



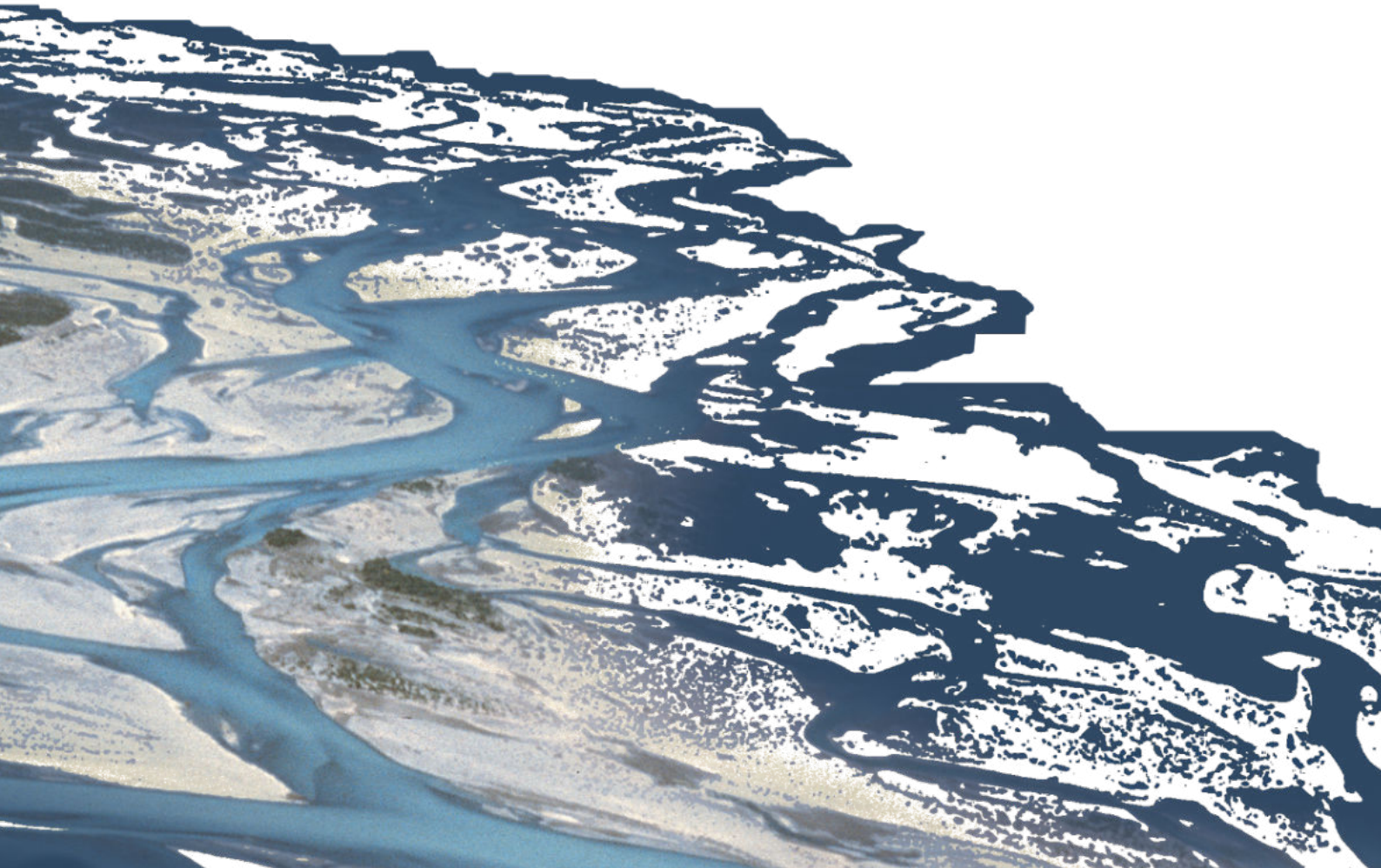
**BASEMENT**

**BASIC SIMULATION ENVIRONMENT  
FOR MODELLING OF ENVIRONMENTAL  
FLOWS AND NATURAL HAZARDS**

---

# **SYSTEM MANUALS**

**VERSION 4.1.0  
JUNE 2024**







---

# Preface

## Preface to Version 4.0

### Back to the roots

In October 2002, we began the development of the software BASEMENT. From the start we have been guided by two principle objectives. Firstly, we wanted to develop a simulation tool that is suitable for practical applications, as well as scientific research. Secondly, we deliberately resolved that the entire modeling workflow should be achievable with freely available software. We will strive to adhere to these principles in the future. Since the beginning, the development team has consisted of very dedicated and talented people – the so-called BASE(wo)men – and the team has always been of a manageable size. We would like to take this opportunity to thank all those who have supported and contributed to BASEMENT. We also greatly appreciate the thanks that we have received from many happy users over the years. The list of contributors is available on our website.

A full 20 years after the birth of the project, we are proud to present version 4 of BASEMENT. This new major version is similar to the very first version of BASEMENT – released in 2006 – which consisted of the two modules BASEchain and BASEplane and had a modular workflow. Version 4 also consists of two modules, BASEMD and BASEHPC; MD stands for “Multi-Domain” and corresponds to the previous version 2.8.2, HPC stands for “High-Performance Computing” and corresponds to the previous version 3.2. Thus, all features from the two previous versions are retained and integrated into a common simulation workflow. With this structure, we proudly find ourselves going “back to our roots”.

The development of the software would not have been possible without the support of the Swiss Federal Office for the Environment (FOEN) and the backing of the BASEMENT advisory board. The FOEN has financially supported the BASEMENT project since 2002, which is greatly appreciated.

On behalf of the project team Dr. D. Vetsch, Project Director Prof. Dr. R. Boes, Director VAW

February, 2023



---

# Preamble

**VERSION 4.1.0**

*June 2024*

## Credits

### Contributors

Over the years, many enthusiastic engineers and developers have contributed to the development, testing and documentation of BASEMENT. An up-to-date overview of the current development team, along with current and former contributors, can be found on our website:

<https://basement.ethz.ch/people>

**Commissioned and co-financed by**

Swiss Federal Office for the Environment (FOEN)

**Contact**

Website: <https://www.basement.ethz.ch>

User forum: <https://people.ee.ethz.ch/~basement/forum>

ETH Zurich / Laboratory of Hydraulics, Glaciology and Hydrology (VAW)

For list of contributors see <https://www.basement.ethz.ch>



Laboratory of Hydraulics,  
Hydrology and Glaciology



Eidgenössische Technische Hochschule Zürich  
Swiss Federal Institute of Technology Zurich

**Citation Advice**

*For System Manuals:*

Vetsch D., Siviglia A., Bacigaluppi P., Brown, A., Bürgler M., Caponi F., Conde D., Gerke E., Halso M., Kammerer S., Koch A., Peter S., Vanzo D., Vonwiller L., Weberndorfer M. 2024. System Manuals of BASEMENT, Version 4.1.0. Laboratory of Hydraulics, Glaciology and Hydrology (VAW). ETH Zurich. Available from <https://www.basement.ethz.ch>. [date of access].

*For Website:*

BASEMENT – Basic Simulation Environment for Computation of Environmental Flow and Natural Hazard Simulation, 2024. <https://www.basement.ethz.ch>

*For Software:*

BASEMENT – Basic Simulation Environment for Computation of Environmental Flows and Natural Hazard Simulation. Version 4.1.0 ETH Zurich, VAW, 2024.



---

# License

## End-User License Agreement (EULA)

THIS EULA IS INTENDED FOR COMMERCIAL AND NON-COMMERCIAL PURPOSES. FOR QUESTIONS RELATED TO THIS AGREEMENT PLEASE CONTACT: Dr. David Vetsch, [basement@ethz.ch](mailto:basement@ethz.ch)

This End-User License Agreement (“EULA”) is a legal agreement between you (“You”) (an individual or acting on behalf of a company) and ETH Zurich, Raemistrasse 101, 8092 Zurich (Switzerland) (“ETH Zurich”) for the binary software code of **BASEMENT** and associated media, and may include “online” or electronic documentation (“SOFTWARE”).

The SOFTWARE simulates water flow, sediment and scalar transport in rivers and according interaction in consideration of movable boundaries and morphological changes. Further information and description of the SOFTWARE is available here: <https://basement.ethz.ch/>

The SOFTWARE is protected by copyright laws. The SOFTWARE is hereby licensed, not sold.

In order to install and use the SOFTWARE, You must indicate agreement with the following terms and conditions by clicking “ACCEPT” at the end of this EULA during the installation process.

### 1 LICENSE GRANT

- (i) ETH Zurich hereby grants to You, and in case You are acting on behalf of a company also to the employees of such company, a free-of-charge, single, non-exclusive, world-wide, non-transferable, non-sublicensable right to install, execute and display the SOFTWARE on device(s) running a validly licensed copy of the operating system for which the SOFTWARE was designed. Such rights are granted for commercial and non-commercial purposes.
- (ii) With respect to electronic documents included with the SOFTWARE, You may make an unlimited number of copies (either in hardcopy or electronic form), provided that such copies shall be used only for internal purposes and are not republished or distributed to any third party.

## 2 USE OF RESULTS GENERATED BY THE SOFTWARE

You are allowed to use the content generated by the SOFTWARE (“SOFTWARE RESULTS”) for commercial and non-commercial purposes.

Note that any attribution (e.g. ETH Zurich logo) on the SOFTWARE RESULTS must be retained. You are not allowed to alter, cancel or fade, after a few seconds, such attribution.

## 3 THIRD PARTY CODE

The SOFTWARE may contain other program code from third parties. A list of other third party code and libraries used by this SOFTWARE is available here: <https://basement.ethz.ch/about/thirdpartysoftware>

Their license applies to such third party code and libraries contained herein. Refer to the above internet site for the licenses and copyrights.

## 4 DURATION OF LICENSE AND TERMINATION

This EULA enters into effect on the date of acceptance of this EULA by You. This EULA,

- (i) may be terminated by ETH Zurich at any time for any reason;
- (ii) will terminate automatically without notice from ETH Zurich if (a) You fail to comply with any term(s) of this EULA or (b) You refuse, after the notification in accordance with clause 10 (i), to accept the new EULA term and conditions provided by ETH Zurich;
- (iii) is terminated as soon as You cease to use the SOFTWARE and destroy all copies, full or partial, of the SOFTWARE;

Upon termination pursuant to (i) and (ii), you must cease all use of the SOFTWARE and destroy all copies, full or partial, of the SOFTWARE.

## 5 OBLIGATIONS OF YOU

- (i) You may not remove or alter any copyright notices on any and all copies of the SOFTWARE.
- (ii) You may not distribute or assign the SOFTWARE or any copy thereof to third parties. You may not rent, lease, sell, lend, transfer, redistribute, or sublicense the SOFTWARE to any third party.
- (iii) You may not reverse engineer, decompile or disassemble the SOFTWARE, except and only to the extent that such activity is expressly permitted by applicable law despite this limitation.
- (iv) You may not reproduce, modify or adapt the SOFTWARE, except and only to the extent that such activity is expressly permitted by applicable law despite this limitation.
- (v) You must comply with all applicable laws.

## **6 OWNERSHIP**

Except as expressly licensed to You in this EULA, ETH Zurich and its licensors retains all right, title, and interest in and to the SOFTWARE. All title and copyrights in and to the SOFTWARE (including but not limited to any images, photographs, animations, video, audio, music, text, and “applets” incorporated into the SOFTWARE), the accompanying materials, and any copies of the SOFTWARE are owned by ETH Zurich and its licensors. The SOFTWARE is protected by copyright laws. Therefore, You must treat the SOFTWARE like any other copyrighted material. All rights not expressly granted are reserved by ETH Zurich.

## **7 MAINTENANCE, SUPPORT, UPGRADES OR NEW RELEASES**

ETH Zurich has no obligation to provide maintenance, support, upgrades, new releases, enhancements or modifications and disclaims all costs associated with service, repair or correction of the SOFTWARE. If any supplemental software code is provided to You by ETH Zurich, this supplemental software code shall be considered part of the SOFTWARE and is subject to the terms and conditions of this EULA if not otherwise explicitly written. It is expressly acknowledged by You that no rights to receive maintenance, support, upgrades, new releases, enhancements or modifications may be derived from this EULA.

## **8 NO WARRANTY**

YOU EXPRESSLY ACKNOWLEDGE AND AGREE THAT USE OF THE SOFTWARE IS AT YOUR SOLE RISK AND THAT THE ENTIRE RISK AS TO SATISFACTORY QUALITY, PERFORMANCE, ACCURACY, AND EFFORT IS WITH YOU. TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, THE SOFTWARE AND ANY SERVICES PERFORMED OR PROVIDED BY THE SOFTWARE ARE PROVIDED “AS IS” AND “AS AVAILABLE”, WITH ALL FAULTS AND WITHOUT WARRANTY OF ANY KIND, AND ETH ZURICH HEREBY DISCLAIMS ALL WARRANTIES AND CONDITIONS WITH RESPECT TO THE SOFTWARE AND ANY SERVICES, EITHER EXPRESS, IMPLIED, OR STATUTORY, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES AND/OR CONDITIONS OF MERCHANTABILITY, OF SATISFACTORY QUALITY, OF FITNESS FOR A PARTICULAR PURPOSE, OF ACCURACY, OF QUIET ENJOYMENT, AND OF NON-INFRINGEMENT OF THIRD-PARTY RIGHTS. ETH ZURICH DOES NOT WARRANT AGAINST INTERFERENCE WITH YOUR ENJOYMENT OF THE SOFTWARE, THAT THE FUNCTIONS CONTAINED IN OR SERVICES PERFORMED OR PROVIDED BY THE SOFTWARE WILL MEET YOUR REQUIREMENTS, THAT THE OPERATION OF THE SOFTWARE OR SERVICES WILL BE UNINTERRUPTED OR ERROR-FREE, OR THAT DEFECTS IN THE SOFTWARE OR SERVICES WILL BE CORRECTED. NO ORAL OR WRITTEN INFORMATION OR ADVICE GIVEN BY ETH ZURICH OR ITS AUTHORIZED REPRESENTATIVE SHALL CREATE A WARRANTY. SHOULD THE SOFTWARE OR SERVICES PROVE DEFECTIVE, YOU ASSUME THE ENTIRE COST OF ALL NECESSARY SERVICING, REPAIR, OR CORRECTION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES OR LIMITATIONS ON APPLICABLE STATUTORY RIGHTS OF A CONSUMER, SO THE ABOVE EXCLUSION AND LIMITATIONS MAY NOT APPLY TO YOU.

## **9 LIABILITY**

In no event shall ETH Zurich be liable for any damages (including, without limitation, lost profits, business interruption, or lost information) arising from the use of or inability

to use the SOFTWARE through You or the employees of the company You are legally representing. In no event will ETH Zurich be liable for loss of data or for indirect, special, incidental, consequential (including loss of profit), or other damages based in contract, tort or otherwise.

The above limitation of liability shall not be construed to amend or limit any party's statutory liability.

## 10 GENERAL PROVISIONS

- (i) ETH Zurich reserves the right to change the terms and conditions of this EULA at any point in time. In such event, ETH Zurich will notify You in due time of the changes to the terms of the EULA.
- (ii) Rights and duties derived from this EULA shall not be transferred to third parties without the written acceptance of the ETH Zurich.
- (iii) You shall not infer from this EULA any other rights, including licenses, than those that are explicitly stated herein.
- (iv) This EULA shall exclusively be governed by and interpreted in accordance with the laws of Switzerland, without reference to its conflict of laws principles. The exclusive place of jurisdiction is Zurich (Switzerland).

## 11 ACKNOWLEDGMENT

You acknowledge that you have read this EULA, understand it, and had an opportunity to seek independent legal advice prior to agreeing to it. In consideration of ETH Zurich agreeing to provide the SOFTWARE, You agree to be bound by the terms and conditions of this EULA. You further agree that it is the complete and exclusive statement of the agreement between you and ETH Zurich, which supersedes any proposal or prior agreement, oral or written, and any other communication between you and ETH Zurich relating to the subject of this EULA.

### **Notice:**

Third party software copyright notices and third party software licenses can be found in the appendix.



**BASIC SIMULATION ENVIRONMENT  
FOR MODELLING OF ENVIRONMENTAL  
FLOWS AND NATURAL HAZARDS**

---

# **INTRODUCTION & INSTALLATION**

**VERSION 4.1.0  
JUNE 2024**



**BASEMENT**



# Contents

<b>1</b>	<b>Summary of Contents</b>	<b>5</b>
1.1	Introduction . . . . .	5
1.2	Content of System Manuals . . . . .	5
1.2.1	Introduction & Installation . . . . .	5
1.2.2	User Manual . . . . .	5
1.2.3	Reference Manual . . . . .	6
1.2.4	Tutorials and Test Cases . . . . .	6
<b>2</b>	<b>Setup and First Start</b>	<b>7</b>
2.1	Setup and First Start . . . . .	7
2.1.1	Operating System Requirements . . . . .	7
2.1.1.1	Microsoft Windows . . . . .	7
2.1.1.2	Linux . . . . .	7
2.1.1.3	Hardware Configuration . . . . .	8
2.1.2	Installing under Windows . . . . .	9
2.1.2.1	Getting the binaries . . . . .	9
2.1.2.2	Installation procedure under Windows 10 and 11 . . . . .	9
2.1.3	Installing under Linux . . . . .	10
2.1.3.1	Getting the binaries . . . . .	10
2.1.3.2	Installation procedure . . . . .	10
2.2	Running BASEMENT . . . . .	12
2.2.1	Windows 10 and 11 . . . . .	12
2.2.2	Linux . . . . .	12
<b>3</b>	<b>Migration Guide</b>	<b>15</b>
3.1	General . . . . .	15
3.2	Migration from version 2.8.2 . . . . .	15
3.2.1	Convertor for command (BMC) files . . . . .	15
3.2.1.1	Conversion in the GUI . . . . .	15
3.2.1.2	Conversion with BMv4_BASEMD_converter . . . . .	16
3.2.2	Restart simulation from result (h5) file . . . . .	16
3.2.3	Adapt setup file . . . . .	17
<b>4</b>	<b>Release Notes</b>	<b>19</b>
4.1	Version 4.1 . . . . .	19
4.1.1	General . . . . .	19
4.1.1.1	Bug fixes . . . . .	19

4.1.1.2	Known Issues . . . . .	19
4.1.2	Module BASEHPC . . . . .	19
4.1.2.1	Bug fixes . . . . .	20
4.1.3	Module BASEMD . . . . .	20
4.1.3.1	Bug fixes . . . . .	20
4.2	Version 4.0.2 . . . . .	21
4.2.1	General . . . . .	21
4.2.1.1	Known Issues . . . . .	21
4.2.2	Module BASEHPC . . . . .	21
4.2.3	Module BASEMD . . . . .	22
4.2.3.1	Bug fixes . . . . .	22
4.2.3.2	Known Issues . . . . .	22
4.3	Version 4.0.1 . . . . .	22
4.3.1	General . . . . .	22
4.3.1.1	Known Issues . . . . .	23
4.3.2	Module BASEHPC . . . . .	23
4.3.2.1	Bug fixes . . . . .	23
4.3.3	Module BASEMD . . . . .	23
4.3.3.1	Bug fixes . . . . .	23
4.3.3.2	Known Issues . . . . .	23
4.4	Version 4.0 . . . . .	24
4.4.1	General . . . . .	24
4.4.2	Known Issues . . . . .	24
4.4.3	Module BASEHPC . . . . .	24
4.4.3.1	Bug fixes . . . . .	24
4.4.4	Module BASEMD . . . . .	25
4.4.4.1	General . . . . .	25
4.4.4.2	Bug fixes . . . . .	25
4.4.4.3	Known Issues . . . . .	25
4.5	Notes on older releases . . . . .	25
<b>5</b>	<b>Summary of Features Version 4.0</b>	<b>27</b>
5.1	Features of module BASEMD . . . . .	27
5.1.1	Submodule BASEchain . . . . .	27
5.1.1.1	Numerics . . . . .	27
5.1.1.2	Geometry . . . . .	27
5.1.1.3	Hydrodynamics . . . . .	27
5.1.1.4	Sediment Transport: Bed load . . . . .	28
5.1.1.5	Sediment transport : Suspended load . . . . .	28
5.1.1.6	Sediment transport : Gravitational induced transport . . . . .	28
5.1.1.7	Output Variables/Formats . . . . .	28
5.1.2	Submodule BASEplane . . . . .	29
5.1.2.1	Numerics . . . . .	29
5.1.2.2	Geometry . . . . .	29
5.1.2.3	Hydrodynamics . . . . .	29
5.1.2.4	Sediment Transport: Bed load . . . . .	30
5.1.2.5	Sediment transport : Suspended load . . . . .	30
5.1.2.6	Sediment transport : Gravitational induced transport . . . . .	30
5.1.2.7	Output Variables/Formats . . . . .	30



---

5.1.3	Submodule BASEsub . . . . .	31
5.1.3.1	Numerics . . . . .	31
5.1.3.2	Geometry . . . . .	31
5.1.3.3	Hydrodynamics . . . . .	31
5.1.3.4	Output Variables/Formats . . . . .	31
5.2	Features of module BASEHPC . . . . .	32
5.2.1	Hydrodynamics . . . . .	32
5.2.2	Morphodynamics . . . . .	34
5.2.2.1	Bed load transport . . . . .	34
5.2.2.2	Suspended load transport . . . . .	36
5.2.3	Tracers . . . . .	37
5.2.4	Temperature . . . . .	38
5.2.5	Vegetation . . . . .	39
5.2.6	Output Variables Summary . . . . .	39
<b>6</b>	<b>References</b>	<b>43</b>



---

## Summary of Contents

### 1.1 Introduction

BASEMENT version 4 is a freeware simulation tool for hydro- and morphodynamic modelling developed at the Laboratory of Hydraulics, Hydrology and Glaciology (VAW) of ETH Zurich. The software provides a powerful tool for the simulation of river hydro- and morphodynamics. BASEMENT system manual provides information about BASEMENT software and guides the user through the process of using BASEMENT version 4.

### 1.2 Content of System Manuals

The documentation is composed of four parts, the Introduction & Installation, the User Manual, the Reference Manual and the Tutorials & Test Cases.

#### 1.2.1 Introduction & Installation

This part aims at introducing BASEMENT version 4 to the user by presenting the novelties and changes of the released version. First, the installation procedure is explained for Windows and Linux operating systems. Then, the differences between BASEMENT modules BASEMD and BASEHPC are described in the migration guide for the users already familiar with BASEMENT. The release notes summarize the changes introduced by BASEMENT version 4 and the summary of features provides an overview of the available functionalities of the new version.

#### 1.2.2 User Manual

The user manual provides information about the simulation environment of BASEMENT version 4. The modelling procedure presents the three-stage process, namely the pre-processing, the numerical simulation and post-processing. The numerical simulation is

carefully described in the simulation workflow section. The graphical user interface (GUI) provides a user-friendly tool to assist the user during the numerical simulation process.

### **1.2.3 Reference Manual**

The reference manual provides information about the mathematical models and numerical approximations implemented in BASEMENT version 4.

### **1.2.4 Tutorials and Test Cases**

This part is composed of three tutorials and two test cases. The tutorials guide the user throughout the pre-processing, the numerical simulation and post-processing stages of BASEMENT version 4 by taking a section of the river Flaz in Graubünden as example for the numerical simulation. The test cases aim at testing the performance and accuracy of the simulations performed with BASEMENT version 4 by standardized test cases, namely the circular dam break and the conical dune.



# 2

---

## Setup and First Start

### 2.1 Setup and First Start

#### 2.1.1 Operating System Requirements

##### 2.1.1.1 Microsoft Windows

BASEMENT version 4 has been tested for MS Windows 10 and 11. For the latest news concerning new features and current changes, please visit the webpage <https://www.basement.ethz.ch>.

##### 2.1.1.2 Linux

BASEMENT version 4 is available for the following Linux (x86-64) systems:

- Ubuntu 18.04 (LTS), alias “Bionic Beaver”:
  - Kernel version 5.4
  - GNU C Library (glibc) version 2.27
  - VTK-version: 6.3
  - GPU driver version:
    - \* Kepler architecture and later: at least 418.39
    - \* Tesla architecture: in [384.111, 385.00) or in [410.72, 411.00)
- Ubuntu 20.04 (LTS), alias “Focal Fossa”:
  - Kernel version 5.13
  - GNU C Library (glibc) version 2.31
  - VTK-version: 9

- GPU driver version:
  - \* Kepler architecture and later: at least 418.39
  - \* Tesla architecture: in [384.111, 385.00) or in [410.72, 411.00)
- Ubuntu 22.04 (LTS), alias “Jammy Jellyfish”:
  - Kernel version 5.15
  - GNU C Library (glibc) version 2.35
  - GPU driver version:
    - \* Kepler architecture and later: at least 418.39
    - \* Tesla architecture: in [384.111, 385.00) or in [410.72, 411.00)

The binaries were compiled and tested on both Linux systems. Binaries without GUI should run on debian-based linux systems.

### 2.1.1.3 Hardware Configuration

We recommend the following hardware configurations:

#### 2.1.1.3.1 CPU multi-core processors (x86/x86-64)

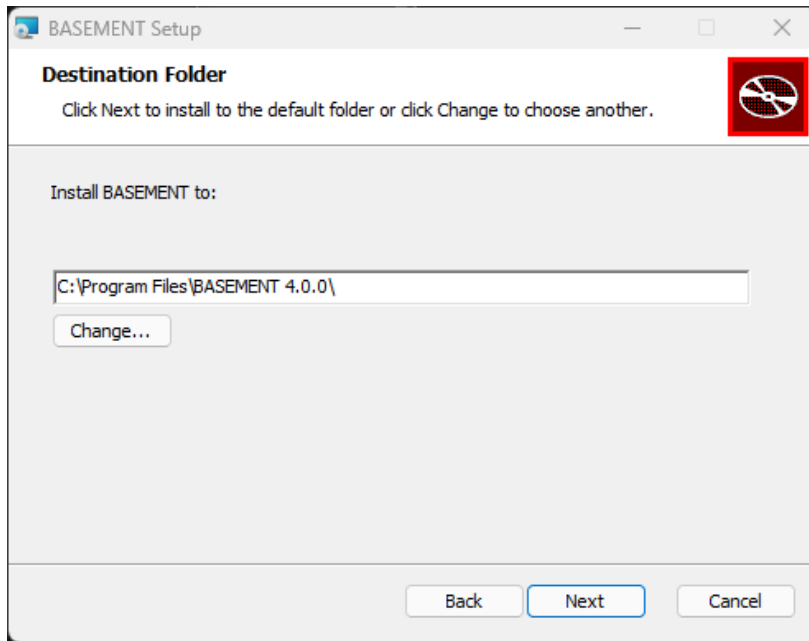
- Intel (Xeon, 12 to 18 Cores, dual socket)
- 1 GB per core
- Minimum of 2.8 GHz

#### 2.1.1.3.2 Graphical Processing Units (GPUs)

Please note that the GPU-support of BASEMENT version 4 is **only** possible for CUDA-enabled (Compute Unified Device Architecture) GPUs produced by NVIDIA. BASEMENT version 4 has been specifically tested with GPUs listed in [2.1](#).

*Table 2.1 GPU hardware used for the numerical simulations*

Card	Tesla K20	Tesla P100	GTX 1080 Ti	GTX 1070 Ti	GTX 1050 Ti	RTX 2080Ti
Memory [GB]	5	12	11	8	4	11
Architecture	Kepler	Pascal	Pascal	Pascal	Pascal	Turing
Bandwidth[GB/s]	208	549	484	256	112	616
CUDA cores	2496	3584	3584	2432	768	4352



*Figure 2.1 Select the installation folder.*

## 2.1.2 Installing under Windows

BASEMENT version 4 is available for Microsoft Windows Windows 10 and 11 operating system.

### 2.1.2.1 Getting the binaries

First of all, you need to get a copy of the latest software package. Therefore go to the project webpage <https://basement.ethz.ch> and download the latest version (BASEMENT version 4.x) free of charge.

### 2.1.2.2 Installation procedure under Windows 10 and 11

Please note, that existing installations may not automatically detected by the installer. Therefore, uninstall any previous BASEMENT version before installing a more recent version e.g. using the link in the start menu. After downloading the version from the project webpage, start the installation by double-clicking on the BASEMENT installer.

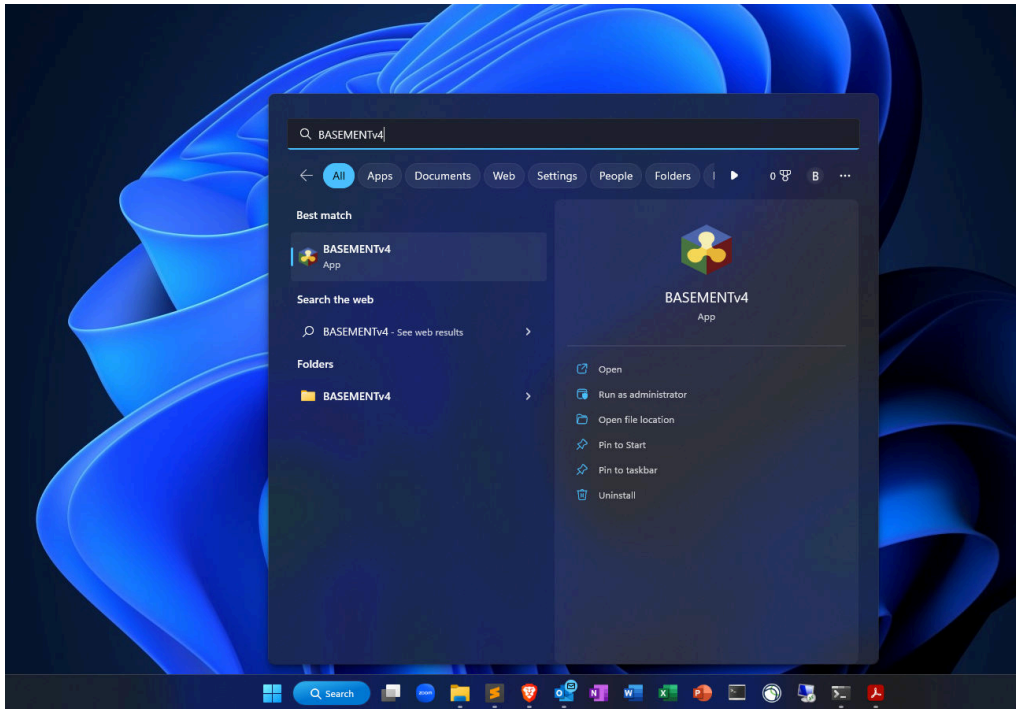
#### *Step 1: Accepting the license agreement*

Please read the License Agreement carefully and click on the 'I accept' button if you accept the terms and conditions and proceed with the installation.

#### *Step 2: Select the installation folder*

After accepting the License agreement, you can choose where to install the binaries. The recommended location is "C:\Program Files\BASEMENT 4.0" (Figure 2.1). You are free to choose any other directory.

#### *Step 3 and 4: Confirming and finishing the installation*



*Figure 2.2 Start BASEMENT over the icon in the Start Menu folder.*

Clicking ‘Install’ will start the installation process. After all files are copied, a final window informs about the success of the installation. Click ‘Finish’ to close the installer.

#### *Step 5: Start BASEMENT*

You can start the program by opening the Start Menu, navigating to the Start Menu folder of BASEMENT and clicking on the program icon of BASEMENT version 4 (Figure 2.2). To create a Desktop shortcut, simply drag the program icon to your Desktop. Clicking on the BASEMENT icon runs the program as a standalone application including a simple graphical user interface (to run BASEMENT in batch mode see the section Run the programm in the User Manual documentation part).

The graphical user interface should appear as in Figure 2.3.

### 2.1.3 Installing under Linux

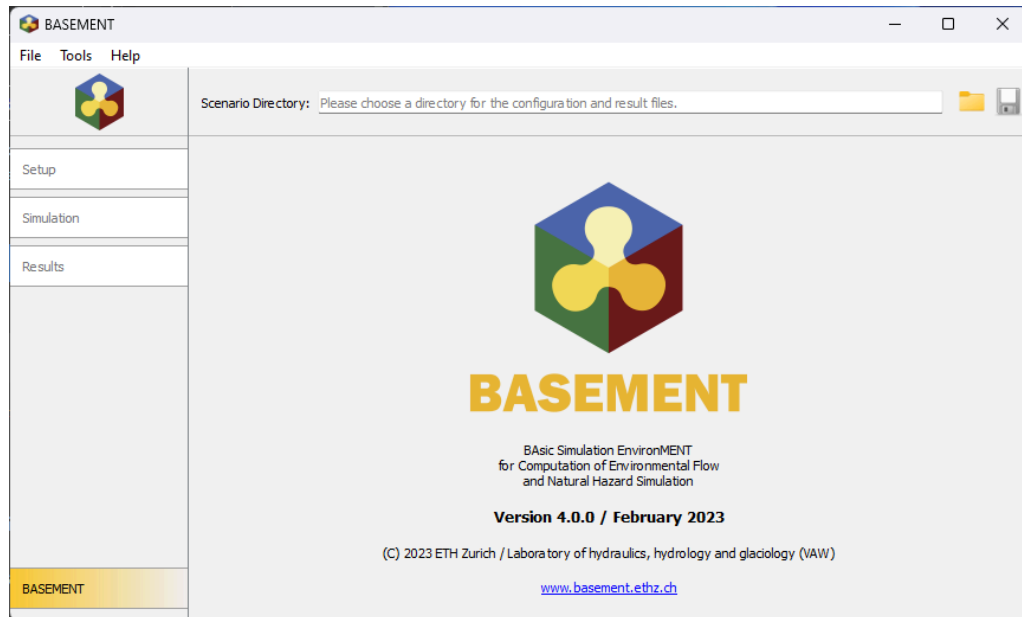
#### 2.1.3.1 Getting the binaries

You need to get a copy of the actual distribution as described in the Windows installation section. You can download the most recent version from the projects webpage <https://basement.ethz.ch>.

#### 2.1.3.2 Installation procedure

##### **Step 1: Preparation of the installation**

Extract the downloaded package and change to the directory containing the installation script. Make the installation script executable by running (replace \* by the BASEMENT and Ubuntu version number):



*Figure 2.3 The graphical user interface.*

```
$ chmod +x BASEMENT_v*_linux64_ubuntu*.sh
```

To run the setup enter

```
$ ./BASEMENT_v*_linux64_ubuntu*.sh
```

and follow the instructions. You have to read and accept the license text.

The BASEMENT Debian package file (\*.deb) is extracted.

### **Step 2: Install the Debian package**

Administrative rights are required to install the package using dpkg. Therefore prefix the command with “sudo” (replace \*\*\* by the version number):

```
$ sudo dpkg -i BASEMENT-***.deb
```

The Debian package automatically detects if your configuration misses one of the required packages. In such a case you can either use

```
$ sudo apt-get -f install
```

to install all missing packages automatically or, in case you want to install dependencies manually, you can perform

```
$ sudo apt-get install MISSING_PACKAGE
```

to install the missing packages from the repository. Replace `MISSING_PACKAGE` with the missing package name.

Previous versions of `basement` are automatically detected by the installer and deleted before the installation starts.

### Step 3: Run basement

If the installation of the package was successful, then the executables are copied to

```
/usr/bin
```

and the other program files are copied to

```
/usr/share/basement
```

You can now run `Basement` from the console by typing:

```
$ basement
```

### Step 4: In case of trouble

Report your issue and get in touch with other users in the `BASEMENT` forum: <https://people.ee.ethz.ch/~basement/forum/>

## 2.2 Running BASEMENT

### 2.2.1 Windows 10 and 11

When running `BASEMENT` under Microsoft Windows operating system, the easiest way to start a simulation is by clicking on `BASEMENT` icon. After running, `BASEMENT` will open the graphical user interface. You have to select the scenario directory that contains all the configuration files and binaries by pressing on the folder icon, where you can load the path to the scenario directory.

The configuration is done in three steps (three `.json` files) that can be edited using `BASEMENT` graphical user interface or `JSON` editors. Each step is executed independently (setup, simulation and results) and the generated files are saved in the scenario directory containing the command and auxiliary files. The procedure to run numerical simulations with `BASEMENT` using the graphical user interface (GUI) or in batch mode is explained in the User manual.

### 2.2.2 Linux

`BASEMENT` runs as a console application without program icon. On Linux, open a console and type:

```
$ basement
```

to start the program (if no environment variables have been set, change into your `'bin'` directory of the installation path). The GUI starting page is appearing. You have now to select or create the scenario directory in the scenario directory field. The configuration

of the .json files (3 steps) is the same as for Microsoft Windows (see Section 2.2.1) and information about the use of the GUI is given in the section ‘BASEMENT Graphical User Interface’ of the User Manual. Selecting the executables and running a simulation on Linux or using batch mode works the same way as it does on Windows explained in the User Manual. *Notice:* **The command filename must not contain any spaces or special characters like ä, ö, ü, è, etc.**





# 3

---

## Migration Guide

### 3.1 General

BASEMENT version 4 consists of two main modules, namely BASEMD (MD stands for Multi Domain) and BASEHPC (HPC stands for High Performance Computing, based on the possibility to use GPGPUs). The BASEMD module refers to the former version 2.8.2 and the BASEHPC module to the version 3.2. Version 4 combines all the functionality of the previous versions, providing an unified GUI and workflow. The migration of existing model setups for versions 2.8.2 and 3.2 will be explained in the following.

### 3.2 Migration from version 2.8.2

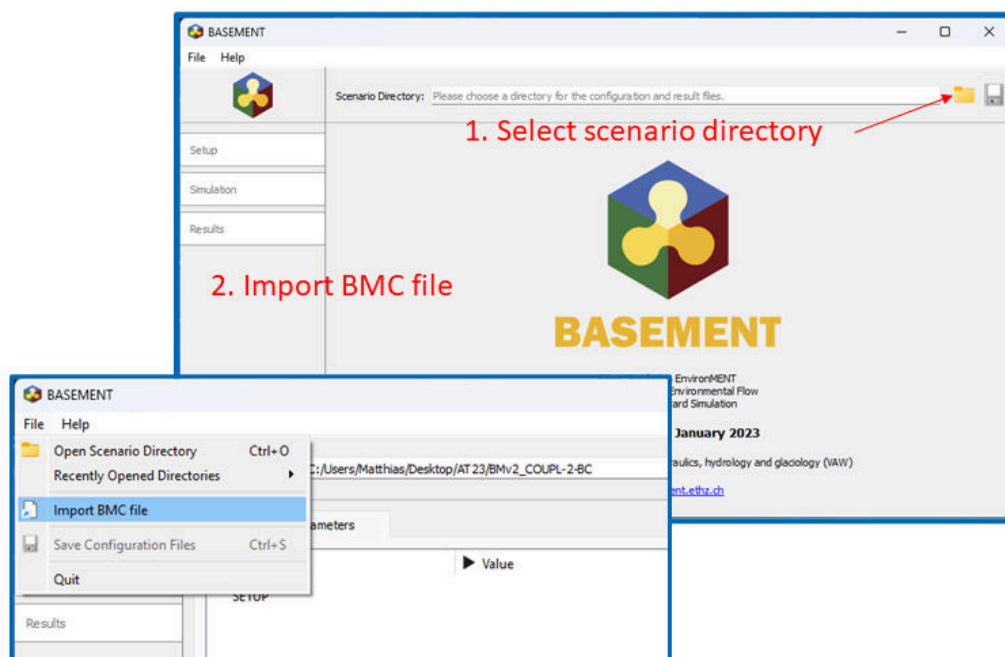
#### 3.2.1 Convertor for command (BMC) files

Running simulations with BASEMENT version 4 requires the model setup to be defined in files with the JSON format (model.json), while for version 2.8.2, model definitions were done in command (BMC) files. In order to migrate an existing model from version 2.8.2, the command file must first be converted into the JSON format. The conversion of a BMC file to a JSON file can be done either with the graphical user interface (GUI) of BASEMENT version 4, or with the standalone executable `BMv4_BASEMD_converter`.

##### 3.2.1.1 Conversion in the GUI

A BMC file can be converted into a model.json file with the following steps:

1. Select the scenarion directory in the GUI, that contains the command (BMC) file.
2. Convert the BMC file via the “Import BMC file” function in the File-Menu.
3. Follow the simulation workflow as described in the User Manual.



*Figure 3.1 Migration from version 2.8.2 by selection of a scenario directory and importing a BMC file.*

### 3.2.1.2 Conversion with BMv4\_BASEMD\_converter

Alternatively, a BMC file can be converted into a model.json file with the executable BMv4\_BASEMD\_converter via the command line.

On linux, run the command:

```
$ BMv4_BASEMD_converter myModel.bmc -o model.json
```

On Windows, run the command (assuming you added the installation location of BASEMENT to the Path environment variable)

```
> BMv4_BASEMD_converter.exe myModel.bmc -o model.json
```

### 3.2.2 Restart simulation from result (h5) file

The output format CGNS became deprecated with BASEMENT version 4 and was replaced by the h5 format. In order to restart a simulation with a restart file in CGNS format, the file must be converted to h5 format. Therefore, the python script cgns\_to\_h5.py is provided on the BASEMENT website <https://basement.ethz.ch/download/tools/python-scripts.html>.

The script requires python 3 and the module h5py, which can be installed, e.g. with 'pip install h5py'. To convert a CGNS file into an h5 file run the following command:

```
$ python cgns_to_h5.py restart.cgns
```



```
{
  "SETUP":{
    "simulation_name":"Flaz_A_run",
    "DOMAIN": {
      "PHYSICAL_PROPERTIES": {
        "gravity": 9.81
      },
      "BASEPLANE_2D": {
        "GEOMETRY": {
          "mesh_file": "Flaz_mesh.2dm",
          "INTERPOLATION": {
            "method": "weighted"
          }
        }
      }
    }
  }
}
```

```
{
  "SETUP":{
    "simulation_name":"Flaz_A_run",
    "BASEHPC": {
      "PHYSICAL_PROPERTIES": {
        "gravity": 9.81
      },
      "BASEPLANE_2D": {
        "GEOMETRY": {
          "mesh_file": "Flaz_mesh.2dm",
          "INTERPOLATION": {
            "method": "weighted"
          }
        }
      }
    }
  }
}
```

**Figure 3.2** Migration from version 3.2 by replacing the keyword “DOMAIN” with “BASEHPC” the setup file.

Then follow the simulation workflow as described in the User Manual. # Migration from version 3.2

### 3.2.3 Adapt setup file

The combination of all the functionalities of the previous versions required a minor modification in the structure of the setup file (model.json). Therefore, the migration of an existing simulation from version 3.2 requires to adapt the setup file (model.json) in a text editor of choice and replace the keyword “DOMAIN” to “BASEHPC” as illustrated in (Figure 3.2).



