1981) performed a series of experiments dealing with aggradations. With this particular test, they determined the coefficients a and b of the empirical power law for sediment transport. This test has been used by many researchers as validation for numerical techniques containing sediment transport phenomena (see e.g. Kassem and Chaudhry (1998), Soulis (2002) and Vasquez et al. (2005)).

The experiments were realized within a rectangular laboratory flume. First, a uniform equilibrium flow is established as starting condition. Then the overloading with sediment starts, resulting in aggradations.

2.2.1.3   Geometry and general data

Length of flume	l_f	20	[m]
Width of flume	w_f	0.2	[m]
Median grain size	d_m	0.32	[mm]
Relative density	s	2.65	[-]
Porosity	p	0.4	[-]

2.2.1.4   Equilibrium experiments

In total, a number of 24 experiments with different initial slopes S have been performed by Soni. Measured values are equilibrium water depth h_{eq} and the sediments equilibrium discharge $q_{B,eq}$. The flow velocity u_{eq} has been computed manually by Soni.

Concerning initial conditions for the equilibrium runs, "the flume was filled with sediment up to a depth of 0.15 m and then was given the desired slope". The equilibrium is reached, when a uniform flow has established respectively "when the measured bed- and water surface profiles were parallel to each other." In the average, the equilibrium needed about 4 to 6 hours to develop.

2.2.1.5   Boundary conditions

water flow: constant upstream flow discharge q_{in} and a weir on level 0 at the downstream end. Alternatively, a ghost cell may be used downstream, where the total flux into the last cell leaves the computational domain.

sediment: periodic boundary conditions are applied. The inflow $q_{B,in}$ upstream equals the outflow $q_{B,out}$ downstream.

From now on, we refer to the Soni test case #1, as the entire data for all experiments would go beyond the scope of this section. Measured data for the equilibrium are

q_{in}	4.0	$[m^3/(s \cdot m)] \times 10^{-3}$
S	3.56	$[-] \times 10^{-3}$
h_{eq}	5.0	$[m] \times 10^{-2}$
$q_{B,eq}$	12.1	$[m^3/(s \cdot m)] \times 10^{-6}$
u_{eq}	0.4	$[m/s]$

The establishment of the same equilibrium conditions is achieved by calibrating e.g. the hydraulic friction. As soon as the simulation data looks similar to the measured results, the aggradation can be started.

2.2.1.6   Aggradation experiments

After the equilibrium has been reached, the sediment feed was increased upstream by an overloading factor of $q_B/q_{B,eq}$. This factor is different for each experiment. For test case #1, it was set to 4.0. The flow discharge remains the same as in the equilibrium case. Due to the massive sediment overloading, aggradation starts quickly at the upstream end. Measured data is available for the bottom elevation at certain times.

2.2.2   ST_2 : Saiedi

2.2.2.1   Intention

Similar to the Soni test case, this validation deals with aggradation due to sediment overloading. However, compared to Soni, the sediment input is considerably greater than the carriage capacity of the flow and an aggradation shock is forming. The test verifies whether the computational model can handle shocks within the sediment phase. Again, a single grain size is used.

2.2.2.2   Description

The experiments were conducted in a laboratory flume located at the Water Research Laboratory, School of Civil Engineering, University of New South Wales, Australia. Geometric details and results are reported in Saiedi (1981a) and Saiedi (1981b). Saiedi did two separate tests, one in a steady flow and one in a rapidly unsteady flow with varying sediment supply. Only the steady case is of our interest here.

2.2.2.3   Geometry and general data

The sediment used was of fairly uniform size with the median grain size $d_{50} = 2$ mm, relative density $s = 2.6$ and porosity at bed $p = 0.37$.

The test section was about 18 m long in a 0.61 m wide glass-sided flume with a slope of 0.1 %. The flow was fed with a sediment-supply of constant rate using a vibratory feeder. The sediment input rate was chosen to be greater than the transport capacity of the flow, therefore resulting in the formation of an aggradation shock sand-wave travelling downstream. The downstream water level was maintained constant by adjusting the downstream gate.

2.2.2.4   Calibration

Before starting the sediment feed, at first, the aim is a uniform flow which fulfils a measured flow discharge q and the corresponding water depth h . This can be achieved by adjusting e.g. the hydraulic friction coefficient from Manning n. From the experiments, Saiedi proposes a calibration estimate of $n=0.0136 +0.0170$ q . Note that there is no sediment layer at the calibration phase.

As initial conditions, start with a fluid at rest and a uniform flow depth of $h_0= 0.223$ m on the sloped flume. The flow feed rate upstream is constant at $q_{in} = 0.098$ m^3/s. At the

downstream boundary, the water level is kept constant at h = 0.223 m "by adjusting the downstream gate".

2.2.2.5   Aggradation

As soon as a uniform flow with constant water depth is established, the sediment feed upstream can be started. It remains constantly on $q_B = 3.81$ kg/min. The results for the steady state computation are available at $t = 30$ min and $t = 120$ min. To avoid cluttering, the plots of the measured bed levels have been averaged.

2.2.3   BeST_3 : Guenter

2.2.3.1   Intention

The Günter test series deals with degradation in a laboratory flume. The experiments were conducted using a multiple grain size distribution. The reported results allow for a validation of the bed topography and the behaviour of heterogeneous sediment transport models.

2.2.3.2   Description

Günter performed some tests at the Laboratory of Hydraulics, Hydrology and Glaciology at the ETH Zurich using sediment input with a multiple grain size distribution. His aim was to determine a critical median shear stress of such a grain composition.

Günter was interested in the behaviour of different grain classes but also in the steady ultimate state of the bed layer, when a top layer has developed and no more degradation occurs with the given discharge. As initial state, a sediment bed with a higher slope than the slope in the steady case is used. There is no sediment feed but a constant water discharge. The bed then starts rotating around the downstream end resulting in a steady, uniform state with an armouring layer.

The validation is based on the resulting bed topography and the measured grain size distribution of the armouring layer at the end of the simulation.

2.2.3.3   Geometry and general data

Guenter actually investigated several test cases with different grain compositions and boundary conditions. This test case corresponds to his experiment No.3 with the grain composition No.1.

The experiment was conducted in a straight, rectangular flume, 40 m long, 1 m width with vertical walls. At the downstream end, the flume opens out into a slurry tank with 8 m length, 1.1 m width and 0.8 m depth (relative to the main flume), where the transported sediment material is held back. The sediment bed has an initial slope of 0.25 % with an initial grain size distribution given by

d_g [cm]	Mass fraction [%]
0.0 - 0.102	35.9
0.102 - 0.2	20.8
0.2 - 0.31	11.9
0.31 - 0.41	17.5
0.41 - 0.52	6.7
0.52 - 0.6	7.2

2.2.3.4   Simulation procedure and boundary/initial conditions

First of all, the domain gets filled slowly with fluid until the highest point is wet. The weir on the downstream end is set such that no outflow occurs. After the bed is completely under water, the inflow discharge upstream is increased successively during 10 minutes up to the constant discharge of 56.0 l/s. At the same time, the weir downstream is lowered down to a constant level until the water surface elevation remains constant in the outflow area. There is no sediment inflow during the whole test.

Using this configuration, the bed layer should rotate around the downstream end of the soil. The experiment needed about 4 to 6 weeks until a stationary state was reached. The data to be compared with the experiment are final bed level and final grain size distribution.

2.3   BASEchain Specific Test Cases

2.3.1   ST_BC_1: Advection of suspended load

2.3.1.1   Intention

This test is intended to verify the quality of the advection schemes. The aim is to minimize the numerical diffusion even over long distances.

2.3.1.2   Description

In a rectangular channel with steady flow conditions the advection of the suspended material is observed, once for a vertical front of concentration (case A) and once for a concentration with a Gaussian distribution (case B). The Quickest, Holly-Preissmann and MDPM-Scheme are tested. The diffusion is set to 0 and there is no sediment exchange with the soil.

2.3.1.3   Geometry and initial conditions

- The computational area is a rectangular channel with 10 km length, 30 m width and a slope of 0.5 ‰ .The cell width Δx is 40 m.

- The initial condition for the hydraulics is a steady discharge of 10 m^3/s

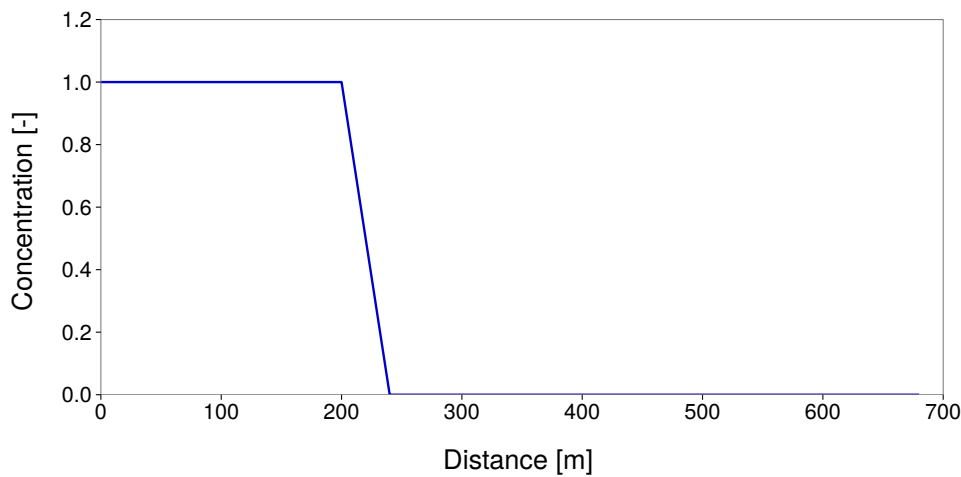- The initial conditions for the suspended load are the following:

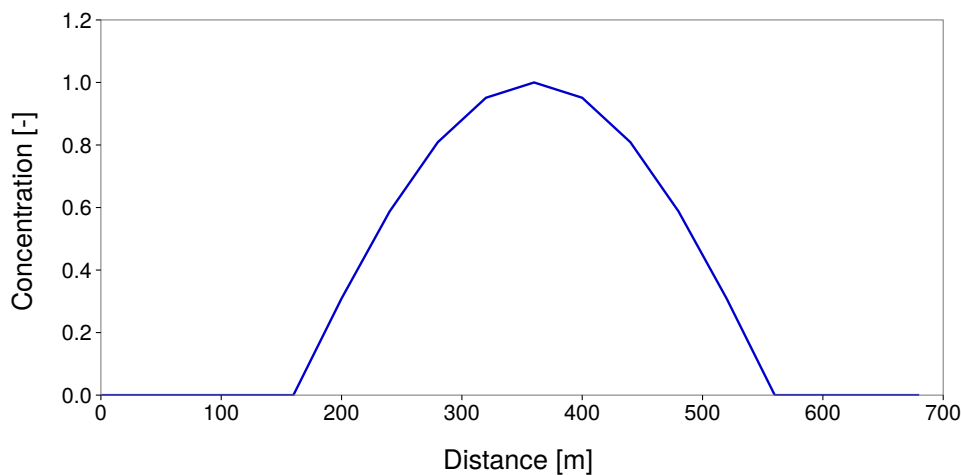*Figure 2.1* *ST_BC_1: Initial condition for case A*

Figure 2.2 *ST_BC_1: Initial condition for case B*

Case A: The concentration is 1 for the first 200 meters of the flume and 0 for the rest.