---

# Release Notes

## 4.1 Version 4.1

### 4.1.1 General

This minor release contains two new features (TEMPERATURE and MIXED-SIZE BEDLOAD TRANSPORT) for the BASEHPC module and some bugfixes; see details below.

#### 4.1.1.1 Bug fixes

- The help menu of the BASEMENT GUI now includes each documentation manual as an individual document. The Introduction and Installation Manual, User Manual, Reference Manuals, Tutorials Manual, and Test Cases Manual can each be viewed separately.

#### 4.1.1.2 Known Issues

- Model setup fails if there is a dot ‘.’ in the working directory path. For example: “MySimulations/Sim1.1/” does not work, while “MySimulations/Sim1\_1/” works.
- Aborting a simulation using Ctrl+C can corrupt the HDF5 result file.
- Abort bottom may no work as intended, i.e. simulation does not stop.

### 4.1.2 Module BASEHPC

- A temperature transport model for the BASEHPC module. The temperature transport model is based on a passive advection and diffusion approach for a scalar quantity and allows the simulation of surface water temperature dynamics. Details on the module and the input parameters are in the BASEHPC Reference Manual.

- A mixed-size bedload transport model for the BASEHPC module. The model is based on the Hirano-Exner model. Details on the module and the input parameters are in the BASEHPC Reference Manual. This new feature affects the structure of the MORPHOLOGY block of BASEHPC simulations, and existing model setups may not work without modifications.
- The SOURCE block for external bedload sources has been moved from MORPHOLOGY/SOURCE to MORPHOLOGY/BEDLOAD/SOURCE. This requires existing model.json files to be updated.
- The approach of Yalin and da Silva (2001) for determining the critical Shields parameter has been implemented for the BASEHPC module. The approach for determining the critical Shields parameter can be selected via the tag `theta_critical_approach` between:
  - constant: Constant critical Shields parameter provided via the tag `theta_critical_approach` (default: 0.047)
  - yalin: Approach of Yalin and da Silva (2001)
- BASEHPC executables for GPUs have been built using CUDA version 12.1 (Ubuntu 18.04) and 12.5 (Ubuntu 20.04, Ubuntu 22.04, Windows). The installation of a recent NVIDIA driver may be required to run simulations on GPUs.

#### 4.1.2.1 Bug fixes

- A bug made it impossible to load XDMF formatted simulation results of a BASEHPC simulations onto a mesh layer in QGIS. This has been fixed.
- The console output of BASEHPC executables did not contain information on written time steps. This has been fixed.
- The implementation of the Engelund and Hansen method for bedload transport of uniform bed material contained a bug, which has been fixed.

**WARNING:** The implementation of the mixed-size bedload transport model for BASEHPC affects the structure of the MORPHOLOGY block. Existing model setups (model.json) may not work without modifications. Existing model.json files containing the block MORPHOLOGY/SOURCE (bedload source) require a modification in order to function with the new version. The SOURCE block is now located at MORPHOLOGY/BEDLOAD/SOURCE.

### 4.1.3 Module BASEMD

#### 4.1.3.1 Bug fixes

- The console output of BASEMD simulations was not displayed properly in the GUI. This has been fixed.

Note: Existing installations of BASEMENT version 3.0.1 (or earlier) are not automatically detected by the updated installer. Therefore, uninstall any previous version of BASEMENT e.g. using the link in the Start Menu before installing the newest version.

## 4.2 Version 4.0.2

### 4.2.1 General

- A new panel has been added to the GUI. This panel displays the information previously shown as tool-tip information, but also includes additional information on units, default values or admissible parameter ranges.

#### 4.2.1.1 Known Issues

- Visualization of results in QGIS requires manual modification of .xdmf file: change `<Xdmf Version="3.0">` to `<Xdmf Version="2.0">` .
- Model setup fails if there is a dot ‘.’ in the working directory path. For example: “MySimulations/Sim1.1/” does not work, while “MySimulations/Sim1\_1/” works.
- Aborting a simulation using Ctrl+C can corrupt the HDF5 result file.
- Abort bottom may no work as intended, i.e. simulation does not stop.

### 4.2.2 Module BASEHPC

- For LINKED boundary conditions and the INTERNAL boundary condition `hqrelation_internal`, the inflow conditions at the downstream boundary can be better controlled by the user via the newly introduced tag `type_downstream`. Four options are available as downstream inflow conditions. For all options, the mass inflow flux is calculated based on outflow at upstream boundary as previously. The difference lies in the calculation of the momentum flux over the boundary:
  - `no_momentum`: No momentum flux over the boundary.
  - `froude_downstream`: Momentum flux calculated based on Froude number in the elements of the downstream nodestring.
  - `froude`: Momentum flux calculated based on the Froude number specified via the parameter `froude_number`.
  - `uniform`: Momentum flux calculated based on uniform flow conditions for the slope specified via the parameter `slope`. In previous versions, the various INTERNAL and LINKED boundary differed in how the momentum flux at the downstream boundary was calculated. The following combinations correspond to the implementation of the boundary conditions in previous versions:

type	type_downstream
<code>hqrelation_internal</code>	<code>no_momentum</code>
<code>weir_linked_constant</code>	<code>froude_downstream</code>
<code>weir_linked_dynamic</code>	<code>froude_downstream</code>
<code>hqrelation_linked</code>	<code>no_momentum</code>
<code>2way_hqrelation_linked</code>	<code>no_momentum</code>
<code>zhydrograph_linked</code>	<code>no_momentum</code>
<code>zhydrograph_linked_kinE</code>	<code>froude_downstream</code>

With this new feature, the LINKED boundary condition `zhydrograph_linked_kinE` became obsolete and has been removed.

### 4.2.3 Module BASEMD

#### 4.2.3.1 Bug fixes

- Absolute file paths on BASEMD simulations resulted in errors. This has been fixed.
- A bugfix implemented in v4.0.1 resulted many parameters of the BASEMD module being mandatory. This has been reverted. Those parameters are again initialized in the background if not explicitly defined by the user, results in the same behaviour as not explicitly specifying them in BASEMENT version 2.8.2 and earlier.

#### 4.2.3.2 Known Issues

##### BASEplane

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

##### BASEchain

- 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).

Note: Existing installations of BASEMENT version 3.0.1 (or earlier) are not automatically detected by the updated installer. Therefore, uninstall any previous version of BASEMENT e.g. using the link in the Start Menu before installing the newest version.

## 4.3 Version 4.0.1

### 4.3.1 General

- Log and error messages during setup, simulation and results steps of the workflow are now written to a file.
- The latest GPU architectures Ada and Hopper should now be supported, but have not been explicitly tested. An upgrade to the latest NVIDIA driver might be necessary.



#### 4.3.1.1 Known Issues

- Visualization of results in QGIS requires manual modification of .xdmf file: change `<Xdmf Version="3.0">` to `<Xdmf Version="2.0">` .
- Model setup fails if there is a dot ‘.’ in the working directory path. For example: “MySimulations/Sim1.1/” does not work, while “MySimulations/Sim1\_1/” works.
- Aborting a simulation using Ctrl+C can corrupt the HDF5 result file.
- Tool tip text is too long to be displayed completely.
- Abort bottom may no work as intended, i.e. simulation does not stop.

#### 4.3.2 Module BASEHPC

##### 4.3.2.1 Bug fixes

- A bug during writing the .xdmf output file resulted in unresponsive behaviour when visualizing the results in ParaView or even errors in QGIS. This bug has been fixed for visualizing the results in ParaView. Since the QGIS mesh layer does not yet support XDMF Version 3.0 it is necessary to manual replace the line `<Xdmf Version="3.0">` at the top of the .xdmf output file by `<Xdmf Version="2.0">` in order to load the simulation results onto the mesh layer.

#### 4.3.3 Module BASEMD

##### 4.3.3.1 Bug fixes

- A bug in the initialization of several model parameters (e.g. `theta_critical`, `theta_critical_approach`, etc.) resulted in BASEMD simulations to crash or not respond. Those parameters were previously intialized in the background if not explicitly defined by the user. The tags of those parameters are now mandatory in the `model.json` and are initialized with default parameters. The use of the default parameters results in the same behaviour as not explicitly specifying them in BASEMENT version 2.8.2 and earlier.

##### 4.3.3.2 Known Issues

#### BASEplane

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

#### BASEchain

- 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).

Note: Existing installations are not automatically detected by the updated installer. Therefore uninstall any previous version of BASEMENT e.g. using the link in the Start Menu before installing the newest version.

## 4.4 Version 4.0

### 4.4.1 General

BASEMENT version 4 consists of two modules, BASEMD and BASEHPC; MD stands for “Multi-Domain” and corresponds to the previous version 2.8.2, HPC stands for “High-Performance Computing” and corresponds to the previous version 3.2. Thus, all features from the two previous versions are retained and integrated into a common simulation workflow. The selection of the modules occurs via the keywords “BASEHPC” or “BASEMD” in the model definition file.

- BASEMENT now provides an installer package for Ubuntu 22.04, while the version for Ubuntu 16.04 was discontinued. Also, BASEMENT now requires CUDA 11.7.0.
- Keys in JSON configuration files were sorted by the GUI. This problem has been fixed.
- The trees representing the JSON files in the GUI are now expanded by default.
- Items in JSON objects can now be reordered using the GUI via drag and drop.
- The 1-D Grid Editor previously contained in version 2.8.2 is now integrated into the GUI of version 4.
- The migration of existing simulations from version 2.8.2 and 3.2 to version 4 are described in the migration guide (Section 3.1)

### 4.4.2 Known Issues

- Model setup fails if there is a dot ‘.’ in the working directory path. For example: “MySimulations/Sim1.1/” does not work, while “MySimulations/Sim1\_1/” works.
- Aborting a simulation using Ctrl+C can corrupt the HDF5 result file.
- Tool tip text is too long to be displayed completely.
- Abort bottom may no work as intended, i.e. simulation does not stop.

### 4.4.3 Module BASEHPC

#### 4.4.3.1 Bug fixes

- A bug causing a segmentation fault when using the MORPHOLOGY and TRACER block has been fixed.

#### 4.4.4 Module BASEMD

##### 4.4.4.1 General

###### BASEplane

- Implementation of time range for controller. Within the time range, the controller will be active (manipulated variable is based on PID/HID control). Outside of the time range, a default value for the manipulated variable must be specified.
- The CGNS format for saving simulation states was discontinued and replaced by the .h5 format. Existing CGNS files may be converted to the .h5 format with a Python-script provided on the website.

##### 4.4.4.2 Bug fixes

###### BASEchain

- A bug in the gravitational transport feature for mixed size sediment transport has been fixed.
- A bug in the implementation of the hiding\_exponent (mpm\_h and parker formulation) in mixed size sediment transport has been fixed.

##### 4.4.4.3 Known Issues

###### BASEplane

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

###### BASEchain

- 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).

Note: Existing installations are not automatically detected by the updated installer. Therefore uninstall any previous version of BASEMENT e.g. using the link in the Start Menu before installing the newest version.

## 4.5 Notes on older releases

Release notes of previous version can be found on the webpage: <https://basement.ethz.ch/download/software-download/release-notes>



---

## Summary of Features Version 4.0

### 5.1 Features of module BASEMD

#### 5.1.1 Submodule BASEchain

##### 5.1.1.1 Numerics

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

##### 5.1.1.2 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)

##### 5.1.1.3 Hydrodynamics

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 input file
Inner boundary	Weir/Gate

---

#### 5.1.1.4 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

---

#### 5.1.1.5 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

---

#### 5.1.1.6 Sediment transport : Gravitational induced transport

---

Approach	Geometrical modelling in longitudinal direction (1D) based on critical failure angles
Features	Single and multiple grain classes

---

#### 5.1.1.7 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: $z_{\text{dam}}$ (left/right), $z_{\text{bottom}}$ (mean), friction, $Q_b^{\text{cap}}$ , $C_{\text{tot}}$ Cross section: Water surface elevation WSE, energy level EL, velocity $v$ , wetted area $A$ , discharge $Q$ , Froude number $Fr$ , friction
Special output	Edge: continuity flux, momentum flux, $Q_{b,i}$ , $Q_s^{\text{adv}}$ , $Q_s^{\text{diff}}$ 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: $Q$ , $Q_b$ , $A$ , 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

---

## 5.1.2 Submodule BASEplane

### 5.1.2.1 Numerics

---

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

---

### 5.1.2.2 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

---

### 5.1.2.3 Hydrodynamics

---

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

---

**5.1.2.4 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

---

**5.1.2.5 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

---

**5.1.2.6 Sediment transport : Gravitational induced transport**


---

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

---

**5.1.2.7 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

---



### 5.1.3 Submodule BASEsub

#### 5.1.3.1 Numerics

---

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

---

#### 5.1.3.2 Geometry

---

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

---

#### 5.1.3.3 Hydrodynamics

---

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

---

#### 5.1.3.4 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

---

## 5.2 Features of module BASEHPC

### 5.2.1 Hydrodynamics

#### Riemann Solver

- HLLC, with hydrostatic reconstruction based on modified states (Duran et al., 2013).

#### Initial Conditions

- Dry
- Continue
- Region defined (regiondef) for water surface elevation or water depth, u and v

#### Boundary Conditions

- WALL : inviscid, default
- STANDARD (in parenthesis user-required data):
  - INFLOW: uniform (discharge; slope), froude (discharge, froude number), hq\_relation (H-Q relation), zhydrograph (water surface elevation, inflowPossible)
  - OUTFLOW: uniform (slope), zero\_gradient (-), weir (weir height, constant or dynamic poleni factor), hq\_relation (H-Q relation), dynamic wall (collapse time), zhydrograph (water surface elevation, inflowPossible)
- INTERNAL: dynamic wall (collapse time), internal wall (-), hq\_relation (H-Q relation)
- LINKED: hq\_relation (H-Q relation), 2 way hq\_relation (2 H-Q relations, time lag, water surface elevation upstream and downstream), weir (weir height, constant or dynamic poleni factor), zhydrograph (water surface elevation)

Inflow boundary conditions for INTERNAL and LINKED boundaries: - no\_momentum: no momentum flux over the boundary. - froude\_downstream: momentum flux calculated based on Froude number in the elements of the downstream nodestring. - froude: momentum flux calculated based on the Froude number specified via the parameter *froude\_number*. - uniform: momentum flux calculated based on uniform flow conditions for the slope specified via the parameter *slope*.

#### Parameters

- CFL
- Minimum water depth
- Fluid density
- Maximum time step

#### Friction

- Type: implicit Runge-Kutta 2nd order integration
- Closure types:
  - Manning
  - Strickler
  - Chezy
  - Bezzola

All require a default (or index defined) friction value.

### **Turbulence**

- Parameters:
  - Turbulence start time
  - On/off switch for diffusion of turbulent species (tke, epsilon)
  - Relaxation parameter for turbulent diffusion
  - On/off switch for friction generated turbulence
- Closure model for Reynolds stress:
  - Constant viscosity (default)
  - k-epsilon model
  - Mixing length model
- Initial conditions:
  - Zero, i.e. no turbulence (default)
  - Uniform values for tke and epsilon
  - Region defined values for tke and epsilon
  - Continue
- Boundary conditions (k-epsilon model)
  - STANDARD: k\_eps\_in, zero\_gradient\_out
- Source:
  - Type: k\_eps (force k-epsilon values directly)
  - Sink behaviours: exact (as prescribed), available (as prescribed or less), infinity (as much as possible)

### **Flood**

- Flood tracking of water front arrival time, maximum water depth, maximum flow velocity, maximum specific discharge, maximum bed shear stress (tracking time step)

### **Source (water volume)**

- Type: total (as discharge, m<sup>3</sup>/s), distributed (as rain, mm/h)
- Sink behaviors:
  - Exact (as prescribed)
  - Available (as prescribed or less)
  - Infinity (as much as possible)

## 5.2.2 Morphodynamics

### 5.2.2.1 Bed load transport

- HLL-type Approximate Riemann Solver (Soares-Frazão and Zech, 2011)
- Simple upwind scheme

### Parameters

- Morphodynamic start time
- Sediment porosity
- Sediment density
- Bed gradient calculation type
- Morphodynamic solver
- Active layer thickness

### Initial conditions

- Mesh file
- Continue

### Incipient motion

- van Rijn (1989) and Chen et al. (2010)
- Angle of repose

### Closure formula

- MPM (coefficient = 8, exponent = 15, critical threshold = 0.047)
- MPM-like (coefficient, exponent, critical threshold are adaptable)
- MPM\_multi (mixed-size sediment, hiding and exposition approach of Ashida and Michiue (1971); coefficient, exponent, critical threshold are adaptable)
- GRASS-like (coefficient, exponent, critical threshold are adaptable)
- Engelund and Hansen (1972)
- Smart and Jaeggi (1983)

- Wilcock and Crowe (2003) (mixed-size sediment)
- Hunziker and Jaeggi (2002) (mixed-size sediment)

### **Direction**

- Lateral bed slope effect (e.g. Ikeda, 1982, p. Talmon1995)
- Curvature effect

### **Bedload boundary conditions**

- WALL: inviscid, default
- STANDARD:
  - INFLOW: equilibrium (reference\_bed\_elevation), sedimentograph (sediment\_discharge), transport capacity (boundary\_factor)
  - OUTFLOW: equilibrium (reference\_bed\_elevation)
  - Weighting scheme for transport capacity and sedimentograph: geometrical (default), wetted area, conveyance
- LINKED: equilibrium (reference\_bed\_elevation at upstream boundary)

### **Bed material**

- Grain class
- Mixture
- Soil definition
- Soil assignment
- Soil initial conditions (restart from previous simulation)
- Fixed bed
- Parameters (correction accuracy and maximum number of iterations for fixed bed correction)

### **Gravitational Transport**

- Critical angle for wet/dry material
- update time step
- maximum settling velocity
- minimum bed elevation change

### **Source (sediment volume)**

- Type: total (as discharge, m<sup>3</sup>/s)
- Sink behaviors:
  - Exact (as prescribed)
  - Available (as prescribed or less)
  - Infinity (as much as possible)

### 5.2.2.2 Suspended load transport

#### Parameters

- Relaxation parameter
- Diffusivity coefficient (isotropic, longitudinal/transversal)
- Turbulent diffusivity coefficient (isotropic, longitudinal/transversal)
- Scaling (beta) factor for the suspended source terms

#### Settling Velocity

- Closure model for the grain settling velocity:
  - Van Rijn
  - Wu and Wang
  - Zhang
- Shape factor parameter (

#### Shear partition

- Skin friction parameter for suspended load computations
- Default value or region defined

#### Erosion rate

- Closure model for the erosion rate:
  - Van Rijn
  - Zyserman and Fredsoe
- Critical shields parameter (both)

#### Deposition rate

- Closure model for the deposition rate:
  - Lin
  - Xu
- Critical shields parameter (Xu)

#### Initial conditions

- Zero, i.e. no suspended load concentration (default)
- Uniform initial concentration

- Region defined values for concentration
- Continue

### **Boundary conditions**

- STANDARD:
  - INFLOW: concentration\_in, discharge\_in, discharge\_in\_warea
  - OUTFLOW: zero\_gradient\_out

### **Source (suspended sediment volume)**

- Type: total (as discharge, m<sup>3</sup>/s) or concentration (-)
- Sink behaviors:
  - Exact (as prescribed)
  - Available (as prescribed or less)
  - Infinity (as much as possible)

## **5.2.3 Tracers**

### **Parameters**

- Tracer start time
- Number of tracers (max. 5)
- Molecular diffusivity coefficient (isotropic, longitudinal/transversal)
- Turbulent diffusivity coefficient (isotropic, longitudinal/transversal)

### **Initial conditions**

- Zero, i.e. no tracer concentration (default)
- Uniform initial concentration
- Region defined values for concentration
- Continue

### **Boundary conditions**

- STANDARD:
  - INFLOW: concentration\_in, discharge\_in, discharge\_in\_warea
  - OUTFLOW: zero\_gradient\_out

### **Source (tracer volume)**

- Type: total (as discharge, m<sup>3</sup>/s) or concentration (-)
- Sink behaviors:
  - Exact (as prescribed)
  - Available (as prescribed or less)
  - Infinity (as much as possible)

### 5.2.4 Temperature

#### Riemann Solver

- Approximate Riemann Solver based on flux splitting technique (Vanzo et al., 2016).

#### Initial conditions

- Zero, i.e. no thermal mass (default)
- Region defined values for temperature
- Continue
- Assigned from file (experimental): temperature values (in column) are read from external file

#### Boundary conditions

- STANDARD:
  - INFLOW: temperature input as value or time series (passive\_in)
  - OUTFLOW: passive\_out

#### Source

- Type: “meteo” containing the meteorological forcing terms
- Sink behaviors:
  - Exact (as prescribed)

#### Parameters

- Temperature module starting time
- Update frequency of the source terms (default is every timestep)
- Factor to calibrate bottom heating dissipation by friction (Beltaos, 2013)
- Soil thickness to reach constant (in vertical direction) soil temperature (Caissie and Luce, 2017)
- Soil thermal conductivity



## Diffusion Parameters

- Type of diffusion (none, constant, dynamic)
- Molecular diffusivity coefficient (isotropic)
- Longitudinal Turbulent diffusivity coefficient (absolute value or parameter, depending on diffusion type)
- Transversal Turbulent diffusivity coefficient (absolute value or parameter, depending on diffusion type)
- Maximum relaxation parameter (numerical parameter with default value, (Vanzo et al., 2016))

### 5.2.5 Vegetation

#### Parameters:

- Flow resistance modification (Strickler approach)
- Shear stress partitioning
- Critical Shields parameter modification (with MPM formula)
- Vegetation uprooting
- Plant burial

#### Initial conditions:

- Continue
- Region defined (regiondef)

#### Output variables:

- Aboveground biomass [-]
- Belowground biomass [-]
- Rooting depth [m]

### 5.2.6 Output Variables Summary

The output are generated on the mesh elements (cell centered) or at nodestrings and are stored in a binary file format (.h5).

The output type available are summarized in Table 5.22.

*Table 5.22 Output types and description.*

Output	Description
<b>Cell-centered</b>	
water_surface	Water surface elevation [m]

<b>Output</b>	<b>Description</b>
water_depth	Water depth [m]
bottom_elevation	Bottom elevation [m]
friction_chezy	Dimensionless squared chezy friction coefficient [-]
delta_z	Change in bottom elevation of during the course of the simulation [m]
spec_discharge	Specific hydraulic discharge ( $q_x, q_y$ ) [ $\text{m}^2/\text{s}$ ]
flow_velocity	Flow velocity ( $u_x, u_y$ ) [m/s]
turb_k	Turbulent kinetic energy ( $\kappa$ ) [ $\text{m}^2/\text{s}^2$ ]
turb_reynolds	Turbulent Reynolds stresses [ $\text{Nm}/\text{m}^2$ ]
flow_velocity	Flow velocity ( $u_x, u_y$ ) [m/s]
flow_curvature	Flow curvature (inverse of flow radius) [ $\text{m}^{-1}$ ]
flow_radius	Flow radius [m]
theta	Non-dimensional effective bed shear stress for bedload transport [-]
theta_critical	Non-dimensional critical bed shear stress [-]
trsp_capacity	Bedload transport capacity ( $q_{b,x}, q_{b,y}$ ) [ $\text{m}^2/\text{s}$ ] as compact volume, no porosity
trsp_capacity_abs	Bedload transport capacity magnitude $q_b$ [ $\text{m}^2/\text{s}$ ] as compact volume, no porosity
active_layer_thickness	Active layer thickness [m]
active_layer_fractions	Grain class fractions in the active layer [-]
sublayer_fractions0	Grain class fractions in the upper sublayer [-]
sublayer_fractions1	Grain class fractions in the lower sublayer [-]
sublayer_id	Index of the sublayer directly below the active layer [-]
sublayer_elevations	Elevation of the lower edges of the sublayers [m]
dm_arithmetic	Arithmetic mean grain size of the active layer [m]
dm_geometric	Geometric mean grain size of the active layer [m]

---

<b>Output</b>	<b>Description</b>
d_16	Characteristic grain size d_16 of the active layer [m]
d_50	Characteristic grain size d_50 of the active layer [m]
d_84	Characteristic grain size d_84 of the active layer [m]
d_90	Characteristic grain size d_90 of the active layer [m]
suspended_conc	Suspended sediment concentration [-]
suspended_theta	Non-dimensional effective bed shear stress for suspended sediment transport [-]
bed_gradient	Local bed gradient ( $\partial z/\partial x$ , $\partial z/\partial y$ ) [-]
flood_tracking	Water front arrival time [s] Max. water depth [m] Max. flow velocity [m/s] Max. specific discharge [m <sup>2</sup> /s] Max. bed shear stress [Pa]
tracer1-5	Concentration of tracer specie 1 [-] ... Concentration of tracer specie 5 [-]
temperature	Water temperature [ $^{\circ}C$ ]
temperature_q	Thermal mass [ $m^{\circ}C$ ]
aboveground_biomass	Above ground biomass of vegetation [-]
belowground_biomass	Above ground biomass of vegetation [-]
rooting_depth	Rooting depth of vegetation [m]
<b>Nodestring</b>	
ns_hyd_discharge	Hydraulic output variables on the Nodestrings
ns_mor_discharge	Morphologic output variables on the Nodestrings

---



# 6

---

## References

- 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.
- Beltaos, S. (2013). River ice formation. *Committee on river ice processes; the environment, Canadian geophysical . . .*.
- Caissie, D. and Luce, C.H. (2017). [Quantifying streambed advection and conduction heat fluxes](#). *Water Resources Research*, 53(2): 1595–1624.
- Chen, X., Ma, J. and Dey, S. (2010). Sediment transport on arbitrary slopes: Simplified model. *Journal of Hydraulic Engineering-ASCE*, 136(5): 311–317.
- Duran, A., Liang, Q. and Marche, F. (2013). On the well-balanced numerical discretization of shallow water equations on unstructured meshes. *Journal of Computational Physics*, 235: 565–586.
- Engelund, F. and Hansen, E. (1972). A monograph on sediment transport in alluvial streams. *Teknisk Forlag, Copenhagen*.
- 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.
- Smart, G.M. and Jaeggi, M.N.R. (1983). Sediment Transport on Steep Slopes. *VAW-Mitteilung 64*, Versuchsanstalt für Wasserbau, Hydrologie und Glaziologie (VAW). Zürich, ETH Zürich.
- Soares-Frazão, S. and Zech, Y. (2011). HLLC scheme with novel wave-speed estimators appropriate for two-dimensional shallow-water flow on erodible bed. *International Journal for Numerical Methods in Fluids*, 66.
- van Rijn, L.C. (1989). Handbook Sediment Transport by Current and Waves. *Delft Hydraulics Laboratory*, Delft, The Netherlands.
- Vanzo, D., Siviglia, A. and Toro, E.F. (2016). Pollutant transport by shallow water equations on unstructured meshes: Hyperbolization of the model and numerical solution via a novel flux splitting scheme. *Journal of Computational Physics*, 321: 1–20.
- Wilcock, P.T. and Crowe, J.C. (2003). Surface-based transport model for mixed-size sediment. *Journal of Hydraulic Engineering*.





**BASIC SIMULATION ENVIRONMENT  
FOR MODELLING OF ENVIRONMENTAL  
FLOWS AND NATURAL HAZARDS**

---

# **USER MANUAL**

**VERSION 4.1.0  
JUNE 2024**



**BASEMENT**





# Contents

<b>1</b>	<b>Basic Simulation Environment</b>	<b>5</b>
1.1	Introduction . . . . .	5
1.2	General Use . . . . .	6
1.2.1	Problem Description . . . . .	6
1.2.2	Product Delineation and Employment Domains . . . . .	6
1.2.2.1	Product Delineation . . . . .	6
1.2.2.2	Employment Domains . . . . .	7
1.3	Modelling Procedure . . . . .	8
1.3.1	Pre-processing . . . . .	8
1.3.1.1	Topographical data . . . . .	8
1.3.1.2	Hydrological data . . . . .	9
1.3.1.3	Sediment data . . . . .	9
1.3.1.4	Dissolved species data . . . . .	9
1.3.2	Simulation . . . . .	10
1.3.3	Postprocessing . . . . .	10
1.3.3.1	Output Files . . . . .	10
1.3.3.2	Result Visualization . . . . .	10
<b>2</b>	<b>Grid Generation</b>	<b>11</b>
2.0.1	General . . . . .	11
2.1	1D grid generation . . . . .	11
2.1.1	The “BASEMENT 1D Grid File Editor” . . . . .	11
2.1.2	Edit Existing Geometry File . . . . .	11
2.1.3	Create New of Edit Cross Section Data . . . . .	12
2.1.4	Tools . . . . .	12
2.1.4.1	Edit Raw . . . . .	13
2.1.4.2	Friction . . . . .	13
2.1.4.3	Remove Nodes . . . . .	13
2.1.4.4	Guess Fixpoints . . . . .	15
2.1.4.5	Interpolation . . . . .	16
2.1.4.6	Export DTM for BASEplane . . . . .	19
2.2	2D grid generation with BASEmesh QGIS plugin . . . . .	19
2.2.1	Installation . . . . .	19
2.2.2	Fundamentals . . . . .	21
2.2.2.1	Breaklines . . . . .	21
2.2.2.2	Boundary Conditions . . . . .	22
2.2.2.3	Mesh Quality . . . . .	22
2.2.3	Mesh generation workflow . . . . .	22

2.2.3.1	Quality mesh generation . . . . .	23
2.2.3.2	Elevation mesh generation . . . . .	24
2.2.3.3	Mesh interpolation . . . . .	24
<b>3</b>	<b>Simulation workflow</b>	<b>25</b>
3.1	General . . . . .	25
3.2	Pre-Simulation . . . . .	26
3.2.1	Command Files . . . . .	26
3.2.2	Model Setup . . . . .	27
3.3	Simulation . . . . .	27
3.3.1	Command File . . . . .	27
3.3.2	Model Backend . . . . .	27
3.4	Post-Simulation . . . . .	27
3.5	Re-Run Simulation . . . . .	28
3.6	Restart Simulation . . . . .	29
<b>4</b>	<b>Graphical User Interface (GUI)</b>	<b>31</b>
4.1	General . . . . .	31
4.2	First Steps . . . . .	31
4.2.1	Scenario Directory . . . . .	31
4.2.2	Load and Save . . . . .	31
4.3	Setup . . . . .	32
4.3.1	Adding and Deleting Items . . . . .	32
4.3.2	Help and Parameter Values . . . . .	34
4.3.3	Run BASEMENT Setup . . . . .	34
4.4	Simulation . . . . .	34
4.4.1	Selecting the Simulation Backend . . . . .	35
4.4.2	Run Simulation . . . . .	35
4.5	Results . . . . .	35
<b>5</b>	<b>Run the program</b>	<b>37</b>
5.1	Graphical user interface (GUI) . . . . .	37
5.2	Batch mode under Linux . . . . .	37
5.2.1	Setup . . . . .	37
5.2.2	Simulation . . . . .	38
5.2.3	Results . . . . .	39
5.3	Batch mode under Windows . . . . .	40
<b>6</b>	<b>Differences between modules BASEMD and BASEHPC</b>	<b>43</b>
6.1	General . . . . .	43
6.2	Grid topology and generation . . . . .	43
6.3	Workflow . . . . .	44
6.4	Setup . . . . .	45
6.5	Simulation . . . . .	50
6.6	Results . . . . .	52
6.7	Case example . . . . .	52
6.7.1	Description . . . . .	52

6.7.2	Computational mesh . . . . .	53
6.7.2.1	Quality mesh . . . . .	53
6.7.2.2	Computational mesh . . . . .	53
6.7.2.3	Import of a 2.x to a compatible 4.x computational mesh . .	54
6.7.3	Setup and simulation . . . . .	57
6.7.4	Results and discussion . . . . .	59
6.7.4.1	Hydraulic results . . . . .	59
6.7.4.2	Boundary conditions . . . . .	60
6.7.4.3	Discussion . . . . .	62
<b>7</b>	<b>Performance on Multi-Core Processors and GPU</b>	<b>63</b>
7.1	General . . . . .	63
7.2	Scalability . . . . .	63
7.3	Computational Time . . . . .	64
<b>8</b>	<b>Specific Features of BASEMD</b>	<b>67</b>
8.1	General . . . . .	67
8.2	Model Coupling . . . . .	67
8.2.1	Introduction . . . . .	67
8.2.2	Coupling Types . . . . .	68
8.2.3	Coupling Mechanisms . . . . .	70
8.2.3.1	Explicit coupling of sub-domains . . . . .	70
8.2.3.2	One-way coupling and two-way coupling . . . . .	70
8.2.4	Definitions of Exchange Conditions . . . . .	71
8.2.4.1	General remarks . . . . .	71
8.2.4.2	Exchange conditions for mixed-dimensional sub-domains .	72
8.2.4.3	Exchange conditions for river junctions in 1-D river networks	72
8.2.4.4	Exchange conditions for river bifurcations in 1-D river networks . . . . .	73
8.2.4.5	Exchange conditions for combined 1-D and 2-D modelling .	74
8.2.4.6	Data exchange for morphological simulations with multiple grain classes . . . . .	75
8.2.5	Synchronization Concept . . . . .	75
8.2.5.1	General remarks on synchronization . . . . .	75
8.2.5.2	“Local time stepping” approach . . . . .	76
8.3	BASEextern - data exchange during runtime . . . . .	78
8.3.1	Introduction . . . . .	78
8.3.1.1	Data exchange over TCP/IP . . . . .	79
8.4	Flow Control in River Systems . . . . .	81
8.4.1	Introduction . . . . .	81
8.4.2	Concept of Flow Control . . . . .	81
8.4.2.1	Monitored Variable . . . . .	82
8.4.2.2	Manipulated Variable . . . . .	82
8.4.3	Controller Types . . . . .	82
8.4.3.1	PID-Controller . . . . .	82
8.5	Built-In GUI Tools . . . . .	83
8.5.1	Interactive Visualization during run time using BASEviz . . . . .	83
8.5.2	Manual Controller Interface (HID) . . . . .	85

**9 References**

**87**

---

# Basic Simulation Environment

## 1.1 Introduction

The software system BASEMENT (BAsic-Simulation-EnvironMENT) provides a functional environment for numerical simulation of river flows with sediment transport in alpine and sub-alpine regions. The main focus of conception and development is the robustness 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).

BASEMENT version 4 consists of two main modules (Figure 1.1), namely BASEMD (MD stands for Multi Domain) and BASEHPC (HPC stands for High Performance Computing, based on the possibility to use GPGPUs). The BASEMD module contains the three simulation submodules BASEchain (1D open-channel flow), BASEplane (2D open-channel flow), BASEsub (3D subsurface flow) and allows for submodule coupling. Further, there is the BASEextern submodule for simulation data exchange during runtime. The BASEHPC module contains only the BASEplane submodule which has a simpler spatial discretization but improved performance compared to BASEMD::BASEplane. The software provides a unified simulation workflow and graphic user interface (GUI) for setting up numerical models with the two modules. # Main Features

The three submodules of the BASEMD module offer a wide range of different modelling features originating from more than 20 years of model development. Both modules BASEMD and BASEHPC have a submodule BASEPLANE but the BASEHPC::BASEPLANE submodule is tuned for performance using GPGPUs. The two main modules contain all the features of the two previous versions 2.8.2 (BASEMD) and 3.2 (BASEHPC). Table 1.1 gives an overview of the various features of both modules.

*Table 1.1 List of BASEMENT main features*

Feature	BASEMD	BASEHPC
1-D model	✓	
2-D model	✓	✓

Feature	BASEMD	BASEHPC
Hydrodynamics	✓	✓
Morphodynamics:		
- Bed load	✓	✓
- Suspended load	✓	✓
Tracer advection		✓
External sub-domain	✓	
Model coupling (multi domain)	✓	
Controller	✓	
Subsurface flow	✓	
Vegetation	✓	✓
Water temperature		✓
SMP hardware	✓	✓
GPU/HPC support		✓

## 1.2 General Use

### 1.2.1 Problem Description

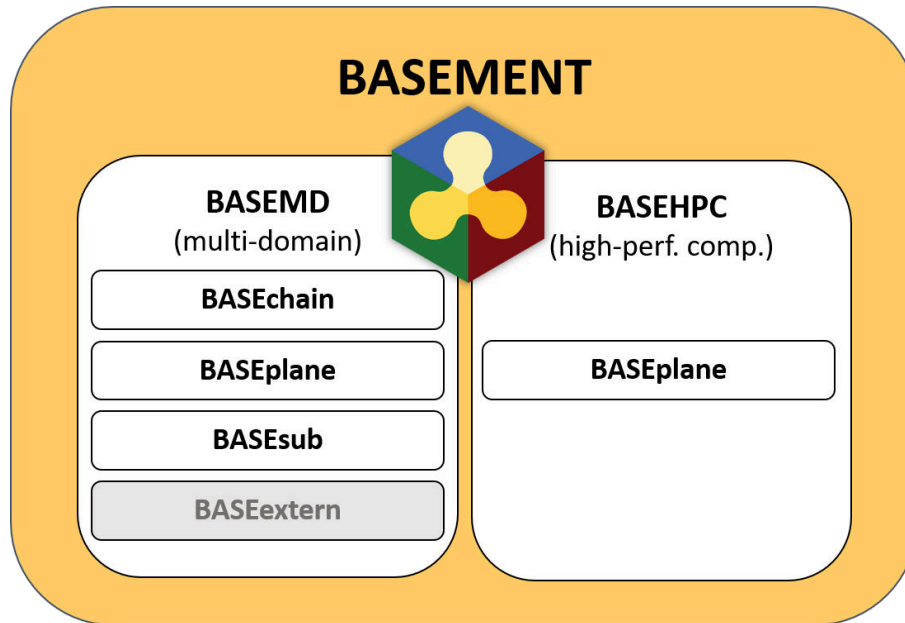
In connection with watercourses and river scapes, increasingly complex problems have to be addressed. The assessment of flood risk, the more frequent planning of river restoration projects or the study of naturally shaped watercourses implicate the examination of larger regions - also outside of the actual waterway - and a more manifold shape of the channels. The simple formulas for the calculation of flow 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 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 and no sediment transport) 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.



*Figure 1.1 Overview of BASEMENT modules*

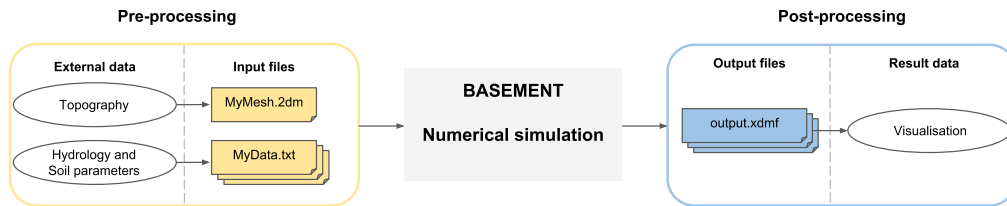
Unlike currently used programs for the simulation of a specific flow behaviour, BASEMENT intends the arrangement of many different problem types with one single tool to gain an integrated understanding for the initial position, the solution process and its results.

#### 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 related 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 aggradation 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 risk 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.
- Ecohydraulic application, such as habitat modelling





**Figure 1.2** Overview of the modelling procedure with BASEMENT

## 1.3 Modelling Procedure

The modelling procedure involves three stages: the pre-processing, the simulation and the post-processing (Figure 1.2). A project is usually based on a topographical region on which one or more scenarios are studied by running appropriate numerical simulations. Each scenario and all representative parameters with the required type of data should be defined in advance. The pre-processing stage consists of gathering the necessary external data in order to obtain the required input file format for the numerical simulation. The simulation generates output files that can be visualized and modified by external softwares (e.g. ParaView) in order to represent and interpret the results of the numerical simulation. It is recommended to organize your files in different directories, considering static and varying input data of the scenarios.

### 1.3.1 Pre-processing

Three main types of external data need to be provided for the numerical simulation: topography, hydrology and sediment data. The pre-processing stage involves the conversion of external data into appropriate input files that are used in the numerical simulation. The topography of the investigated region has to be transformed into a computational mesh. The topographical data types are manifold and may come from a cluster of point with (x,y,z)-coordinates, cross sections, height contour lines or raster data like a digital elevation model (DEM). Beside the computational mesh, hydrological and morphological data have to be determined for the numerical simulation and therefore converted into series data, constant or dynamic value (e.g. weir activation). The hydrology is characterized by inflow discharge, friction, water level or local sources and sink. The soil parameters include the mean grain size, the porosity, sediment density, the roughness, the angle of rest and the sediment flow.

#### 1.3.1.1 Topographical data

The numerical methods used in BASEMENT are based on a discretization of the domain topography into river cross sections (for 1D model) or using an unstructured mesh consisting of triangular elements (for 2D models). These elements are the control volumes (finite volume of 1st order) for the computation of flow equations and the complex of these elements forms the computational mesh. The procedure to setup computational meshes for BASEMENT is introduced in Section 2.0.1.

### 1.3.1.2 Hydrological data

The hydrology of the domain can be specified at boundary conditions in case of water fluxes or over a defined region of the computational mesh if an external source (mass) like rainfall, local source or sink is considered. The water flux can be implemented as discharge ( $m^3/s$ ), h-q relation or as water surface elevation and the external source can be implemented as discharge or as rainfall precipitation ( $mm/h$ ).

The type of data can be assigned as a single constant value (lake level, constant discharge, . . .) or as a time series like a hydrograph or series variable (e.g. h-q relation) or as dynamic in case of weir activation or dam collapse. In case of variable water flux (e.g. discharge hydrograph or rating curve), the hydrological data is stored in a time series data file (MyData.txt, see Figure 1.2). The simulation module will then interpolate the desired values to the actual computational time. The source data is either defined as constant or in a time series.

Initial hydraulic conditions can be defined as dry or defined by setting the values of the water surface elevation (wse), the velocity in x direction (u) and y direction (v) over the regions.

### 1.3.1.3 Sediment data

The river bed is characterized by a porosity and a mean grain size diameter ( $m$ ) determined from sediment or line samples. In BASEHPC, the simulation works only for uniform sediments.