Case B: The concentration has a Gauss-distribution on the upstream part of the flume.

2.3.1.4 Boundary conditions

- The friction expressed as k_{Str} is 80.
- The hydraulic upper Boundary is a hydrograph with steady discharge of 10 m^3/s.
- The hydraulic outflow boundary is h-q-relation directly computed with the slope.
- The upper boundary condition for suspended load is a constant inflow concentration of 1 for case A and 0 for case B.
- The lower boundary condition is an outflow concentration corresponding to the concentration in the last cell.
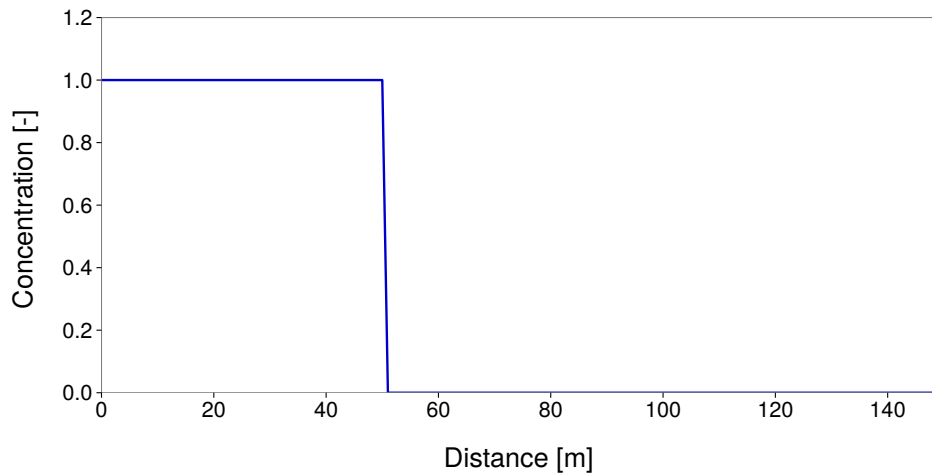
***Figure 2.3** ST_BC_2: Initial condition for case A*

2.3.2   ST_BC_2: Advection-Diffusion

2.3.2.1   Intention

This test is intended to verify the combination of advection and diffusion for suspended load for a given diffusion factor Γ.

2.3.2.2   Description

In a rectangular channel with steady flow conditions the behaviour of the suspended material is observed, once for a vertical front of concentration (case A) and once for a concentration with a Gaussian distribution (case B). The Quickest, Holly-Preissmann and MDPM-Scheme are tested. There is no sediment exchange with the soil.

2.3.2.3   Geometry and initial conditions

- The computational area is a rectangular channel with 1000 length, 20 m width and a slope of 1 ‰. The friction expressed as k_{Str} is 30. The cell width Δx is 1 m.
- The initial condition for the hydraulics is a steady discharge of 50 m^3/s.
- The initial conditions for the suspended load are:

Case A: The concentration is 1 for the first 50 meters of the flume and 0 for the rest.

Case B: The concentration has a Gauss-distribution on the upstream part of the flume.

2.3.2.4   Boundary conditions

- The friction is expressed as k_{Str} is 30.
- The hydraulic upper boundary is hydrograph with steady discharge of 50 m^3/s.
- The hydraulic outflow boundary is h-q-relation directly computed with the slope.
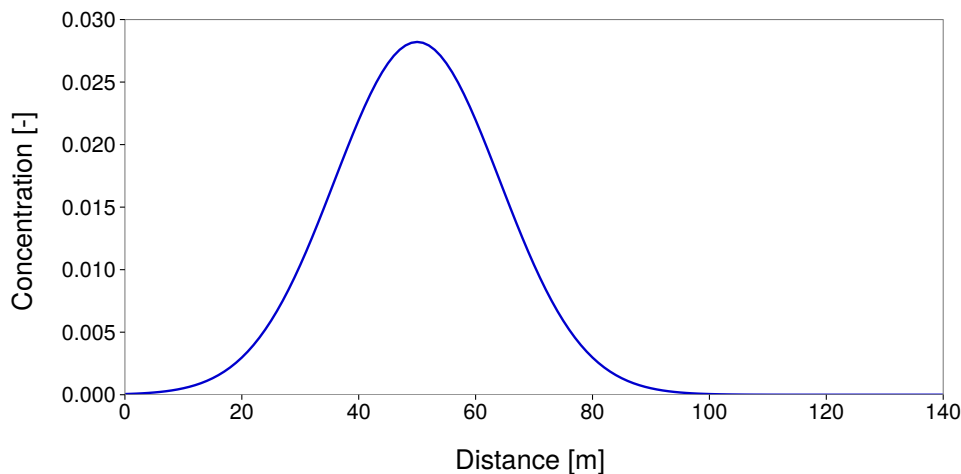
Figure 2.4 ST_BC_2: Initial condition for case B

- The upper boundary condition for suspended load is a constant inflow concentration of 1 for case A and 0 for case B.

- The lower boundary condition is an outflow concentration corresponding to the concentration in the last cell.

2.4 BASEplane Specific Test Cases

2.4.1 ST_BP_1: Advection of suspended load

2.4.1.1 Intention

This test is intended to verify the quality of the advection scheme. The aim is to minimize the numerical diffusion even over long simulation time or distance.

2.4.1.2 Description

In a rectangular channel with steady flow conditions the advection of the suspended material is observed, once for a vertical front of concentration (case A) and once for a concentration with a Gaussian distribution (case B). The MDPM-Scheme is tested. The diffusion is set to 0 and there is no sediment exchange with the soil.

2.4.1.3 Geometry and initial conditions

- The computational area is a rectangular channel with 1 km length, 10 m width and a slope of 7 ‰.

- The initial condition for the hydraulics is a steady discharge of 50 m^3/s.

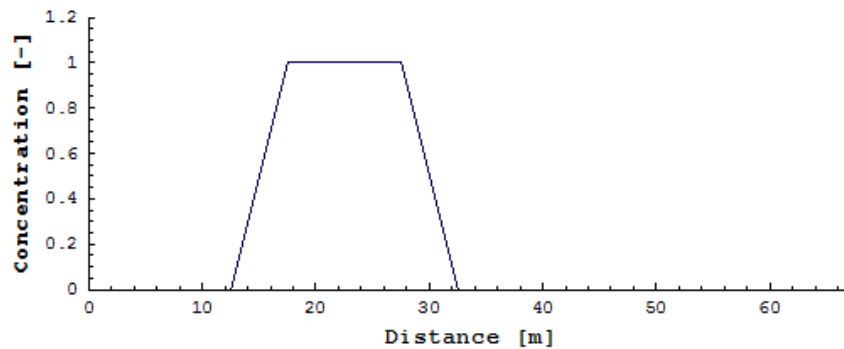- The initial conditions for the suspended load are the following:

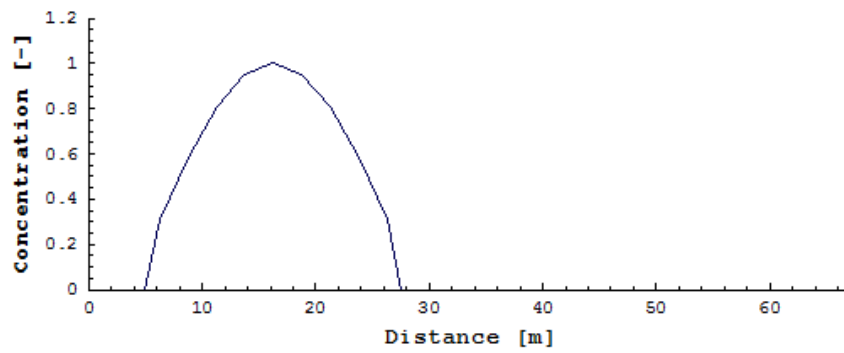***Figure 2.5*** *ST_BP_1: Initial condition for case A*

Figure 2.6 *ST_BP_1: Initial condition for case B*

Case A: The concentration is 1 for the first 30 meters at the upper end of the channel. During the simulation a constant sediment source is added to maintain between 15 and 20 meters from the upstream end to maintain a constant concentration.

Case B: The concentration has a Gauss-distribution on the upstream part of the flume.

2.4.1.4   Boundary conditions

- The friction is expressed as Manning factor n is 0.03.
- The upper boundary is a hydrograph with steady discharge of 50 m^3/s with concentration 0.
- The hydraulic outflow boundary is zero_gradient.
- The lower boundary condition for suspended load is an outflow concentration corresponding to the concentration in the last cell.
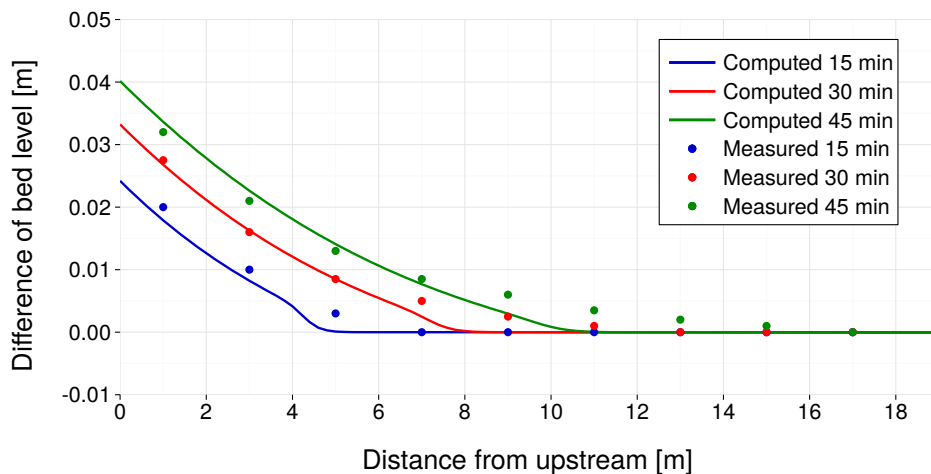
Figure 2.7 *Soni Test with MPM-factor = 6.44 (BC)*

2.5 Results

2.5.1 Common Test Cases

2.5.1.1 ST_1: Soni

2.5.1.1.1 Results obtained by BASEchain

The Soni test case has been simulated with the MPM approach like the one used for the next test case of Saiedi (see Section 2.2.2). As the equilibrium bed load of $2.42 \cdot 10^{-7}$ m^3/s is known, the MPM-factor has been calibrated to obtain a good agreement for the equilibrium state. This leads to a value of 6.44 for the prefactor in the MPM-formula.

The results show a quite good agreement between experiment and simulation.