The sediment flow is defined as a specific bedload and/or suspended load flux, which is averaged and evenly distributed over the stringdef length (sediment flow boundary). The sediment boundaries are of type standard (external boundaries). The type of data for the specific bed load flux is either set constant or defined in a time series as sedimentograph [ $m^3/s$ ] or in a transport capacity formula, without porosity. For the suspended load boundary fluxes, the suspended sediment concentration to be used at the boundary must be specified as either a constant or time series. The reference bed elevation has to be provided at inflow and outflow boundary conditions of type equilibrium.

### 1.3.1.4 Dissolved species data

The presence of dissolved species in the flow can be defined as either boundary conditions, local sources or sinks and also as initial conditions. At present, the maximum number of transported species is 5. The fluxes of each specie can be defined as discharges ( $m^3/s$ ) or, alternatively, the concentration [-] of each specie can be set according to user specified values.

Similarly to the hydrological and sediment data, the tracer discharges or target concentrations can either be set as constant or defined as a time series. For the case of fluxes prescribed as boundary discharges, these are distributed evenly along the boundary length or weighted according to the wet area of the boundary section. In the case of region-defined local sources, the total discharge is distributed evenly across the region's area. In the case of a prescribed target concentration at boundaries or sources this value is uniformly applied to the entirety of the boundary length or region area, respectively.

## 1.3.2 Simulation

The software system BASEMENT encompasses the numerical simulation, composed of numerical subsystems, executables binary files and interfaces to the infrastructural software like the pre- and post-processors. More details concerning the simulation workflow are described in Section 3.

## 1.3.3 Postprocessing

### 1.3.3.1 Output Files

The BASEMD::BASEchain and BASEMD:BASEplane models provide different output variables and formats. BASEchain results are in ASCII format in tabular form ready to drag and drop to spread sheet software. BASEplane results are stored in a binary HDF file format (.h5) to be converted to xdmf format using the result binary of the simulation workflow. Available output variables and formats are summarized in the “Introduction and Installation” part of the manual.

### 1.3.3.2 Result Visualization

The visualization of results is separated from the software system BASEMENT and can be done with independent products using a well-defined common interface. The output are available as an extensible data model format “results.xdmf” (see Figure 1.2) for the cell centered outputs or in a text format (.csv) for the nodestring output. The software ParaView and QGIS enables to visualize the results stored in “results.xdmf”.

# 2

---

## Grid Generation

### 2.0.1 General

Computational meshes must be generated to perform numerical simulations with the 1D (BASEMD::BASEchain) or 2D (BASEMD::BASEplane or BASEHPC::BASEplane) models. For BASEchain, the computational mesh consist of river cross sections. River cross section geometry data can be imported from survey data, generated by tools or defined by hand (see Section 2.1). For BASEplane, an unstructured mesh consisting of triangular elements must be generated. For that purpose, BASEMENT provides the tool BASEmesh (Section 2.2) but other grid generation software may work as well.

## 2.1 1D grid generation

### 2.1.1 The “BASEMENT 1D Grid File Editor”

The 1D grid file editor of BASEMENT is integrated into the GUI of version 4 (Tools-menu) and can be used to create or adjust computational grids for BASEchain. If you have an existing 1D grid from HEC-RAS or in GEWISS format, please refer to the conversion tools provided on the BASEMENT website ([basement.ethz.ch/download/tools](http://basement.ethz.ch/download/tools)). If you would like to create a 1D grid from scratch, we suggest to use the tool BASEchange from the [BASEmesh package](#).

### 2.1.2 Edit Existing Geometry File

After opening a 1D grid file with the 1D Grid File Editor (see Figure 2.1), 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

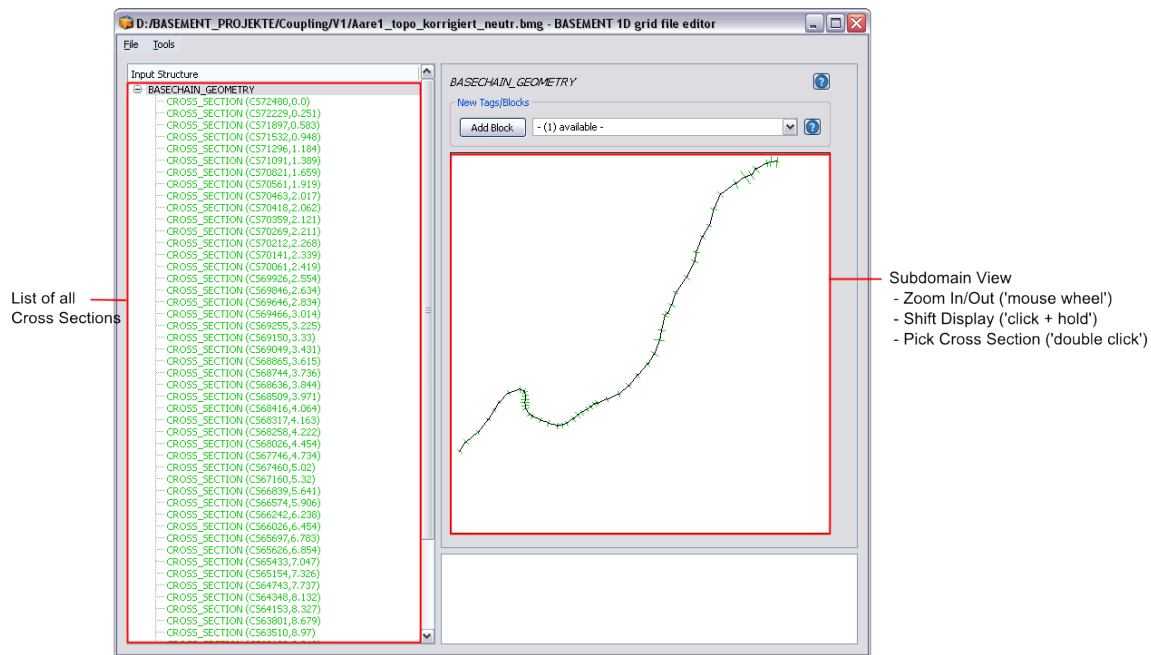


Figure 2.1 Grid File Editor – Subdomain View

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.

### 2.1.3 Create New or Edit Cross Section Data

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 2.2). With this visualization of the profile one can easily check for input errors in the geometrical definition of the cross section profile. 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 1D tutorial, hydrodynamics and sediment transport at the river Thur.

### 2.1.4 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

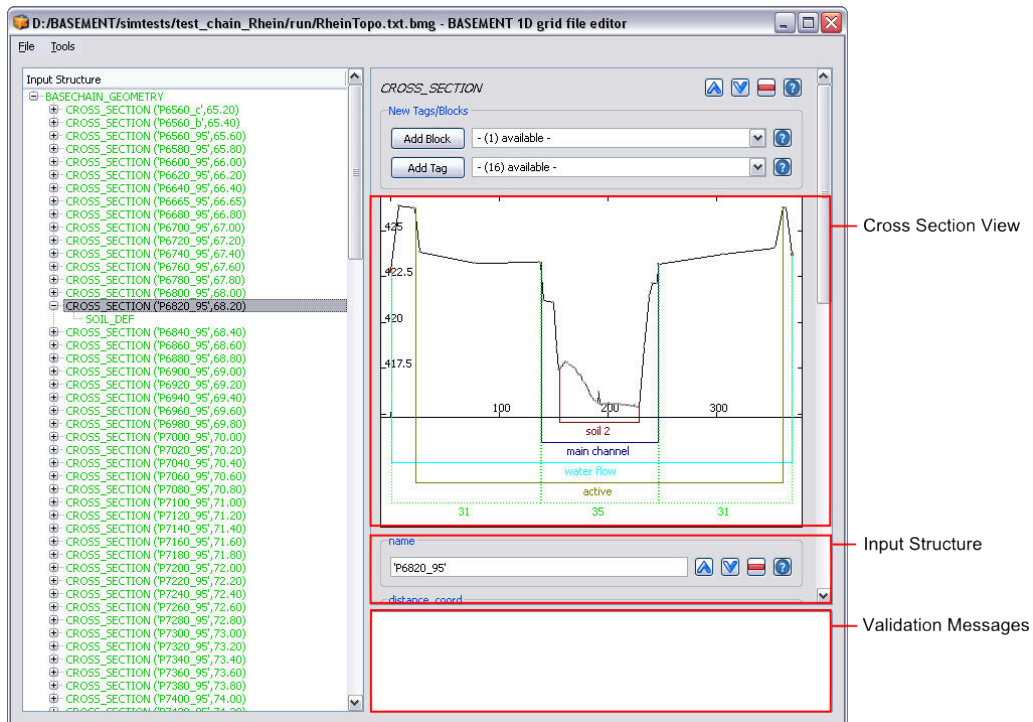


Figure 2.2 Grid File Editor – Cross Section View

of these tools are given. Please be aware that some of the offered tools are still in a beta-status.

#### 2.1.4.1 Edit Raw

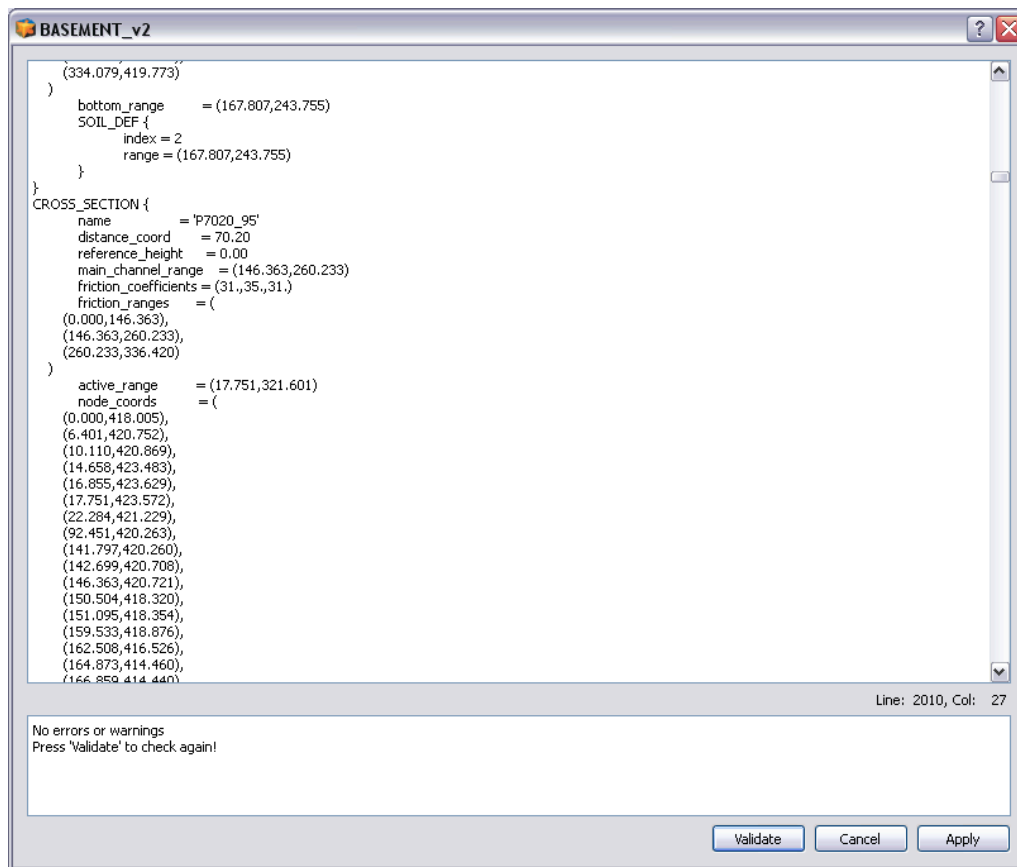
Another way to edit a grid file is to edit the grid file in raw text mode. For this purpose choose *Tools* on the menu bar and select *Edit Raw*. The Editor window will pop up as shown in Figure 2.3. In the lower part of the window the Input is validated and possible parse errors are indicated. For the sake of completeness it is mentioned here that the grid file can still be built up and edited with a simple text editor.

#### 2.1.4.2 Friction

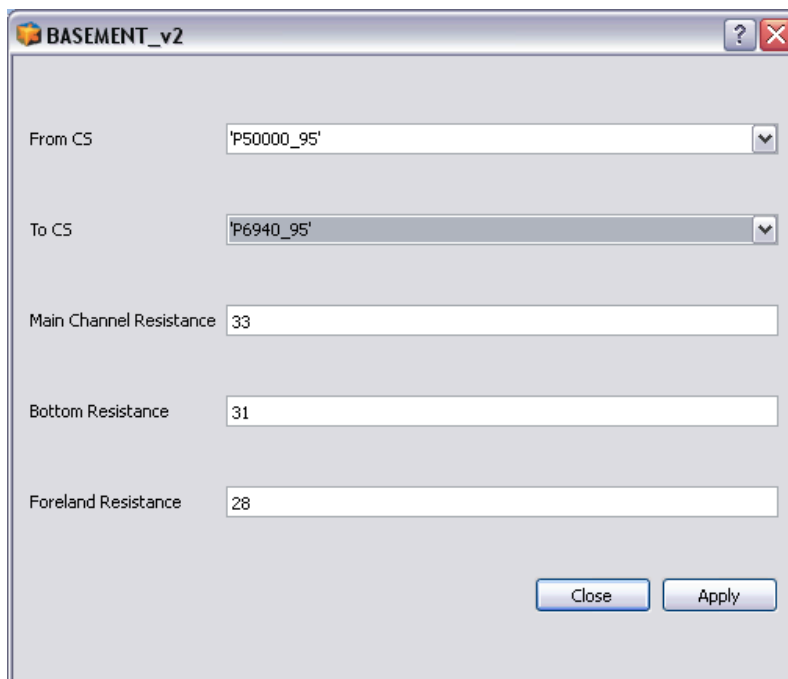
With this tool (see Figure 2.4) the friction value of the main channel, the forelands and the bottom can be assigned to a range of consecutive cross sections. The names of the first and the last cross section of the range have to be given.

#### 2.1.4.3 Remove Nodes

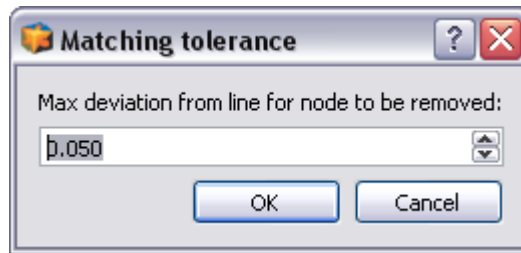
Cross sections often consist of a large number of nodes and slices which consumes significant computational performance. If multiple nodes lie on a straight line within the cross section there is redundant information present which can be removed without reducing the accuracy of the computations. Therefore 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.



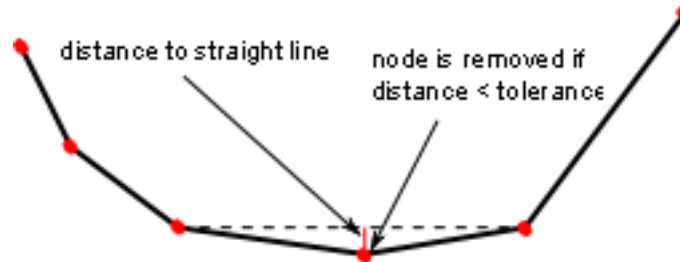
*Figure 2.3 Grid File Editor: Raw Edit Window*



*Figure 2.4 Friction assignment*



*Figure 2.5 Node removal dialog for setting up the tolerance*



*Figure 2.6 Schematic sketch of node removal*

To access this tool open the *Tools* menu and click on the *Remove nodes option*. The dialog opens as shown in Figure 2.5:

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 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 neighbored nodes (considering the given tolerance). If this is the case the node is removed from the profile (see Figure 2.6). 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

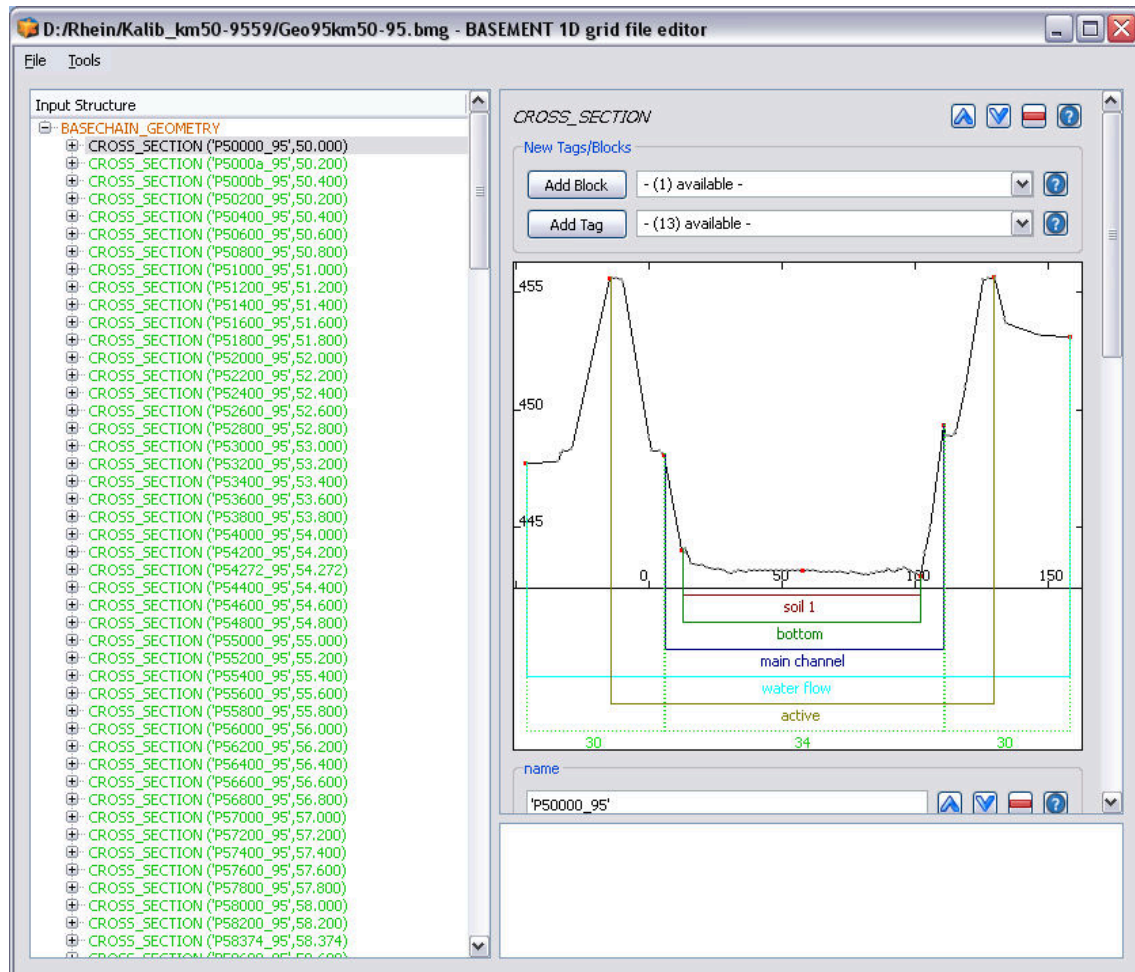
This tool provides a definition of the active range where it is not yet defined. For this purpose the lowest point of the cross section is searched and then the highest points to the left and the right of it (usually the dam crests) are set as limits of the active range. The definition of the active range can always be changed manually by the user in the interface. As an example on how the active range is set see Figure 2.7.

#### 2.1.4.4 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





*Figure 2.7 In red the guessed fix points for cross section interpolation*

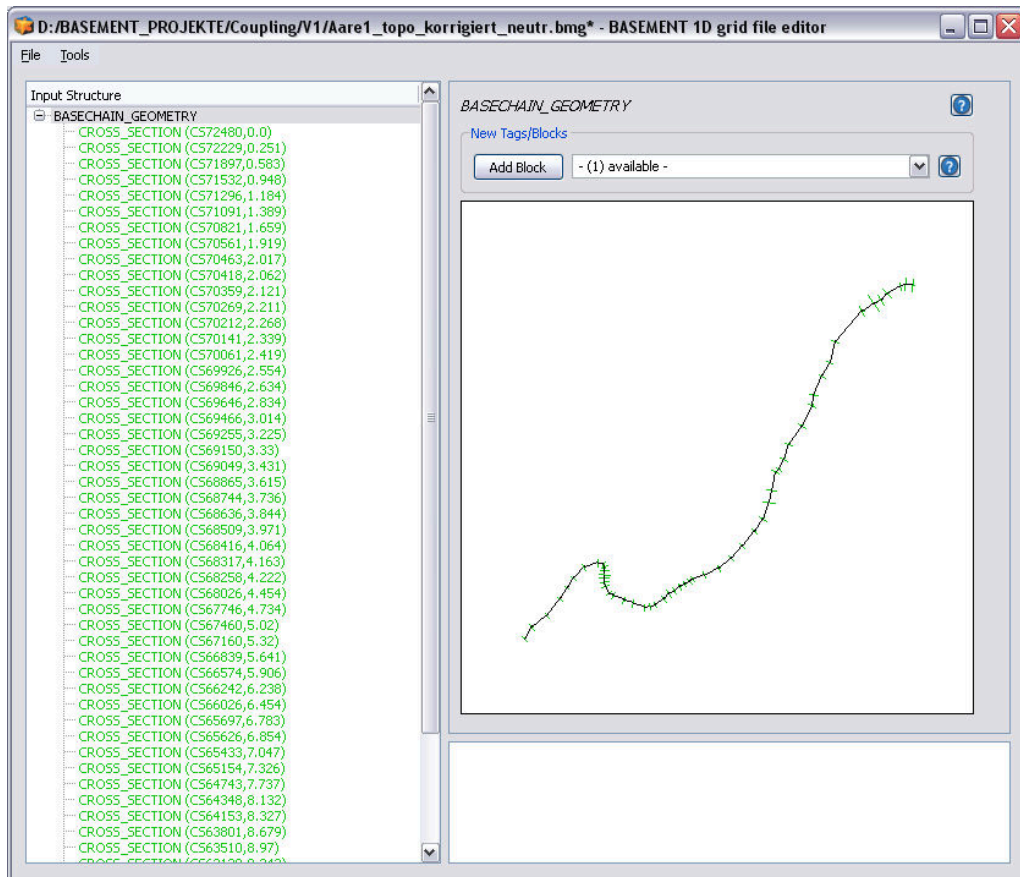
- The limits of the main channel
- The limits of the active range
- The midpoint of the main channel.

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.

#### 2.1.4.5 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.

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



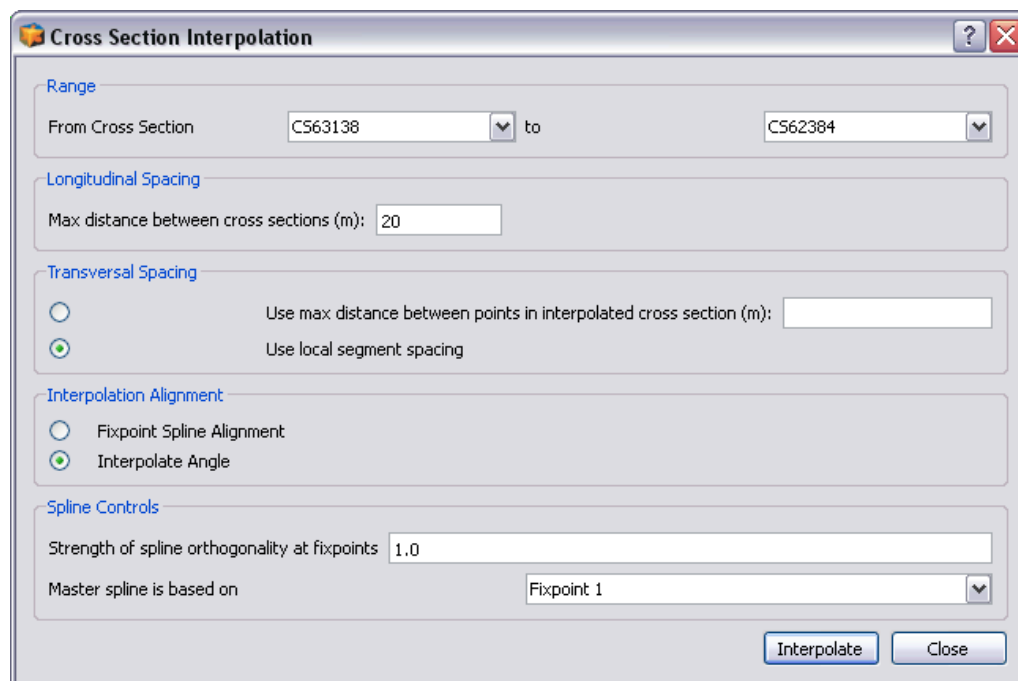
**Figure 2.8** Curved alignment of the cross sections.

*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 2.8).

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 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



**Figure 2.9** Setup dialog for the cross section interpolation

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 2.9 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 spline's tangent direction. Alternatively, by 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. Finally, also the fixpoint that 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.

#### 2.1.4.6 Export DTM for BASEplane

This tool enables the user to convert a 1-D BASECHAIN\_GEOMETRY (Figure 2.10) 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 Section 2.1.4.5): 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.

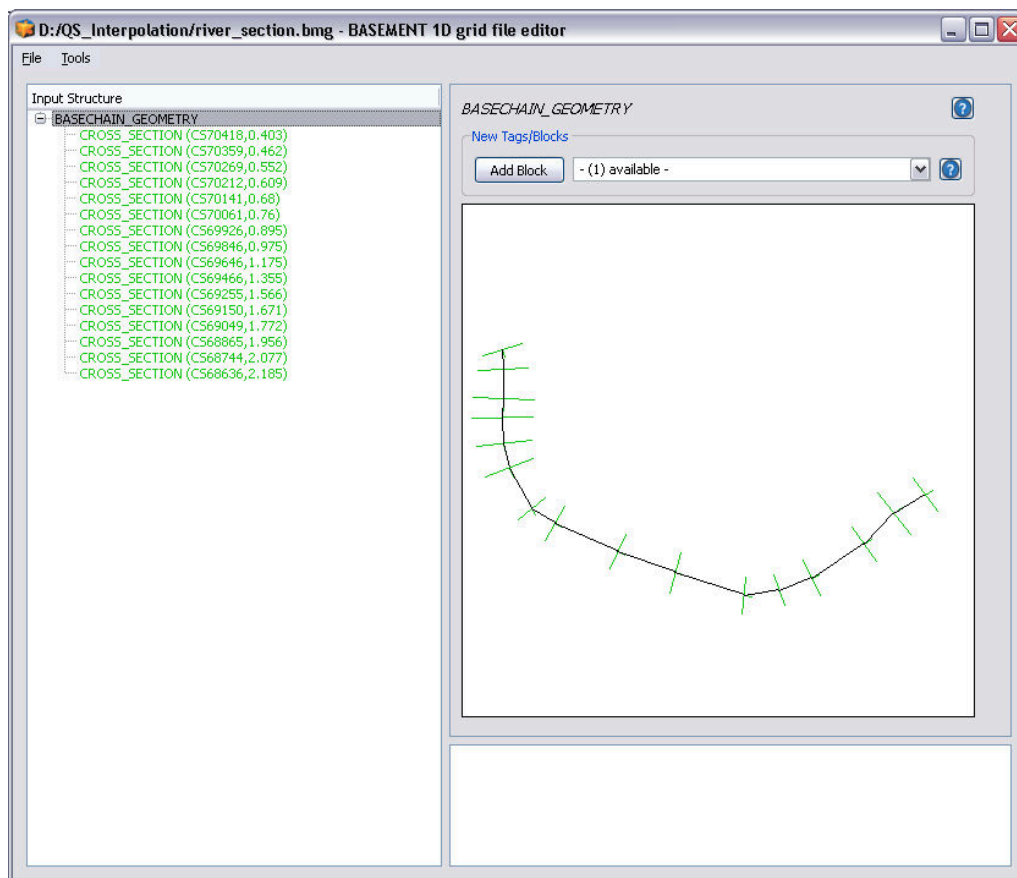
## 2.2 2D grid generation with BASEmesh QGIS plugin

In order to provide a free and open source solution for the creation of computational meshes, the plugin BASEmesh for the open source geographic information (GIS) software QGIS was developed. The plugin utilises Jonathan R. Shewchuk's advanced mesh generator Triangle (Shewchuk, 1996) as its meshing algorithm.

BASEmesh version 2.x is compatible with BASEMD and BASEHPC, but requires QGIS version 3.10 or higher. For versions of BASEmesh compatible with QGIS v2.18 or versions lower than v3.10, please refer to the compatibility section on the [BASEmesh Website](#).

### 2.2.1 Installation

BASEmesh is available for installation through a custom plugin repository which is not included in QGIS upon installation. The BASEmesh plugin repository must be added to the QGIS plugin manager by the user prior to installation.



**Figure 2.10** 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`.

To install BASEmesh, follow these steps:

1. Start QGIS
2. Load the QGIS plugin manager by choosing *Manage and Install Plugins...* in the *Plugins* category of the QGIS toolbar
3. Select *Settings* from the left panel
4. Click on *Add...* and provide a descriptive name, e.g. 'BASEmesh Plugin Repository'
5. Specify the repository address: [https://people.ee.ethz.ch/~basement/qgis\\_plugins/qgis\\_plugins.xml](https://people.ee.ethz.ch/~basement/qgis_plugins/qgis_plugins.xml)
6. Press OK to confirm; a new entry has been added to the list of plugin repositories (make sure the *Status* reports as *connected* before continuing)
7. Select *All* from the left panel of the plugin manager and search for 'BASEmesh'
8. Choose the BASEmesh plugin (if several are available, choose the one with the highest version number) and press *Install Plugin*
9. Close the plugin manager. A new toolbar should have appeared and a *BASEmesh* entry added to the *Plugins* category of the QGIS toolbar

## 2.2.2 Fundamentals

Computational meshes used with BASEplane (applies to BASEMD::BASEplane and BASEHPC::BASEplane) are of the format "2dm" (see [Aquaveo XMS Wiki](#)). The plugin BASEmesh for the free and open source geographic information system software Quantum GIS (QGIS) provides automated routines for mesh generation in case of a small or large meshes. The breaklines, the definition of boundaries and the generation of the quality mesh are steps of the mesh generation process using BASEmesh.

### 2.2.2.1 Breaklines

Breaklines affect the quality mesh outcome by preventing the meshing of elements over them during the meshing process. Breaklines enable to delineate the limits of the quality mesh as well as relevant regions like buildings or zones of local mesh refinement. These regions are characterized by marker points (Regiondefs) that allow the user to divide the computational mesh into areas of common features for the numerical simulation, e.g. setting different initial friction values or definition of an external source over a specific region of the mesh.

Breaklines are important and should be carefully defined with the consideration that the computational mesh attributes a single elevation to the cell center. There is a risk of lost geometrical accuracy at locations of distinct change of slope (e.g. levee crests or river side walls) or where the cells are required to have a determined and fixed elevation (riverbed, bank crest, etc.). In order to overcome this issue, areas of fixed or known elevation need to be delimited by breaklines as regions to ensure that the right elevation is assigned to the cell.



### 2.2.2.2 Boundary Conditions

Boundary conditions control the water and sediment flow into and out of the domain. Boundary conditions are defined on STRINGDEFs, i.e. on a selected sequence of successive vertices (with direction) located either inside or at the boundary of the computational mesh. The sequence of vertices along the stringdef gives the stringdef's direction with a left and right side. The upstream flow direction must be defined by the user during the setup stage of the numerical simulation and has to be set according to the stringdef definition, i.e. direction.

There exists three types of boundary conditions, the external (standard), internal and linked boundaries. The external boundaries are defined on the domain boundary, while the internal boundary is defined inside the domain. Linked boundary conditions connect two stringdefs inside or on the boundary of the domain. More information about the type of boundaries and their features can be found in the Reference Manual. The “upstream\_direction” is determined by placing yourself on the first node of the stringdef and looking into the direction of the second node of the nodestring. Then, determine whether “upstream” is on your left or right side.

STRINGDEFs can be defined on nodestrings, which are listed at the end of the computational mesh file “MyMesh.2dm”.

Please Note: In BASEHPC, the number of nodes per nodestring is limited to 40, i.e. larger nodestrings must be split up.

### 2.2.2.3 Mesh Quality

The quality of the mesh is defined by the size and number of mesh elements that compose the computational mesh. Regions of high interest need some mesh refinement to get higher accuracy and regions of lower interest often have a coarser mesh. Two parameters are characterizing the mesh quality: the maximum element area and the minimum element angle.

The maximum element area is assigned to cluster of elements, i.e. specific region surrounded by breaklines and can vary among the zones. The minimum element angle is a parameter defined over the entire mesh. Smaller angles lead to less elements, while larger angles lead to more elements.

## 2.2.3 Mesh generation workflow

The following section covers the basics of mesh generation using version 2.0 of the BASEmesh plugin. For in-depth parameter explanations and advanced use-cases, refer to the [BASEmesh Manual](#) and the Tutorial *Introduction to BASEmesh*.

Mesh generation in BASEmesh v2.x is performed in two steps. First, a 2D quality mesh is generated using Triangle, which is then interpolated using one or more elevation sources. Elevation sources are either existing meshes containing elevation data (TIN), or raster data in the form of a digital elevation model (DEM). This interpolation can be performed for the mesh nodes (BASEMENT v2.8, BASEMD), the mesh elements (BASEMENT v3.x, BASEHPC), or both, which allows use of the same computational grid for both environments.

### 2.2.3.1 Quality mesh generation

The quality meshing utility provides a QGIS interface to the Triangle advanced mesh generator. As Triangle is two-dimensional, the generated mesh will not contain any elevation information.

The following constraints are available to control the mesh generation process:

- *Break lines*: A map layer containing lines or line strings representing distinct interruptions of the surface slope (e.g. dyke crests, river side walls, ...) which will be preserved in the computational mesh. Note that you do not have to include break lines for node string definitions (see *String definitions* description below).
- *Dividing constraints*: An integer layer attribute used to split a break line before meshing. This is useful when using inner boundaries in BASEMENT, as the number of mesh elements at the upstream and downstream interface must be equal.
- *Constrained points*: Additional points to enforce during triangulation, such as a known measurement point.
- *Minimum angle constraint*: The minimum angle enforced for any mesh elements generated. This heavily affects the element count of the resulting mesh.
- *Maximum area constraint*: A global maximum area for any mesh elements generated. This will be overridden by any region-specific area constraints defined (see below).

In addition to the global mesh quality constraints, additional constraints may be defined for individual mesh regions. A region is any closed loop of break lines, the constraints are then applied by placing a point marker within a region.

These markers may specify up to three flags:

- *Hole marker*: Regions marked as holes will be carved out of the resulting mesh. This flag is mutually exclusive with the other flags.
- *MATID*: Specify the material ID for any mesh elements generated within this region.
- *Maximum area*: This allows overriding the global maximum area constraint for mesh elements in this region.

In BASEMENT an ordered list of neighbouring node IDs is called a *string definition* (aka StringDef) or *node string*. In BASEmesh v2.x, the quality meshing process can be used to define the StringDefs of break lines. They are defined through line strings in a separate map layer and will be preserved in the resulting mesh as break lines. The following outputs can be generated related to StringDefs:

- *String definitions layer*: A map layer containing lines defining the node strings.
- *String definition ID field*: The unique name attribute of a given string definition. Required for node string identification.
- *Include in 2DM node strings*: If checked, the node strings will be written into the 2DM mesh file using *NS* tags. Required for BASEMENT v3.x.



- *Write to sidecar file*: If checked, the node strings will be written into a separate text file. Required for BASEMD.

Note that BASEHPC does not allow more than 200 nodes per node string; split your string definition lines if your meshing parameters generate meshes exceeding this limit.

### 2.2.3.2 Elevation mesh generation

The elevation meshing utility generates mesh geometries in the SMS 2DM format from existing 3D input geometries. It is provided to allow generation of TIN elevation data from geometries and is not necessary if you already have raster (DEM) elevation data for your quality mesh.

You can use the *BASEmesh/Converters/Convert legacy layer* utilities in the QGIS processing toolbox to create 3D geometries from 2D geometries with elevation attributes as used in previous versions of BASEmesh. Only layers containing elevation information will be displayed for this step.

Key parameters for the elevation meshing utility:

- *Line segments*: A map layer containing 3D lines or line strings constraining the generated output geometry.
- *Fixed points*: A map layer containing 3D points used to further constrain the triangulation.
- *Keep convex hull*: If selected, the convex hull of the input data is kept and used as the mesh boundary.
- *Shrink to segments*: If selected, only closed areas enclosed by break lines are included in the generated mesh.

Note that in BASEmesh v2.x, there is no more differentiation between the mesh domain (aka. mesh boundary polygon) and the mesh break lines layer. For behaviour similar to previous versions of BASEmesh, merge the mesh boundary polygon lines into the break lines layer and select the *Shrink to segments* option as your mesh domain.

### 2.2.3.3 Mesh interpolation

The interpolation step converts the flat quality mesh generated by Triangle into a suitable computational mesh for BASEMENT. For BASEMD, this means adding elevation information to the mesh nodes, for BASEHPC, the elevation information is added for the mesh elements instead. This interpolation is done from one or more interpolation sources, i.e. elevation meshes (TIN) or raster data (DEM).

In basic mode, a single elevation source may be selected, though multiple elevation sources are allowed in advanced mode - refer to the Interpolation utility's help panel for details.

Be aware that the interpolation process can be time consuming for large meshes. While it is possible to interpolate both the mesh nodes and elements, this will also double the time required to complete the interpolation process.

---

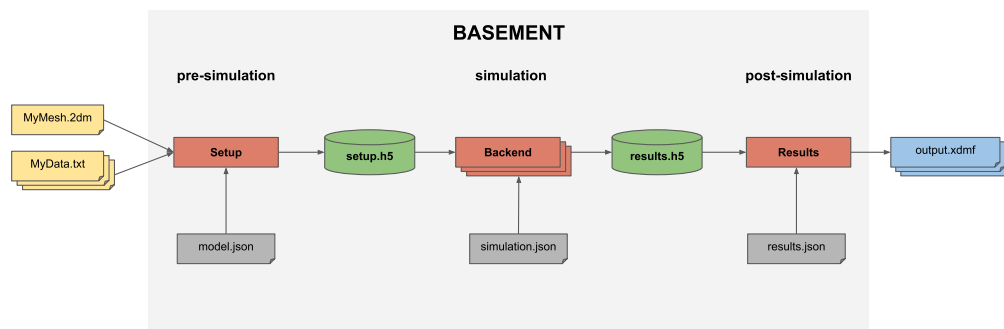
## Simulation workflow

### 3.1 General

The simulation workflow of the software system BASEMENT (light grey rectangular background on Figure 3.1) is composed of three parts: the pre-simulation, the simulation and the post-simulation. Each part contains an executable (red rectangles) and a command file (.json). The command files are in standardized file format of type JavaScript Object Notation (.json) with an independent language and syntaxe. Binary files (green cylinders) of HDF5 type (Hierarchical Data Format version 5, [www.hdfgroup.org](http://www.hdfgroup.org)) work like containers that can store large amount of data and thus allow the division of the numerical simulation in three parts. The input and output data files are located outside of the simulation environment (Figure 3.1).

The pre-simulation consists on setting up the model for the simulation. The hydro- and morphodynamic parameters are defined inside the command file `model.json`. The setup executable combines the computational mesh (`MyMesh.2dm`), external required data (`MyData.txt`) and the command file (`model.json`), validates the model and stores it inside the binary `setup.h5`.

The simulation part runs the simulation on a selected backend type. It combines the



*Figure 3.1* BASEMENT simulation workflow

model (setup.h5) stored in the first container with the command file simulation.json, where the simulation parameters are defined (e.g. execution time, output). The results of the simulation are stored in the second container (results.h5).

The post-simulation part transforms the simulation result file into output data that can be processed by the user. The type of output format (e.g. xdmf) is specified inside the command file results.json. The post-simulation process is based on python scripts.

## 3.2 Pre-Simulation

### 3.2.1 Command Files

The first command file model.json defines the parameters required to run a numerical simulation on the generated computational mesh. In this command file, either the BASEMD or the BASEHPC module must be selected.

The geometry block gives information on the computational mesh used for the 2D simulation. The name of the computational mesh or its path have to be specified. If a computational mesh of module BASEMD::BASEplane is used, an elevation interpolation method has to be defined. If no interpolation method is specified, the default interpolation method “mean” is selected. In the STRINGDEF block, the stringdefs must be listed by their name and the upstream flow direction should be indicated as either left or right (see Section 2.2.2.2). The in the REGIONDEF block, regions can be defined by listing a region name and assigning cells to that region via the MatID from the .2dm mesh file. Currently, each MatID should only be assigned to one region. Assigning an already assigned MatID to another region will overwrite the assignment to the previous region.

The hydraulics block contains the information about the initial conditions (dry, continue, region\_defined), the parameters (CFL, minimum water depth, . . .), the boundary conditions, friction values, external sources and flood tracking. If the initial conditions are defined via regions, dry initial conditions are assigned for cells which do not belong to any regions from REGIONDEF or whose region is not specifically assigned initial conditions. The boundary conditions are defined by giving the corresponding STRINGDEF name and the required type (standard, linked or internal). The friction value of a cell is set to the default friction value unless specified otherwise via regions. Regions can further be used to specify external sources. The flood tracking feature will track the maximum values of the water depth, flow velocity, specific discharge, bed shear stress and the flood arrival time.

The morphology block contains all information for setting a morphological simulation with uniform bedload transport. The bed material, the bedload transport formula, initial conditions and parameters like porosity and sediment density are required. Standard bedload boundary conditions characterize sediment inflow and outflow. The curvature and lateral bed slope effects can be activated in order to influence the bedload transport direction. Further, gravitational transport processes can be activated.

The command file model.json does not give any information about the duration of the simulation or the type of output. These are implemented in the next command files.

### 3.2.2 Model Setup

The setup executable gathers the different input files and generates the run file for the simulation stored in binary format (setup.h5). It validates the model before starting the simulation.

## 3.3 Simulation

### 3.3.1 Command File

The command file simulation.json contains information about the simulation time, the type of output (see Table 6.8) and optionally the minimum and maximum time step allowed. The user can define the start time, the output timestep and the end of the numerical simulation. The water surface, the water depth, the flow velocity or the change in bed elevation are examples of specific output that can be defined inside the command file. The output is generally defined on the mesh elements except for the hydraulic and sediment discharge, calculated at flow boundaries (nodestrings).

The command file simulation.json is coupled to the setup file stored inside the first container (setup.h5) in order to run the numerical simulation on a selected backend type. The results are stored as “results.h5” inside the second container.

### 3.3.2 Model Backend

The backend type can be selected between central processor unit (CPU), graphics processor unit (GPU) or a combination of GPU and CPU. The CPU provides sequential or multi-threading (OpenMP) backends. The backend types that support the numerical simulation are:

- seq: sequential execution on the CPU
- omp: multi-threading using OpenMP technology
- cuda: GPU
- cudaC: GPU with some kernels running sequentially on the CPU
- cudaO: GPU with some kernels running in parallel (OpenMP) on the CPU

All the backends execute the numerical simulations in double precision (default) and can be changed to single precision. For simulation running on CPU, the number of cores has to be given as argument.

## 3.4 Post-Simulation

The post-simulation converts the simulation results stored in the second container (results.h5) into a defined output format. The name and the output format are specified inside the command file results.json. At the moment, only the .xdmf file type is available

(Figure 3.1). The output.xdmf file can be modified by the user using the software ParaView to present the simulation results in a proper way.

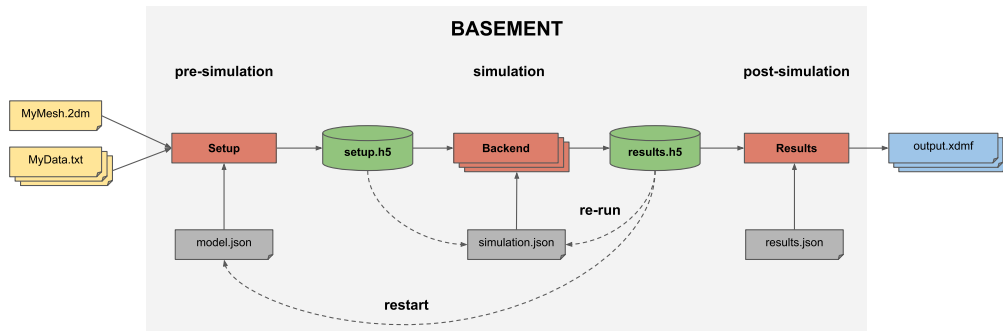
A python script [BMv3NodestringResulty.py](#) is available for extracting the stringdefs results (discharge) stored in the results.h5 binary and converts them in a text format (.csv). The available outputs are listed and described in Table 3.1.

**Table 3.1** Output of the post-processing python script *BMv3NodestringResulty.py*

Output	Description
Mean wse	mean water surface elevation [m]
Discharge	total normal water discharge Q [m <sup>3</sup> /s]
Wetted area	total wetted area of the edges belonging to the NS [m <sup>2</sup> ]
Mean bottom elevation	mean bed elevation of wetted edges [m]
Reference elevation	reference elevation (talweg) [m]
Wetted geometric length	wetted geometric length [m]
Total water volume stored in cells	total water volume stored in the cells belonging to the NS
Total cells conveyance	total conveyance of the cells belonging to the NS
Morphological flux	total normal morphological flux [m <sup>3</sup> /s] as compact volume, no porosity (output)
Bedload transport	total bed load transport capacity [m <sup>3</sup> /s] as compact volume, no porosity

## 3.5 Re-Run Simulation

The concept of rerun is to execute the same setup file (setup.h5) by fetching the initial conditions from the result file (results.h5) without parsing the command file model.json. It allows to continue a simulation from given results, thus obtaining a longer simulation without starting from the beginning. Other parameters can be modified like setting different output time step or adding/removing an output type. The rerun is activated by setting a start time larger than zero and the initial conditions are taken from the result file (results.h5) that should be copied inside the setup file.



*Figure 3.2* BASEMENT simulation workflow with restart and re-run processes

## 3.6 Restart Simulation

Restarting a simulation (Figure 3.2) means to modify the parameters of the command file `model.json`, while fetching initial conditions from an existing result file (`results.h5`). It allows, for example to run two different simulations one after the other, e.g. by adding bed load transport after a purely hydraulic simulation that reached steady state.

The block containing the initial conditions (`model.json`) is set as `continue` and the existing result file name with the time at which the new simulation start is specified inside the command file. The command file `simulation.json` indicates the desired end of the simulation and the output time step. The starting time is still required and should be set to 0.0.



---

# Graphical User Interface (GUI)

## 4.1 General

The BASEMENT graphical user interface assists the user with model configuration, numerical simulation and result export. For this purpose, the application provides a convenient way to edit the JSON configuration files and to select and run the backend executables.

## 4.2 First Steps

Once started, the BASEMENT user interface application displays the welcome screen (see Figure 4.1). Notice that all the tabs except for ‘BASEMENT’ are deactivated. The first and most important step when using the application is to select the scenario directory. This directory will contain all the configuration and output files that the application reads and writes. To select a scenario directory, click the button with the “Open” icon and select a folder using the folder selection dialog.

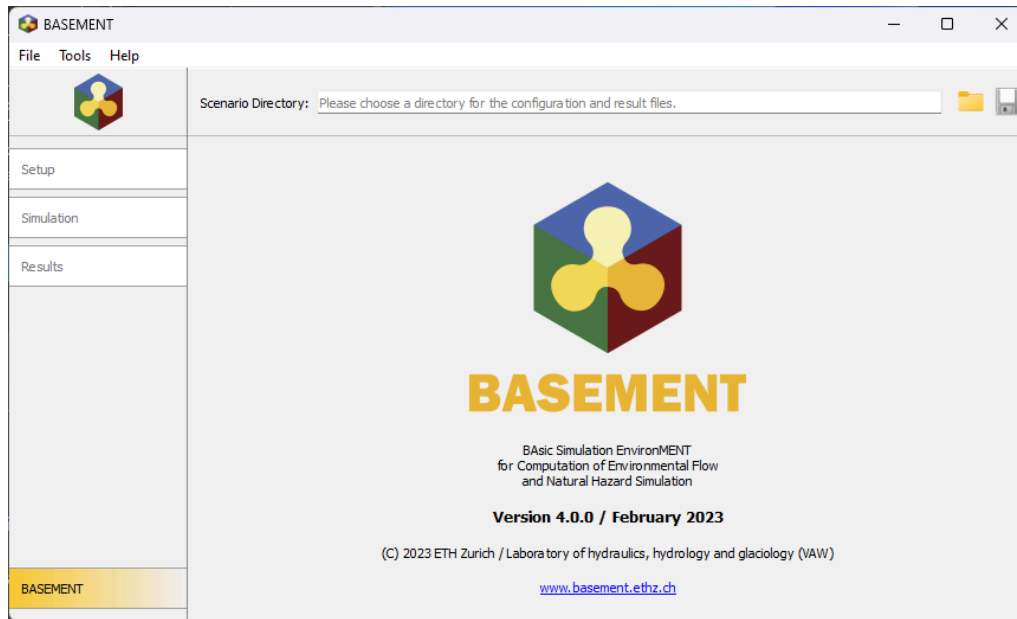
### 4.2.1 Scenario Directory

A scenario directory can only be opened by a single instance of the application at a time. A temporary ‘scenario\_directory.lock’ file is created in the scenario directory to enforce this constraint. This file signals that the directory is locked until the application is closed. If the scenario directory does not exist (this is checked regularly by the application) then an error icon is displayed in the scenario directory text field.

### 4.2.2 Load and Save

The JSON configuration files stored in a directory are loaded when it is selected as a new scenario directory. All currently unsaved changes are discarded after the user accepts the





*Figure 4.1 Welcome Screen*

corresponding warning. To save the three JSON configuration files for setup, simulation, and results into the current scenario directory click the button with the “Save” icon.

The tab ‘Setup’ is activated and selected as soon as a valid scenario directory has been chosen.

## 4.3 Setup

The setup screen (Figure 4.2) is designed for scenario parameter definition. The main part, the JSON editor, contains three columns: ‘Parameter’, ‘Value’, and ‘Validation’. The name of a JSON item (a parameter or a group of parameters) is displayed in the column ‘Parameter’, its value is displayed in the column ‘Value’ and the corresponding validation messages are shown in the ‘Validation’ column. Note that the button ‘Write’ is deactivated as long as the validation fails due to invalid parameters. Initially, only the item ‘Setup’ is present.

### 4.3.1 Adding and Deleting Items

To add a subitem to a parameter group (i.e. a JSON array or a JSON array), right-click on the item to open a context menu as shown in Figure 4.3. Select the item that you want to add for JSON objects or click the generic ‘Add item’ for JSON arrays. Once selected, the new subitem and all required sub-subitems are created automatically with default values (if available). Press Ctrl+Shift+A to expand all parameter groups quickly.

To delete a JSON item, use the context menu and select ‘Delete item’. Deleting parameter groups deletes the group and all contained items (after displaying a warning).

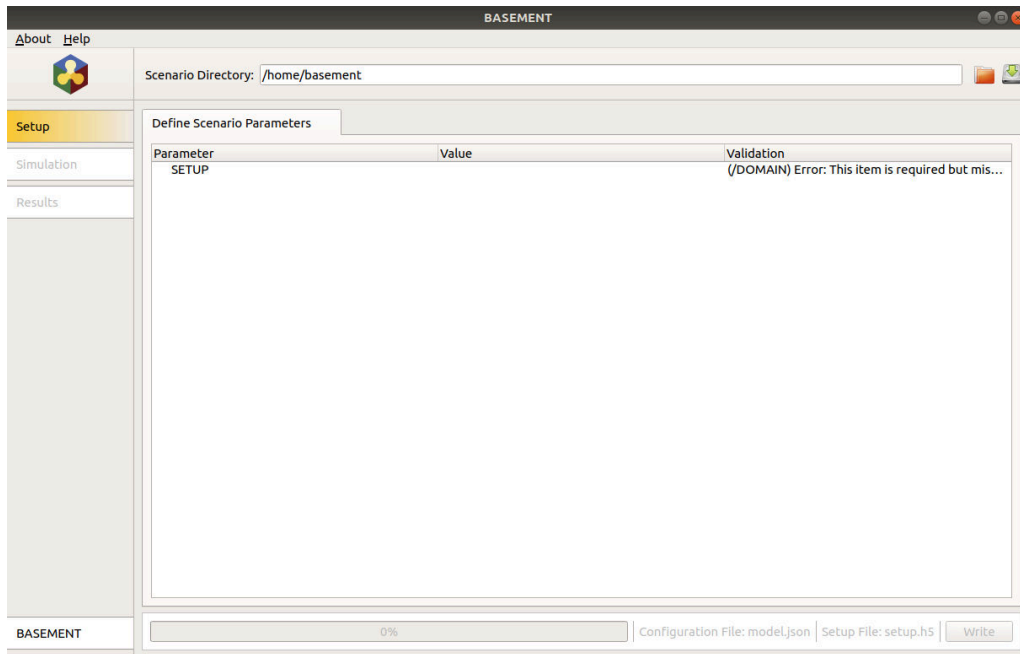


Figure 4.2 Setup Screen

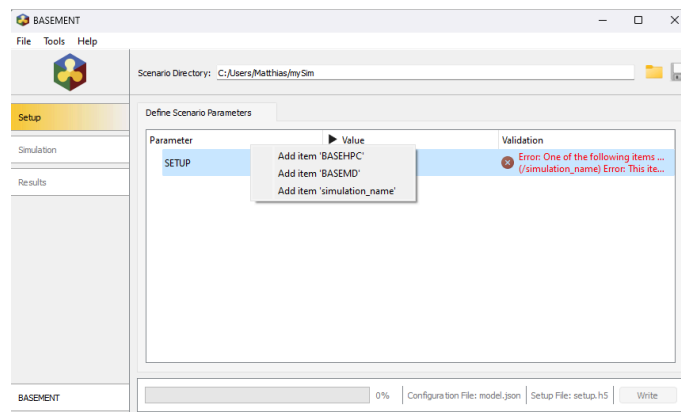
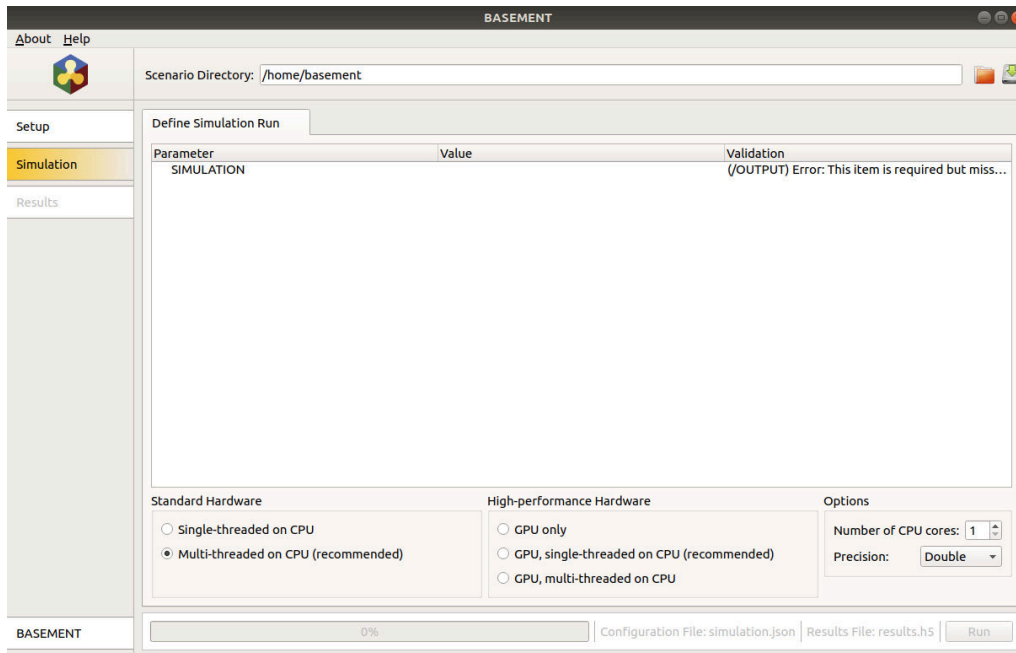


Figure 4.3 Adding JSON Items



*Figure 4.4 File Name Editor*



*Figure 4.5 Simulation Screen*

### 4.3.2 Help and Parameter Values

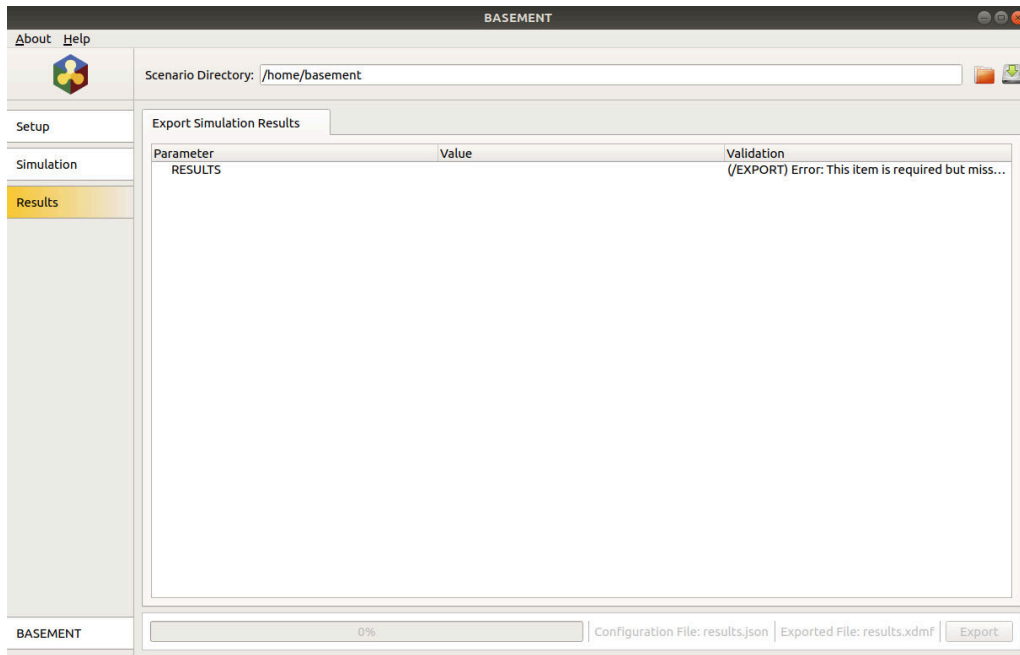
If you want to see the help for a parameter, mouse-over the parameter name and a tooltip with a parameter description appears. Double-clicking a parameter value opens a type-specific editor. In particular, you can click the “Open” icon to select a file for parameters that expect a file name (see Figure 4.4).

### 4.3.3 Run BASEMENT Setup

Click the ‘Write’ button to write the JSON file and to run the setup executable in the background when you are done with configuring the scenario parameters (the names of the written files are displayed next to this button). A closable console tab is opened. This tab contains two views: ‘Console Output’ and ‘Error Output’. The first view contains information about the status from the running BASEMENT setup process. The second view, ‘Error Output’, contains error messages from this process. If everything went well, all the files are successfully written and the ‘Simulation’ tab is activated.

## 4.4 Simulation

The simulation screen (Figure 4.5) is enabled if the file ‘setup.h5’ exists in the scenario directory. Use this screen to edit and review the parameters required to run the numerical simulation. The JSON editor works just like the editor in ‘Setup’, but of course the available parameters are different and only the item ‘Simulation’ is present initially.



*Figure 4.6 Results Screen*

#### 4.4.1 Selecting the Simulation Backend

The simulation screen also provides a way to select the simulation executable and command line flags: Choose the number of CPU cores that shall be used for the simulation, whether you want to compute on the GPU and the precision of the simulation using the controls on the lower end of the screen. Clearly, the number of CPU cores can only be set for multithreaded simulation backends.

#### 4.4.2 Run Simulation

When all the parameters are defined and valid, click the button ‘Run’ to launch the numerical simulation. Again, this will save the JSON configuration file and start the simulation backend in the background (the names of the files that are written are displayed next to the button). Track the progress of the simulation using the progress bar or click ‘Abort’ to abort. If everything went well, all the files are successfully written and the ‘Results’ tab is activated.

### 4.5 Results

The results tab (Figure 4.6) is enabled if the file ‘results.h5’ exists in the scenario directory. It can be used to define the export parameters. Again, the JSON editor works just like the editor in ‘Setup’. Initially, only the item ‘Results’ is present.

When all the parameters are defined and valid, click the button ‘Export’ to save the JSON configuration file and generate the output. If everything went well, the exported file (and an auxiliary results file in the case of export to ‘xdmf’) is successfully written to the scenario directory and is available for post-processing.



---

## Run the program

### 5.1 Graphical user interface (GUI)

The installation and executing of the BASEMENT software is described in the part Setup and First Start of the “Introduction and Installation” of this manual. Further details concerning the GUI of BASEMENT are explained in Section [4.1](#).

### 5.2 Batch mode under Linux

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. 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 command line arguments. The execution of BASEMENT is split in three steps, the setup, the simulation and the results having their own backend and parameters.

#### 5.2.1 Setup

The setup parameters of the numerical model are defined in the json file (“model.json”). The setup is executed from the command prompt (console) using the following line:

```
$ BMv4_setup model.json -o setup.h5
```

The arguments of the setup can be obtained in the command prompt (console) with the help flag ‘-h’. Table 5.1 shows the setup arguments.

```
$ BMv4_setup -h
```

**Table 5.1** Command line flags and arguments for the setup. Some flags are restricted to model definitions making use of the BASEHPC module.

Setup flag	Definition and arguments
-h , -help	display help information
-v , -version	displays version information
-o , -output	path to HDF5 output file
-l , -log	level of debug messages [BASEHPC]
-n , -nthreads	number of threads [BASEHPC]

### 5.2.2 Simulation

The execution of the simulation is executed from the command prompt (console) using the following line:

```
$ BMv4_simulation simulation.json setup.h5 -o results.h5
```

Different option for running the simulation are available via command line flags, while some flags, such as the backend type, are only available for simulations with the module BASEHPC. The following backends are available for simulations with the module BASEHPC:

- seq
- omp
- cuda
- cudaC
- cudaO

The backend “omp” stands for parallel execution with OpenMP and the number of thread should be specified. The backend “cuda” stands for GPU simulation. The backend “cudaC” executes the simulation using a coupled GPU and sequential processor and finally “cudaO” uses a coupled GPU and parallel processor. Simulations with the BASEMD module are always executed with the “omp” backend and the selected number of threads.

The various backends available for the BASEHPC module can also be run with single precision.

*Please note: Using single precision can lead to less accurate results!*

The command line arguments of the simulation executable are shown in Table 5.2.

**Table 5.2** Command line flags and arguments for the simulation

Setup flag	Definition and arguments
-h , -help	display help information
-v , -version	displays version information
-p , -progress	print simulation progress
-o , -output	path to HDF5 output file
-b , -backend name	the computational backend to use [BASEHPC]
-s , -single-precision	use single-precision arithmetic [BASEHPC]
-n , -nthreads number	number of threads (int)
-l , -log level	level of debug messages [BASEHPC]

### 5.2.3 Results

The last backend converts the simulation results into XDMF format. This executable must only be applied for simulations with the BASEHPC module, or for the BASEMD module involving a BASEplane domain. The result file (“results.json”) is executed as follow:

```
$ BmV4_results results.json results.h5 -o results.xdmf
```

The command line arguments for the output generation are listed in Table 5.3

**Table 5.3** Command line flags and arguments for the results

Setup flag	Definition and arguments
-h , -help	display help information
-v , -version	displays version information
-o , -output	path to HDF5 output file
-l , -log	level of debug messages [BASEHPC]
-n , -nthreads	number of threads (int)

The command line argument can be supported in any order.

Note that the ‘xdmf’ output file format contains a reference to the simulation results instead of copying the data. Also, an auxiliary results file (named ‘output\_aux.h5’ if the output name is ‘output’) is generated when exporting this file format. This has the advantage of using less storage space, but it also means that the three files (i.e. the simulation results file, the auxiliary results file, and the generated output file) are required to display the results. When opening such an output file, the file with the simulation results will be read from the path specified using the ‘-results’ command line parameter. Therefore provide a relative path to the simulation results file if you want to be able to move these files to different locations together.

Of particular interest is the possibility to run BASEMENT in the batch mode without the GUI to be started. Under Linux this can be done with a shell script. In a shell script, the three steps as well as several simulations can be run consecutively (for example over the weekend). To generate a shell script just create an empty text file and replace the ending ‘.txt’ by ‘.sh’. In this file several command lines can be defined as for example:

```
# Project 1
```



```

BMv4_setup /home/MyUser/Project_1/model.json \
  --output /home/MyUser/Project_1/setup.h5

BMv4_simulation /home/MyUser/Project_1/simulation.json \
  /home/MyUser/Project_1/setup.h5 \
  --output /home/MyUser/Project_1/results.h5 \
  --backend omp --nthreads 4

BMv4_results /home/MyUser/Project_1/results.json \
  /home/MyUser/Project_1/results.h5 \
  -o /home/MyUser/Project_1/mySim_output

# Project 2
BMv4_setup /home/MyUser/Project_2/model.json \
  --output /home/MyUser/Project_2/setup.h5

BMv4_simulation /home/MyUser/Project_2/simulation.json \
  /home/MyUser/Project_2/setup.h5 \
  --output /home/MyUser/Project_2/results.h5 \
  --backend omp --nthreads 4

BMv4_results /home/MyUser/Project_2/results.json \
  /home/MyUser/Project_2/results.h5 \
  --output /home/MyUser/Project_2/results.xdmf

```

To make the shell script executable open to console in the same directory of the shell script and run

```
chmod +x myShellScript.sh
```

Then run the shell script in the console with

```
./myShellScript.sh
```

## 5.3 Batch mode under Windows

Running BASEMENT 4.x in with a graphical user interface under Microsoft Windows can be done with the same work flow as described in Section 5.2. The syntax of the PowerShell is slightly different from that of the console. Further, the different backends of the BASEMENT software package have to be called with the full path of the installation folder. Note: Folder paths with whitespaces must be written in quotation marks (“”).

For example in the case you installed BASEMENT 4.x in under the path “C:\Program Files\BASEMENT 4.1.0” and your simulation scenario is stored on drive “F:\” in the folder “Project\_1”, then you should run the simulation with the following three commands:

```
C:\"Program Files\BASEMENT 4.1.0"\bin\BMv4_setup.exe `
```

```
F:\Project_1\model.json `
-o F:\Project_1\setup.h5

C:"Program Files\BASEMENT 4.1.0"\bin\BMv4_simulation.exe `
F:\Project_1\simulation.json `
F:\Project_1\setup.h5 `
--output F:\Project_1\results.h5 --backend omp --nthreads 4

C:"Program Files\BASEMENT 4.1.0"\bin\BMv4_results.exe `
F:\Project_1\results.json `
F:\Project_1\results.h5 `
--output F:\Project_1\results.xdmf
```

Of particular interest is the possibility to run BASEMENT in the batch mode without the GUI to be started. Under Microsoft Windows this can be done with a batch file. In a batch file, the three steps of the simulation workflow as well as several simulations can be run 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:"Program Files\BASEMENT 4.1.0"\bin\BMv4_setup.exe `
F:\Project_1\model.json `
-o F:\Project_1\setup.h5

C:"Program Files\BASEMENT 4.1.0"\bin\BMv4_simulation.exe `
F:\Project_1\simulation.json `
F:\Project_1\setup.h5 `
--output F:\Project_1\results.h5 --backend omp --nthreads 4

C:"Program Files\BASEMENT 4.1.0"\bin\BMv4_results.exe `
F:\Project_1\results.json `
F:\Project_1\results.h5 `
--output F:\Project_1\results.xdmf

C:"Program Files\BASEMENT 4.1.0"\bin\BMv4_setup.exe `
F:\Project_2\model.json `
-o F:\Project_2\setup.h5

C:"Program Files\BASEMENT 4.1.0"\bin\BMv4_simulation.exe `
F:\Project_2\simulation.json `
F:\Project_2\setup.h5 `
--output F:\Project_2\results.h5 --backend omp --nthreads 4

C:"Program Files\BASEMENT 4.1.0"\bin\BMv4_results.exe `
F:\Project_2\results.json `
F:\Project_2\results.h5 `
--output F:\Project_2\results.xdmf
```

Then run the batch file by double clicking on it.



# 6

---

## Differences between modules BASEMD and BASEHPC

The specific differences between the BASEMD and BASEHPC modules regarding the grid generation, the workflow, the model setup, the simulation and result configuration, are listed in this chapter by tables. Further, the differences are illustrated by a case study.

### 6.1 General

The specific differences between the BASEMD and BASEHPC modules regarding the grid generation, the workflow, the model setup, the simulation and result configuration, are listed in this chapter by tables. Further, the differences are illustrated by a case study. It is worth to remark that, given the differences summarized later, the numerical solutions of BASEMD and BASEHPC are not identical, even when conducted on the same domain. Nevertheless, both versions are tested for robustness, accuracy and convergence.

### 6.2 Grid topology and generation

*Table 6.1 Main differences regarding the computational mesh*

BASEMD	BASEHPC
Triangular and quadrilateral cells	Triangular cells
Dual mesh (cell vertex and cell centered)	Cell centered mesh
Variable bottom elevation over the cell	Constant bottom elevation over the cell

<b>BASEMD</b>	<b>BASEHPC</b>
Computational mesh in 2dm format (SMS), including material indices (stringdefs defined separately in *.bmc file)	Computational mesh in 2dm format (SMS), including material indices and stringdefs
Domain differentiation with <code>element_ids</code>	Domain differentiation with <code>regiondef</code>

*Table 6.2 Main differences regarding the grid generation with BASEmesh*

<b>BASEMD</b>	<b>BASEHPC</b>
Single procedure to generate a .2dm file with BASEmesh	Single procedure to generate a .2dm file with BASEmesh
Elevation information stored per mesh node (node z-coordinate)	Elevation information stored per cell
Stringdefs can be saved in separate *.bmc file for further usage	Stringdefs must be included at the end of the .2dm file
Manual editing of mesh in Qgis	Not available
View of the mesh in 3D	View of the mesh in 2D

## 6.3 Workflow

*Table 6.3 Major changes in workflow*

	<b>BASEMD</b>	<b>BASEHPC</b>
Configuration files	one command file with arbitrary name: *.bmc	three command files with fixed name: model.json, simulation.json and results.json
Data storage	results stored in a specified format	setup and result stored in HDF5 container (.h5)
Rerun	modify *.bmc file and run simulation	modify simulation.json and run simulation
Restart	modify *.bmc and select restart file	modify model.json and select restart file (.h5)

	<b>BASEMD</b>	<b>BASEHPC</b>
Executables	one executable (basement.exe) for CPU & SMP computing	separate executables for GUI, setup, results and for each simulation backend, e.g. for CPU, SMP and GPU

## 6.4 Setup

*Table 6.4 Main changes regarding model setup*

	<b>BASEMD</b>	<b>BASEHPC</b>
Command file type	run.bmc	model.json
Physical properties	gravity viscosity rho_fluid	gravity - -
Geometry	mesh file stringdef movable bed index_table -	mesh file stringdef - regiondef interpolation

*Table 6.5 Main changes in the **hydraulics** block of the domain BASEplane*

	<b>BASEMD</b>	<b>BASEHPC</b>
<b>Parameters:</b>		
Riemann Solver	exact, HLL and HLLC	HLLC
Fluid density	no (physical properties block)	yes
Max time step	no (timestep block)	yes
CFL	no (timestep block)	yes
Dynamic depth solver	water depth from left and right side of the cell edge and from center of the right and left cells	water depth from center of the right and left cells

	BASEMD	BASEHPC
Safe mode	no	yes
<b>Friction</b>		
Type	Manning Strickler Chezy Yalin Darcy-Weissbach Bezzola	Manning Strickler Chezy - - Bezzola
Wall friction	yes	no
Grain size friction	yes	no
<b>Boundary</b>		
Type	- hydrograph	<b>Standard</b> uniform_in uniform_out
	- zhydrograph	<b>Standard</b> zhydrograph <b>Linked</b> zhydrograph_linked, zhydrograph_linked_kinE
	zero_gradient weir	zero_gradient_out weir_out_constant, weir_out_dynamic <b>Linked</b> weir_linked_constant, weir_linked_dynamic
	gate	-
	- HQ_relation	<b>Standard</b> hqrelation_out <b>Linked</b> 2way_hqrelation_linked, hqrelation_linked
	coupling	-
	- wall	<b>Internal:</b> wall_internal

	<b>BASEMD</b>	<b>BASEHPC</b>
		wall_internal hqrelation_internal
File type	hydrograph, weir, gate, hqrelation	discharge, weir elevation, hqrelation, wse
Boundary inside the computational domain	Inner boundary (weir, gate and hqrelation)	<b>Internal</b> boundary:  wall, dynamic wall and h-Q relation <b>Linked</b> boundary: weir, h-Q relation
<b>Turbulence model</b>	yes	yes
<b>External source</b> Type	source discharge	total and distributed
Sink behavior	negative source discharge values	exact, available, infinity
<b>Initial</b> Type	dry continue index_table	dry continue region_defined
<b>Flood tracking</b>	no	yes

*Table 6.6 Main changes in the **morphology** block of the domain BASEplane*

	<b>BASEMD</b>	<b>BASEHPC</b>
<b>Parameter</b>		
Active layer	yes (control_volume)	yes
Porosity	porosity	sediment_porosity
Density	density	sediment_density



	<b>BASEMD</b>	<b>BASEHPC</b>
Starting time	- (bedload)	morphodynamic_start
morph_cycle	yes	no
morphological factor	no	yes
time scaling	no	with morphological factor
Create new layers	yes	Up to 2 layers
Grid perturbation (random)	distortion	-
<b>Bedmaterial</b>		
Grain class	Single or multi grain classes	Single or multi grain classes
Layer	Multiple layers	Up to 2 layers
Fix bed elevation	.2dm mesh or node list	via SOILDEF, .2dm mesh or over region (index)
<b>Bedload</b>		
Bedload transport	Simple upwind scheme	HLL-type Approximate Riemann Solver (Soares-Frazão and Zech, 2011) Simple upwind scheme Simple upwind scheme
Closure formula	mpm - engelundhansen mpmh power_law mpm_multi wilcockcrowe ashidamichiue	MPM MPM_like (adaptable) engelundhansen - GRASS_like (adaptable) MPM_multi (adaptable) wilcockcrowe -

	<b>BASEMD</b>	<b>BASEHPC</b>
	parker	-
	rickenmann	-
	smartjaeggi	smartjaeggi
	smartjaeggi_multi	-
	wu	-
	vanriijn	-
<b>Boundary</b>		<b>Standard</b>
- Inflow	-	sedimentograph
	sediment_discharge	sedimentograph_warea
	-	sedimentograph_conveyance
	-	equilibrium_in
	IOUp	transport_capacity
	transport_capacity	transport_capacity_warea
	-	transport_capacity_conveyance
	-	
- Outflow	IODown	equilibrium_out
Parameters	upwind factor	-
	cell average bedload flux	cell average bedload flux (default)
Direction	lateral_bed_slope	LATERAL_SLOPE
	curvature_effect_static	-
	curvature_effect_dynamic	CURVATURE
Inner boundary	weir, open	<b>Internal:</b>
	-	equilibrium_linked
<b>Incipient motion</b>	angle_of_repose	repose_angle
	local_slope_vanriijn	van_riijn
	local_slope_chen	chen_et_al
<b>Gravitational transport</b>	yes	yes
<b>Source</b>		
Type	sediment_discharge	sediment_discharge
	dredge	-

**Table 6.7** Introduction of the *tracers* block in the domain *BASEplane*

	<b>BASEMD</b>	<b>BASEHPC</b>
<b>Parameters:</b>		
Number of species	no	num_tracers (max. 5)
Starting time	no	tracers_start
<b>Boundary</b>		
Type	no	<b>Standard</b> discharge_in discharge_in_warea concentration_in zero_gradient_out
<b>External source</b>		
Type	no	total and concentration
Sink behavior	no	exact, available, infinity
<b>Initial</b>		
Type	no	zero uniform continue region_defined

## 6.5 Simulation

**Table 6.8** Main changes regarding simulation parameters

	<b>BASEMD</b>	<b>BASEHPC</b>
Command file type	run.bmc	simulation.json
Simulation time	start_time total_run_time output_time_step restart_time_step console_time_step reference_time	start end out - - -
Timestep	initial_time_step minimum_time_step	init minimum

	<b>BASEMD</b>	<b>BASEHPC</b>
Simulation outputs	wse	water_surface
	depth	water_depth
	velocity	flow_velocity
	abs_velocity	flow_velocity_abs
	abs_momentum	-
	z_element	bottom_elevation
	z_node	-
	friction	friction_chezy
	deltaz	delta_z
	tau	-
	specific_discharge	spec_discharge
	concentration	-
	susp_load	-
	susp_net_deposition_rate	-
	susp_grain_conc	suspended_conc
	susp_deltaz	delta_z
	susp_total_pickup	-
	susp_total_deposition	-
	susp_grain_pickup	-
	susp_grain_deposition	-
	-	suspended_theta
	theta_critical	theta_critical
	grain_size	-
	grain_bedload	-
	bedload_vec	-
	saturation	-
	sediment_sum	-
	-	bed_gradient
	-	theta
	-	trsp_capacity
	-	trsp_capacity_abs
	-	active_layer_thickness
	-	active_layer_fractions
	-	sublayer_fractions0
	-	sublayer_fractions1
	-	sublayer_elevations
	-	sublayer_id
	-	dm_arithmetic
	-	dm_geometric
	-	d_16
	-	d_50
	-	d_84
	-	d_90
	-	flow_radius
	-	flow_curvature
	-	flood_tracking
	pore_pressure	-
	-	ns_hyd_discharge
	-	ns_mor_discharge
version 4.1.0	external_sediment_discharge	-
	radius_curvature	-
	radius_curvature_abs	-
	-	turb_k
	-	turb_reynolds

**BASEMD****BASEHPC**

## 6.6 Results

*Table 6.9 Main changes regarding the results parameters*

	<b>BASEMD</b>	<b>BASEHPC</b>
Command file type	run.bmc	results.json
Format	ascii, sms, tecplot, shape, vtk	xdmf
Output Type	node_centered element_centered BASEviz node_history element_history stringdef_history edge_history boundary_history balance avs_ucd sediment_grid	- element_centered - - - nodestring - nodestring - - -

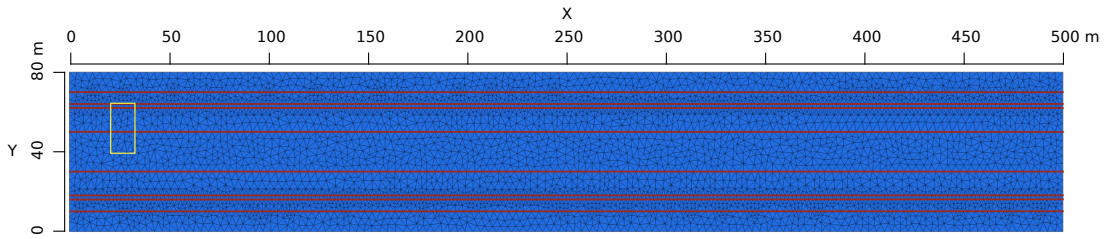
## 6.7 Case example

### 6.7.1 Description

This section provides helpful hints for the users already familiarised with BASEMENT. For beginners, please have a look at the User Manual and the Tutorials first. The objective of this test case is to illustrate the main differences between BASEMD::BASEplane and BASEHPC::BASEplane. A hydraulic simulation of a simple straight trapezoidal channel illustrates the changes and differences between the two versions. The geometry of the channel is specified in Table 6.10.

*Table 6.10 Geometry of trapezoidal channel*

<b>Type</b>	<b>Value</b>	<b>Unit</b>
Length	500	m
Bed width	20	m
Bank slope	1/3	-
Bank height	4	m
Bank crest width	2	m
Bed slope	0.2	%
Flood plain width	10	m



**Figure 6.1** Quality mesh used for the case example with breaklines (red)

## 6.7.2 Computational mesh

The topology of the computational mesh used for BASEHPC::BASEplane is different than for BASEMD::BASEplane, see Tables 6.1 and 6.2. This section describes the differences between the two mesh types and provides a guideline on how to import a v2.x (or BASEMD::BASEplane) mesh into BASEMENT version 4.

### 6.7.2.1 Quality mesh

**Table 6.11** Quality mesh attributes

Type	Value
Number of cells	9418
Number of vertices	4862
Minimum triangle angle	30
Cell maximum area	10
Number of breaklines	8
Regiondefs	3 (channel bed, banks and floodplains)

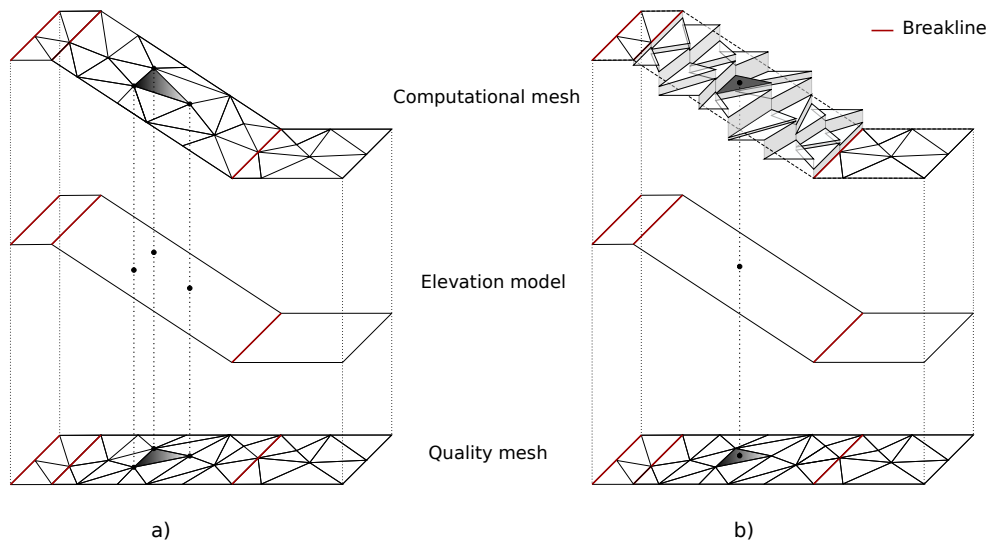
The quality mesh contains all the mesh attributes defined by the user, i.e. cell size, breaklines, regiondefs, minimum triangle angle and maximum cell area, but has no elevation information. The quality mesh of the simple straight trapezoidal channel (Figure 6.1) is identical for both versions, BASEMD and BASEHPC and its attributes are listed in Table 6.11. The procedure to generate a quality mesh with QGIS using the BASEmesh plugin is explained in the Tutorial of BASEMENT v2.8 documentation.

### 6.7.2.2 Computational mesh

The elevation information can be provided by cross sections, height contour lines, raster data or elevation functions. The computational mesh is generated by interpolating the elevation data at specific points of the quality mesh.

The main difference between the computational mesh of BASEHPC and BASEMD lies in the process of attributing the elevation information to the mesh cells. A small surface area (yellow rectangle, Figure 6.1) is schematically reproduced on Figure 6.2 in order to illustrate the two approaches used to create the computational mesh.

In BASEMD, the topographic elevation is attributed to the cell vertices (Figure 6.2 a). The quality mesh defines the location on the elevation model at which the elevation information



**Figure 6.2** Schematic sketch of the elevation interpolation methods with breaklines (red):  
a) BASEMD b) BASEHPC

will be assigned to create the computational mesh. It results in a continuous interpolation of the topography between the vertices, displaying a variable elevation over the cell. In contrast, for BASEHPC, the elevation information is assigned to the coordinate of the cell center, resulting in a constant elevation over the cell surface (Figure 6.2 b).

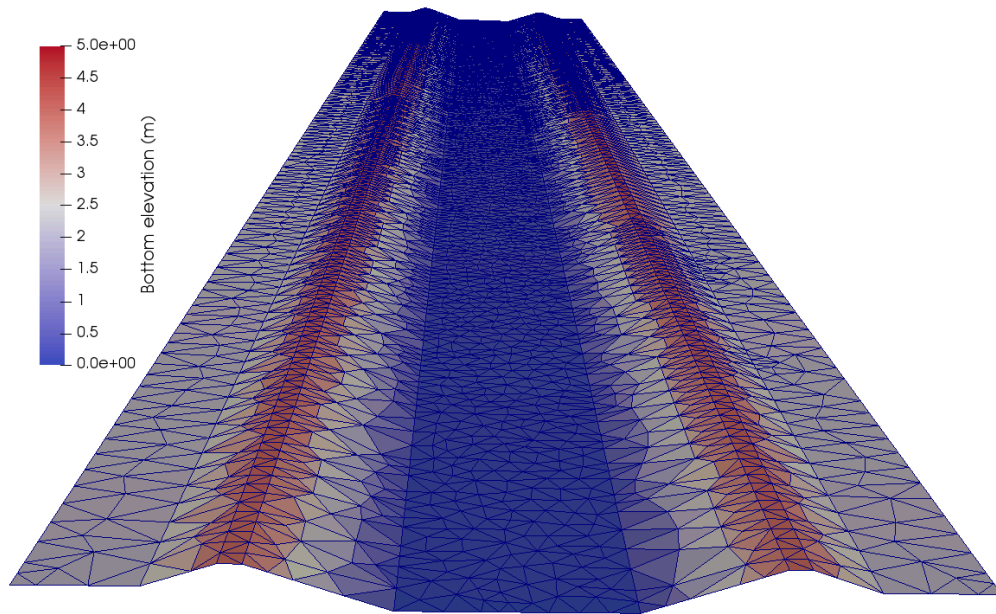
Breaklines are used to shape the mesh by separating the domain into specific zones (river bed, banks and floodplains) of similar feature (e.g. friction, cell mesh density, ...). The edges of cells adjacent to the breakline lie on the breakline. In BASEMD, the elevation information of the breakline is exactly similar to that of the vertices along it, which allows to represent clear changes in slope as for example between the bed and the bank. This is not the case in version 3.x, as the elevation information is not assigned to vertices anymore but to the coordinate of the cell center. Therefore, the definition of breaklines deserves some particular attention in BASEHPC, where two or more breaklines need to be defined in order to obtain cells at desired elevation (e.g. the elevation at the bank crest has to be guaranteed by two breaklines).

The computational mesh of the trapezoidal channel for the simulation with BASEMD is represented on Figure 6.3 and the computational mesh for the simulation with BASEHPC on Figure 6.4. The flow direction is from top to bottom.

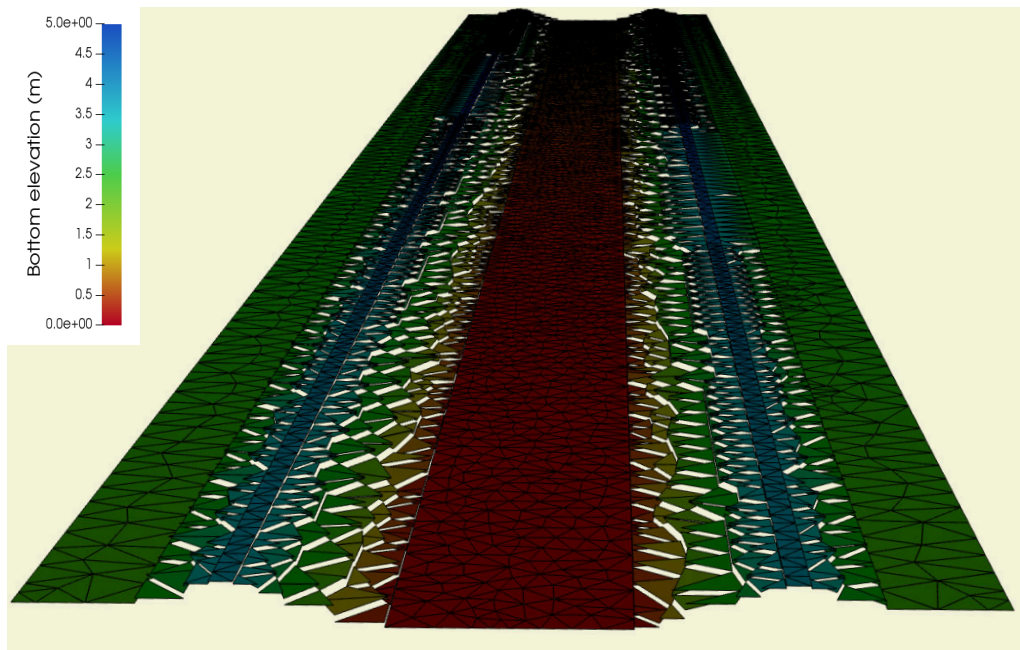
### 6.7.2.3 Import of a 2.x to a compatible 4.x computational mesh

The computational mesh of BASEHPC::BASEplane (version 4) can be obtained using a computational mesh of BASEMENT version 2.x. The import of a 2.x mesh to a mesh compatible with BASEMENT version 4.x consists of defining a unique elevation value to each cell from the elevation information of the 2.x mesh vertices.

First of all, the computational mesh version 2.x has to be composed of triangular elements. The QGIS plugin BASEmesh is used to generate a computational mesh for BASEMENT version 2.x, the tutorial is provided in the Tutorial of BASEMENT v2.8 documentation. The computational mesh is saved in a .dm file and the stringdefs list is saved in a



**Figure 6.3** Computational grid BASEMD with breaklines (view from downstream)



**Figure 6.4** Computational grid BASEHPC with breaklines (view from downstream)