2.5.1.1.2 Results obtained by BASEplane

The Soni test case was modelled as single grain computation on a mobile bed with the transport formula of Meyer-Peter & Müller (MPM). The grain diameter is chosen as the mean diameter of the grain mixture used by Soni. The simulations were performed on an unstructured mesh with 1202 triangular elements.

In Soni's experiments at first an equilibrium transport was established within the laboratory flume. The corresponding transport rate observed by Soni is known to be $2.42*10^{-7}$ m^3/s. Then the sediment inflow was increased to 4 times the equilibrium transport.

To be able to reproduce the experiments, the transport formula must be calibrated to achieve the same equilibrium transport rate. The calibration resulted in a reduction of the pre-factor of the MPM formula from 8 to about 3.3. The critical dimensionless shear stress for incipient motion in the MPM formula is not calibrated here and therefore, per default, determined from the Shields diagram.

The numerical results for the bed aggradations and the water levels are compared with the measurements by Soni. The situations after 15min, 30min and 40min are plotted in the following figures. Generally the numerical results show an acceptable agreement
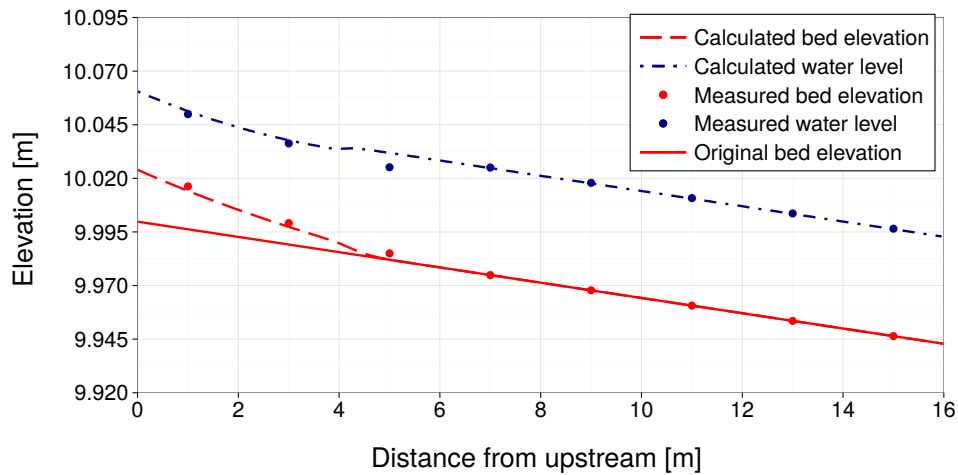
Figure 2.8 *ST_1: Soni Test – Bed aggradation and water level after 15 min (BP)*
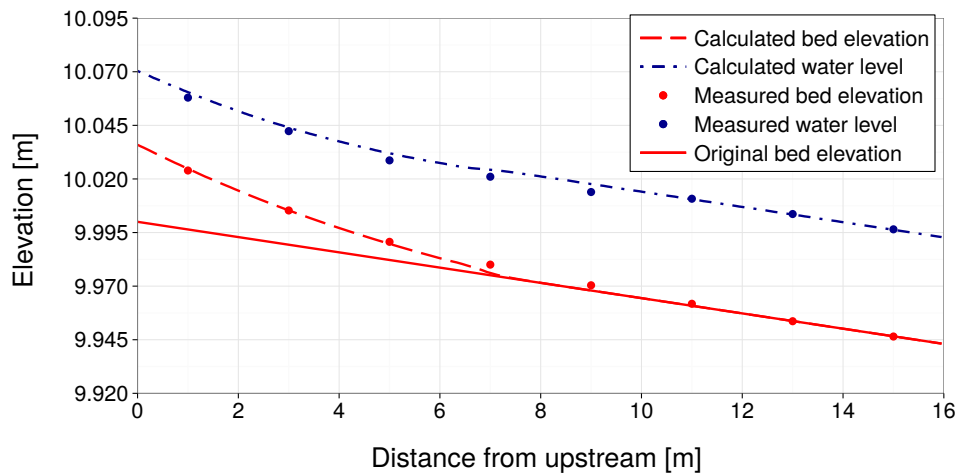

Figure 2.9 *ST_1: Soni Test – Bed aggradation and water level after 30 min (BP)*

with the measured values. Compared to the measurements, the toe of the aggradation front shows a less diffusive smearing behaviour. This may be attributed to the finer grain fractions within Soni's grain mixtures, whose behaviour is not adequately modelled in single grain computation. Also the rather fine diameter of 0.32 mm can be seen as problematic concerning the applicability of the MPM formula, which is best suited for coarse sands and gravel.

2.5.1.2 ST_2: Saiedi

2.5.1.2.1 Results obtained by BASEchain

The Saiedi test was simulated using the Meyer-Peter Müller approach for the sediment flux, which can also be formulated as:

$$q_B = factor(\theta - \theta_{cr})^{3/2}\sqrt{(s-1)g}d_m^{3/2}$$

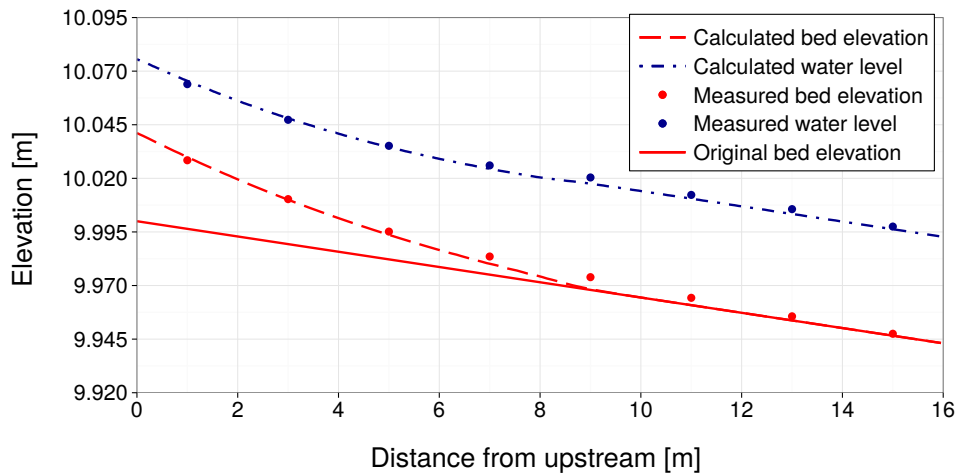The factor is usually set around 8.0 and should be between 5 and 15 (Wiberg and Smith,

Figure 2.10 *ST_1: Soni Test – Bed aggradation and water level after 40 min (BP)*
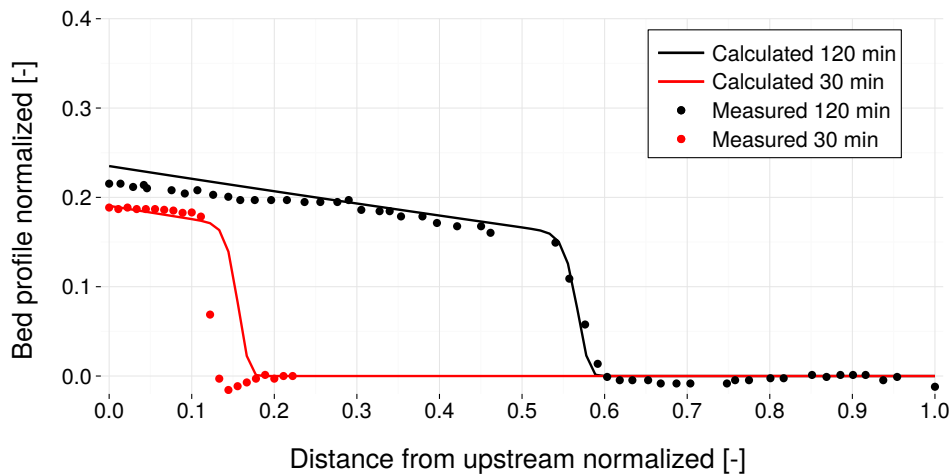

Figure 2.11 *Saiedi Test, with MPM-factor = 13 (BC)*

1989). In this case it was calibrated to 13.

The results present a good fit. This example shows that, to reproduce this type of transport, the approach of MPM approach needs calibration.

2.5.1.2.2 Results obtained by BASEplane

The Saiedi test case was simulated using a single grain approach on an unstructured mesh made of 768 triangles. The Meyer-Peter & Müller formula was used to determine the bedload transport. The simulation was performed on a fixed bed, where no erosion can take place.

To obtain a reasonable fit between the measured values and the simulation results the pre-factor of the transport formula again had to be reduced from 8 to about 5. Furthermore, the simulations show that the sediment is transported too rapidly out of the flume compared to the experiments. To compensate for this effect the critical Shields factor for incipient motion is increased to 0.055, instead of using the value from the Shields diagram.
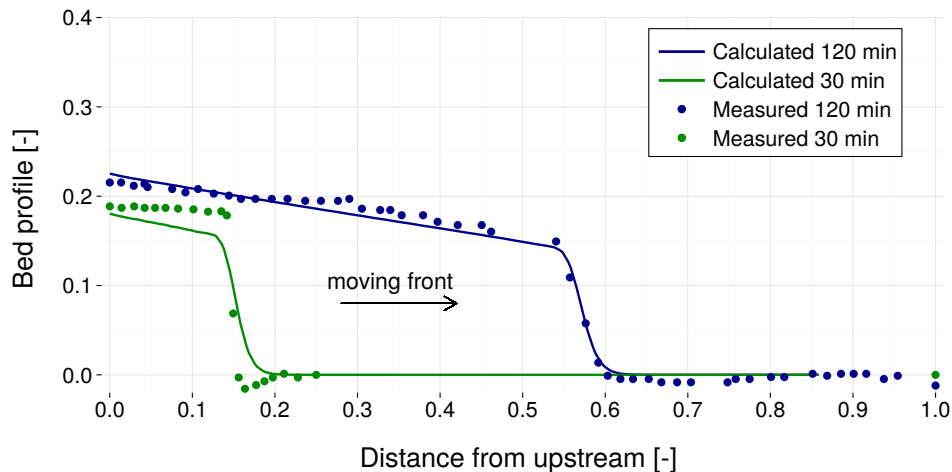
Figure 2.12 *ST_2 : Saiedi Test, propagation of sediment front (BP)*

With these adjustments, the shape and the propagation speed of the sediment front are captured well and Saiedi's results can be reproduced with good accuracy. The front of the sediment bore does not smear over the time but remains steep during the propagation.

2.5.1.3   ST_3: Guenter

2.5.1.3.1   Results obtained by BASEchain

For mixed materials, the Guenter test case has been performed. The initial slope was chosen to be 0.25 % - this corresponds to the full experiment of Guenter.

The eroded material is eliminated from the end basin by a sediment sink. The active layer height is 5 mm and the critical dimensionless shear stress (for beginning of sediment transport) is set to the default value of 0.047. After 200 hours, there are no more important changes recognizable. After the results of Guenter the slope at the final equilibrium state should be the same as the initial one.

The final slope in the numerical model is in good agreement with the final slope observed in the physical experiment. Moreover the grain sorting effect can be modelled quite well. The final grain size distribution is slightly coarser than the observed grain size distribution in the physical model (Figure 2.14).

There are many parameters which influence this result, especially the choice of the critical shear stress and the active layer thickness. Again, this example shows, one has to use different approaches for sediment transport with different values for the free parameters. Quite often, a sensitivity analysis may give a better insight in the behaviour of certain formulas and their parameters.

2.5.1.3.2   Results obtained by BASEplane

The Guenter test has been numerically modelled with 6 different grain classes. The sediment transport capacity has been determined with Hunziker's formula for graded transport. The critical dimensionless shear stress for beginning of sediment transport is set to the default value of 0.047 in Hunziker's formula. This bedload formula was not further adjusted for
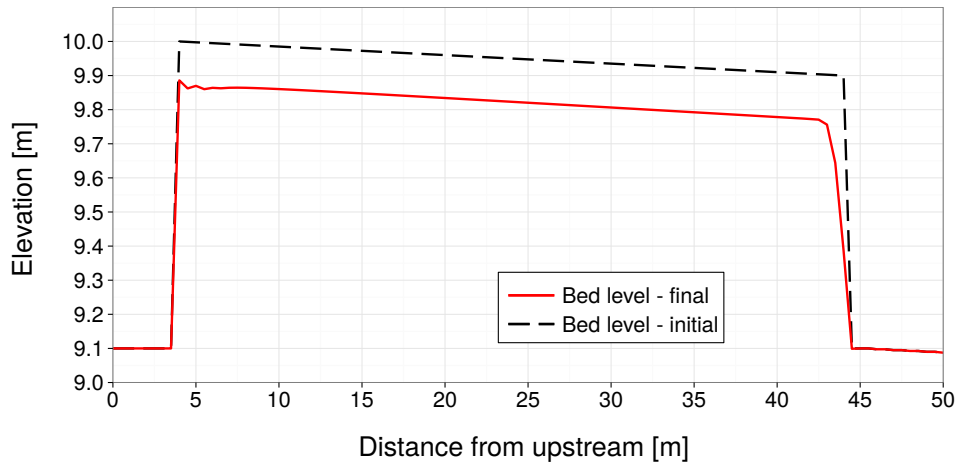
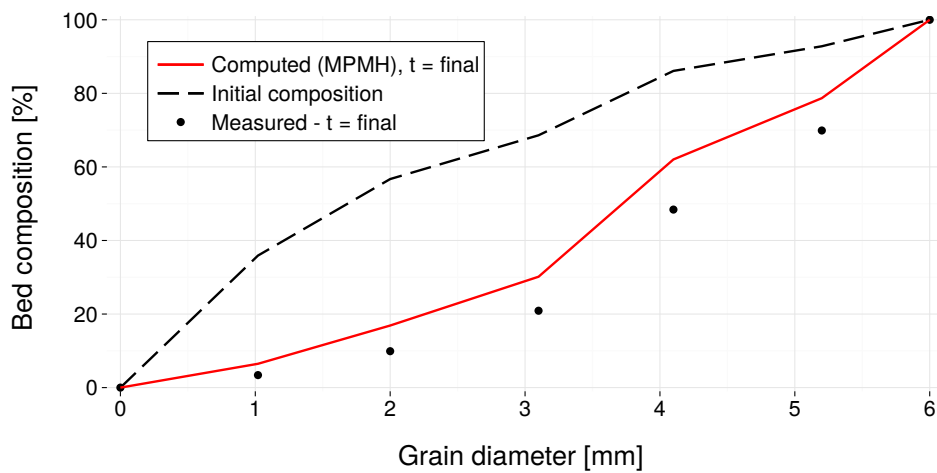***Figure 2.13*** *ST_3: Guenter Test: equilibrium bed level (BC)*

Figure 2.14 *ST_3: Guenter Test: grain size distribution (BC)*
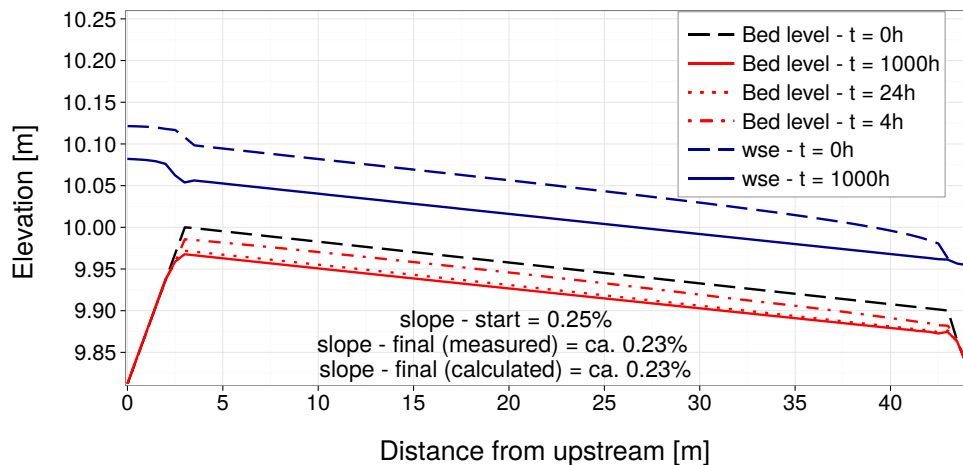
Figure 2.15 *ST_3: Guenter Test: equilibrium bed level (BP)*

the simulations and used with its default values for the pre-factor (=1.0) and the exponent of the hiding exponent (=-1.5).

The experimental flume has a mobile bed and is modelled with 312 rectangular elements. The channel parts in front and behind the experimental flume, are set to fixed bed elevations in this simulation. The sediment which leaves the mobile bed and enters the fixed bed section is removed using a sediment sink. The thickness of the bedload control volume is set to a constant value of 5 mm. The choice of the thickness of the bedload control volume shows large impacts on the simulation results. An increased thickness here results in larger erosion and a finer composition of the final bed armour.

The mobile bed is eroded during the simulation until finally an armouring layer of coarser materials is formed which prevents further erosions. The final slope of the bed is nearly constant with a value of 0.23%, which is slightly smaller than the original slope of 0.25%. This result is in agreement with the observations made by Guenter. Also the rotation of the bed surface around the downstream end of the flume can be observed during the simulation.

Figure 2.16 shows the measured and computed grain compositions, whereas the latter was taken from the upstream end of the flume. The simulated grain size distribution (red curve) shows a good qualitative agreement with the measured distribution by Guenter (blue dots). But a trend can be seen that the computed composition is slightly too fine. Finally, it can be said, that the numerical model seems capable of reproducing the sorting effects and seems able to simulate the formation of an armouring bed layer.

Beside the results obtained with the Hunziker's transport formula, Figure 2.16 presents also the resulting grain distributions obtained with Wu's transport formula (green curve). Here it can be seen that the finer fractions are eroded too strongly, but a qualitative agreement is obtained. For the simulations with the Wu formula the same critical shear stress of 0.047 was used, and the default settings of the bed load factor (=1.0) and the default exponent of the hiding-and exposure coefficient (=-0.6) were set. But to enable similar erosion volumes as obtained with the Hunziker's formula, the thickness of the bed load control volume was increased to 2.5 cm.

To check for the mass conservation properties a comparison was made between the initial
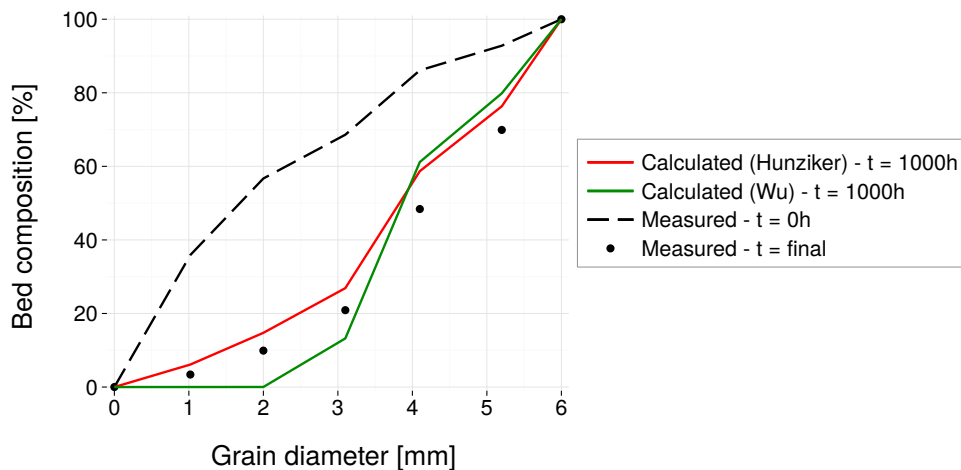
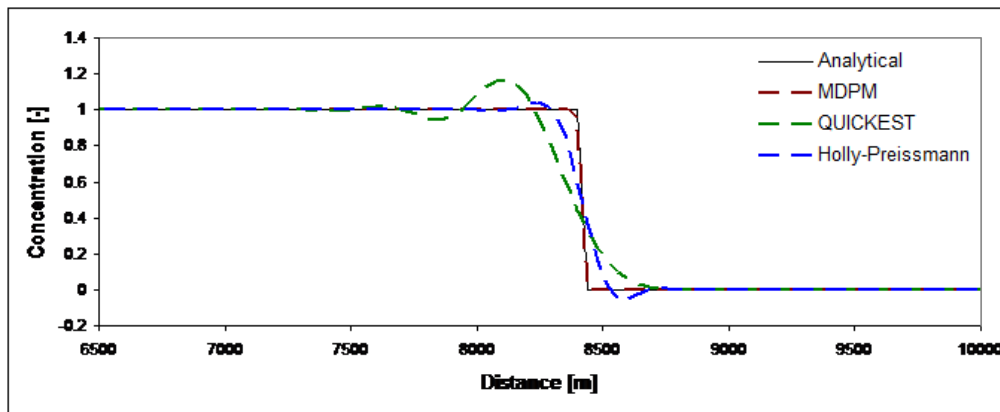**Figure 2.16** *ST_3: Guenter Test: grain size distribution (BP)*

Figure 2.17 *ST_BC_1: case A, concentration after 9060 s*

sediment volumes in the flume and the final sediment volumes, under regard of all removed sediment volumes. Despite the long run time, the simulation proves to be conservative regarding the sediment transport. Since the experiment has a constant inflow rate and quasi-steady flow conditions, the 'hydro_step' approach can be used here which allows a reduction of several orders of magnitude in simulation time.

2.5.2 BASEchain specific Test Cases

2.5.2.1 ST_BC_1: Advection of suspended load

The advection test has been executed for the QUICK, the QUICKEST, the Holly-Preissmann and the MDPM-scheme. The results of the QUICK-scheme are not illustrated as they are very instable.

Case A: Figure 2.17 shows the concentration front after 9060 s or after 8200 m of way. The results show that the big issue of the advection simulation, the numerical diffusion, is almost completely avoided by the MDPM scheme.