```

MESH2D #created automatically via meshModel tool
NUM_MATERIALS_PER_ELEM 1
E3T 1 1155 861 1154 2
E3T 2 137 3166 2145 3
... ..
... ..
... ..
ND 3510 401.701104 0.719666 0.803402
ND 3511 292.228530 35.734722 2.584457
NS 3 6 34 65 123 654 -7 Stringdef_name

```

**Figure 6.5** Lines to add manually to the 2dm mesh file (orange)

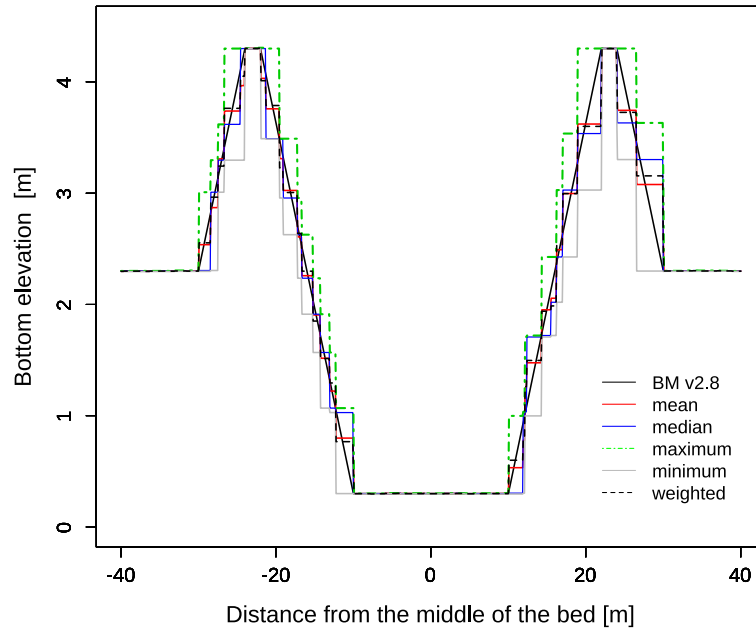
separate .txt file. In order to use the computational mesh version 2.x for simulations with BASEHPC::BASEplane (version 4), the .2dm mesh file has to be modified:

1. Add manually the line `NUM_MATERIALS_PER_ELEM 1` after the 1st line of the 2dm file and copy the stringdefs (list of nodes or nodestring) saved in the separate text file to the end of the 2dm file (see example Figure 6.5). The “Stringdef\_name” must be replaced accordingly. Please Note: The number of nodes per nodestring is limited to 40. Larger nodestrings must be split up.
2. Inside the model.json file (model setup, see Section 6.7.3), give the name of the modified .2dm mesh file in the GEOMETRY block and choose between the interpolation methods:
  - Mean: the average elevation of the three cell vertices is calculated
  - Median: the median elevation of the three cell vertices is calculated
  - Maximum: the maximum elevation value of the cell vertices is allocated to the cell.
  - Minimum: the minimum elevation value of the cell vertices is allocated to the cell.
  - Weighted: same as for the mean interpolation method, it calculates the average elevation of the three vertices after applying a weight factor that accounts for the cell geometry (triangle). The mean and weighted interpolation methods give the same results in case of equilateral triangle.

The interpolation method defines how the elevation information stored on the nodes of the computational mesh version 2.x is interpolated in order to generate a computational mesh compatible with BASEHPC::BASEplane (version 4). The choice of the interpolation method and its relevance in the numerical simulation is let to the user.

The result of the different interpolation methods is displayed in Figure 6.6, where a cross section of the trapezoidal mesh illustrates the local differences between the mesh of BASEMD::BASEplane and the different interpolated meshes used in simulations with BASEHPC::BASEplane.

Moreover, Figure 6.7 represents the same cross section on the trapezoidal mesh for the same mesh resolution but with only 2 breaklines defined on each side of the bank crest. The change in slope at the levee bottom and crest is less distinct compared to Figure 6.6 and most of the interpolation methods can't preserve the bank elevation. In the case



**Figure 6.6** Comparison of interpolation methods with the mesh of BASEMENT::BASEplane on a cross section at  $x=150$  m of the trapezoidal channel with breaklines

of a numerical simulation where the exact elevation of the bank is required (e.g. to calculate the bankfull discharge), the definition of breaklines ensures the conservation of the bank elevation, independently from the chosen interpolation methods. Otherwise, the interpolation methods “maximum” and “median” can be appropriate in the situation with only one breakline defined at the crest.

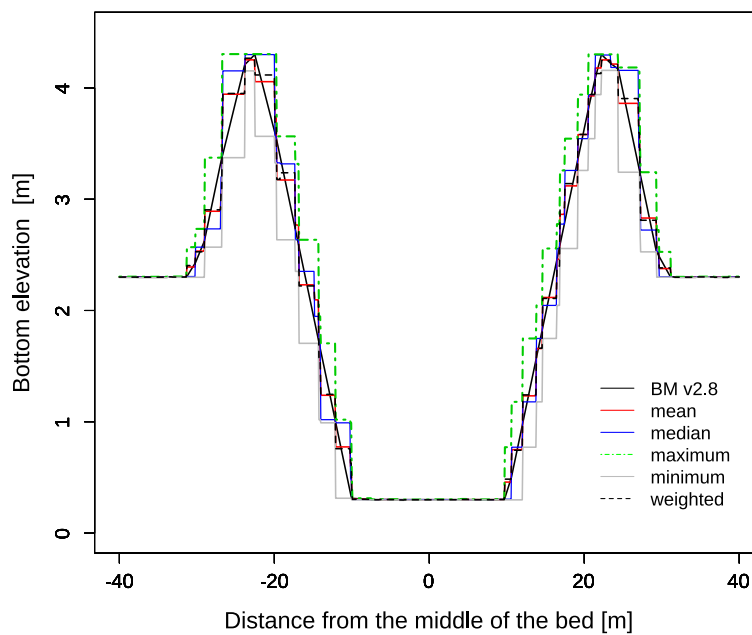
The regions delimited by breaklines e.g. the levees or the river bed, can be assigned to different interpolation methods over the computational mesh. Figure 6.8 illustrates the same cross section but for the trapezoidal mesh with a coarser mesh resolution and with breaklines. In this example, the bank side facing the river bed could be defined as “mean” while the other sides (facing the floodplain) could be defined as “maximum”.

### 6.7.3 Setup and simulation

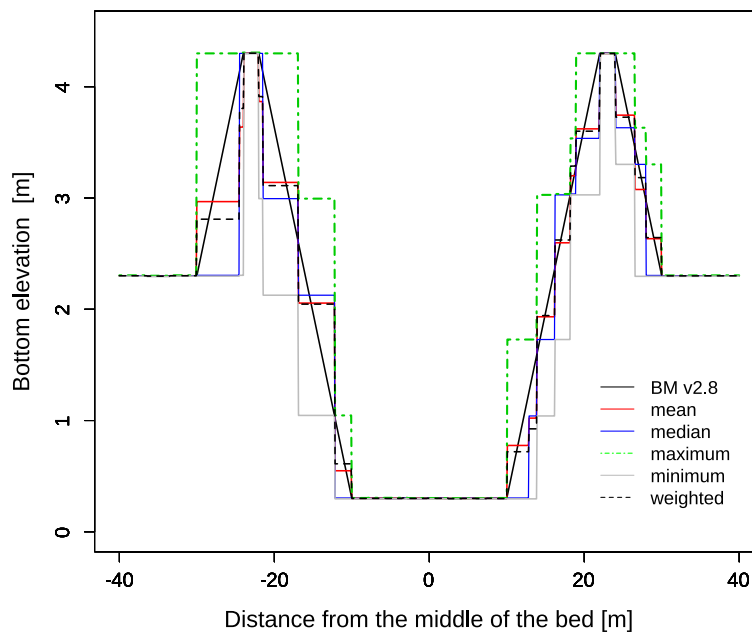
The simulations were performed for all the interpolation methods using BASEMENT version 4. A simple hydraulic simulation starting from dry initial conditions and with a progressive discharge from zero to the bankfull discharge (water depth around 4 m) was running for 20000 seconds. The output data was recorded every 2000 seconds for which the steady state condition was ensured. The Strickler friction type is used with a value of 30. Standard boundaries are used with the inflow boundary of type ‘uniform\_in’ and the outflow boundary defined as ‘uniform\_out’. The numerical simulation is performed with the HLLC Riemann solver.

Different files are needed to setup the numerical simulation of BASEMENT version 3:

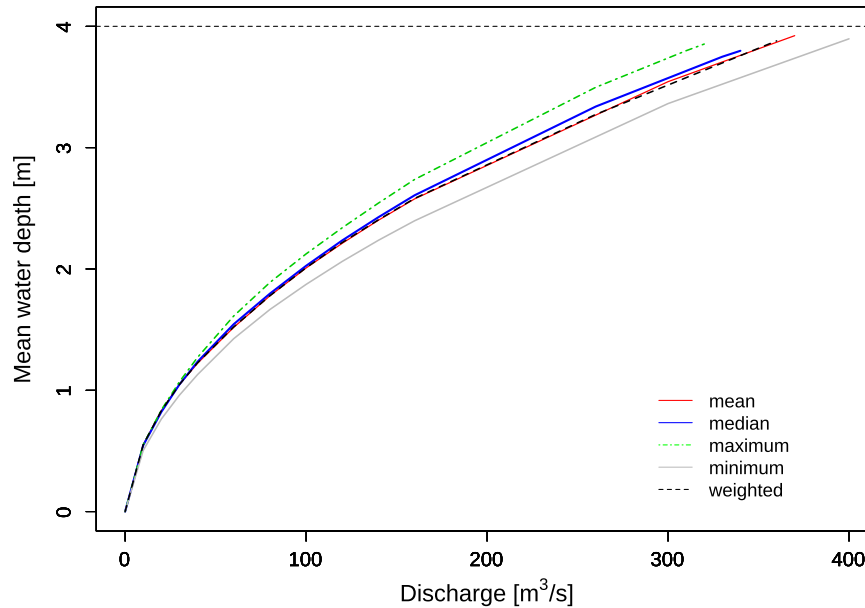
- Computational mesh (2dm), including stringdef specification



**Figure 6.7** Comparison of interpolation methods with the mesh of `BASEMD::BASEplane` on a cross section at  $x=150$  m of the trapezoidal channel with only one breakline defined at the bank crest



**Figure 6.8** Comparison of interpolation methods with the mesh of `BASEMD::BASEplane` on a cross section at  $x=150$  m of the trapezoidal channel with breaklines and for a coarser mesh resolution



**Figure 6.9** Comparison of  $H$ - $Q$  relations between the simulations of BASEMENT v3 for different interpolation types on the trapezoidal channel at  $x = 150$  m.

- Configuration files (model.json, simulation.json and results.json)
- Boundary condition data (.txt)

Three configuration files, model.json, simulation.json and results.json replace the command file (.bmc) of BASEMENT version 2.x. See the User Manual for more information about their attributes. As described in Section 6.7.2.3, the specification of stringsdefs, i.e. the list of nodes is included in the computational mesh (.2dm) in BASEHPC::BASEplane (version 4).

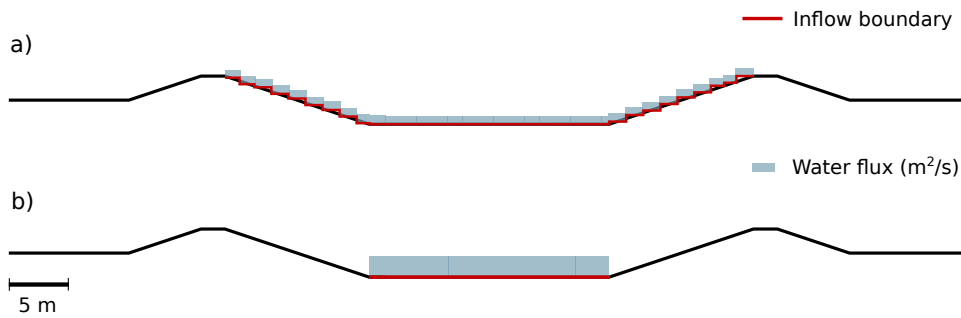
## 6.7.4 Results and discussion

In BASEMENT version 4.x, the output data are generated either on cells (cell centered) or at the boundaries (stringdefs). Various results are available (see Table 6.9 and Table 6.8).

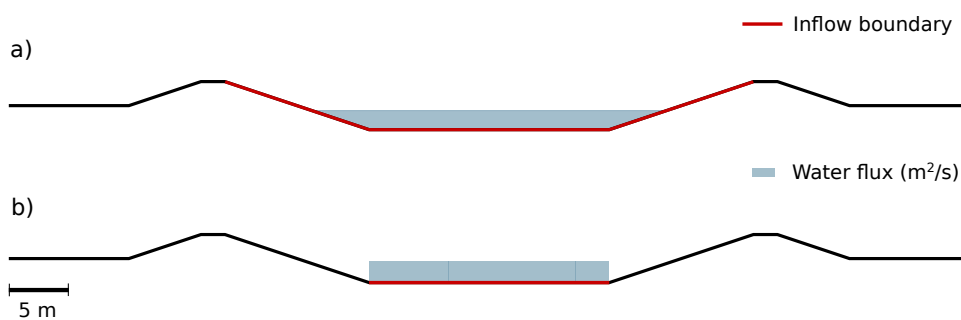
### 6.7.4.1 Hydraulic results

The result of the simulations with BASEMENT version 4 for different interpolation methods are compared in a stage discharge rating curve (Figure 6.9). The mesh features are summarized in Tables 6.10 and 6.11.

The bankfull water depth is 4 m and is represented by the dashed horizontal line. The bankfull discharge represents the capacity maximum of the channel before water overflows the channel banks. The smaller channel capacity is reached with the interpolation type “maximum” and the maximum capacity with the interpolation type “minimum”.



**Figure 6.10** Channel cross section and inflow boundary limit in `BASEHPC::BASEplane`  
 a) Inflow boundary limit set at levee's highest point b) Reduced inflow boundary limit



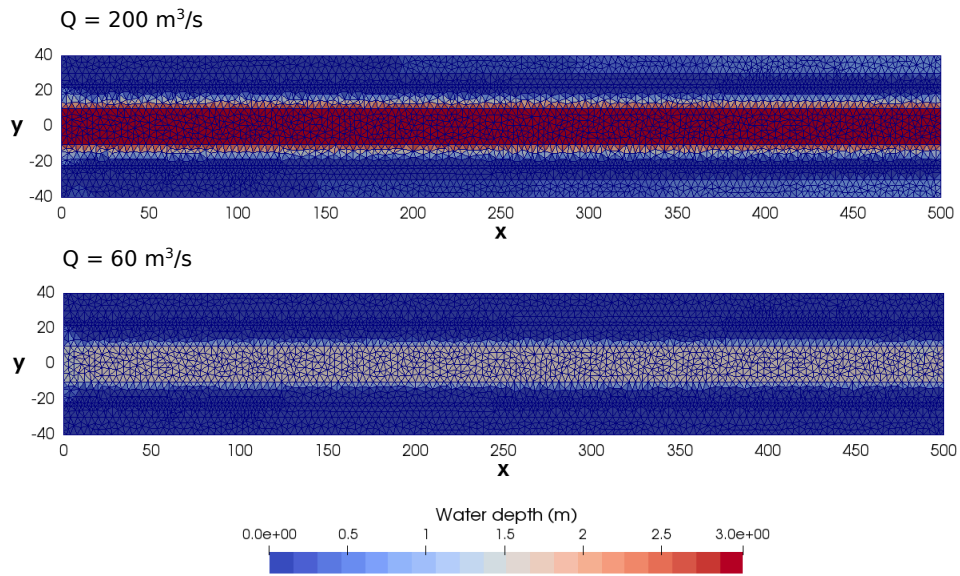
**Figure 6.11** Channel cross section and inflow boundary limit in `BASEMD::BASEplane`  
 a) Inflow boundary limit set at levee's highest point b) Reduced inflow boundary limit

#### 6.7.4.2 Boundary conditions

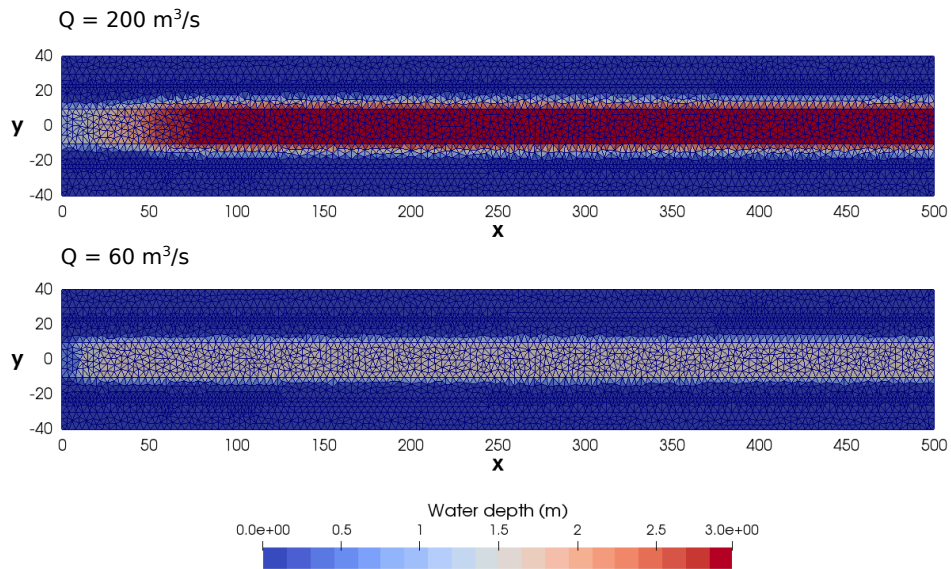
In `BASEHPC::BASEplane`, the inflow data is averaged over the boundary length and the mean value is uniformly distributed over the cell edges. This assumption simplifies the boundary conditions compared to `BASEMD::BASEplane`. Figure 6.10 and Figure 6.11 show two simplified representation of the averaged discharge value distribution on the element edges of the inflow boundary cross section for `BASEMD::BASEplane` and `BASEMD::BASEplane` respectively.

An inflow boundary defined between the top elevation of the two levees in `BASEHPC` (red line in Figure 6.10, a) generates an undesired converging flow from the bank towards the channel center ( $Q = 60 \text{ m}^3/\text{s}$ ) and small flux towards the floodplains as represented on Figure 6.12 for a discharge value  $Q = 200 \text{ m}^3/\text{s}$ . An inflow boundary restricted to the channel bed width (Figure 6.10, b) will locally increase the flow velocity at the inflow boundary as the discharge increases. In this case, stable flow conditions are obtained after a distance of 20-30 meters from the inflow boundary. Figure 6.13 illustrates the location of high flow velocity by an area of low water level. The water depth at boundary conditions (inflow and outflow) depends on the stringdef length, the friction value and the boundary condition type (froude, uniform, ...).

The boundary conditions in `BASEMENT v4` are more sensitive to the domain geometry and boundary parameters than those in `BASEMENT v2.8`, therefore, the resulting values located near the boundary conditions should be interpreted with caution and enough space should be provided to reach stable flow conditions. The stringdef length is limited to a maximum of 40 nodes. In case of large computational mesh with fine resolution, the



**Figure 6.12** Planar view of the simulation results using BASEMENT version 4 of the trapezoidal channel with breaklines and for two discharge stages. Inflow boundary ( $x=0.0$  m) defined between the top elevation of the levees (Figure 6.10, a), inducing a converging flow from the levee towards the channel center and small fluxes towards the floodplains for higher discharge  $Q=200$  m<sup>3</sup>/s



**Figure 6.13** Planar view of the simulation results using BASEHPC of the trapezoidal channel with breaklines and for two discharge stages. Inflow boundary ( $x=0.0$ m) restricted to the channel bed (Figure 6.10, b), inducing an increase of the flow velocity.

boundaries shall be split into several smaller stringdef of equal length and consequently, the discharge applied to the boundaries has to be adapted.

#### **6.7.4.3 Discussion**

This case example of the hydraulic simulation of a trapezoidal channel pointed out the differences between BASEMD::BASEplane and BASEHPC::BASEplane for the topology and the boundary setup. The use of a BASEMENT v2.x mesh into BASEMENT v4.x is possible by interpolation but the simulation results may differ between the two versions due to the new topology. The simulation with BASEMENT v4.x based on the topology of version 2.x must be calibrated and must be considered as a new model. Moreover, the inflow boundary should be carefully defined in order to avoid unexpected flow behaviour at the boundary. Finally, additional breaklines might be required in order to attribute a precise elevation to the edges or to some parts of the mesh.

---

# Performance on Multi-Core Processors and GPU

## 7.1 General

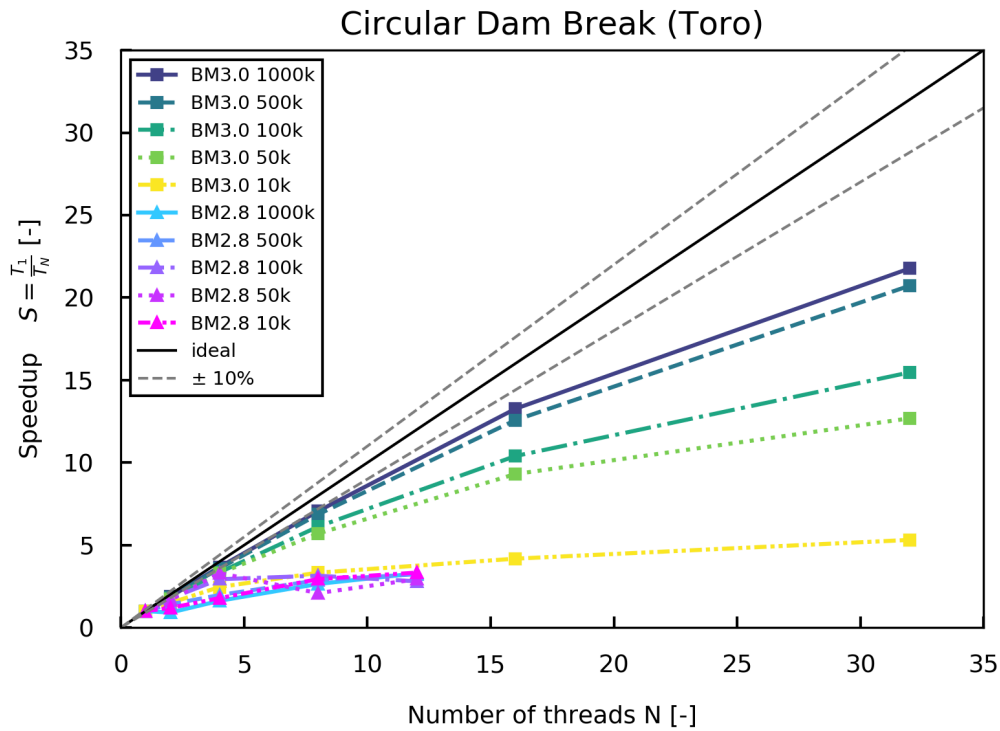
The performance of BASEMD and BASEHPC is assessed by comparing the execution time of simulations based on a common test case. The circular dam break test case is introduced here but explained in more detail in the “Test case” section of this documentation. The circular dam break is a hydrodynamic simulation that reproduces the wave propagation induced by the break of a circular dam located at the center of the computational mesh. The reference solution of the circular dam break is given by Toro (2001). The simulation was performed for BASEMD on multi-core processors (there is no GPU option for BASEMD), i.e. using the CPU backend on 1, 2, 4, 8 and 12 cores. For BASEHPC the CPU backends with up to 32 cores and the GPU backend using different GPU cards was used. The backend types are listed and described in more detail in the section “Test case”. Besides, five different mesh resolutions were defined for the circular dam break, with 10'000 cells (10k), 50'000 cells (50k), 100'000 cells (100k), 500'000 cells (500k) and 1'000'000 cells (1000k).

## 7.2 Scalability

The speedup of the simulations performed on CPU hardware is shown in Figure 7.1. The speedup  $S$  of the respective version is calculated as the division of the sequential runtime  $T_1$  by the runtime with a certain number of cores  $T_N$ . The black line represents the ideal speedup according to the increasing number of threads. The speedup is a measure for the parallelizability of the respective version and indicates how the computing time scales with the number of used processor cores. A linear or ideal increase in speed  $S$  results for  $S = N$ .

For the smallest computational grid (10k), the speedup of both BASEMENT modules only scale linearly up to approximately 4 threads before reaching a plateau (due to overhead).



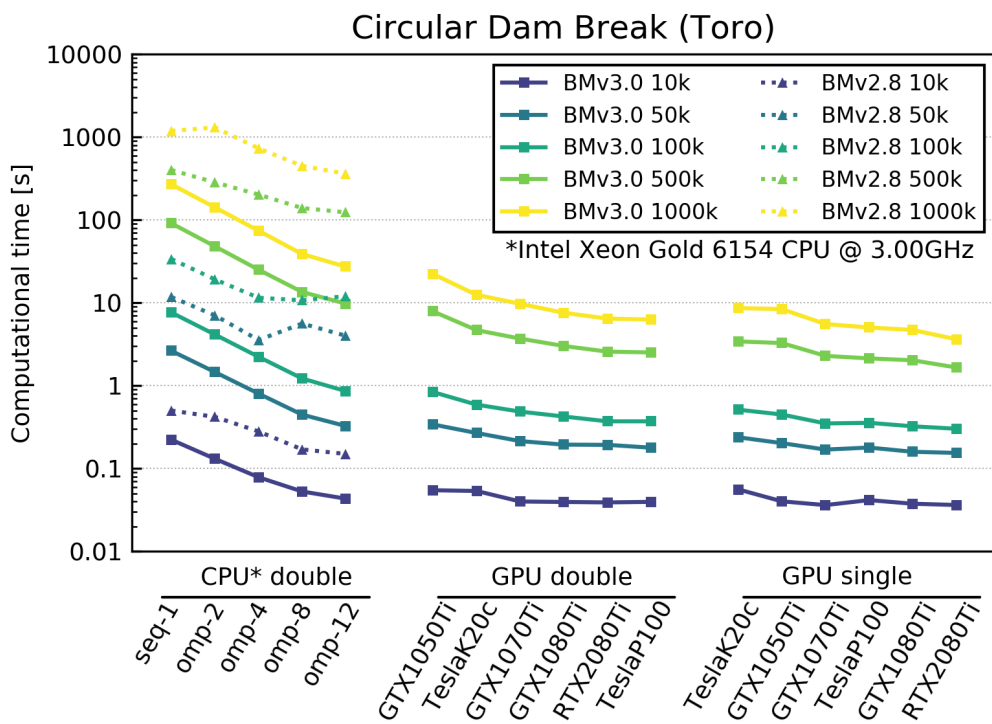


**Figure 7.1** Speedup of the circular dam break test case performed on CPU for an increasing number of threads and different mesh sizes

For BASEMD, the scalability does not change significantly for the larger mesh sizes and hence, the performance does not increase significantly anymore when using more than 4 threads. In contrast, the speedup of BASEHPC scales almost linearly up to 16 threads for the four larger meshes and up to 32 cores for the two largest meshes. Overall, BASEHPC exhibits significantly improved scalability compared to BASEMD.

### 7.3 Computational Time

The execution time of all the simulations is shown in Figure 7.2. The execution time increases with the computational mesh size for all backends. The execution times obtained on the CPU hardware indicate the significantly improved performance of BASEHPC compared BASEMD. This increase in performance by a factor of up to 13 is the results of completely restructuring the software. The performance of BASEHPC can be improved even further by the use of GPU hardware. For example, the runtime for the largest grid (1000 k) on the Intel processor with 32 cores is 8.7 s, while with the RTX2080Ti graphics card (single precision) only 3.6 s are required, which corresponds to a reduction of the runtime by a factor of 2.4. It should be noted that the results of simulations with single and double precision can vary greatly depending on the problem. When using GPUs, however, the significantly better price/performance ratio should be emphasized. For example, the GeForce GTX1080Ti card with double precision has about the same performance as the Intel Xeon Gold 6154 processors when using 32 cores, but with a 6 times lower purchasing price.



*Figure 7.2 Execution time of the circular dam break test case for different backends and mesh sizes*



---

# Specific Features of BASEMD

## 8.1 General

This chapter details some specific features of the BASEMD module which are not available for the BASEHPC module. The specific features of BASEMD include the coupling of the various submodules and with external programs during runtime via BASEextern, the automatic flow control using controllers, and the built-in runtime GUI tools BASEviz and HID.

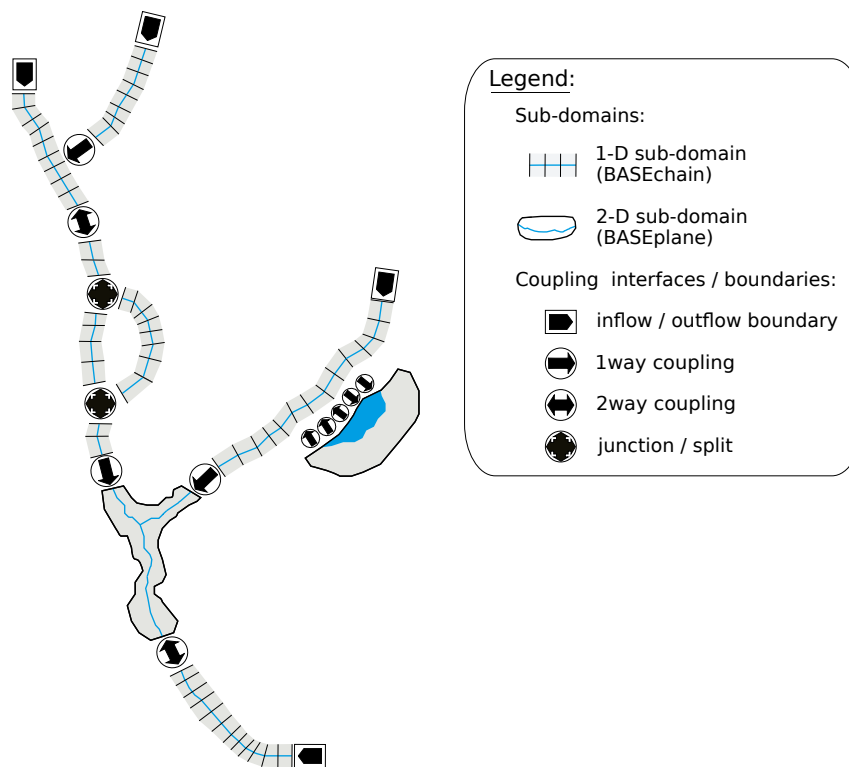
## 8.2 Model Coupling

### 8.2.1 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 8.1 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.



**Figure 8.1** River network with multiple BASEchain (1-D) and BASEplane (2-D) sub-domains and several coupling interfaces.

- 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.

### 8.2.2 Coupling Types

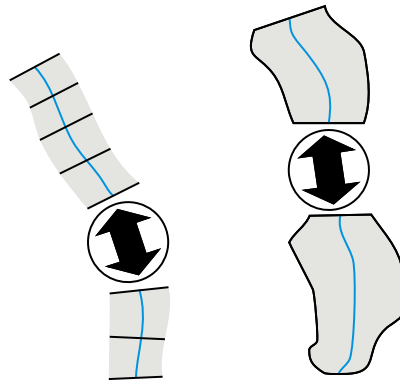
The implemented coupling types are briefly sketched below.

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

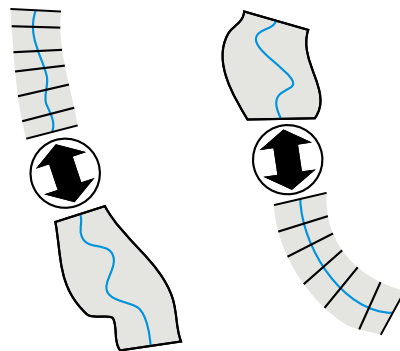
Single sub-domains can be combined sequentially via coupling interfaces at the upstream or downstream boundaries (Figure 8.2). This can also be done for sub-domains with mixed dimensionalities (1-D / 2-D, 2-D / 1-D), see Figure 8.3.

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



**Figure 8.2** 1-D / 1-D coupling (left), 2-D / 2-D coupling (right)



**Figure 8.3** 1-D / 2-D coupling (left), 2-D / 1-D coupling (right)

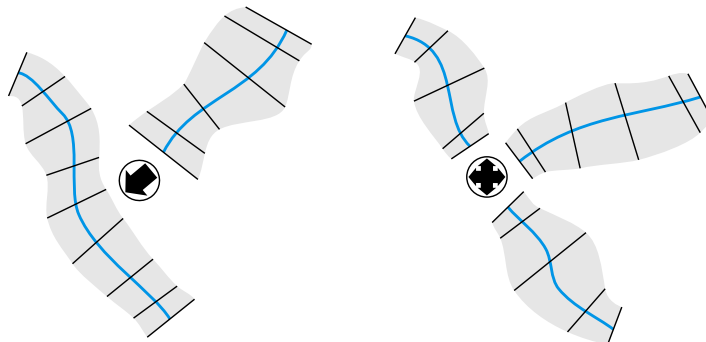
flow in any direction. This coupling type requires special ‘connection’ boundaries at the coupling interfaces.

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

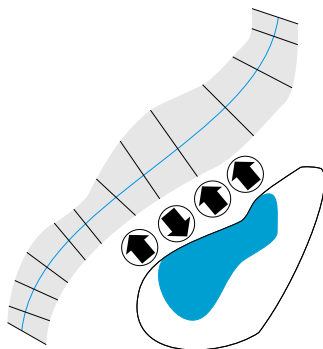
Coupling interfaces for river **junctions** or river **bifurcations** allow a simplified modelling of conjunctions of river branches within a 1-D river network (Figure 8.4).

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.**



**Figure 8.4** junction / bifurcation / confluence WSE (1-D)



*Figure 8.5 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 coupling interfaces which connect cross sections (1-D) with corresponding mesh elements (2-D), see Figure 8.5.

### 8.2.3 Coupling Mechanisms

#### 8.2.3.1 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.

#### 8.2.3.2 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.

## 8.2.4 Definitions of Exchange Conditions

### 8.2.4.1 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 8.1** Possible exchange conditions between sub-domains.

direction of exchange	type of coupling	exchange variables
in downstream direction	boundary conditions & sources	Q : discharge q <sub>b,g</sub> : bed load C <sub>g</sub> : concentration
in upstream direction	boundary conditions & sources	z <sub>s</sub> : 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.



### 8.2.4.2 Exchange conditions for mixed-dimensional sub-domains

Exchange via boundary/sources:

**Table 8.2** *Exchange conditions for mixed-dimensional coupling  
( $i$ =index of 2-D edge or 2-D element)*

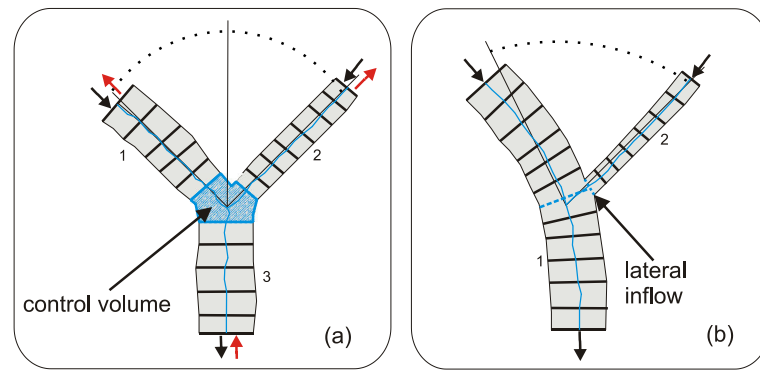
Exchange variable	Exchange equations	Nr of exchange terms
Discharge	$1D \rightarrow 2D: q_i^{2D} = \omega_i Q^{1D}$ ( $\omega_i$ =area/length weighting or conveyance weighting) $2D \rightarrow 1D: Q^{1D} = \sum_i^n q_i^{2D}$	1
Water surface	$1D \rightarrow 2D: z_{S,i}^{2D} = z_S^{1D}$ $2D \rightarrow 1D: z_S^{1D} = \frac{1}{n} \sum_i^n z_{S,i}^{2D}$	1
Bed load	$1D \rightarrow 2D: q_{b,g,i}^{2D} = \omega_i q_{b,g}^{1D}$ ( $\omega_i$ =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

### 8.2.4.3 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 8.6). 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.



**Figure 8.6** 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.

**Table 8.3** 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	$Q_{up1}C_{up1,g} + Q_{up2}C_{up2,g} = Q_{down}C_{down,g}$	$g=1..ng$

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.

#### 8.2.4.4 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  $\phi$  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.

**Table 8.4** 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

#### 8.2.4.5 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 8.7 which illustrates river cross sections of the BASEchain sub-domain and the 2-D mesh of a floodplain modelled with a BASEplane sub-domain.

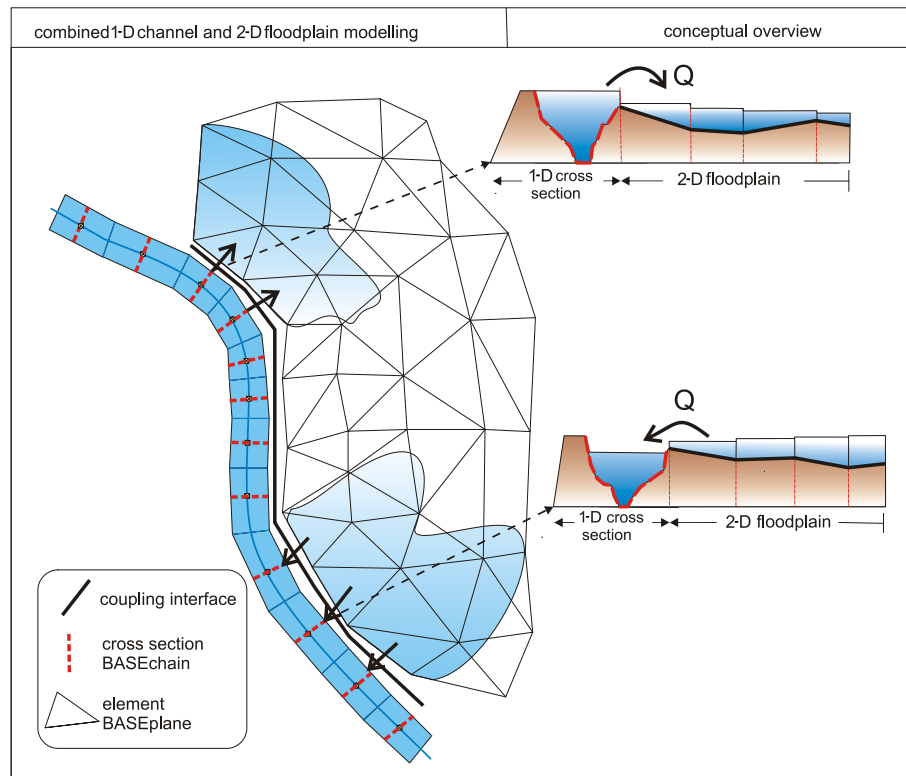
The coupling interfaces between the sub-domains are implemented as one-way couplings via source terms. As a consequence, only discharges are exchanged between the sub-domains. The exchange between the sub-domains is calculated as weir flow over the dykes of the 1-D cross section or as weir flow over 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 8.5** Exchange conditions for lateral coupling

	Exchange conditions	Nr. of equations
Discharge	$1D \rightarrow 2D: Q = \delta \mu \frac{2}{3} b_{weir} \sqrt{2gh}^{3/2}$ if $(z_{S,1D} \geq z_{S,2D})$ (side weir, $\delta$ = side weir reduction factor)	1
	$2D \rightarrow 1D: Q = \mu \frac{2}{3} b_{weir} \sqrt{2gh}^{3/2}$ 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



**Figure 8.7** Conceptual overview of combined 1-D river flow and 2-D floodplain modeling

two adjacent 1-D cross-sections.

#### 8.2.4.6 Data exchange for morphological simulations with multiple grain classes

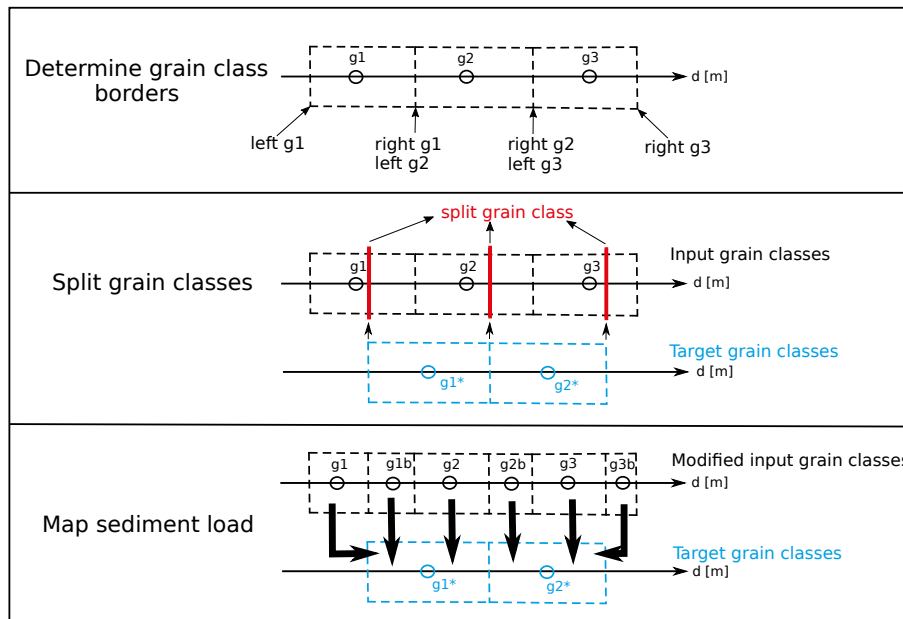
In morphological, coupled simulations the possibility exists that sub-domains can have differing grain compositions. The handling of data exchange for such cases is not trivial and unclear. But such situations may arise in coupled large-scale simulations where grain classes get finer along course of 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 8.8. The sediment mass balance is thereby fulfilled.

### 8.2.5 Synchronization Concept

#### 8.2.5.1 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



**Figure 8.8** Mapping of grain compositions from one sub-domain to another

of combined 1-D and 2-D simulations the spatial extends and time step sizes can vary considerable.

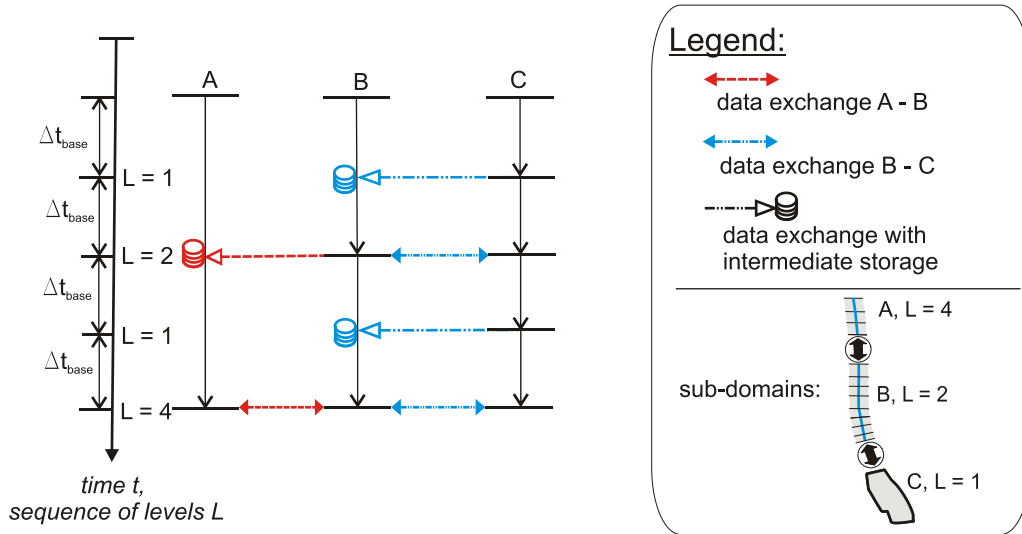
Mainly two different coupling concepts are often encountered for the selection of the time step sizes of the sub-domains.

- 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.

### 8.2.5.2 “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



**Figure 8.9** 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.

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 chosen as power-of-two multiples of the base time step size  $\Delta t_{base}$ . 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 8.9 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.

## 8.3 BASEextern - data exchange during runtime

### 8.3.1 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:

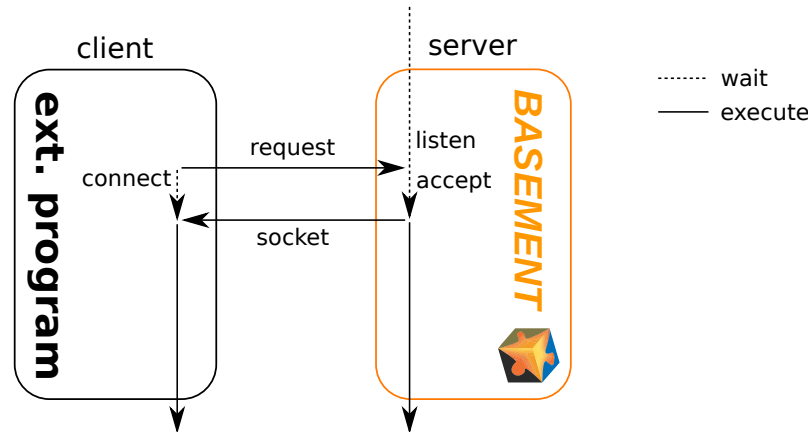
- An external Program may receive data which is sent by BASEMENT. Therefore the external program must be defined as an external sub-domain in the command file. In the OUTPUT block an output must be defined and connected with this external sub-domain. While executing, BASEMENT sends data as soon as it 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, BASEMENT 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.



**Figure 8.10** Connection request from external program (client) to BASEMENT (server)

- 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.

#### 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 programming 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.

#### 8.3.1.1 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.

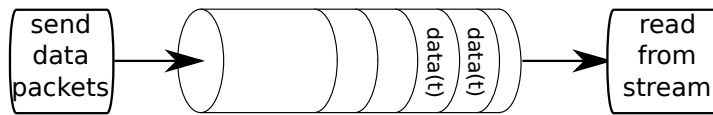
##### Create connection

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





**Figure 8.11** Sending and receiving data from socket stream (FIFO pipe)

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>
```

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.

## 8.4 Flow Control in River Systems

### 8.4.1 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 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.

### 8.4.2 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  $f(\cdot)$ , determining the values of the manipulated variables  $u(t)$  from the values of the monitored variables  $m(t)$ . This can be expressed using the following mathematical expression:

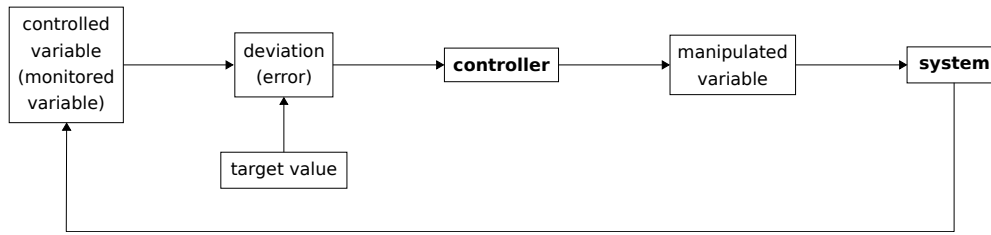


Figure 8.12 Basic Control Cycle

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

Logically, there can be multiple monitored and manipulated variables.

#### 8.4.2.1 Monitored Variable

A monitored variable is defined by

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

Here,  $\nu_i$  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.  $\tau_i$  is a delay time controlling when the information of the measured variable is fed into the controller.

#### 8.4.2.2 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).

### 8.4.3 Controller Types

#### 8.4.3.1 PID-Controller

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

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

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 Föh and Kühne (1987). Recommendations on how to choose the coefficients are given in the integrated help of the software.

## 8.5 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.

### 8.5.1 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 8.13 and Figure 8.14). 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.

- 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.

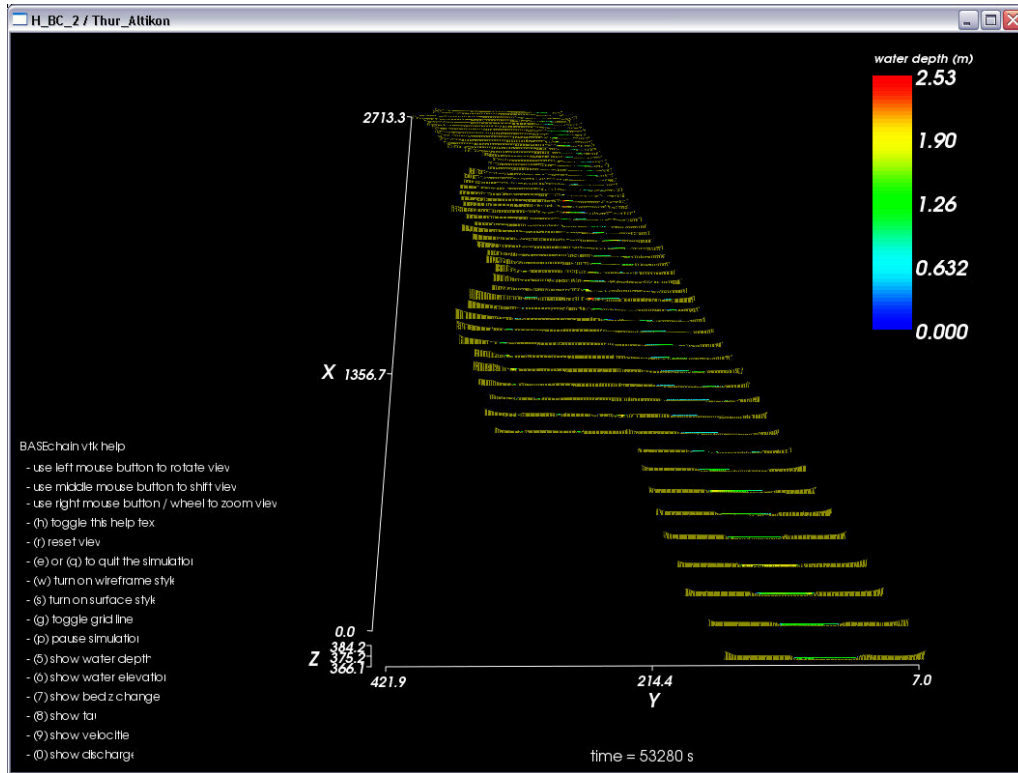


Figure 8.13 Visualization of BASEchain with BASEviz

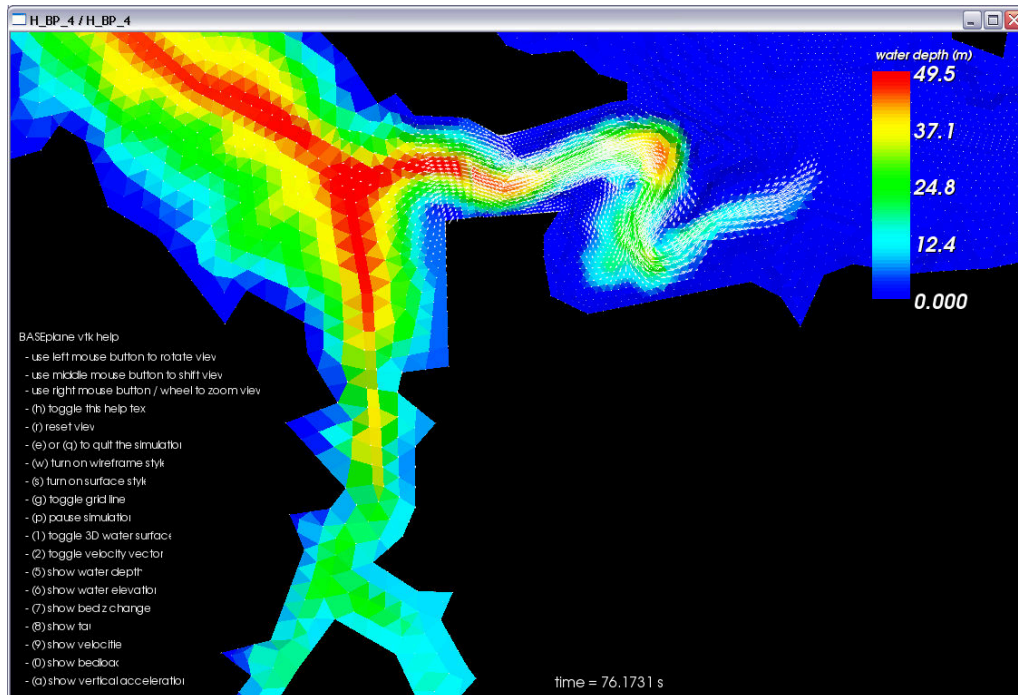
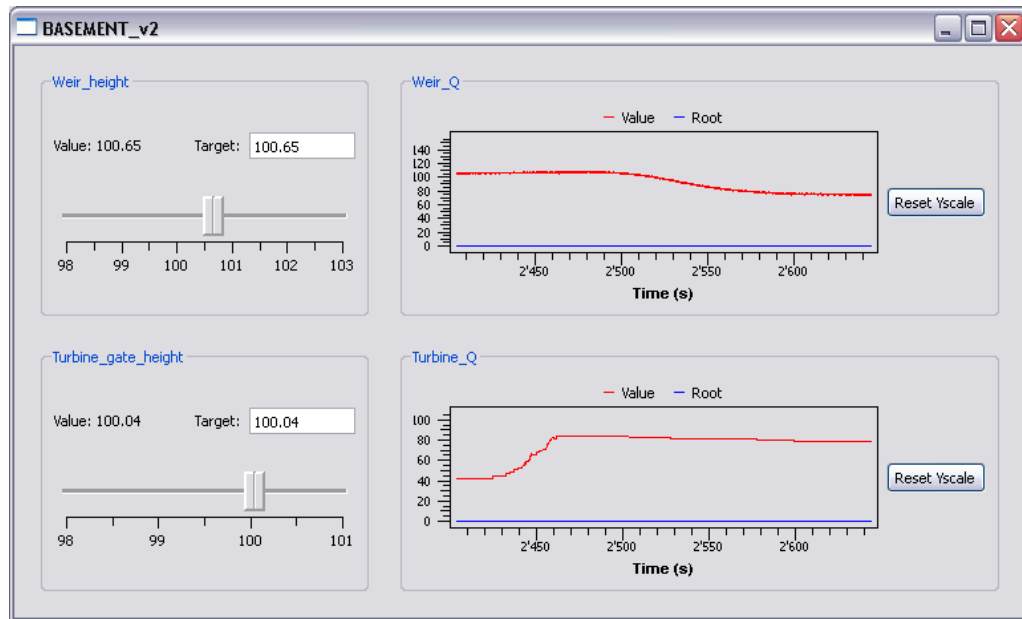


Figure 8.14 Visualization of BASEplain with BASEviz



**Figure 8.15** Interface for the manual control and monitoring of the selected variables.

## 8.5.2 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 8.15). 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 8.15). 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 8.5.1).



---

## References

- Beffa, C. (2002). Integration ein- und zweidimensionaler Modelle zur hydrodynamischen Simulation von Gewässersystemen. *Int. Symposium Moderne Methoden und Konzepte im Wasserbau*, ETH Zürich.
- Fäh, R. and Kühne, A. (1987). Numerische Simulation automatischer Stauregelungen bei Laufwasserkraftwerken. (p. 93. Heft 5/6, Ed.) *Wasser, Energie, Luft*, 79(5/6): 93–98.
- 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.
- Sanders, B.F. (2008). Integration of a shallow water model with a local time step. *Journal of Hydraulic Research*, 46(4): 466–475.
- Soares-Frazão, S. and Zech, Y. (2011). HLLC scheme with novel wave-speed estimators appropriate for two-dimensional shallow-water flow on erodible bed. *International Journal for Numerical Methods in Fluids*, 66.
- Toro, E.F. (2001). Shock-Capturing Methods for Free-Surface Shallow Flows. *John Wiley*, Chichester, New York.







**BASIC SIMULATION ENVIRONMENT  
FOR MODELLING OF ENVIRONMENTAL  
FLOWS AND NATURAL HAZARDS**

---

# **REFERENCE MANUAL BASEMD**

**VERSION 4.1.0  
JUNE 2024**



**BASEMENT**



# Contents

<b>1</b>	<b>Mathematical Models</b>	<b>5</b>
1.1	Governing Flow Equations . . . . .	5
1.1.1	Saint-Venant Equations . . . . .	5
1.1.1.1	Introduction . . . . .	5
1.1.1.2	Conservative Form of SVE . . . . .	5
1.1.1.3	Source Terms . . . . .	9
1.1.1.4	Closure Conditions . . . . .	9
1.1.1.5	Boundary Conditions . . . . .	11
1.1.2	Shallow Water Equations . . . . .	13
1.1.2.1	Introduction . . . . .	13
1.1.2.2	Closure Conditions . . . . .	17
1.1.2.3	Conservative Form of SWE . . . . .	18
1.1.2.4	Source Terms . . . . .	19
1.1.2.5	Boundary Conditions . . . . .	20
1.2	Sediment and Pollutant Transport . . . . .	22
1.2.1	Fundamentals of Sediment Motion . . . . .	22
1.2.1.1	Threshold Condition for Sediment Transport . . . . .	22
1.2.1.2	Influence of Bed Forms on Bottom Shear Stress . . . . .	23
1.2.1.3	Bed Armouring . . . . .	24
1.2.1.4	Settling Velocities of Particles . . . . .	25
1.2.1.5	Bed load Propagation Velocity . . . . .	26
1.2.2	Suspended Sediment and Pollutant Transport . . . . .	26
1.2.2.1	One Dimensional Advection-Diffusion-Equation . . . . .	26
1.2.2.2	Two Dimensional Advection-Diffusion-Equation . . . . .	26
1.2.2.3	Source Terms . . . . .	27
1.2.3	Bed Load Transport . . . . .	28
1.2.3.1	One Dimensional Bed Load Transport . . . . .	28
1.2.3.2	Two Dimensional Bed Load Transport . . . . .	30
1.2.3.3	Sublayer Source Term . . . . .	33
1.2.3.4	Closures for Bed Load Transport . . . . .	34
1.2.3.5	Abrasion in 1D . . . . .	41
1.2.3.6	Sediment Boundary Conditions . . . . .	41
1.2.4	Enforced Bed Movement . . . . .	41
1.2.5	Random Bed Perturbation . . . . .	42
1.3	Sub-surface flow . . . . .	42
1.3.1	Introduction . . . . .	42
1.3.2	Governing equations . . . . .	42
1.3.3	Constitutive relationships . . . . .	43
1.4	Morphodynamics and Vegetation . . . . .	44

1.4.1	Introduction . . . . .	44
1.4.2	Mathematical model . . . . .	45
1.4.3	Vegetation dynamics . . . . .	45
1.4.4	Feedback . . . . .	46
1.4.4.1	Vegetation effects on hydro-morphodynamics . . . . .	46
1.4.4.2	Hydro-morphodynamic effects on vegetation . . . . .	47
1.4.5	Range of the parameters used in previous works . . . . .	47
1.4.6	Reconstruction of the position of the mean water table . . . . .	48
<b>2</b>	<b>Numerics Kernel</b>	<b>51</b>
2.1	General View . . . . .	51
2.2	Methods for Solving the Flow Equations . . . . .	52
2.2.1	Fundamentals . . . . .	52
2.2.1.1	Finite Volume Method . . . . .	52
2.2.1.2	The Riemann Problem . . . . .	53
2.2.1.3	Exact Riemann Solvers . . . . .	55
2.2.1.4	Approximate Riemann Solvers . . . . .	56
2.2.2	Saint-Venant Equations . . . . .	59
2.2.2.1	Discretisation . . . . .	59
2.2.2.2	Discrete Form of Equations . . . . .	59
2.2.2.3	Discretisation of Source Terms . . . . .	60
2.2.2.4	Discretisation of Boundary conditions . . . . .	66
2.2.2.5	Solution Procedure . . . . .	73
2.2.3	Shallow Water Equations . . . . .	77
2.2.3.1	Discrete Form of Equations . . . . .	77
2.2.3.2	Discretisation of Source Terms . . . . .	79
2.2.3.3	Conservative property (C-Property) . . . . .	88
2.2.3.4	Discretisation of Boundary Conditions . . . . .	89
2.2.3.5	Solution Procedure . . . . .	98
2.3	Solution of Sediment Transport Equations . . . . .	99
2.3.1	Vertical Discretisation . . . . .	99
2.3.1.1	General . . . . .	99
2.3.1.2	Determination of Mixing Layer Thickness . . . . .	99
2.3.2	One Dimensional Sediment Transport . . . . .	103
2.3.2.1	Spatial Discretisation . . . . .	103
2.3.2.2	Discrete Form of Equations . . . . .	106
2.3.2.3	Discretisation of Source Terms . . . . .	115
2.3.2.4	Solution Procedure . . . . .	118
2.3.3	Two Dimensional Sediment Transport . . . . .	118
2.3.3.1	Spatial Discretisation . . . . .	118
2.3.3.2	Discrete Form of Equations . . . . .	119
2.3.3.3	Discretisation of Source Terms . . . . .	126
2.3.3.4	Gravitational Transport . . . . .	128
2.3.3.5	Management of Soil Layers . . . . .	131
2.3.3.6	Solution Procedure . . . . .	132
2.4	Time Discretisation and Stability Issues . . . . .	132
2.4.1	Explicit Schemes . . . . .	132
2.4.1.1	Euler First Order . . . . .	132
2.4.2	Determination of Time Step Size . . . . .	133

2.4.2.1	Hydrodynamic . . . . .	133
2.4.2.2	Bedload Transport . . . . .	134
2.4.2.3	Suspension Transport . . . . .	134
2.4.3	Implicit Scheme . . . . .	134
2.4.3.1	Introduction . . . . .	134
2.4.3.2	Time Discretisation . . . . .	134
2.4.3.3	Solution . . . . .	136
2.4.3.4	Integral Terms . . . . .	137
2.4.3.5	General Description of the Derivatives for the Matrix A . .	139
2.4.3.6	Determination of the Derivatives with Upwind Flux Determination . . . . .	141
2.4.3.7	Determination of the Derivatives with Roe Flux Determination . . . . .	143
2.4.3.8	Derivatives of the fluxes for an inner Weir . . . . .	146
2.5	Numerical Solution of Sub-surface Flow . . . . .	147
2.5.1	Introduction . . . . .	147
2.5.2	Lattice-Boltzmann Method . . . . .	148
2.5.3	Solution procedure . . . . .	149
2.5.3.1	Advection step . . . . .	149
2.5.3.2	Collision step . . . . .	150
2.5.3.3	Update of macroscopic variables . . . . .	150
2.5.4	Equilibrium functions . . . . .	150
2.5.5	Boundary and initial conditions . . . . .	151

### 3 References

153





---

## 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:

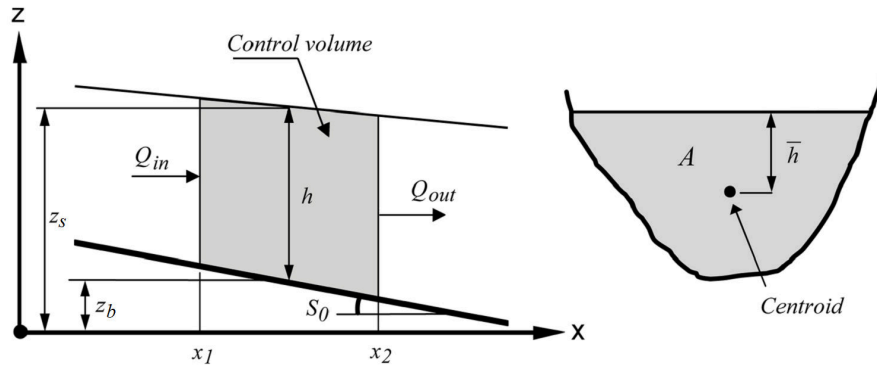
- Hydrostatic distribution of pressure: this is fulfilled if the streamline curvatures are small and the vertical accelerations are negligible
- Uniform velocity over the cross section and horizontal water surface across the section
- Small slope of the channel bottom, so that the cosine of the angle 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  $\rho$  is constant (incompressible flow). This leads basically to a



**Figure 1.1** Definition Sketch

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

$$\frac{d}{dt} \int_{x_1}^{x_2} A \, dx + Q_{out} - Q_{in} - q_l(x_2 - x_1) = 0 \quad (1.1)$$

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

$$\frac{d}{dt} \int_{x_1}^{x_2} A \, dx = \int_{x_1}^{x_2} \frac{\partial A}{\partial t} \, dx = \frac{\partial A}{\partial t} (x_2 - x_1)$$

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

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

### 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

$$p = m u$$

$$\frac{dp}{dt} = m a = \sum F$$

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

$$\frac{dp}{dt} = \sum F = \frac{d}{dt} \int_{x_1}^{x_2} u \rho A dx + u_2 \rho A_2 u_2 - u_1 \rho A_1 u_1 - u_x \rho q_l (x_2 - x_1) \quad (1.3)$$

where:

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

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

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

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

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

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

leads to the following formulation:

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

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:  $F_1 = -\rho g A_1 \bar{h}_1$  and  $F_2 = \rho g A_2 \bar{h}_2$  (hydrostatic pressure is  $p = \rho g h$ , the force is then  $F = pA$ )
- the weight of water (gravitational force) in x-direction:  $F_3 = \rho g \int_{x_1}^{x_2} A S_B dx$
- and the frictional force:  $F_4 = \rho g \int_{x_1}^{x_2} A S_f dx$

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

$$\frac{\sum F}{\rho(x_2 - x_1)} = \frac{\rho g A_2 \bar{h}_2 - \rho g A_1 \bar{h}_1}{\rho(x_2 - x_1)} = g \frac{\partial}{\partial x} (A \bar{h})$$

For the gravitational force and the friction force, the mean value theorem is applied:

$$\int_{x_1}^{x_2} A(S_B - S_f) dx = A(S_B - S_f)(x_2 - x_1)$$

This results in the following momentum equation:

$$\frac{\partial Q}{\partial t} + \frac{\partial}{\partial x} \left( \frac{Q^2}{A} \right) - q_l u_x = -g \frac{\partial}{\partial x} (A \bar{h}) + g A (S_B - S_f) \quad (1.6)$$

There is still an unknown  $\bar{h}$  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.

$$-g \frac{\partial}{\partial x} (A \bar{h}) = -g A \frac{\partial h}{\partial x} \quad (1.7)$$

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.

$$\frac{\partial Q}{\partial t} + \frac{\partial}{\partial x} \left( \frac{Q^2}{A} \right) + g A \frac{\partial z_S}{\partial x} + g A S_f - q_l u_x = 0 \quad (1.8)$$

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  $\beta$  accounting for the velocity distribution in the cross section (Cunge et al. (1980)), eq. 1.9 is obtained:

$$\frac{\partial Q}{\partial t} + \frac{\partial}{\partial x} \left( \beta \frac{Q^2}{A_{red}} \right) + g A_{red} \frac{\partial z_S}{\partial x} + g A_{red} S_f - q_l u_x = 0 \quad (1.9)$$

where  $A_{red}$  [m<sup>2</sup>] 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

$$\beta = \frac{A \sum_i k_{str_i}^2 h_i^{7/3} b_i}{\left( \sum_i k_{str_i} h_i^{5/3} b_i \right)^2} \quad (1.10)$$

If the equivalent roughness height is used

$$\beta = \frac{A \sum_i k_{s_i}^2 h_i^2 b_i}{\left( \sum_i k_{s_i} h_i^{3/2} b_i \right)^2} \quad (1.11)$$

### 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

$$W = gA \frac{\partial z_S}{\partial x} \quad (1.12)$$

- The bottom friction:

$$Fr = gA_{red} S_f \quad (1.13)$$

- The influence of lateral in- or outflow:

$$q_l u_x \quad (1.14)$$

However, in BASEchain, the influence of the lateral inflow on the momentum equation is neglected. Exceptions are the sideweir, coupling\_sideweir and bottom outflow source terms, where the influence of the lateral outflow on the momentum equation can be taken into account.

### 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:

$$\frac{\tau_B}{\rho} = gRS_f \quad (1.15)$$

As the unit of  $\tau_B/\rho$  is the square of a velocity, a shear stress velocity can be defined as:

$$u_* = \sqrt{\frac{\tau_B}{\rho}} \quad (1.16)$$

The velocity in the channel is proportional to the shear flow velocity and thus:

$$u = c_f \sqrt{gRS_f} \quad (1.17)$$

where  $c_f$  is the dimensionless Chézy coefficient. It is defined as  $c_f = C/\sqrt{g}$ , where  $C$  is the Chézy coefficient [ $\text{m}^{1/2}/\text{s}$ ].

If  $u$  is replaced by  $Q/A$ :

$$\frac{Q}{A} = c_f \sqrt{gRS_f} \quad (1.18)$$

results, with

$$S_f = \frac{Q|Q|}{gA^2c_f^2R} \quad (1.19)$$

Introducing the conveyance  $K$  :

$$K = \frac{Q}{\sqrt{S_f}} = Ac_f\sqrt{Rg} \quad (1.20)$$

$$S_f = \frac{Q|Q|}{K^2} \quad (1.21)$$

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. Therefore 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

$$c_f = \frac{k_{str}R^{1/6}}{\sqrt{g}} \quad (1.22)$$

### Logarithmic Friction Laws

The following approaches are implemented to determine the coefficient  $c_f$ :

Chézy:

$$c_f = 5.75 \log \left( 12 \frac{R}{k_s} \right) \quad (1.23)$$

Yalin:

$$c_f = \frac{1}{\kappa} \ln \left( 11 \frac{R}{k_s} \right) \quad (1.24)$$

Bezzola:

This approach uses the roughness sublayer  $y_R$  as relevant roughness height. Usually for rivers  $y_R \approx 1.0d_{90}$  is a good approximation. This approach takes small relative roughness heights into account Bezzola (2002).

$$\begin{aligned} c_f &= 2.5 \sqrt{1 - \frac{y_R}{R}} \ln \left( 10.9 \frac{R}{y_R} \right), & \text{for } \frac{R}{y_R} > 2 \\ c_f &= 1.25 \sqrt{\frac{R}{y_R}} \ln \left( 10.9 \frac{R}{y_R} \right), & \text{for } \frac{R}{y_R} \leq 2 \\ c_f &= 1.5, & \text{for } \frac{R}{y_R} < 0.5 \end{aligned} \quad (1.25)$$

Darcy-Weissbach:

$$c_f = \sqrt{\frac{8}{f}} \quad \text{with} \quad f = \frac{0.24}{\log \left( \frac{12R}{k_s} \right)} \quad (1.26)$$

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  $d_{90}$ :

$$k_{str} = \frac{\text{factor}}{\sqrt[6]{d_{90}}} \quad \text{default value of factor} = 21.1 \quad (1.27)$$

or

$$k_s = \text{factor} \cdot d_{90} \quad \text{default value of factor} = 3 \quad (1.28)$$

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

$$c = \sqrt{gh} \quad (1.29)$$

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:

$$C_- = \frac{dx}{dt} = u - c \quad (1.30)$$

$$C_+ = \frac{dx}{dt} = u + c \quad (1.31)$$

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):

$$w_s = \frac{\partial z_S}{\partial t} + u_s \frac{\partial z_S}{\partial x} + v_s \frac{\partial z_S}{\partial y} \quad (1.32)$$

Dynamic boundary condition:

$$\tau = (\tau_{Sx}, \tau_{Sy}) \quad \text{and} \quad P = P_{atm} \quad (1.33)$$

2. At the bottom of water body:

Kinematic boundary condition (this condition describes that no flow through the bed surface can take place):

$$w_B = u_B \frac{\partial z_B}{\partial x} + v_B \frac{\partial z_B}{\partial y} \quad (1.34)$$

### 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

$$\int_{z_B(x,y)}^{z_S(x,y)} \frac{\partial f(x,y,z)}{\partial x} dz = \frac{\partial}{\partial x} \int_{z_B(x,y)}^{z_S(x,y)} f(x,y,z) dz + f(x,y,z_B) \frac{\partial z_B}{\partial x} - f(x,y,z_S) \frac{\partial z_S}{\partial x}$$

The 3-D Reynolds-averaged mass conservation equation is integrated over the flow depth from the bed bottom  $z_B$  to the water surface  $z_S$ .

$$\int_{z_B}^{z_S} \left[ \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} \right] dz = 0 \quad (1.35)$$

Applying Leibniz's rule on the first term, the velocity gradient in x-direction, leads to

$$\int_{z_B}^{z_S} \frac{\partial u}{\partial x} dz = \frac{\partial}{\partial x} \int_{z_B}^{z_S} u dz + u_B \frac{\partial z_B}{\partial x} - v_s \frac{\partial z_S}{\partial x}$$

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 analogously.

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:

$$\int_{z_B}^{z_S} \frac{\partial w}{\partial z} dz = w_S - w_B$$

Assembling these terms, one can identify and eliminate the kinematic boundary conditions as stated above.

$$\frac{\partial}{\partial x} \int_{z_B}^{z_S} u dz + \frac{\partial}{\partial y} \int_{z_B}^{z_S} v dz + \underbrace{u_B \frac{\partial z_B}{\partial x} + v_B \frac{\partial z_B}{\partial y} - w_B}_{=0} - \underbrace{u_S \frac{\partial z_S}{\partial x} - v_S \frac{\partial z_S}{\partial y} + w_S}_{= \frac{\partial z_S}{\partial t} = \frac{\partial h}{\partial t}} = 0$$

Evaluating the integrals over the depth as

$$\int_{z_B}^{z_S} u \, dz = \bar{u}h$$

finally leads to the depth integrated formulation of mass conservation:

$$\frac{\partial h}{\partial t} + \frac{\partial(\bar{u}h)}{\partial x} + \frac{\partial(\bar{v}h)}{\partial y} = 0$$

### Derivation of momentum conservation

The Reynolds-averaged 3-D momentum equation in x-direction of the Navier-Stokes equation is integrated over the depth

$$\int_{z_B}^{z_S} \left[ \frac{\partial u}{\partial t} + \frac{\partial u^2}{\partial x} + \frac{\partial uv}{\partial y} + \frac{\partial uw}{\partial z} \right] dz = \int_{z_B}^{z_S} \left[ -\frac{1}{\rho} \frac{\partial p}{\partial x} + \frac{1}{\rho} \frac{\partial \tau_{xx}}{\partial x} + \frac{1}{\rho} \frac{\partial \tau_{xy}}{\partial y} + \frac{1}{\rho} \frac{\partial \tau_{xz}}{\partial z} \right] dz \quad (1.36)$$

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.

$$\int_{z_B}^{z_S} \frac{\partial u}{\partial t} dz = \frac{\partial}{\partial t} \int_{z_B}^{z_S} u dz + u_B \frac{\partial z_B}{\partial t} - u_S \frac{\partial z_S}{\partial t} = \frac{\partial \bar{u}h}{\partial t} + u_B \frac{\partial z_B}{\partial t} - u_S \frac{\partial z_S}{\partial t}$$

The Leibniz rule is also applied on the advective terms as follows:

$$\int_{z_B}^{z_S} \frac{\partial u^2}{\partial x} dz = \frac{\partial}{\partial x} \int_{z_B}^{z_S} u^2 dz + u_B^2 \frac{\partial z_B}{\partial x} - u_S^2 \frac{\partial z_S}{\partial x} \quad , \quad \int_{z_B}^{z_S} \frac{\partial uv}{\partial y} dz = \frac{\partial}{\partial y} \int_{z_B}^{z_S} uv dz + u_B v_B \frac{\partial z_B}{\partial y} - u_S v_S \frac{\partial z_S}{\partial y}$$

And the fundamental theorem of calculus allows the evaluation of the fourth term on the left hand side.

$$\int_{z_B}^{z_S} \frac{\partial uw}{\partial z} dz = u_S w_S - u_B w_B$$

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.

$$\bar{u}h + \frac{\partial}{\partial x} \int_{z_B}^{z_S} u^2 dz + \frac{\partial}{\partial y} \int_{z_B}^{z_S} uv dz + u_B \underbrace{\left( \frac{\partial z_B}{\partial t} + u_B \frac{\partial z_B}{\partial x} + v_B \frac{\partial z_B}{\partial x} - w_B \right)}_{=0} -$$

$$u_S \underbrace{\left( \frac{\partial z_S}{\partial t} + u_S \frac{\partial z_S}{\partial x} + v_S \frac{\partial z_S}{\partial y} + w_S \right)}_{=0}$$

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.

$$\frac{1}{\rho} \int_{z_B}^{z_S} \frac{\partial p}{\partial x} dz = gh \frac{\partial z_S}{\partial x} = gh \left( \frac{\partial z_B}{\partial x} + \frac{\partial h}{\partial x} \right)$$

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.

$$\frac{1}{\rho} \int_{z_B}^{z_S} \frac{\partial \tau_{xx}}{\partial x} dz + \frac{1}{\rho} \int_{z_B}^{z_S} \frac{\partial \tau_{yx}}{\partial y} dz = \frac{1}{\rho} \frac{\partial \bar{\tau}_{xx} h}{\partial x} + \frac{1}{\rho} \frac{\partial \bar{\tau}_{xy} h}{\partial y}$$

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  $\tau_{Bx}$ , e.g. due to wind flow over the water surface, are neglected from here on.

$$\frac{1}{\rho} \int_{z_B}^{z_S} \frac{\partial \tau_{zx}}{\partial z} dz = \frac{1}{\rho} (\tau_{Sx} - \tau_{Bx})$$

Putting the terms together one obtains

$$\frac{\partial \bar{u}h}{\partial t} + \frac{\partial}{\partial x} \int_{z_B}^{z_S} u^2 dz + \frac{\partial}{\partial y} \int_{z_B}^{z_S} uv dz + gh \frac{\partial h}{\partial x} = -gh \frac{\partial z_B}{\partial x} - \frac{1}{\rho} \tau_{Bx} + \frac{1}{\rho} \frac{\partial \bar{\tau}_{xx} h}{\partial x} + \frac{1}{\rho} \frac{\partial \bar{\tau}_{xy} h}{\partial y}$$

The depth integrals of the advective terms still need to be solved. By dividing the velocities in a mean velocity  $\bar{u}$  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.

$$u = \bar{u} + u' \Rightarrow \frac{\partial}{\partial x} \int_{z_B}^{z_S} u^2 dz = \frac{\partial \bar{u}^2 h}{\partial x} + \frac{\partial \overline{u' u' h}}{\partial x} = \frac{\partial \bar{u}^2 h}{\partial x} + \frac{1}{\rho} \frac{\partial D_{xx} h}{\partial x}$$

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:

$$\frac{\partial \bar{u}}{\partial t} + \bar{u} \frac{\partial \bar{u}}{\partial x} + \bar{v} \frac{\partial \bar{u}}{\partial y} + g \frac{\partial h}{\partial x} = -g \frac{\partial z_B}{\partial x} - \frac{1}{\rho h} \tau_{Bx} + \frac{1}{\rho h} \frac{\partial [h(\bar{\tau}_{xx} + D_{xx})]}{\partial x} + \frac{1}{\rho h} \frac{\partial [h(\bar{\tau}_{xy} + D_{yx})]}{\partial y}$$

### Shallow Water Equations

Conclusive, as shown before, the complete set of SWE is derived in the form:

$$\frac{\partial h}{\partial t} + \frac{\partial(\bar{u}h)}{\partial x} + \frac{\partial(\bar{v}h)}{\partial y} = 0 \quad (1.37)$$

$$\frac{\partial \bar{u}}{\partial t} + \bar{u} \frac{\partial \bar{u}}{\partial x} + \bar{v} \frac{\partial \bar{u}}{\partial y} + g \frac{\partial h}{\partial x} = -g \frac{\partial z_B}{\partial x} - \frac{1}{\rho h} \tau_{Bx} + \frac{1}{\rho h} \frac{\partial [h(\bar{\tau}_{xx} + D_{xx})]}{\partial x} + \frac{1}{\rho h} \frac{\partial [h(\bar{\tau}_{xy} + D_{yx})]}{\partial y} \quad (1.38)$$

$$\frac{\partial \bar{v}}{\partial t} + \bar{u} \frac{\partial \bar{v}}{\partial x} + \bar{v} \frac{\partial \bar{v}}{\partial y} + g \frac{\partial h}{\partial y} = -g \frac{\partial z_B}{\partial y} - \frac{1}{\rho h} \tau_{By} + \frac{1}{\rho h} \frac{\partial [h(\bar{\tau}_{yx} + D_{yx})]}{\partial x} + \frac{1}{\rho h} \frac{\partial [h(\bar{\tau}_{yy} + D_{yy})]}{\partial y} \quad (1.39)$$

where:

$h$	$[m]$	water depth
$g$	$[m/s^2]$	gravity acceleration
$P$	$[Pa]$	pressure
$\bar{u}$	$[m/s]$	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)
$\bar{v}$	$[m/s]$	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
$\tau_{Sx}, \tau_{Sy}$	$[N/m^2]$	surface shear stress in x- and y direction (here neglected)
$\tau_{Bx}, \tau_{By}$	$[N/m^2]$	bed shear stress in x- and y direction
$\bar{\tau}_{xx}, \bar{\tau}_{xy}, \bar{\tau}_{yx}, \bar{\tau}_{yy}$	$[N/m^2]$	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

$$\tau_{xx} = 2\rho v \frac{\partial u}{\partial x} \quad , \quad \tau_{yy} = 2\rho v \frac{\partial v}{\partial y} \quad , \quad \tau_{xy} = \rho v \left( \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) \quad (1.40)$$

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):  $\nu = \nu_t + \nu_m$ .