Case B: Figure 2.18 shows the concentration front after 9772.5 s or after 8843 m of way. For
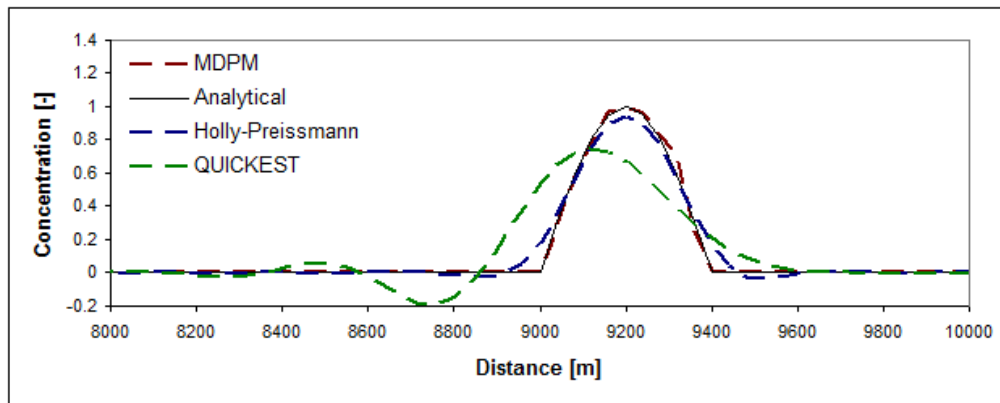
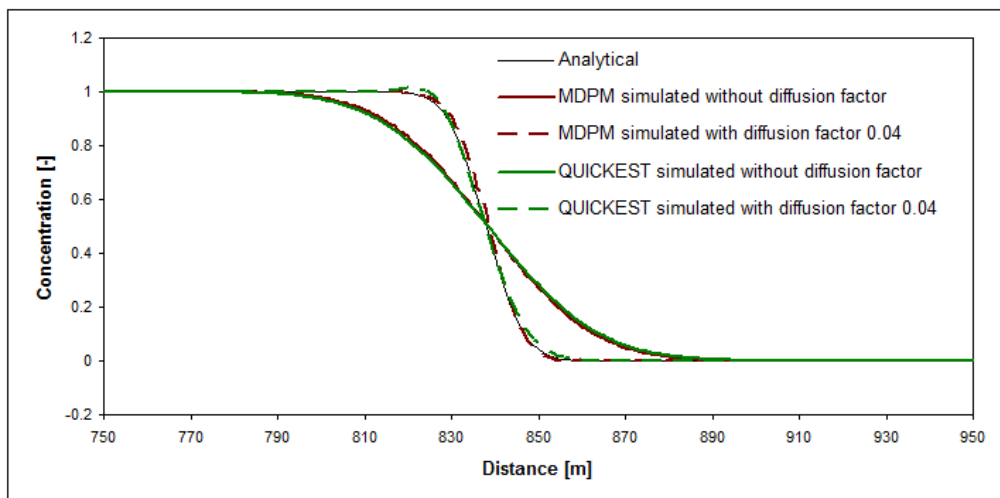***Figure 2.18*** *ST_BC_1: case B, concentration after 9772.5 s*

Figure 2.19 *ST_BC_2: case A, concentration after 605 s*

the MDPM scheme there are some local deviations from the analytical solution. These are due to the discretisation and do not increase with time and distance. The initial maximal concentration is conserved.

2.5.2.2   ST_BC_2: Advection-Diffusion

The Advection Test has been executed for the QUICKEST, and the MDPM-scheme. Figure 2.19 and Figure 2.20 show the concentration front of case A and the Gauss distribution for case B after 605 s or after 788 m away. The diffusion factor is 0.04 for case A and 0.1 for case B. The result of the simulation with the internally computed diffusion is also illustrated, but there is no analytical solution to compare.

The results are satisfying. The QUICKEST scheme gives here a result which is nearly as good as the one of the MDPM scheme, but this is due to the short simulation time. Its deviation from the exact solution due to numerical diffusion increases with the time like shown in test ST_BC_1.
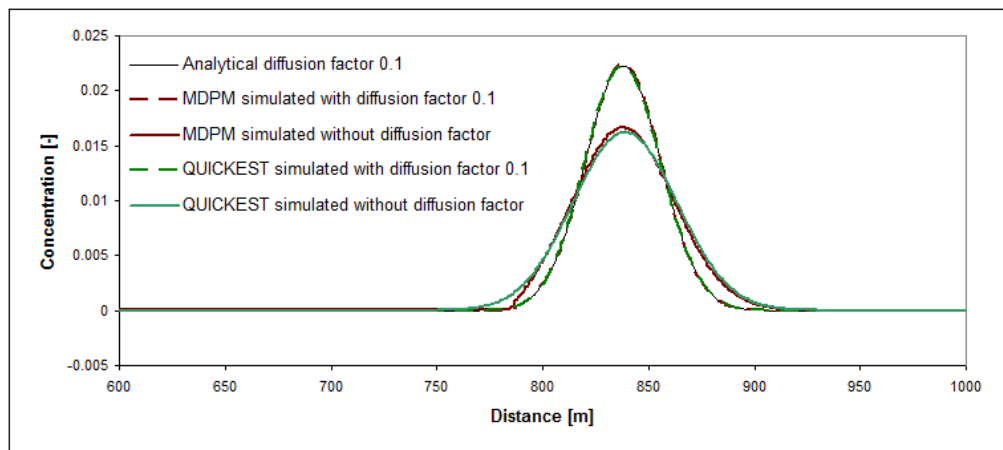
***Figure 2.20** ST_BC_2: case B, concentration after 605 s*

2.5.3 BASEplane specific Test Cases

2.5.3.1 ST_BP_1: Advection of suspended load

To test the advection of suspended sediment transport, the MDPM-scheme has been applied.

Case A: Figure 2.21 shows the concentration front after 200 s or after 730 m of distance travelled. The results show that there is no oscillation before and after the front. The diffusion is small and manly due to the irregular discretisation.

Case B: Figure 2.22 shows the concentration front after 200 s or after 700 m of distance travelled. The maximum of the Gauss distribution is conserved and there are no oscillations near the steep gradients.
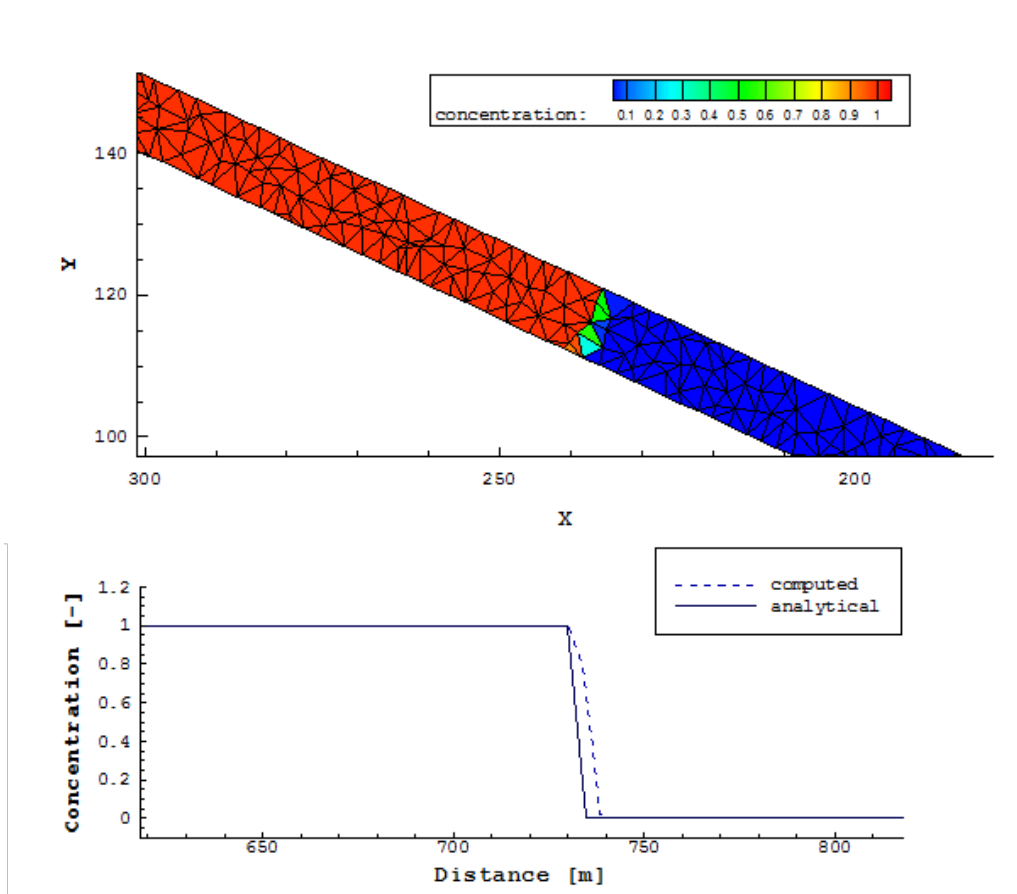
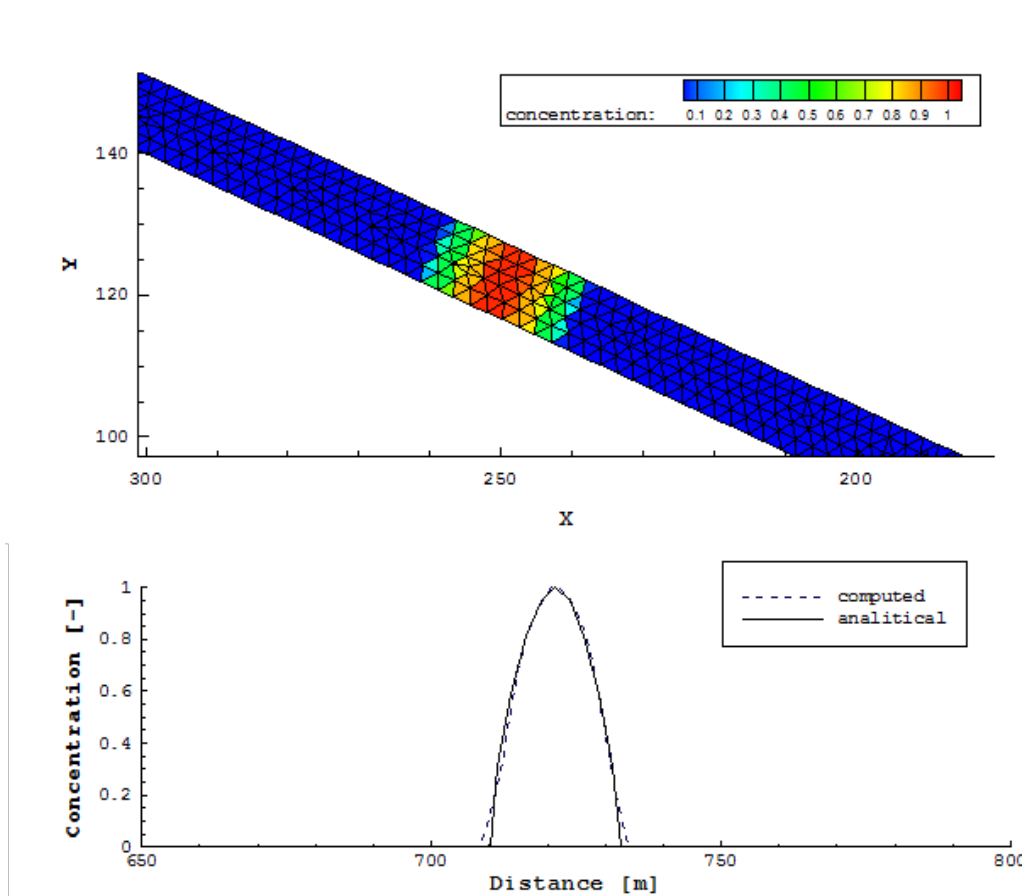**Figure 2.21** *ST_BC_1: case A, concentration after 200 s*

Figure 2.22 *ST_BP_1: case B, concentration after 200 s*

3

Model Coupling

3.1 Coupling of Domains of Same Type

3.1.1 COUPL_1: Sequential two-way coupling with backwater effects

3.1.1.1 Intention

This test is intended to verify and test the coupling mechanism of a sequential coupling with mutual data exchange between the sub-domains.

3.1.1.2 Description

Two sub-domains are simulated in a combined simulation. Sub-domain A is situated upstream of sub-domain B (Figure 3.1). Both sub-domains are coupled using a two-way coupling mechanism via an outflow hydrograph boundary and an inflow hydrograph boundary. Discharges are passed to the downstream sub-domain and water surface elevations are passed in upstream direction. Backwater effects travel in direction of the upstream sub-domain, caused by an outflow weir at the downstream end.

3.1.1.3 Geometry and initial conditions

- Case A: 1-D $\rightarrow$ 1-D coupling

  Both channels have trapezoidal cross sections and 1 km length, 40 m base width and a slope of 1.25 ‰.

- Case B: 2-D $\rightarrow$ 2-D coupling

  Both channels have rectangular cross sections, 1 km length, 40m width and a slope of 1.25 ‰.
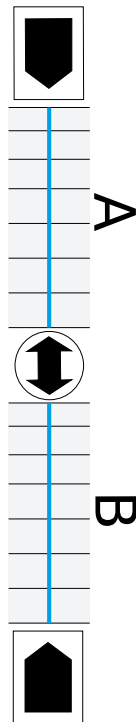
- Initially the sub-domains are dry.

Figure 3.1 *Sub-domains A and B*

3.1.2   COUPL_2: River network modelling

3.1.2.1   Intention

This test is intended to verify and test the coupling mechanism of a junction and a bifurcation in a simple river network configuration.

3.1.2.2   Description

Five sub-domains are simulated in a combined simulation (Figure 3.2). Sub-domains A and D are situated upstream and have an inflow hydrograph defined. Both river branches merge together into the sub-domain B. At its downstream end sub-domain B splits up into sub-domains E and C. The junction and the bifurcation are simulated with one-way couplings, i.e. the upstream sub-domains are not influenced by the water elevations in the downstream sub-domains.

3.1.2.3   Geometry and initial conditions

- All channels have trapezoidal cross sections, 1 km length, 40 m base width and a slope of 1.25 ‰.
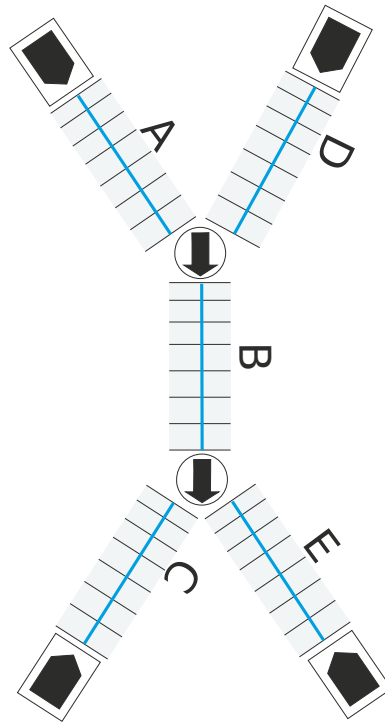- Initially all sub-domains are dry.

Figure 3.2 *Five sub-domains*

3.1.3   COUPL_3: Sequential coupling with sediment transport

3.1.3.1   Intention

This test is intended to verify and test the coupling mechanism of a sequential coupling in a morphological simulation with bed load and suspended load transport.

3.1.3.2   Description

Sub-domain A is situated upstream of sub-domain B (Figure 3.1). Both sub-domains are coupled using a 1-way coupling mechanism via an outflow hq-relation boundary and an inflow hydrograph boundary. Both sub-domains simulate single grain bed load transport. The bed load and suspended load are exchanged over the sequential coupling using corresponding outflow and inflow boundary conditions. At the upstream end of sub-domain A discharge, bed load and suspended load enter with constant rates.

3.1.3.3   Geometry and initial conditions

- Both channels have trapezoidal cross sections and 1 km length, 40 m base width and a slope of 1.25 ‰.
- The initial conditions for the hydraulics are uniform flow conditions in both sub-domains.
- The initial conditions for the bed load transport are set to equilibrium conditions (using IODown and IOup boundary conditions).
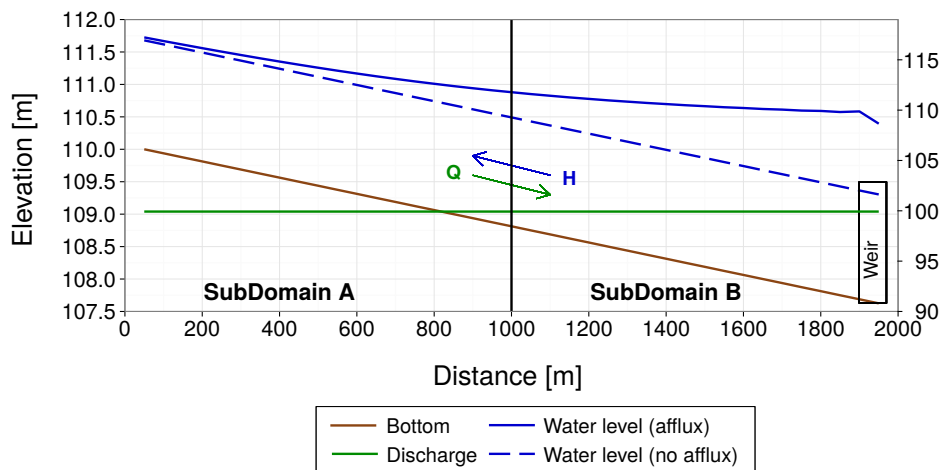
Figure 3.3 *COUPL_1_BC: Water surface elevation and discharge profiles of coupled BASEchain sub-domains A and B*

- The initial condition for the suspended sediment transport is set to concentration 0.0.

3.2 Results

3.2.1 Coupling Test Cases for Domains of Same Type

3.2.1.1 COUPL_1: Sequential two-way coupling with backwater effects

3.2.1.1.1 COUPL_1_BC: Results obtained by BASEchain

The coupling test case has been simulated starting from dry conditions in sub-domains A and B. Discharge is passed into downstream direction and water surface elevations are passed into upstream direction (two-way coupling).

After some time steady state conditions are reached in the coupled simulation. The backwater curve travels seamlessly from the downstream sub-domain into the upstream sub-domain. The discharge is constant over the whole domain. This indicates a correct two-way coupling which is capable to consider backwater effects from downstream.

3.2.1.1.2 COUPL_1_BP: Results obtained by BASEplane

As in the 1-D case, this coupling test case has been simulated starting from dry conditions in sub-domains A and B. Discharge is passed in downstream direction and water surface elevations are passed in upstream direction (two-way coupling).

After some time steady state conditions are reached in the coupled simulation. The backwater curve again enters seamlessly the upstream sub-domain. The velocity has no jumps at the interface between the sub-domains and it reduces in direction of the downstream weir.
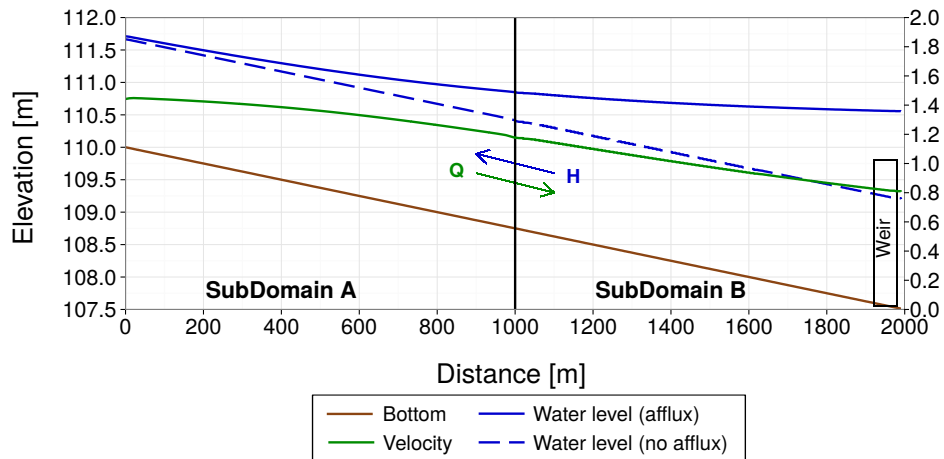
Figure 3.4 *COUPL_1_BP: Water surface elevation and discharge profiles of coupled BASEplane sub-domains A and B*
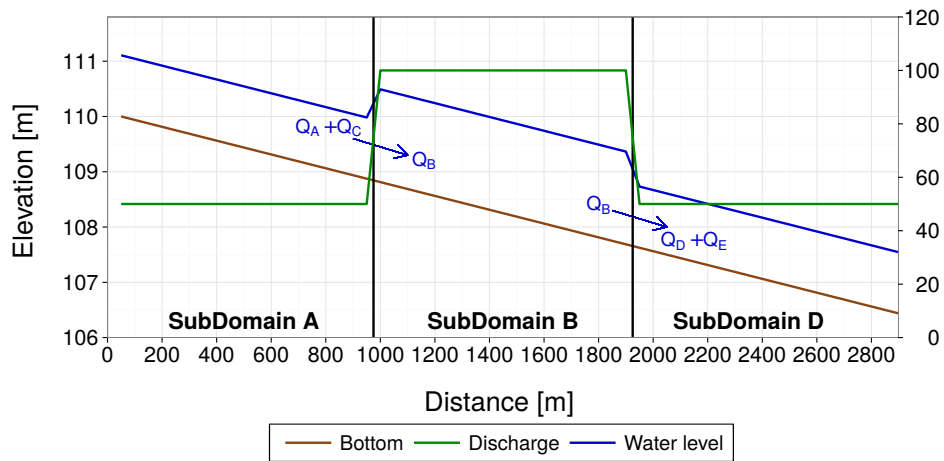
Figure 3.5 *COUPL_2: Water surface elevation and discharge profiles in the coupled river network*

3.2.1.2   COUPL_2: River network modelling

This test case has been simulated starting from dry conditions in all sub-domains. Sub-domains A and C have inflow hydrographs defined and sub-domains D and E have uniform outflow conditions defined. The discharge is passed in downstream direction using a one-way coupling, i.e. influences of downstream water levels are neglected here.