Turbulent eddy viscosity may be dynamically calculated as  $\nu_t = \kappa u_* h / 6$  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

$$\tau_{Bx} = \rho \frac{|\mathbf{u}|u}{c_f^2} \quad ; \quad \tau_{By} = \rho \frac{|\mathbf{u}|v}{c_f^2} \quad (1.41)$$

in which  $|\mathbf{u}| = \sqrt{u^2 + v^2}$  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:

$$\mathbf{U}_t + \nabla(\mathbf{F}, \mathbf{G}) + \mathbf{S} = 0 \quad (1.42)$$

where  $\mathbf{U}, \mathbf{F}(\mathbf{U}), \mathbf{G}(\mathbf{U})$  and  $\mathbf{S}$  are the vectors of conserved variables, fluxes in the  $x$  and  $y$  directions and sources respectively, given by:

$$\mathbf{U} = \begin{pmatrix} h \\ uh \\ \nu h \end{pmatrix} \quad (1.43)$$

$$\mathbf{F} = \begin{pmatrix} uh \\ u^2h + \frac{1}{2}gh^2 - \nu h \frac{\partial u}{\partial x} \\ u\nu h - \nu h \frac{\partial u}{\partial y} \end{pmatrix} ; \quad \mathbf{G} = \begin{pmatrix} \nu h \\ u\nu h - \nu h \frac{\partial \nu}{\partial x} \\ \nu^2h + \frac{1}{2}gh^2 - \nu h \frac{\partial \nu}{\partial y} \end{pmatrix} \quad (1.44)$$

$$\mathbf{S} = \begin{pmatrix} 0 \\ gh(S_{fx} - S_{Bx}) \\ gh(S_{fy} - S_{By}) \end{pmatrix} \quad (1.45)$$

where  $\nu$  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:

$$S_f = \frac{u|u|}{gc_f^2R} \quad (1.46)$$

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

$$S_{fx} = \frac{u\sqrt{u^2 + v^2}}{gc_f^2R} ; \quad S_{fy} = \frac{v\sqrt{u^2 + v^2}}{gc_f^2R} \quad (1.47)$$

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) to be determined. Bed slope terms represent the gravity forces

$$S_{B,x} = -\partial z_B / \partial x \quad ; \quad S_{B,y} = -\partial z_B / \partial y \quad (1.48)$$

The viscous fluxes are treated as source terms (The superscript ‘d’ refers to the diffusion)

$$\mathbf{S}_d = \frac{\partial \mathbf{F}^d}{\partial x} + \frac{\partial \mathbf{G}^d}{\partial y}$$

where the diffusive fluxes read

$$\mathbf{F}^d = \begin{pmatrix} 0 \\ \nu h \frac{\partial u}{\partial x} \\ \nu h \frac{\partial \nu}{\partial x} \end{pmatrix}$$

$$\mathbf{G}^d = \begin{pmatrix} 0 \\ \nu h \frac{\partial u}{\partial y} \\ \nu h \frac{\partial \nu}{\partial y} \end{pmatrix}$$

The total viscosity  $\nu$  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.

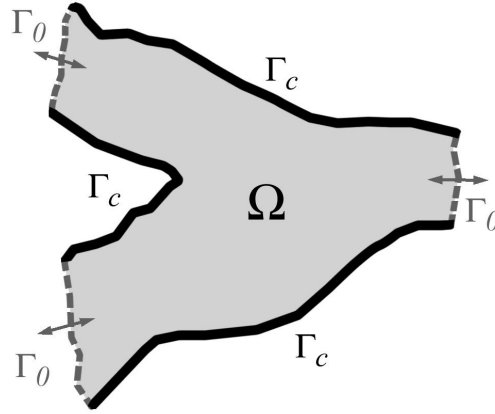
The turbulent eddy viscosity can either be set to a constant value or calculated dynamically as Turbulent eddy viscosity may be dynamically calculated as  $\nu_t = \kappa u_* h / 6$  with the friction velocity  $u_* = \sqrt{\tau_B / \rho}$ .

### 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* ( $\Gamma_c$ ), the other *open* ( $\Gamma_o$ ) (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  $\Gamma_c$  :

$$\rho \mathbf{u} \cdot \mathbf{n} = 0 \quad \frac{\partial \mathbf{u}}{\partial \mathbf{n}} = 0 \quad (1.49)$$

where

- $\mathbf{n}$  [m] the normal (directed outward) unit vector on  $\Gamma_c$
- $\mathbf{u}$  [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  $Fr \leq 1.0$  and hyperbolic for  $Fr \geq 1.0$  , where  $Fr$  is Froude number.

On the open boundary ( $\Gamma_o$ ) the two types inflow and outflow can be respectively distinguished as follows:

$$\Gamma_{in} = (x \in \Gamma_o; \mathbf{u} \cdot \mathbf{n} < 0) \quad (1.50)$$

$$\Gamma_{out} = (x \in \Gamma_o; \mathbf{u} \cdot \mathbf{n} > 0) \quad (1.51)$$

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  $\theta_{cr}$  is a function of the shear Reynolds number  $Re^*$  (Shields, 1936). The critical Shields parameter  $\theta_{cr}$  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,  $\theta_{cr}$  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):

$$\begin{aligned} \theta_{cr} &= 0.24(D^*)^{-1} & \text{for } 1 \leq D^* \leq 4 \\ \theta_{cr} &= 0.14(D^*)^{-0.64} & \text{for } 4 < D^* \leq 10 \\ \theta_{cr} &= 0.04(D^*)^{-0.1} & \text{for } 10 < D^* \leq 20 \\ \theta_{cr} &= 0.013(D^*)^{0.29} & \text{for } 20 < D^* \leq 150 \\ \theta_{cr} &= 0.055 & \text{for } D^* > 150 \end{aligned} \quad (1.52)$$

where the dimensionless grain diameter  $D^*$  is defined as

$$D^* = \left( \frac{(\rho_s - \rho)g}{\rho v^2} \right)^{1/3} d \quad (1.53)$$

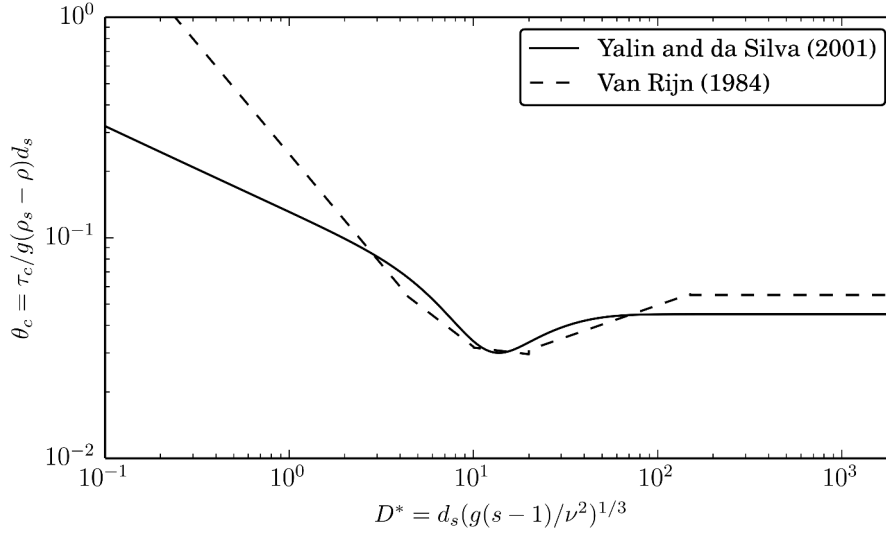
Another explicit formulation of the Shields curve was proposed by Yalin and Silva (2001):

$$\theta_{cr} = 0.13D^{*-0.392} \exp(-0.015D^*) + 0.045(1 - \exp(-0.068D^*)) \quad (1.54)$$

##### 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  $k_\beta$  and  $k_\gamma$  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

$$\tau_{B_{cr}} = k_\beta k_\gamma \tau_{B_{cr},Shields} \quad (1.55)$$



**Figure 1.3** Transformed Shields diagram to determine critical Shields parameter

The correction factors are calculated as suggested by van Rijn (1989):

$$k_{\beta} = \begin{cases} \frac{\sin(\gamma - \beta)}{\sin\gamma} & \text{if slope} < 0 \\ \frac{\sin(\gamma + \beta)}{\sin\gamma} & \text{if slope} > 0 \end{cases} \quad (1.56)$$

$$k_{\gamma} = \begin{cases} \cos\delta \sqrt{1 - \frac{\tan^2\delta}{\tan^2\gamma}} \end{cases}$$

where  $\beta$  is the angle between the horizontal and the bed in flow direction,  $\delta$  is the slope angle transversal to the flow direction and  $\gamma$  is the angle of repose of the sediment material.

Furthermore, Chen et al. (2010) proposed a correction  $k = k_{\beta}k_{\gamma}$  as follows

$$k = \frac{\tau_{B_{cr}}}{\tau_{B_{cr, Shields}}} = \frac{1}{\tan\gamma} \left( \cos^2\left(\frac{\pi}{2} - \beta\right) - 1 + \frac{1}{(1 + \tan^2\beta + \tan^2\gamma)} + \frac{\tan^2\gamma}{(1 + \tan^2\beta + \tan^2\gamma)} \right)^{0.5} + \cos\left(\frac{\pi}{2} - \beta\right) \quad (1.57)$$

### 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  $\theta$  determined from

the present flow conditions can differ from the effective dimensionless bottom shear stress  $\theta'$ , 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  $\mu$  is introduced for the determination of the effective bottom shear stress  $\theta'$  from the bottom shear stress  $\theta$  as

$$\mu = \frac{\tau'_B}{\tau_B} = \frac{\theta'}{\theta} \quad \text{with} \quad \begin{cases} \mu = 1.0 & \text{(no bed forms)} \\ \mu < 1.0 & \text{(bed forms)} \end{cases} \quad (1.58)$$

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  $\mu$ .

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.

$$\mu = \left( \frac{k_{str}}{k'_{str}} \right)^{3/2} \quad (1.59)$$

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  $\theta_{cr}$  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).

$$w = \frac{(s-1)gd^2}{18\nu} \quad \text{for } 0.001 < d \leq 0.1 \quad \text{mm} \quad (1.60)$$

$$w = \frac{10\nu}{d} \left( \sqrt{1 + \frac{0.01(s-1)gd^3}{\nu^3}} - 1 \right) \quad \text{for } 0.1 < d \leq 1 \quad \text{mm} \quad (1.61)$$

$$w = 1.1\sqrt{(s-1)gd} \quad \text{for } d \geq 1 \quad \text{mm} \quad (1.62)$$

$d$  is the diameter of the grain,  $\nu$  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):

$$w = \frac{M\nu}{Nd} \left[ \sqrt{\frac{1}{4} + \left( \frac{4N}{3M^2} (D^*)^3 \right)^{1/n}} - \frac{1}{2} \right]^n \quad (1.63)$$

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).

$$w = \sqrt{\left(13.95 \frac{\nu}{d}\right)^2 + 1.09(s-1)gd} - 13.95 \frac{\nu}{d} \quad (1.64)$$

### 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

$$u_B = 7.5(\sqrt{\theta'} - C_0\sqrt{\theta_{cr}})\sqrt{(s-1)gd} \quad (1.65)$$

where  $\theta'$  is the dimensionless effective bottom shear stress and  $\theta_{cr}$  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 critical 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:

$$\frac{\partial(AC_g)}{\partial t} + \frac{\partial(QC_g)}{\partial x} - \frac{\partial}{\partial x} \left( A\Gamma \frac{\partial C_g}{\partial x} \right) - S_g - Sl_g = 0 \quad \text{for } g = 1, \dots, ng \quad (1.66)$$

Introducing the continuity equation  $\frac{\partial A}{\partial t} + \frac{\partial Q}{\partial x} = 0$  in eq. 1.66 one becomes:

$$A \frac{\partial(C_g)}{\partial t} + Q \frac{\partial(C_g)}{\partial x} - \frac{\partial}{\partial x} \left( A\Gamma \frac{\partial C_g}{\partial x} \right) - S_g - Sl_g = 0 \quad \text{for } g = 1, \dots, ng \quad (1.67)$$

### 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:

$$\frac{\partial}{\partial t} C_g h + \frac{\partial}{\partial x} \left( C_g q - h\Gamma \frac{\partial C_g}{\partial x} \right) - \frac{\partial}{\partial y} \left( C_g r - h\Gamma \frac{\partial C_g}{\partial y} \right) - S_g - Sl_g = 0 \quad \text{for } g = 1, \dots, ng \quad (1.68)$$

where  $C_g$  is the concentration of each grain size class and  $\Gamma$  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  $q_{e_k}$ .

$$S_g = q_{e_g} - q_{d_g} \quad (1.69)$$

The deposition rate is expressed as convection flux of the sink rate:

$$q_{d_g} = w_g C_{d_g} \quad (1.70)$$

$w_g$  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:

$$q_{e_g} = w_g \beta_g C_{e_g} \quad (1.71)$$

The outcome of this is:

$$S_g = w_g (\beta_g C_{e_g} - C_{d_g}) \quad (1.72)$$

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):

$$C_{e_g} = 0.015 \frac{d_g}{a} \frac{T_g^{1.5}}{(D^*)_g^{0.3}} \quad (1.73)$$

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):

$$C_{e_g} = \frac{0.331(\theta' - 0.045)^{1.75}}{1 + 0.72(\theta' - 0.045)^{1.75}} \quad (1.74)$$

with the dimensionless effective bottom shear stress

$$\theta' = u_*^2 / [(s - 1)gd] \quad (1.75)$$

The reference concentration for the deposition rate is calculated after Lin (1984):

$$C_{d_g} = \left( 3.25 + 0.55 \ln \left( \frac{w_g}{\kappa u_*} \right) \right) C_g \quad (1.76)$$

where:

$C_g$  is the mean concentration over depth of suspended particles of grain class  $g$ ,

$u_*$  is the shear velocity,

$\kappa$  the Von Karmann constant.

Another approach is the one of Minh Duc (1998):

$$\alpha_c = \frac{(h - \delta)}{\int_{\delta}^h \left( \frac{h - z}{z} \frac{\delta}{h - \delta} \right)^{\omega/\kappa U_*} dz}$$

The erosion and deposition rates can also be computed using critical shear stresses, as does Xu (1998).

$$q_d = \begin{cases} \omega_s C_a \left( 1 - \frac{\tau}{\tau_{c,S}} \right) & \tau < \tau_{c,S} \\ 0 & \tau \geq \tau_{c,S} \end{cases}$$

$$\tau_{c,S} = \frac{\rho_s - \rho}{\rho_s} \frac{gh\omega_s C_k}{T_k U}$$

$T_k$  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  $C_k$ .

$$q_e = \begin{cases} M \left( \frac{\tau}{\tau_{c,E}} - 1 \right)^n & \tau > \tau_{c,E} \\ 0 & \tau \leq \tau_{c,E} \end{cases}$$

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:

$$q_{B_g} = \beta_g q_B(\xi_g) \quad (1.77)$$

Details about the transport laws for evaluation of  $q_B$  with or without the consideration of the hiding factor  $\xi_g$  can be found in Section 1.2.3.4.

### 1.2.3.1.3 Gravitational Slope Collapse in Longitudinal Direction

Gravitational induced failures in longitudinal direction are significant aspects concerning erosion and transport modelling. This process may play an important role e.g. for delta formation in estuaries. 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 between two cross-sections is flattened if its angle  $\alpha$  becomes steeper than a given critical slope angle  $\gamma_{crit}$ .

$$q_{B_g, xgrav} = \begin{cases} 0 & \text{if } (\alpha \leq \gamma_{crit}) \\ f(\alpha, \gamma_{crit}) & \text{if } (\alpha > \gamma_{crit}) \end{cases} \quad (1.78)$$

The sliding material is moved from the cross-section with higher elevation to the lower situated cross-section. Only one critical slope angle can be defined in this one-dimensional approach.

### 1.2.3.1.4 Bed Material Sorting

For each fraction  $g$  a mass conservation equation can be written, the so called “bed-material sorting equation”:

$$(1 - p) \frac{\partial}{\partial t} (\beta_g \cdot h_m) + \frac{\partial q_{B_g}}{\partial x} + s_g - sf_g - sl_{B_g} = 0 \quad \text{for } g = 1, \dots, n_g \quad (1.79)$$

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).

### 1.2.3.1.5 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:

$$(1-p)\frac{\partial z_B}{\partial t} + \sum_{g=1}^{ng} \left( \frac{\partial q_{B_g}}{\partial x} + s_g - sl_{B_g} \right) = 0 \quad (1.80)$$

### 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):

$$q_{B_g,x} = q_{B_g,xx} + q_{B_g,xlateral} + q_{B_g,xgrav} \quad (1.81)$$

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

$$q_{B_g,xx} = \beta_g q_B(\xi_g) \cdot e_x \quad (1.82)$$

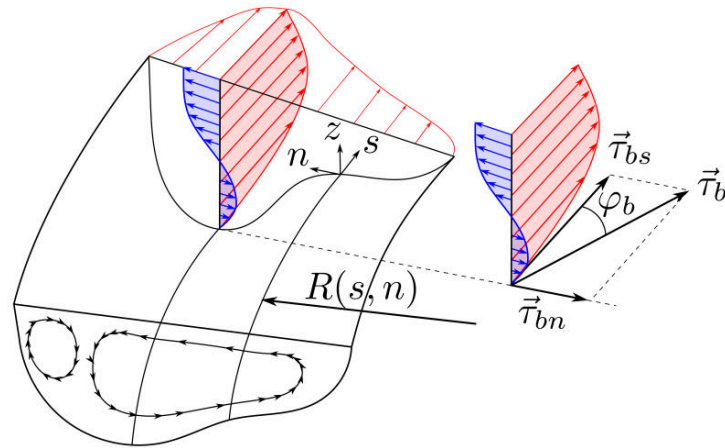
The specific bed load discharge  $q_{B_g}$  of the  $g^{th}$  grain class has to be evaluated by a suitable transport law. Approaches for the bed load discharge  $q_{B_g}(\xi_g)$  with or without the consideration of a hiding factor  $\xi_g$  are discussed in Section 1.2.3.4.

#### 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)):

$$\tan \varphi_b = -N_l \left( \frac{\tau'_{B_{cr,g}}}{\tau'_B} \right)^\gamma \vec{s} \cdot \vec{n}_q \quad (1.83)$$

where:  $\varphi_b$  = bed load direction with respect to the flow vector  $\vec{q}$ ,  $N_l$  = lateral transport factor ( $1.2 \leq N_l \leq 2.4$ ),  $\gamma$  = 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 ( $\vec{s} \cdot \vec{n}_q < 0$ ),  $\tau'_B$  = dimensionless bed shear stress, and  $\tau'_{B_{cr,g}}$  = dimensionless critical shear stress of the individual grain class.



**Figure 1.4** Effect of spiral motion in river bend on bed shear stress  $\tau_b$  with deviation angle from main flow direction  $\varphi_b$ , adapted from Blanckaert (2011)

#### 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  $\varphi_b$  (positive counterclockwise and vice versa) from the main flow direction is determined as

$$\tan \varphi_b = N_* \frac{h}{R} \quad (1.84)$$

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).

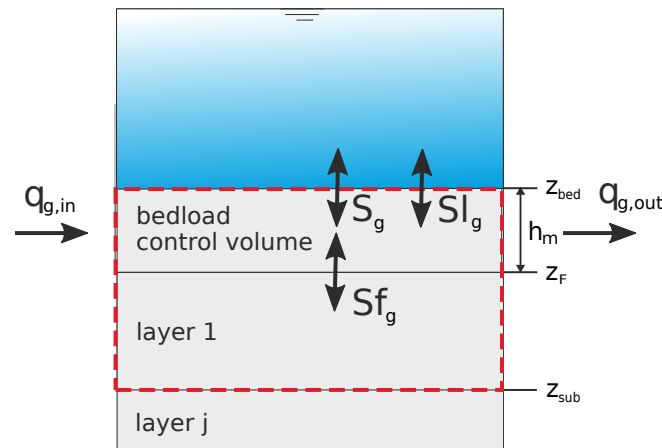
Note that the curvature factor  $N_*$  mainly depends on bed roughness. Therefore,  $N_* \approx 7$  for natural streams (Engelund, 1974), and values up  $N_* \approx 11$  for laboratory channels (Rozovskii, 1957).

#### 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 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  $\alpha$  becomes steeper than a given critical slope angle  $\gamma_{crit}$ .



**Figure 1.5** Definition sketch of overall control volume (red) of bed material sorting equation

$$q_{B_g, xgrav} = \begin{cases} 0 & \text{if } (\alpha \leq \gamma_{crit}) \\ f(\alpha, \gamma_{crit}) & \text{if } (\alpha > \gamma_{crit}) \end{cases} \quad (1.85)$$

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

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. Furthermore 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  $\beta_g$  at the new time level

$$\begin{aligned} \frac{\partial V}{\partial t} &= (1-p) \left[ \underbrace{\frac{\partial(\beta_g \cdot h_m)}{\partial t}}_{\text{change of grain volume in bedload control volume}} + \underbrace{\frac{\partial(\beta_{sub_g} \cdot (z_F - z_{sub}))}{\partial t}}_{\text{change of grain volume in layer } 1=sf_g} \right] \\ &= \underbrace{-\frac{\partial q_{B_g,x}}{\partial x} - \frac{\partial q_{B_g,y}}{\partial y}}_{\text{fluxes over boundary}} \underbrace{-s_g + sl_{B_g}}_{\text{source terms}} \end{aligned}$$

Rearranging this sorting equation leads to following formulation which is used from here on

$$(1-p) \frac{\partial}{\partial t} (\beta_g \cdot h_m) + \frac{\partial q_{B_g,x}}{\partial x} + \frac{\partial q_{B_g,y}}{\partial y} + s_g - sf_g - sl_{B_g} = 0 \quad \text{for } g = 1, \dots, ng \quad (1.86)$$

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  $z_B$ :

$$(1-p) \frac{\partial z_B}{\partial t} + \sum_{g=1}^{ng} \left( \frac{\partial q_{B_g,x}}{\partial x} + \frac{\partial q_{B_g,y}}{\partial y} + s_g - sl_{B_g} \right) = 0 \quad (1.87)$$

### 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:

$$sf_g = -(1-p) \frac{\partial}{\partial t} ((z_F - z_{sub}) \beta_{sub_g}) \quad (1.88)$$

### 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:

$$q_{B_g} = \alpha \sqrt{(s-1)gd_g^3} (\theta_g - \theta_{cr,g})^m \quad (1.89)$$

Herein,  $\alpha$  denotes the bed load factor (originally  $\alpha = 8$ ),  $m$  the bed load exponent (originally  $m = 1.5$ ),  $q_{B_g}$  is the specific bed load transport rate of grain class  $g$ ,  $\theta_g$  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  $\alpha$  to 4.93 and  $m$  to 1.6, the bed load formula can be adapted according to Wong and Parker (2006).

Meyer-Peter and Müller observed in their experiments that the first 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 were 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  $\xi_g$  proposed by Ashida and Michiue (1971):

$$\xi_g = \begin{cases} [\log(19)/\log(19d_g/d_m)]^2 & d_g/d_m \geq 0.4 \\ 0.843d_m/d_g & d_g/d_m < 0.4 \end{cases} \quad (1.90)$$

$d_g$  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:

$$\theta_{cr,g} = \theta_{cr,ref} \xi_g, \quad (1.91)$$

where  $\theta_{cr,ref}$  usually is assigned to a fix value (e.g.  $\theta_{cr,ref} = 0.047$ ) or the critical Shields parameter 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

$$q_{B_g} = 17 \sqrt{(s-1)gd_g^3} (\theta_g - \xi_g \theta_{cr,ref}) (\sqrt{\theta_g} - \sqrt{\xi_g \theta_{cr,ref}}) \quad (1.92)$$

where  $q_{B_g}$  is the specific bed load transport rate of grain class  $g$ ,  $\theta_g$  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  $\xi_g$  is the hiding function according to eq. 1.90.

#### 1.2.3.4.3 Parker

Parker extended his empirical substrate-based bed load relation for gravel mixtures 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.

$$W_{si}^* = 0.00218 G[\xi_s \omega \phi_{sg0}] \quad ; \quad W_{si}^* = \frac{Rgq_{bi}}{(\tau_B/\rho)^{3/2} F_i} \quad (1.93)$$

where:

$$\xi_s = \left( \frac{d_i}{d_g} \right)^{-0.0951} \quad ; \quad \phi_{50} = \frac{\tau_{sg}^*}{\tau_{rsg0}^*} \quad ; \quad \tau_{sg}^* = \frac{\tau_B}{\rho Rg d_g} \quad ; \quad \tau_{rsg0}^* = 0.0386$$

$$\omega = 1 + \frac{\sigma}{\sigma_0(\phi_{sg0})} [\omega_0(\phi_{sg0}) - 1] \quad ; \quad \sigma = \sum F_i \left[ \frac{\ln(d_i/d_g)}{\ln(2)} \right]^2 \quad ; \quad d_g = e^{\sum F_i \ln(d_i)}$$

$\xi_s$  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  $d_i$  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  $\tau_{rsg0}^*$  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  $\tau_{rsg0}^*$  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 and Crowe, 2003), similar to Parker's model (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 observed among earlier hiding functions implemented in other transport models, such as the Oak Creek and the Cambridge ones (Parker and Sutherland, 1990). Wilcock and Crowe model (Wilcock and Crowe, 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.

$$W_{si}^* = G(\phi_i) \quad ; \quad W_{si}^* = \frac{Rgq_{bi}}{(\tau_B/\rho)^{3/2}F_i} \quad (1.94)$$

where:

$$G(\phi_i) = \begin{cases} 0.002 \phi_i^{7.5} & \phi_i < 1.35 \\ 14 \left(1 - \frac{0.894}{\phi_i^{0.5}}\right)^{4.5} & \phi_i \geq 1.35 \end{cases}$$

and:

$$\begin{aligned} \phi_i &= \frac{\tau_{sg}^* d_i^{-b}}{\tau_{ssrg}^* d_g} & ; & \quad \tau_{sg}^* = \frac{\tau_B}{\rho Rg d_g} \\ \tau_{ssrg}^* &= 0.021 + 0.015 \exp(-20F_s) & ; & \quad b = \frac{0.67}{1 + \exp(1.5 - \frac{d_i}{d_g})} \end{aligned}$$

The non-linear effect of sand content  $F_s$  on gravel transport is taken into account in  $\tau_{ssrg}^*$ . Wilcock and Crowe (Wilcock and Crowe, 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  $\tau_{ssrg}^*$  (a surrogate for a critical Shields number) and the fraction sand in the active layer  $F_s$ . Note that  $\tau_{ssrg}^*$  decreases as  $F_s$  increases, causing an increase of  $\phi_i$  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:

$$q_{Bg} = 5\beta_g [\xi_g (\theta'_{dms} - \theta_{cdms})]^{3/2} \sqrt{(s-1)gd_{ms}^3} \quad (1.95)$$



where  $\theta'_{dms}$  denotes the Shields parameter of the mean grain size of the surface bed material  $d_{ms}$  according to eq. 1.96,  $\xi_g$  denotes the hiding function applied on the excess shear stress  $(\theta'_{dms} - \theta_{cdms})$ .

$$\theta'_{dms} = \frac{\tau'_b}{\rho_w (s - 1) d_{ms}} \quad (1.96)$$

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  $\theta_{cdms}$  of the mean grain size diameter is determined according to

$$\theta_{cdms} = \theta_{ce} \left( \frac{d_{mo}}{d_{ms}} \right)^{0.33} \quad (1.97)$$

where  $\theta_{ce}$  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

$$\frac{d_{ms}}{d_{mo}} = 0.0163\theta'^{-1.45}_{dms} + 0.6 \quad (1.98)$$

Finally, the hiding function is determined as

$$\xi_g = \left( \frac{d_g}{d_{ms}} \right)^{-\alpha} \quad (1.99)$$

where  $\alpha$  is an empirical parameter depending on the Shields parameter (see also Hunziker and Jaeggi (2002)) according to eq. 1.100, which is limited to a range between  $-0.4$  and  $2.0$ .

$$\alpha = 0.011\theta'^{-1.5}_{dms} - 0.3 \quad (1.100)$$

#### 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.

$$\Phi_B = 3.1 \left( \frac{d_{90}}{d_{30}} \right)^{0.2} \theta'^{0.5} (\theta' - \theta_{cr}) Fr^{1.1} (s - 1)^{-0.5} \quad (1.101)$$

$$q_B = \Phi_B ((s - 1) g d_m^3)^{0.5} \quad (1.102)$$

$\theta'$  is the dimensionless shear stress,  $\theta_{cr}$  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.

$$q_B = \frac{4}{s-1} \left( \frac{d_{90}}{d_{30}} \right)^{0.2} J^{0.6} Ru (J - J_{cr}) \quad (1.103)$$

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

$$J_{cr} = \frac{\theta_{cr}(s-1)d_m}{R} \quad (1.104)$$

where  $\theta_{cr}$  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:

$$\theta_{cr} = \theta_{cr,Ref}(\cos(\arctan J)) \left( 1 - \frac{J}{\tan \psi} \right) \quad (1.105)$$

where  $J$  is the local bed slope,  $\psi$  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:

$$q_{B,g} = \frac{4}{s-1} \left( \frac{d_{90}}{d_{30}} \right)^{0.2} J^{0.6} Ru (J - J_{cr,g}) \quad (1.106)$$

Compared to the original eq. 1.103 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  $1.06 \leq \alpha \leq 1.53$ . 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. 1.107 and eq. 1.108.

$$\zeta_g = \begin{cases} 0.85 \left( \frac{d_g}{d_m} \right)^{-1} & \text{for } \frac{d_g}{d_m} \leq 0.4 \\ \left( \frac{\log(19)}{\log\left(19 \frac{d_g}{d_m}\right)} \right)^2 & \text{for } \frac{d_g}{d_m} > 0.4 \end{cases} \quad (1.107)$$

$$\theta_{cr,g} = \zeta_g \theta_{cr} \quad (1.108)$$

The critical shields parameter  $\theta_{cr}$  is calculated according to eq. 1.105.

#### 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

$$p_{hid_g} = \sum_{i=1}^{ng} \beta_i \frac{d_i}{d_g + d_i}, \quad p_{exp_g} = \sum_{i=1}^{ng} \beta_i \frac{d_g}{d_g + d_i} \quad (1.109)$$

The critical dimensionless shields parameter for each grain class  $g$  can be calculated with the hiding and exposure factor  $\eta_g$  and the shields parameter of the mean grain size  $\theta_{cr_m}$  as

$$\theta_{cr_g} = \theta_{cr_m} \underbrace{\left( \frac{p_{exp_g}}{p_{hid_g}} \right)^m}_{\eta_g} \quad (1.110)$$

The transport capacity now can be determined with Wu's formula in dimensionless form as

$$\Phi_{B_g} = 0.0053 \left[ \frac{\theta'}{\theta_{cr_g}} - 1 \right]^{2.2} \quad (1.111)$$

Finally the bed load transport rates calculates for each grain fraction as

$$q_{b_g} = \beta_g \sqrt{(s-1)gd_g^3} \Phi_{B_g} \quad (1.112)$$

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  $\theta_{cr_g}$  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  $\theta_{cr_g}$  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. 1.113.

$$q_B = 0.053 \sqrt{(s-1)g} \frac{d_{50}^{1.5} T^{2.1}}{D_*^{0.3}} \quad (1.113)$$

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

$$T = (u_*/u_{*cr})^2 - 1 \quad (1.114)$$

where  $u_*$  is the effective bed shear velocity determined as

$$u_* = u\sqrt{g}/C'_h \quad (1.115)$$

with  $C'_h = 18 \log(4h/d_{90})$ .

$u_{*cr}$  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

$$q_B = 0.05 \sqrt{(s-1)g} c_f^2 \theta^{2.5} d_f^{1.5} \quad (1.116)$$

where  $d_f$  denotes the mean fall diameter of the bed material and  $\theta$  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. Therefore, no critical shear stress is taken into account and bed load transport only depends on the flow velocity  $u$ .

$$q_b = au^b \quad (1.117)$$

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:

$$M(x) = M_0 e^{-c(x-x_0)} \quad (1.118)$$

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.

$$w_{S,t+1}^{forced} = z_{B,t}^{forced} + h_t \quad (1.119)$$

### 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

$$z_b^t = z_b^{t-1} + \varepsilon \quad (1.120)$$

where  $z_b^{t-1}$  denotes the bed elevation from the previous time step and  $\varepsilon$  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  $\theta$  [-] 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  $\theta$  as primary variable as

$$\frac{\partial \theta}{\partial t} - \nabla(K \nabla z) - \nabla(D \theta) = Q \quad (1.121)$$

$$D = K \frac{\partial h}{\partial \theta}$$

It is implemented also in a mixed moisture and pressure head formulation for  $\theta$  and  $h$  as

$$\frac{\partial \theta}{\partial t} - \nabla(K \nabla z) - \nabla(D \nabla h) = Q \quad (1.122)$$

$$D = K$$

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  $\Delta x$  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

$$h(\theta) = h_s \theta^{-1/\lambda}$$

$$k_r = (h/h_s)^{-(4\lambda+2)} \quad (1.123)$$

$$\frac{\partial h}{\partial \theta} = -\frac{h_s}{\lambda} \theta^{-1/\lambda-1}$$

with  $\lambda$  [-] being a soil parameter.

For the modified VGM model the functions and derivate are as follows

$$\begin{aligned}
h(\theta) &= \frac{-1}{\alpha} \left[ -\varkappa \left( \frac{\beta}{\theta} \right) \right]^{1/n} \\
k_r &= \sqrt{\theta} \left( \frac{1 - \varkappa^m \theta / \beta}{1 - \varkappa^m 1 / \beta} \right)^2 \\
\frac{\partial h}{\partial \theta} &= \frac{1 - m}{\beta m \alpha} \frac{\theta^{-1/m-1}}{\beta} - \varkappa \left( \frac{\beta}{\theta} \right)^{-m}
\end{aligned} \tag{1.124}$$

with  $\varkappa(x) = 1 - x^{1/m}$  and  $m = 1 - 1/n$ . The empirical constants  $\alpha$  [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  $\theta$  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

$$h(\theta) = (\theta - 1) \left. \frac{\partial h}{\partial \theta} \right|_{\theta=1} + h_S \quad \theta \geq 1 - 0 \tag{1.125}$$

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).

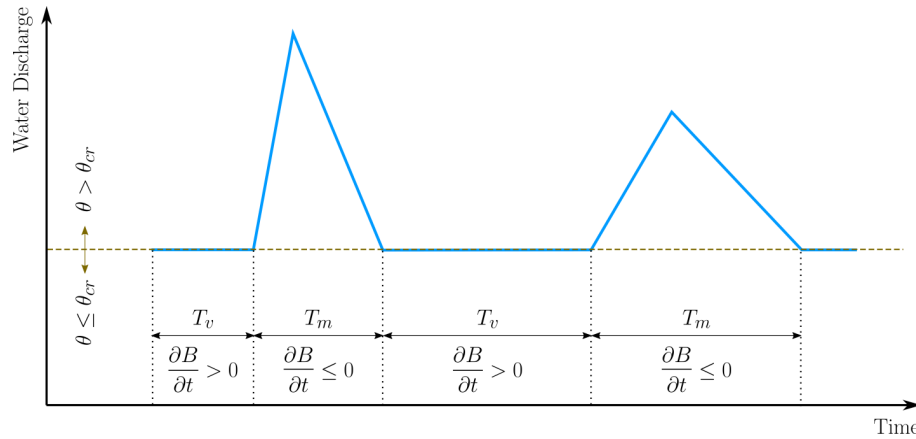
## 1.4 Morphodynamics and Vegetation

### 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





**Figure 1.6** Schematic illustration of the time frame concept used in the vegetation module.

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  $\theta$  reaches its critical value  $\theta_{cr}$ , 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 ( $T_v$ ).

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  $B_{max}$  [ $\text{kg m}^{-2}$ ], representing its maximum carrying capacity. Following Marani et al, we can define an equilibrium biomass  $B_{eq}$ , normalized by its maximum, as

$$B_{eq} = \frac{1}{\exp[\lambda_1(z - z_0)] + \exp[-\lambda_2(z - z_0)]} \quad (1.126)$$

in which  $z$  is the bed elevation, the parameters  $\lambda_1$  and  $\lambda_2$  define the rate at which vegetation fitness diminishes away from its 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  $\lambda_1$ ,  $\lambda_2$  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)

$$\frac{dB}{dt} = \sigma B(t) \left[ 1 - \frac{B(t)}{B_{eq}(z)} \right] \quad (1.127)$$

where  $\sigma$  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 ( $\sigma$ ), without affecting hydro-morphodynamic processes.

From a computational point of view, eq. 1.127 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

$$K_s = K_{s,g} + (K_{s,g} - K_{s,v}) \frac{B}{B_{max}} \quad (1.128)$$

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  $\tau_g$ , 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  $\theta$  (Shield parameter), it can be derived as

$$\frac{\theta}{\theta'} = \left( \frac{K_{s,g}}{K_{s,v}} \right)^2 \quad (1.129)$$

where  $\theta'$  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

$$\theta_{cr} = \theta_{cr,g} + (\theta_{cr,v} - \theta_{cr,g}) \frac{B}{B_{max}} \quad (1.130)$$

in which  $\theta_{cr}$  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  $\Delta z$  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

$$\hat{z}_w(x_j, y_j) = \sum_{i=1}^N \alpha_i z_w(x_i, y_i) \quad (1.131)$$

where the estimated value  $\hat{z}_w(x_j, y_j)$  is a linear combination of the weights  $\alpha_i$  and the known value  $z_w(x_i, y_i)$ .  $\alpha_i$  can be written as

$$\alpha_i = \frac{d_{j,i}^{-\gamma}}{\sum_{i=1}^N d_{j,i}^{-\gamma}} \quad (1.132)$$

with

$$\sum_{i=1}^N \alpha_i = 1 \quad (1.133)$$

in which  $d_{j,i}$  is the distance between  $S_i$  and  $S_j$  and  $\gamma$  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  $\alpha_i$  are given to the points closest to the location  $S_j$ . As such,  $\gamma$  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).