Figure 3.5 shows steady state conditions reached in the coupled simulation. The discharges leaving sub-domains A and C flow together at a junction and pass their discharge into sub-domain B. Sub-domain B splits up into two downstream sub-domains D and E (Figure 3.2). The mass conservation is fulfilled in this simulation and the discharges are distributed correctly among the sub-domains. The water levels show jumps at the coupling interfaces because their influence on the upstream sub-domains was neglected here.
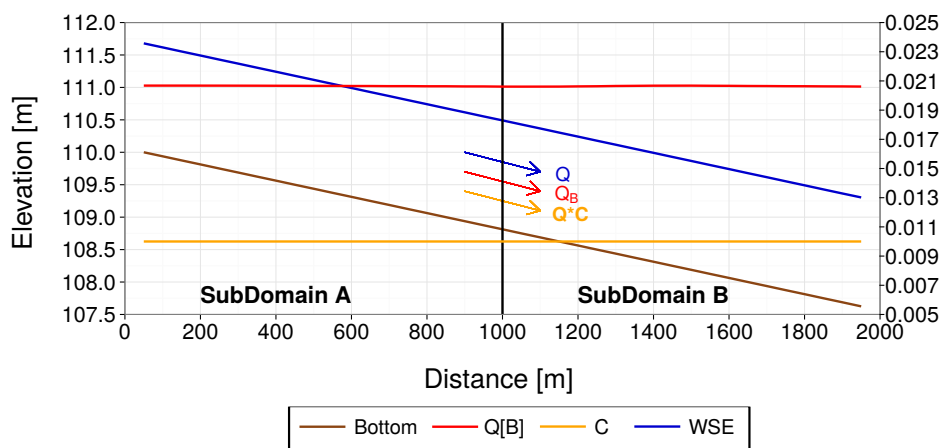
Figure 3.6 *COUPL_3: Bed load transport rate and concentration profile of coupled BASEchain sub-domains A and B*

3.2.1.3 COUPL_3: Sequential coupling with sediment transport

The sub-domains have been simulated starting from uniform flow conditions. Discharge, transported bed load and suspended loads are passed from the upstream sub-domain into downstream direction via boundary conditions. Inflowing discharge and inflow concentration profiles are constant over time. The bed load inflow is set in a way that equilibrium transport is achieved.

The simulation is run until steady state conditions are reached. As can be seen in Figure 3.6, the bed load transport rate is nearly constant in both sub-domains. Also the concentration profiles are the same in the upstream and downstream sub-domains and equal the input concentration profile.

4

Subsurface Flow

4.1 BASEsub1: Saturated, confined water flow in soil

As a first test case a steady-state subsurface flow on a mesh of 1x0.1x1 m extension is selected with a cell size of $\Delta x = 0.01$ m. The soil's hydraulic conductivity is set to $k_f = 0.001$ m/s in the computational domain.

At the west boundary a hydrostatic pressure head of $h_w = 5$ m is set as boundary condition. At the east boundary a hydrostatic pressure head of $h_e = 2$ m is set. The other boundary cells are treated as bounce-back boundaries. The numerical constant ϑ is set to 3 and the time step size is selected as $\Delta t = 1$ s. This configuration results in a confined subsurface flow with constant pressure gradient of $\partial h/\partial x = 3$.

The obtained results are illustrated in Figure 4.1, for a cross sectional cut along the y-axis through the domain. The model is able to reproduce the analytical linear gradient of the pore-water pressure head within the domain.

4.2 BASEsub2: Water infiltration into partially-saturated soil

To test the flow in the partially-saturated zone, the unsteady downward propagation of an infiltration front is simulated. This test case was chosen as described by Vogel et al. (2001), in order to allow for comparison with their results obtained with a FE-model. The same test was also successfully modelled with a LBM-approach by Ginzburg et al. (2004).

The domain is discretized with 1x0.1x1 m extension. The cell size is set to $\Delta x = 0.01$ m. As retention model the approach after Van-Genuchten and Mualem (VGM) is applied. The soil parameters are chosen as $\alpha = 0.8$ 1/m and $n_\nu = 1.09$ based on the tabulated values given in Vogel et al. (2001). The hydraulic conductivity is set to $k_f = 5.55$E–7 m/s. A negative hydrostatic pore-water pressure distribution (= soil suction) is set as initial condition with a total head of $h_w + z = -10$ m. At the top of the domain a constant infiltration source
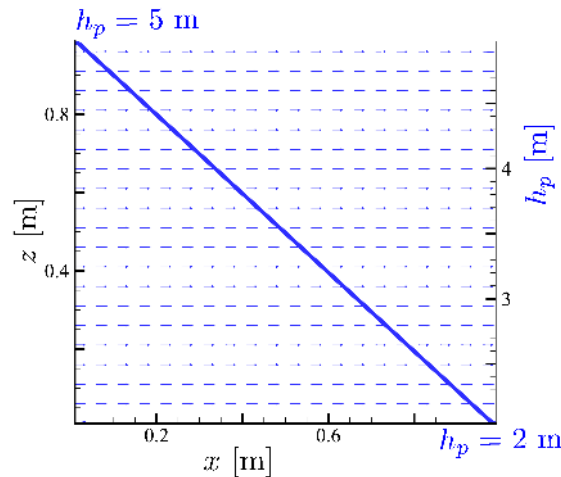
Figure 4.1 *BASEsub1: Saturated confined flow through homogenous domain with linear pore-water pressure gradient.*

is placed with q_{inf} = 2.78E–7 m/s, leading to the formation of an infiltration front. The moisture-formulation is used with $\vartheta = 3$ and a time step size of $\Delta t = 10$ s is selected.

The temporal evolution of the infiltration front was simulated with the air-entry pressure head h_s= -1E7 m. The propagation of the infiltration front through is depicted in Figure 4.2 and Figure 4.3. The propagation of the infiltration front obtained with BASEsub (left) is compared with the results obtained with the FE-model by Vogel (right), indicating only negligible deviations. The results also confirm the previously obtained results by Ginzburg.

4.3 BASEsub3: Unsaturated seepage flow

In this setup the 3-D unsteady seepage flow through a laboratory dam is modelled. The experimental investigations were made at the TU Berlin (see Pham Van (2009) for details). The homogeneous dam is 0.6 m high, 4.0 m long and 0.4 m wide. The sand dam material has a hydraulic conductivity of k_f = 9.5E-4 m/s (the value given in the reference of 0.95E-4 m/s is supposed to be a typo). The saturation moisture content of $\theta_s = 0.49$ and a residual water content of $\theta_R = 0.01$ were measured in the laboratory.

The dam is discretized with cubic cells and a cell size $\Delta x = 0.01$ m. As initial condition the measured water content of $\theta_0 = 0.115$ is applied within the whole dam. The air entry pressure is set to $h_s = $ -0.035 m. At the upstream embankment slope a time dependent pressure boundary is applied, simulating the rising water level in the reservoir left to the dam. At the downstream embankment slope a seepage boundary is set. The VGM-model is used with $\alpha = 14.5$ 1/m and $n = 2.68$ for sand material. These VGM parameters were chosen with respect to the listed values in Vogel et al. (2001) and are not based on laboratory measurements. For the simulation, the moisture-formulation of the Richard's equation is used with $\vartheta = 3$ and a time step size of $\Delta t = 0.25$ s is selected.

Measured and simulated results of the seepage line are depicted in Figure 4.4 . The temporal development of the seepage line is in accordance between measurement and simulation throughout time.
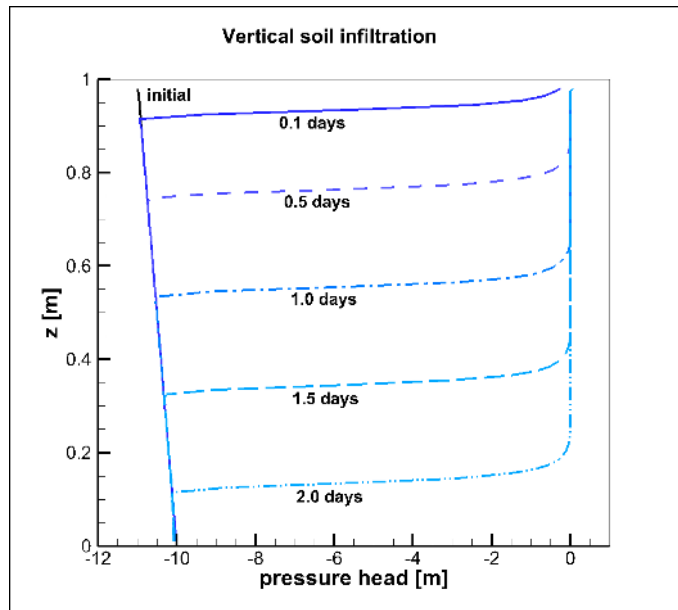
Figure 4.2 *BASEsub2: Downward propagation of infiltration front into partially-saturated soil. Simulation results of BASEsub.*
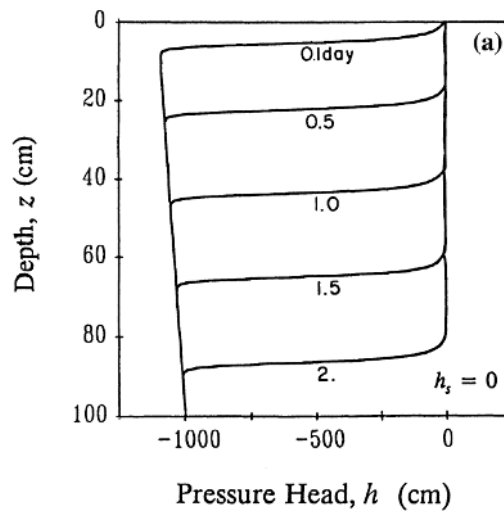

Figure 4.3 *BASEsub2: Downward propagation of infiltration front into partially-saturated soil. Results of Vogel et al.*
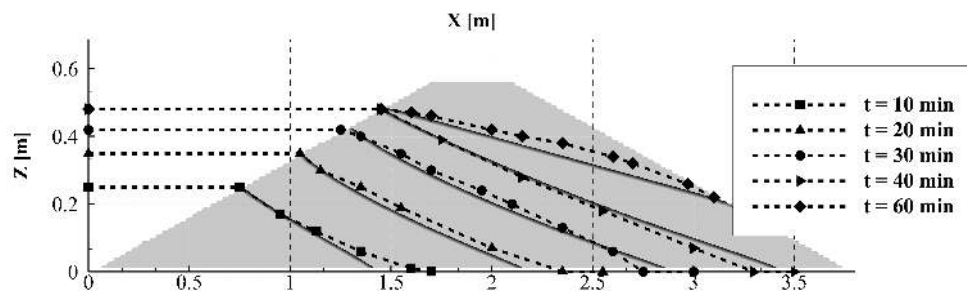

Figure 4.4 *BASEsub3: Cross sectional view of measured (dashed) and simulated (bold) development of seepage line, caused by water infiltration due to a rising reservoir at the east dam side.*

5

References

Ginzburg, I., Carlier, J. and Kao, C. (2004). Lattice Boltzmann approach to Richard's equation. *Proceedings of the CMWR XV, CT Miller*, 583–597. Chapel Hill, NC, USA.

Kassem, A.A. and Chaudhry, M.H. (1998). Comparison of Coupled and Semicoupled Numerical Models for Alluvial Channels. *Journal of Hydraulic Engineering-Asce*, 124(8): 794–802.

Pham Van, S. (2009). Application of Different Model Concepts for Simulation of Two-Phase Flow Processes in Porous Media with Fault Zones [PhD thesis]: Book Series of Institute of Civil Engineering, Volume 3, Technische Universität Berlin.

Saiedi, S. (1981b). A non-dimensional coupled numerical model of alluvial flow. *International Journal of Sediment Research*, 9(2): 59–79.

Saiedi, S. (1981a). Experience in design of a laboratory flume for sediment studies. *International Journal of Sediment Research*, 8(3): 89–101.

Soni, J.P. (1981). Laboratory Study of Aggradation in Alluvial Channels. *Journal of Hydrology*, 49(1-2): 87–106.

Soni, J.P., Garde, R.J. and Ranga Raju, K.G. (1980). Aggradation in Streams Due to Overloading. *Journal of the Hydraulics Division-Asce*, 106(1): 117–132.

Soulis, J.V. (2002). A fully coupled numerical technique for 2-D bed morphology calculations. *International Journal for Numerical Methods in Fluids*, 38(1): 71–98.

Toro, E.F. (2001). Shock-Capturing Methods for Free-Surface Shallow Flows. *John Wiley*, Chichester, New York.

Tseng, M.-H. (1999). Verification of 1-D Transcritical Flow Model in Channels. *Natl. Sci. Counc. ROC(A)*, 23(5): 654–664.

Valiani, A., Caleffi, V. and Zanni, A. (2002). Case Study: Malpasset Dam-Break Simulation using a Two-Dimensional Finite Volume Method. *Journal of Hydraulic Engineering*, 128(5): 460–472.

Vasquez, J.A., Steffler, P.M. and Millar, R.G. (2005). River 2D Morphology, Part I: Straight Alluvial Channels. *17th canadian hydrotechnical conference*, Edmonton, Alberta.

Vogel, T., van Genuchten, M.T. and Cislerova, M. (2001). Effect of the shape of the soil hydraulic functions near saturation on variably saturated flow predictions. *Advances in Water Resources*, 24: 133–144.

Wiberg, P.L. and Smith, J.D. (1989). Model for Calculating Bed-Load Transport of Sediment. *Journal of Hydraulic Engineering-ASCE*, 115(1): 101–123.

Yoon, T. and Kang, S.-K. (2004). Finite Volume Model for Two-Dimensional Shallow Water Flows on Unstructured Grids. *Journal of Hydraulic Engineering*, 130(7): 678–688.

APPENDIX

of **BASEMENT**

Contents

1

Notation

1.1 Super- and Subscripts

$(.)_B$	Property of top most soil layer for bed load (active layer)
$(.)_{cr}$	Critical value
$(.)_i, (.)_j, (.)_k$	Index corresponding to three dimensional Cartesian coordinate system $\mathbb{R}^3$ with coordinates(x, y, z)
$(.)_g$	Property corresponding to g^{th} grain size class
$(.)_L$	Property of the left hand side
$(.)_l$	Lateral property
$(.)^n$	n^{th} step of time integration
$(.)_R$	Property on the right hand side
$(.)_S$	Property at water surface
$(.)_{Sub}$	Property of bed material storage layer (sub layer)
$(.)_x, (.)_y, (.)_z$	Property corresponding to three dimensional Cartesian coordinate system $\mathbb{R}^3$ with coordinates (x, y, z)
$(.)_{(.),(.)}$	Dual property, i.e. $Q_{B,x}$ = bed load discharge in x direction
$(.)_{(.)_{(.)},(.)}$	Triple property, i.e. $Q_{B_g,x}$ = bed load discharge of grain size class g in x direction

1.2 Differential Operators

$\dfrac{\mathrm{d}}{\mathrm{d}x}$ Differential operator for derivation with respect to variable x

$\dfrac{\mathrm{d}^n}{\mathrm{d}x^n}$ Differential operator for derivation of order n w. r. to var. x

$\dfrac{\partial}{\partial x}$ Partial differential operator for derivation w. r. to variable x

$\dfrac{\partial^n}{\partial x^n}$ Partial differential operator for derivation of order n w. r. to var. x

∇ Nabla operator. In three-dimensional Cartesian coordinate system $\mathbb{R}^3$ with coordinates $(x, y, z): \nabla = \left(\dfrac{\partial}{\partial x}, \dfrac{\partial}{\partial y}, \dfrac{\partial}{\partial z}\right)$

1.3   English Symbols

Symbol	*Unit*	*Definition*
A	$[m^2]$	Wetted cross section area
$\boldsymbol{a}$	$[m/s^2]$	Acceleration
A_{red}	$[m^2]$	Reduced area
c	$[m/s]$	Wave speed
c_f	$[-]$	Friction coefficient
C	$[-]$	Concentration
c_μ	$[-]$	Dimensionless coefficient (used for turb. kin. viscosity)
d_g	$[m]$	Mean grain size according to Meyer-Peter and Müller
d_m	$[m]$	Arithmetic mean grain size according to Meyer-Peter and Müller
$\boldsymbol{F}(\boldsymbol{U}), \boldsymbol{G}(\boldsymbol{U})$	$[-]$	Flux vectors
$\boldsymbol{F}$	$[N]$	Force
$\boldsymbol{g}$	$[m/s^2]$	Gravity
h	$[m]$	Water depth, flow depth
h_B	$[m]$	Thickness of active layer
K	$[m^3/s]$	Conveyance factor $K = k_{str} A R^{2/3}$
k_{str}	$[m^{1/3}/s]$	Strickler factor
k_s	$[mm]$	Equivalent roughness height
m	$[kg]$	Mass
$\boldsymbol{n}$	$[m/s]$	Normal (directed outward) unit flow vector of a computational cell
ng	$[-]$	Total number of grain size classes
M	$[Ns]$	Momentum
P	$[N/m^2]$	Pressure
P	$[m]$	Hydraulic perimeter
p	$[-]$	Porosity
p_B	$[-]$	Porosity of bed material in active layer
p_{Sub}	$[-]$	Porosity of bed material in sub layer
Q	$[m^3/s]$	Stream- or surface discharge
Q_B	$[m^3/s]$	Total bed load flux for cross section
q_{B_g}	$[m^3/s/m]$	Total bed load flux of grain size class g per unit width
$q_{B_g,x}, q_{B_g,y}$	$[m^3/s/m]$	Cartesian components of total bed load flux q_{B_g}
$q_{B_g,xx}, q_{B_g,yy}$	$[m^3/s/m]$	Cartesian comp. of bed load flux due to stream forces
$q_{B_g,xy}, q_{B_g,yx}$	$[m^3/s/m]$	Cartesian comp. of lateral bed load flux
q_l	$[m^2/s]$	Specific lateral discharge (discharge per meter of length)
R	$[m]$	Hydraulic radius
$\boldsymbol{S}(\boldsymbol{U})$	$[-]$	Vector of source terms
S_f	$[-]$	Friction slope
S_B	$[-]$	Bed slope
S_g	$[m/s]$	Suspended load source per cell and grain size class
Sf_g	$[m/s]$	Active layer floor source per cell and grain size class
Sl_g	$[m/s]$	Local sediment source per cell and grain size class
t	$[s]$	Time
$\boldsymbol{U}$	$[-]$	Vector of conserved variables
$\boldsymbol{u}$	$[m/s]$	Flow velocity vector with Cartesian components (u, v, w)

u_*	$[m/s]$	Shear stress velocity
u, v, w	$[m/s]$	Cartesian components of flow velocity vector $\boldsymbol{u}$
$\bar{u}, \bar{v}$	$[m/s]$	Cartesian components of depth averaged flow velocity
u_B, v_B, w_B	$[m/s]$	Cartesian components of flow velocity at bottom
u_S, v_S, w_S	$[m/s]$	Cartesian components of flow velocity at water surface
V	$[m^3]$	Volume
x, y, z	$[-]$	Cartesian coordinate axes
x, y, z	$[m]$	Distance in corresponding Cartesian direction
z_B	$[m]$	Bottom elevation
z_S	$[m]$	Water surface elevation
z_F	$[m]$	Elevation of active layer floor

1.4 Greek Symbols

Symbol	Unit	Definition
α_B	$[-]$	Empirical parameter depending on the dimensionless shear stress of the mixture
β_g	$[-]$	Volumetric fraction of grain size class g in active layer
$\beta_{g,Sub}$	$[-]$	Volumetric fraction of grain size class g in active layer
Γ	$[-]$	Eddy diffusivity
κ	$[-]$	Kármán constant
Δt	$[s]$	Computational time step
Δt_h	$[s]$	Time step for hydraulic sequence
Δt_s	$[s]$	Time step for sediment transport sequence
Δt_{seq}	$[s]$	Overall, sequential time step
$\Delta x, \Delta y, \Delta z$	$[m]$	Grid spacing according to three dimensional Cartesian Coordinate system $\mathbb{R}^3$ with coordinates (x, y, z)
η	$[kg/ms]$	Molecular viscosity
ν	$[m^2/s]$	Kinematic viscosity μ/ρ
ν_ε	$[m^2/s]$	Isotropic eddy viscosity
ν_t	$[m^2/s]$	Turbulent kinematic viscosity $\nu_t = \nu_0 + c_\mu u^* h$
ν_0	$[m^2/s]$	Base kinematics eddy viscosity
ρ	$[kg/m^3]$	Mass density (fluid)
ρ_s	$[kg/m^3]$	Bed material density
$\tau_{B,x}, \tau_{B,y}$	$[N/m^2]$	Cartesian components of bottom shear stress vector
$\boldsymbol{\tau}_B$	$[N/m^2]$	Vector of shear stress at bottom due to water flow
$\tau_{B_g,crit}$	$[N/m^2]$	Critical shear stress related to grain size d_g
$\tau_{S,x}, \tau_{S,y}$	$[N/m^2]$	Cartesian components of surface shear stress vector
$\boldsymbol{\tau}_S$	$[N/m^2]$	Vector of shear stress at water surface (e.g due to wind)
Ω	$[m^2]$	Area of an element
ξ_g	$[-]$	Hiding factor