---

# 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

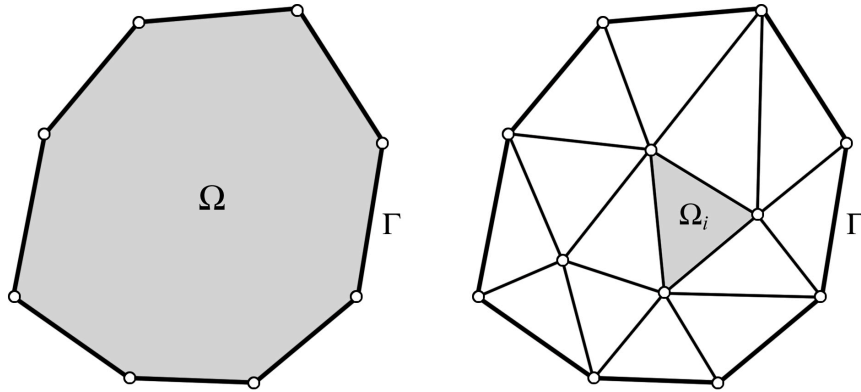
### 2.2.1 Fundamentals

#### 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):

$$\left. \begin{aligned} \mathbf{U}_t + \mathbf{F}_x(\mathbf{U}) &= 0 \\ \mathbf{U}(x, 0) &= \begin{cases} \mathbf{U}_L & \forall x < 0 \\ \mathbf{U}_R & \forall x > 0 \end{cases} \end{aligned} \right\} \quad (2.1)$$

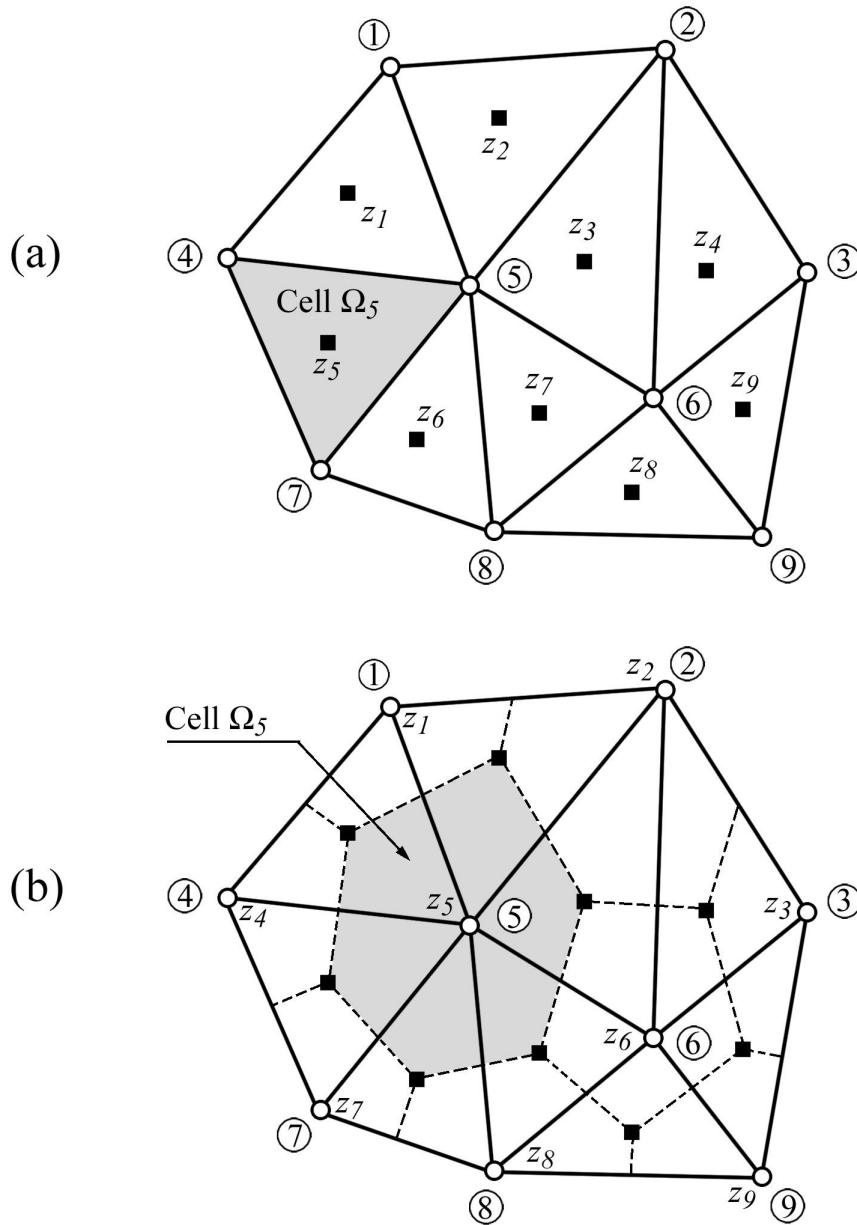
Here eq. 1.36 and eq. 1.37 are considered for the x-split of SWE. The initial states  $\mathbf{U}_L$ ,  $\mathbf{U}_R$  at the left or right side of an triangle edge

$$\mathbf{U}_L = \begin{pmatrix} h_L \\ u_L h_L \\ \nu_L h_L \end{pmatrix}$$

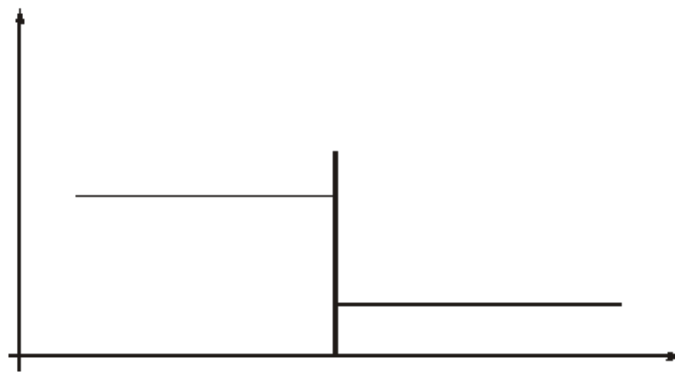
$$\mathbf{U}_R = \begin{pmatrix} h_R \\ u_R h_R \\ \nu_R h_R \end{pmatrix}$$

are constant vectors and present conditions to the left of axes  $x = 0$  and to the right  $x = 0$ , respectively (Figure 2.3).

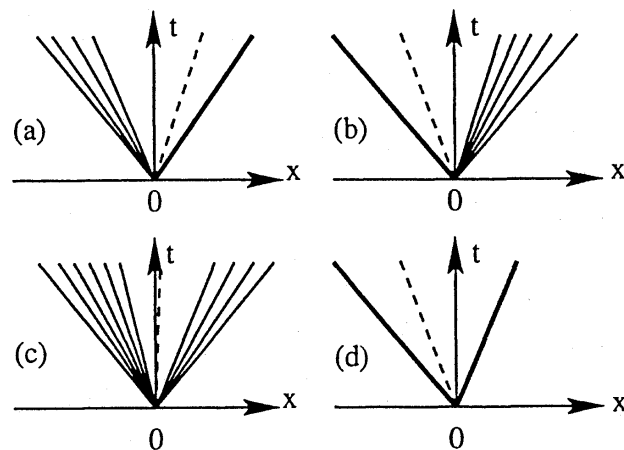
There are four possible wave patterns that may occur in the solution of the Riemann



**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  $\mathbf{J} = \partial \mathbf{F}(\mathbf{U})/\partial \mathbf{U}$  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).

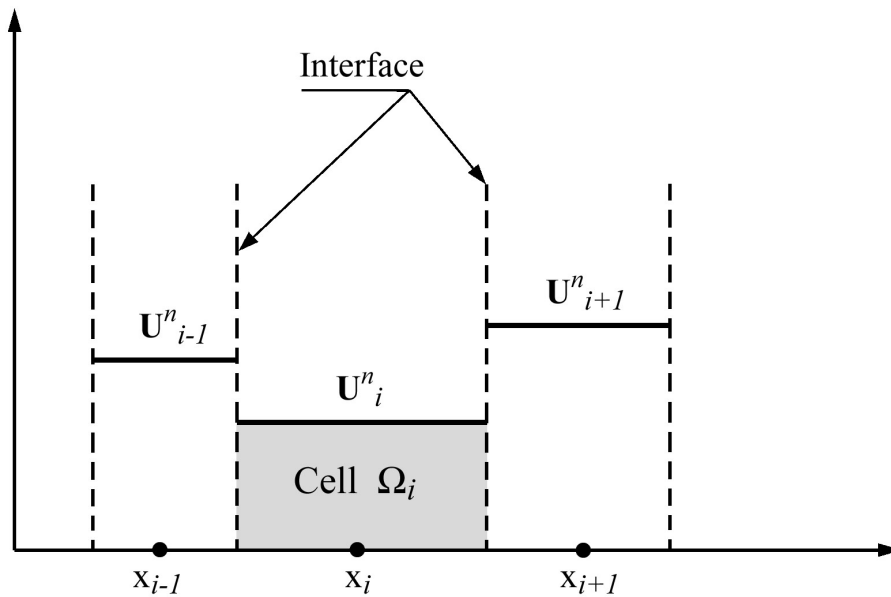
By solving the IVP of Riemann (eq. 2.1), the desired outward flux  $\mathbf{F}(\mathbf{U})$  which is at the origin of local axes  $x = 0$  and the time  $t = 0$  can be obtained.

### 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

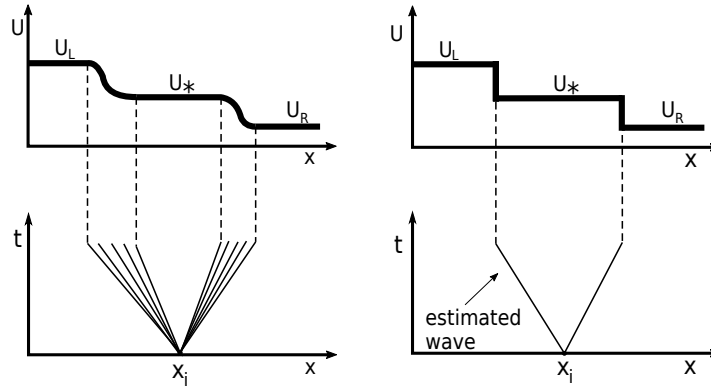
Correctness seems to be the overriding criterion; however the applicability can be also very important. Toro argued that for the shallow water equations, the argument of computational cost is not as strong as for the Euler equations. For the SWE approximate Riemann solvers may lead to savings in computation 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.

$$\mathbf{F}_{i+1/2} = \begin{cases} \mathbf{F}_L & \text{if } 0 \leq S_L, \\ \mathbf{F}^{hll} = \frac{S_R \mathbf{F}_L - S_L \mathbf{F}_R + S_R S_L (\mathbf{U}_R - \mathbf{U}_L)}{S_R - S_L} & \text{if } S_L \leq 0 \leq S_R, \\ \mathbf{F}_R & \text{if } 0 \geq S_R. \end{cases} \quad (2.2)$$

Furthermore, the left and right wave speed velocities are estimated as

$$S_L = u_L - \sqrt{gh_L} q_L; \quad S_R = u_R + \sqrt{gh_R} q_R \quad (2.3)$$

where  $q_K (K = L, R)$  are

$$q_K = \begin{cases} \sqrt{\frac{1}{2} \left[ \frac{(h_* + h_K) h_*}{h_K^2} \right]} & \text{if } h_* > h_K, \\ 1 & \text{if } h_* \leq h_K. \end{cases} \quad (2.4)$$

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

$$h_* = \frac{1}{2}(h_L + h_R) - \frac{1}{4}(u_R - u_L)(h_L - h_R)/(\sqrt{gh_L} + \sqrt{gh_R}) \quad (2.5)$$

In case of dry-bed conditions, the wave speed estimations are modified as follow:

$$S_L = \begin{cases} u_R - 2\sqrt{gh_R} & \text{if } h_L = 0, \\ \text{usualestimate} & \text{if } h_L > 0, \end{cases} \quad (2.6)$$

$$S_R = \begin{cases} u_L + 2\sqrt{gh_L} & \text{if } h_R = 0, \\ \text{usualestimate} & \text{if } h_R > 0. \end{cases}$$

#### 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 then 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

$$\mathbf{F}^{i+1/2} = \begin{cases} \mathbf{F}_L & \text{if } 0 \leq S_L, \\ \mathbf{F}_{*L} = \mathbf{F}_L + S_L(\mathbf{U}_{*L} - \mathbf{U}_L) & \text{if } S_L \leq 0 \leq S_*, \\ \mathbf{F}_{*R} = \mathbf{F}_R + S_R(\mathbf{U}_{*R} - \mathbf{U}_R) & \text{if } S_* \leq 0 \leq S_R, \\ \mathbf{F}_R & \text{if } 0 \geq S_R. \end{cases} \quad (2.7)$$

The states  $\mathbf{U}_{*L}, \mathbf{U}_{*R}$  are obtained, as proposed by Toro (1997), from the relations

$$\mathbf{U}_{*L} = h_L \begin{pmatrix} \frac{S_L - u_L}{S_L - S_*} \\ S_* \\ \nu_L \end{pmatrix} ; \quad \mathbf{U}_{*R} = h_R \begin{pmatrix} \frac{S_R - u_R}{S_R - S_*} \\ S_* \\ \nu_R \end{pmatrix} \quad (2.8)$$

$S_L, S_*$  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

$$S_* = \frac{S_L h_R (u_R - S_R) - S_R h_L (u_L - S_L)}{h_R (u_R - S_R) - h_L (u_L - S_L)} \quad (2.9)$$

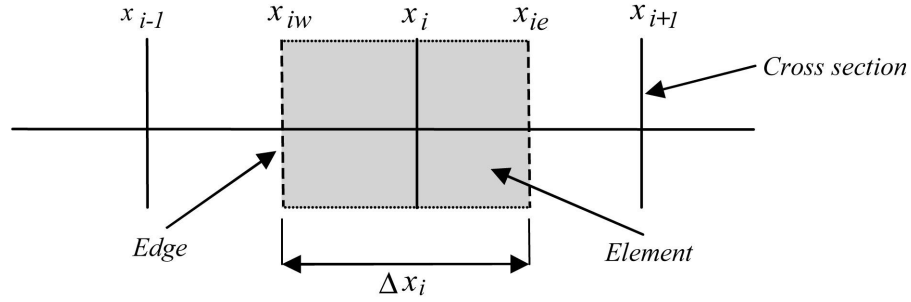


Figure 2.7 Definition sketch

## 2.2.2 Saint-Venant Equations

### 2.2.2.1 Discretisation

The temporal 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:

$$\int_{x_{iL}}^{x_{iR}} f(x) dx \approx f(x_i)(x_{iR} - x_{iL}) = f_i \Delta x_i \quad (2.10)$$

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:

$$\int_{x_{iL}}^{x_{iR}} \left( \frac{\partial A}{\partial t} + \frac{\partial Q}{\partial x} - q_l \right) dx = 0 \quad (2.11)$$

The different parts of the equation are discretized as follows:

$$\int_{x_{iL}}^{x_{iR}} \frac{\partial A_i}{\partial t} dx \approx \frac{\partial A_i}{\partial t} \Delta x_i \approx \frac{A_i^{t+1} - A_i^t}{\Delta t} \Delta x_i \quad (2.12)$$

$$\int_{x_{iL}}^{x_{iR}} \frac{\partial Q_i}{\partial x} dx = Q(x_{iR}) - Q(x_{iL}) = \Phi_{c,iR} - \Phi_{c,iL} \quad (2.13)$$

$$\int_{x_{iL}}^{x_{iR}} q_l dx \approx q_{iR}(x_{iR} - x_i) + q_{iL}(x_i - x_{iL}) \quad (2.14)$$

$\Phi_{c,iR}$  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:

$$A_i^{t+1} = A_i^t - \frac{\Delta t}{\Delta x_i} (\Phi_{c,iR} - \Phi_{c,iL}) - \frac{\Delta t}{\Delta x_i} (q_{iR}(x_i - x_{i,R}) + q_{iL}(x_{iL} - x_i)) \quad (2.15)$$

### 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:

$$\int_{x_{iL}}^{x_{iR}} \left( \frac{\partial Q}{\partial t} + \frac{\partial}{\partial x} \left( \beta \frac{Q^2}{A_{red}} \right) + \sum Sources \right) dx = 0 \quad (2.16)$$

The different parts of the equation are discretized as follows:

$$\int_{x_{iL}}^{x_{iR}} \frac{\partial Q_i}{\partial t} dx \approx \frac{\partial Q_i}{\partial t} \Delta x_i \approx \frac{Q_i^{t+1} - Q_i^t}{\Delta t} \Delta x_i \quad (2.17)$$

$$\int_{x_{iL}}^{x_{iR}} \frac{\partial Q_i}{\partial x} dx = \beta \frac{Q^2}{A_{red}} \Big|_{x_{iR}} - \beta \frac{Q^2}{A_{red}} \Big|_{x_{iL}} = \Phi_{m,iR} - \Phi_{m,iL} \quad (2.18)$$

$\Phi_{m,iR}$  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:

$$Q_i^{t+1} = Q_i^t - \frac{\Delta t}{\Delta x_i} (\Phi_{m,iR} - \Phi_{m,iL}) + \sum Sources \quad (2.19)$$

### 2.2.2.3 Discretisation of Source Terms

#### 2.2.2.3.1 Bed Slope Source Term

The bed slope source term:

$$W = gA \frac{\partial z_S}{\partial x} \quad (2.20)$$

is discretized as follows:



$$\int_{x_{iL}}^{x_{iR}} gA_{red} \frac{\partial z_S}{\partial x} dx \approx gA_{redi} \left. \frac{\partial z_S}{\partial x} \right|_{x_i} \Delta x_i \approx gA_{redi} \left( \frac{z_{S,i+1} - z_{S,i-1}}{x_{i+1} - x_{i-1}} \right) \Delta x_i \quad (2.21)$$

With the subtraction of the bed slope source term, eq. 2.19 becomes

$$Q_i^{t+1} = Q_i^t - \frac{\Delta t}{\Delta x_i} (\Phi_{m,ir} - \Phi_{m,il}) - \Delta t gA_{redi} \left( \frac{z_{S,i+1} - z_{S,i-1}}{x_{i+1} - x_{i-1}} \right) \quad (2.22)$$

### 2.2.2.3.2 Friction Source Term

The friction source term:

$$Fr = gA_{red}S_f$$

is simply calculated with the local values:

$$\int_{x_{iL}}^{x_{iR}} gA_{red}S_{fi} dx \approx gA_{redi}S_{fi}\Delta x_i \quad (2.23)$$

and

$$S_{fi} = \frac{Q_i^t |Q_i^t|}{(K_i^t)^2} \quad (2.24)$$

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:

$$S_{fi} = \frac{Q_i^{t+1} |Q_i^t|}{(K_i^t)^2} \quad (2.25)$$

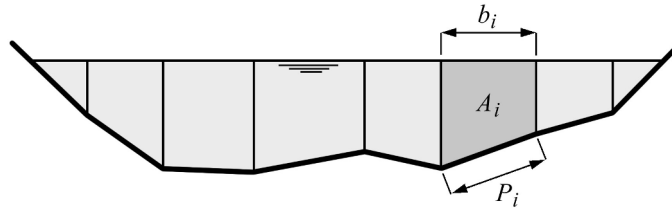
Consequently instead of simply subtracting the source term from eq. 2.22, the following operation is executed on the new discharge  $Q_i^{t+1}$ :

$$Q_i^{t+1} = \frac{Q_i^{t+1}}{1 + \frac{\Delta t |Q_i^t| gA_i^t}{(K_i^t)^2}} \quad (2.26)$$

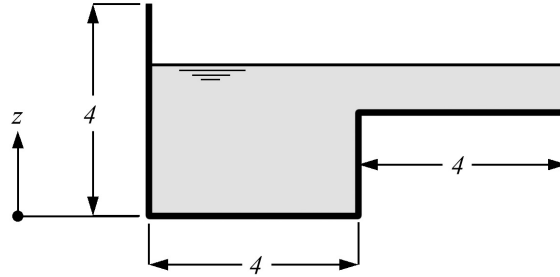
With this approach the discharge tends to zero for small conveyances.

### 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,  $c_f = f(\bar{k})$ .

$$R = \frac{\sum A_i}{\sum P_i} \quad (2.27)$$

$$K = c_f \sqrt{gRA} \quad (2.28)$$

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)

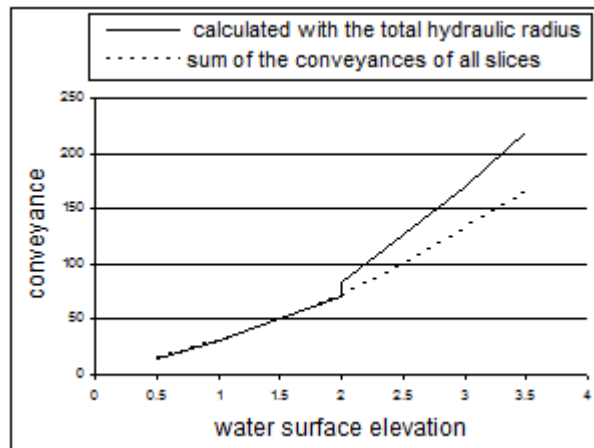


Figure 2.10 Conveyance computation of a channel with a flat zone

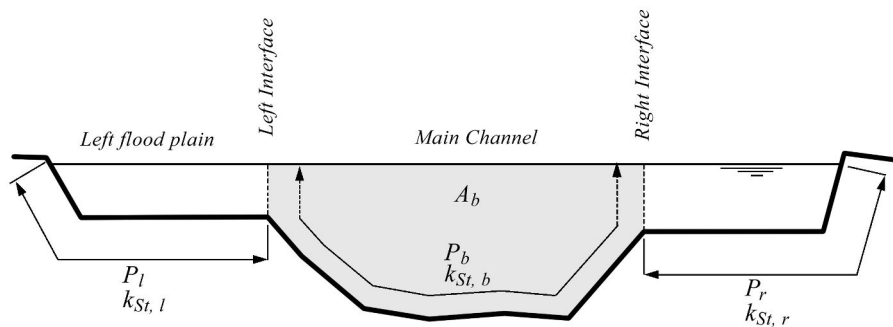


Figure 2.11 Cross section with flood plains and main channel

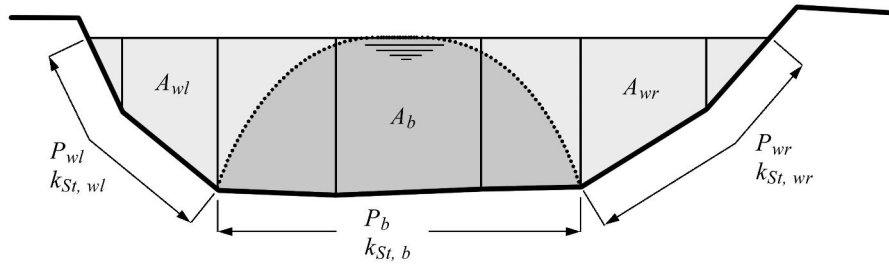


Figure 2.12 Cross section with definition of a bed

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:

$$A_b = R_b P_b \quad (2.29)$$

$$A_{tot} = A_{wl} + A_{wr} + A_b \quad (2.30)$$

$$R_b = \frac{A_{tot}}{k_{stb}^{3/2} \left( \frac{P_{wl}}{k_{StWl}^{3/2}} + \frac{P_b}{k_{Stb}^{3/2}} + \frac{P_{wr}}{k_{StWr}^{3/2}} \right)} \quad (2.31)$$

$$K_b = cf \sqrt{g R_b A_b} \quad (2.32)$$

$$U_{mb} = \frac{K_b}{A_b} \sqrt{S} \quad (2.33)$$

$$Q_b = U_{mb} A_b \quad (2.34)$$

$$Q_w = U_{mb} \left( \frac{A_{tot}}{1.05} - R_b P_b \right) \quad (2.35)$$

$$Q = Q_b + Q_w \quad (2.36)$$

$$K = Q / \sqrt{S} \quad (2.37)$$

$$K = K_b + K_w = \frac{Q_b}{\sqrt{S}} + \frac{Q_w}{\sqrt{S}} = \frac{K_b}{A_b} R_b P_b + \frac{K_b}{A_b} \left( \frac{A_{tot}}{1.05} - R_b P_b \right) = K_b + K_b \left( \frac{A_{tot}}{1.05 A_b} - 1 \right) \quad (2.38)$$

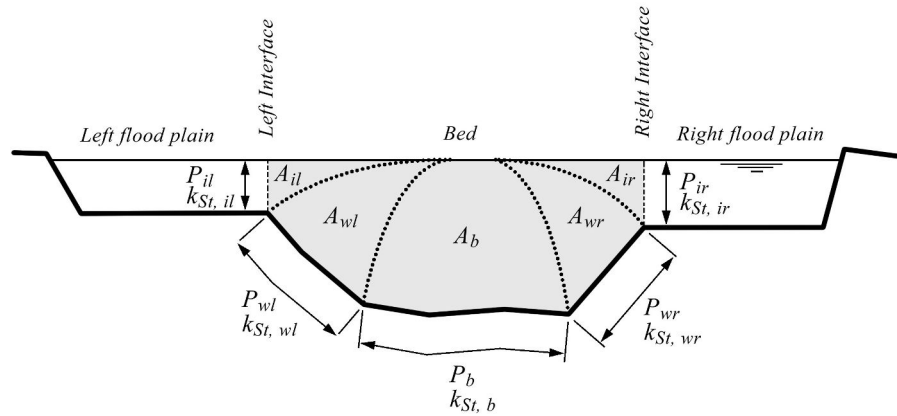


Figure 2.13 Cross section with flood plains and definition of a bed bottom

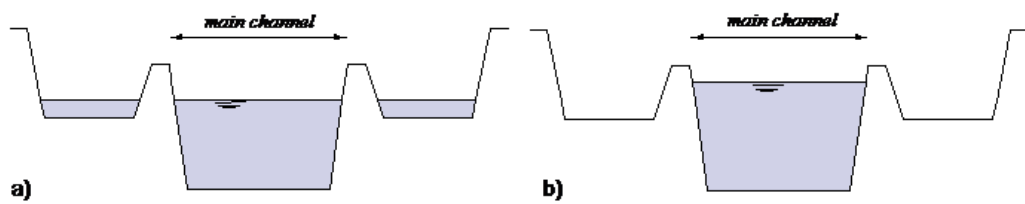


Figure 2.14 Water level in a cross section with internal levees: a) without consideration of the levees, b) with consideration of the levees

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 and:

$$R_b = \frac{A_{tot}}{k_{Stb}^{3/2} \left( \frac{P_{il}}{k_{Stil}^{3/2}} + \frac{P_{wl}}{k_{Stwl}^{3/2}} + \frac{P_b}{k_{Stb}^{3/2}} + \frac{P_{wr}}{k_{Stwr}^{3/2}} + \frac{P_{ir}}{k_{Stir}^{3/2}} \right)} \quad (2.39)$$

### 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.

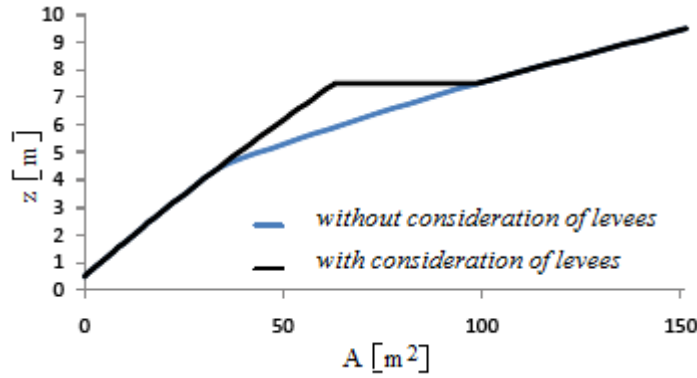


Figure 2.15 Function  $z(A)$  with and without consideration of the levees

### 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  $\beta$  are determined in the first cross section (which has the same location).

$$\Phi_{c,1R} = Q_{bound} \quad (2.40)$$

$$\Phi_{m,1R} = \beta_1 \frac{(Q_{bound})}{A_{red1}} \quad (2.41)$$

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:

$$W_1 = A_{red1} \left( \frac{z_{S,2} - z_{S,1}}{x_2 - x_1} \right) \quad (2.42)$$

#### 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:

$$\Phi_{c,NL} = Q_{weir} \quad (2.43)$$

$$\Phi_{m,NL} = \frac{(Q_{weir})^2}{A_{weir}} \quad (2.44)$$

- $z(t)$  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  $\beta$  in the last cross section.

To satisfy the desired water level, the required inflow from the reservoir into the simulation domain is calculated from:

$$Q_{res}(t) = Q_{in}(t) - \frac{\Delta x}{\tau} (A_{target} - A(t)) \quad (2.45)$$

This approach converges to the target water surface elevation within the characteristic time  $\tau$ .

$$\Phi_{c,NL} = Q_{res} \quad (2.46)$$

$$\Phi_{m,NL} = \beta_{bound} \frac{(Q_{res})^2}{A_{bound}} \quad (2.47)$$

- 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:

$$\Phi_{c,Ll} = \Phi_{c,NR} \quad (2.48)$$

$$\Phi_{m,NL} = \Phi_{m,NR} \quad (2.49)$$

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:

$$W_N = A_{redN} \left( \frac{z_{S,N} - z_{S,N-1}}{x_N - x_{N-1}} \right) \quad (2.50)$$

### 2.2.2.4.3 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 2.16).

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,

$$w_{eff} = w - 2(n_p c_p + c_a) H_{ref} \quad (2.51)$$

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 2.16), this weir formula is used

$$q = \frac{2}{3} \mu \sigma_{uv} \sqrt{2g(z_{Ref} - z_{weircrest})^3} \quad (2.52)$$

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  $\sigma_{uv}$  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 2.16). 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  $\sigma_{uv}$  is calculated according to the Diagram in Figure 2.17.  $z_{Ref}$  is the water surface elevation at the reference cross section. The weir bottom level (2) is the lowest possible weir crest level (Figure 2.16).

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:

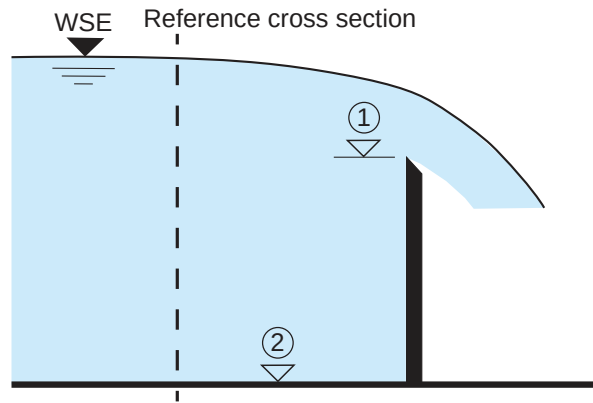
- I) "Standard": The wetted area above the weir crest is calculated as

$$A_{Weir} = (z_{Ref} - z_{weircrest}) w_{eff}$$

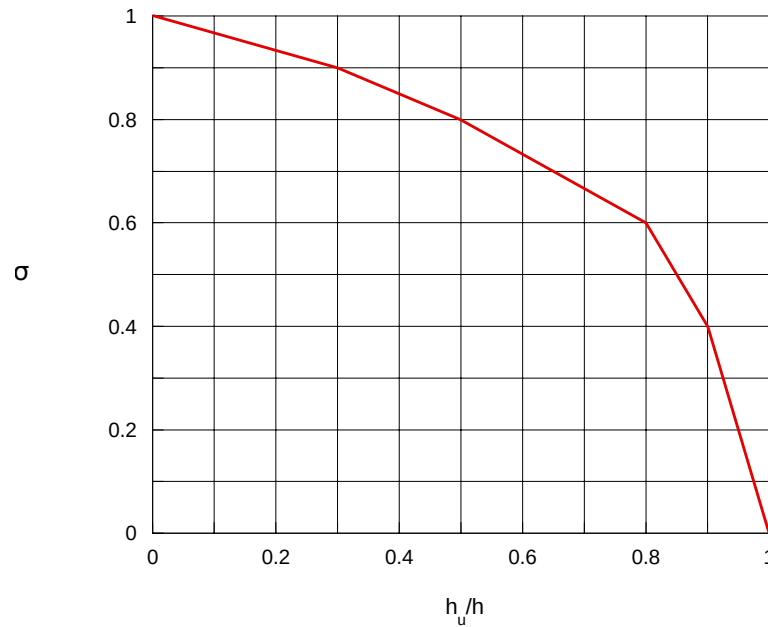
- II) "Critical": It is assumed that the critical flow depth is at the weir crest. The critical flow depth is calculated as  $h_{crit} = 2/3[z_{Ref} - z_{weircrest}]$ . The wetted area above the weir crest is calculated as  $A_{Weir} = h_{crit} w_{eff}$ . For a model calibration the water surface elevation  $z_{crit}$  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  $f_{crit} = a + b h_{crit}$ .

- Inner Gate:

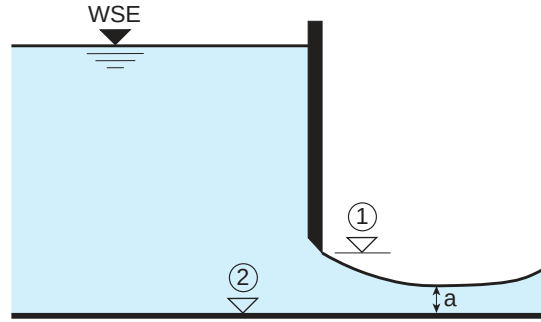




**Figure 2.16** Inner weir with a complete weir flow. (1) weir crest level, (2) weir bottom level, Reference cross section where water surface is unaffected by the flow acceleration at the weir



**Figure 2.17** Reduction factor  $\sigma_{uv}$  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 2.18** Inner sluice gate. (1) gate level, (2) gate bottom level,  $a$  = water depth at vena contracta

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 2.18). 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 2.18). Similar to the gate boundary condition, the calculation of the specific discharge is

$$q = \mu h_{gate} \sqrt{2gh_0} \quad (2.53)$$

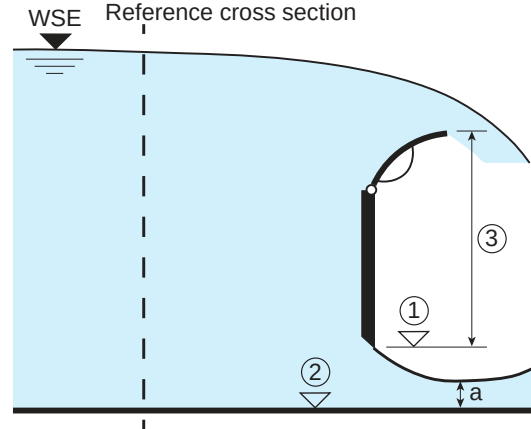
The gate opening is defined as  $h_{gate} = \text{gate level (1)} - \text{gate bottom level (2)}$  (Figure 2.18),  $h_0$  denotes the water depth at the cross section upstream of the gate. The discharge coefficient  $\mu$  depends on the contraction factor  $\delta$  which is the ratio of the water depth  $a$  at the vena contracta to the gate opening  $h_{gate}$  and is calculated as

$$\mu = \delta / \sqrt{1 + \frac{\delta h_{gate}}{h_0}} \quad (2.54)$$

Derived from the conjugate depths at a hydraulic jump the criterion for considering the backwater effect at the gate is defined as

$$\frac{h_d}{h_{gate}} > \frac{\delta}{2} \left( \sqrt{1 + \frac{16}{\delta \left(1 + \frac{\delta h_{gate}}{h_0}\right)} \frac{h_0}{h_{gate}}} - 1 \right) \quad (2.55)$$

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 2.19** 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

$$\mu = \delta \sqrt{\left[1 - 2 \frac{\delta h_{gate}}{h_0} \left(1 - \frac{\delta h_{gate}}{h_d}\right)\right] - \sqrt{\left[1 - 2 \frac{\delta h_{gate}}{h_0} \left(1 - \frac{\delta h_{gate}}{h_d}\right)\right]^2 + \left(\frac{h_d}{h_0}\right)^2} - 1} \quad (2.56)$$

As for the momentum, the velocity through the gate is taken into account in both, downstream and upstream direction.

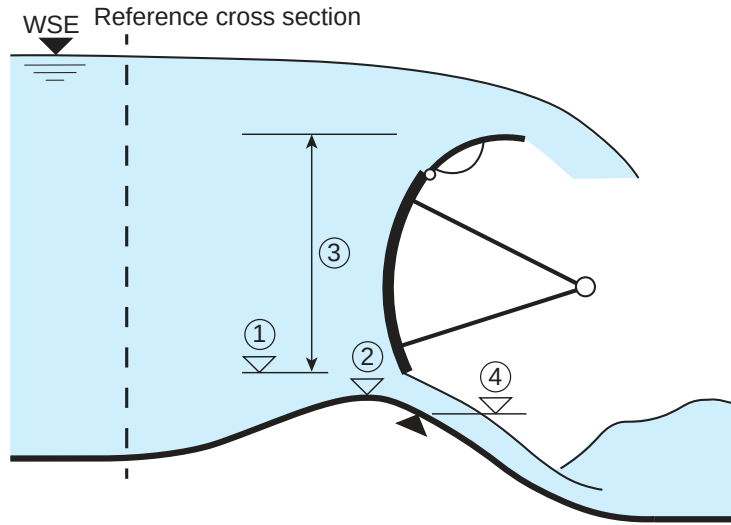
The second gate type is the gate with flap. Three modes are equal to the sluice gate:

- 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 2.19).

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)

$$q = \psi h_{gate} \sqrt{2g \left(H_{Ref} - \frac{h_{gate}}{2}\right)} \quad (2.57)$$



**Figure 2.20** 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  $\psi$  is the discharge coefficient which is defined as  $\psi = \alpha(h_{gate}/H_{Ref})^\beta$ . The user has to define the calibration parameters  $\alpha$  and  $\beta$ . 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 2.20).

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 2.20). In this case the inner boundary acts either as a wall or 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.

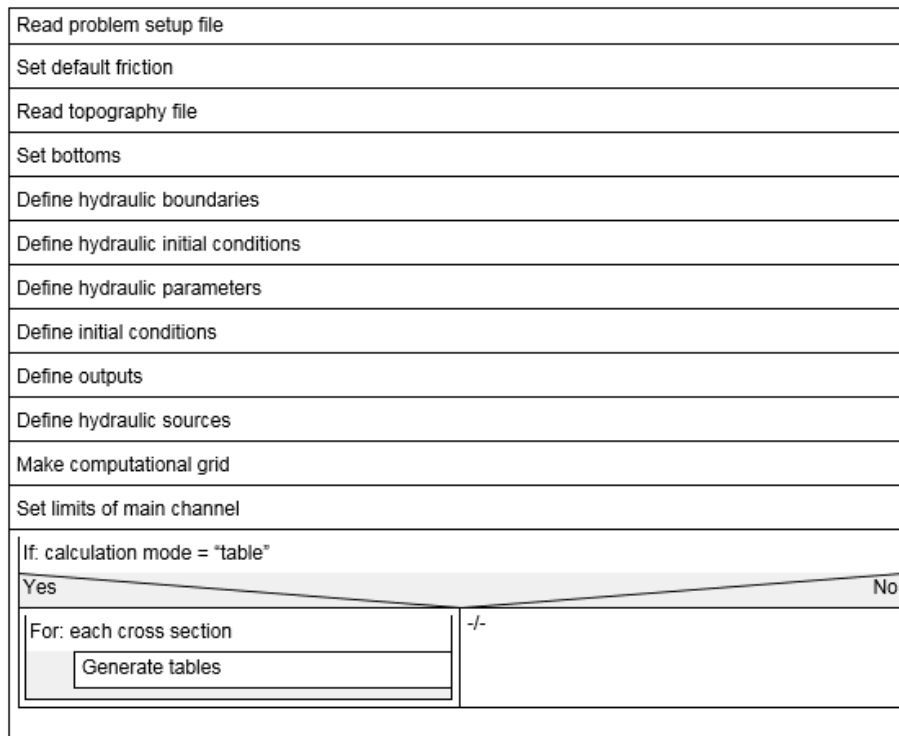
As soon as the radial gate is open and the water surface elevation is lower than the gate level the inner boundary acts as a weir with the weir bottom level (2, Figure 2.20). For the calculation of the weir flux the energy height above the bottom elevation at the reference cross section is used.

#### 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



**Figure 2.21** General solution procedure of BASEchain

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:

$$W_i = A_{red,i} \left( \frac{z_{S,i+1} - z_{S,i}}{x_{i+1} - x_i} \right) \quad (2.58)$$

### 2.2.2.5 Solution Procedure

- General solution procedure of BASEchain in detail (Figure 2.21 and Figure 2.22)
- Time loop (Figure 2.23)
- Hydrodynamic equations (Figure 2.24)
- Morphodynamic equations (Figure 2.25)
- Suspended load equations (Figure 2.26)

If sediment transport is active ( bed load or suspended load )	
Yes	No
Define general sediment parameters	-/-
Define grain classes	-/-
Define mixtures	-/-
Define soils	
If <u>bedload</u> is active	
Yes	No
Set <u>bedload</u> 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
Create exchange source term	
Print topography output	
Print initial conditions to output files	
Generate the needed sources	

Figure 2.22 General solution procedure of BASEchain

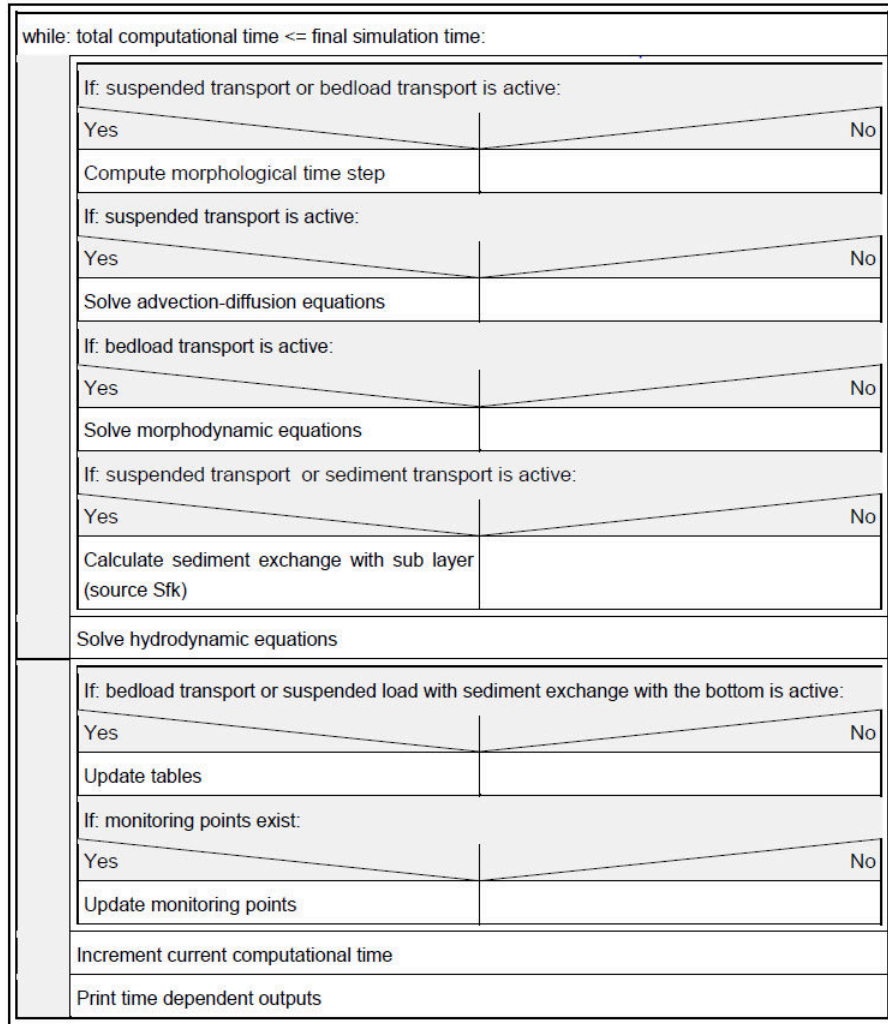


Figure 2.23 Time loop

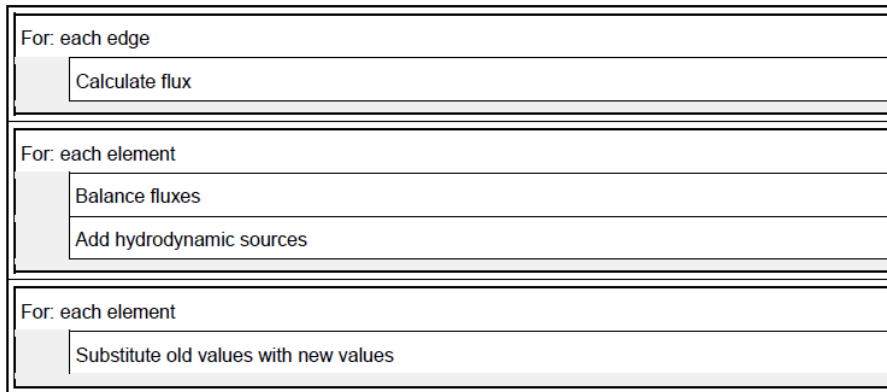
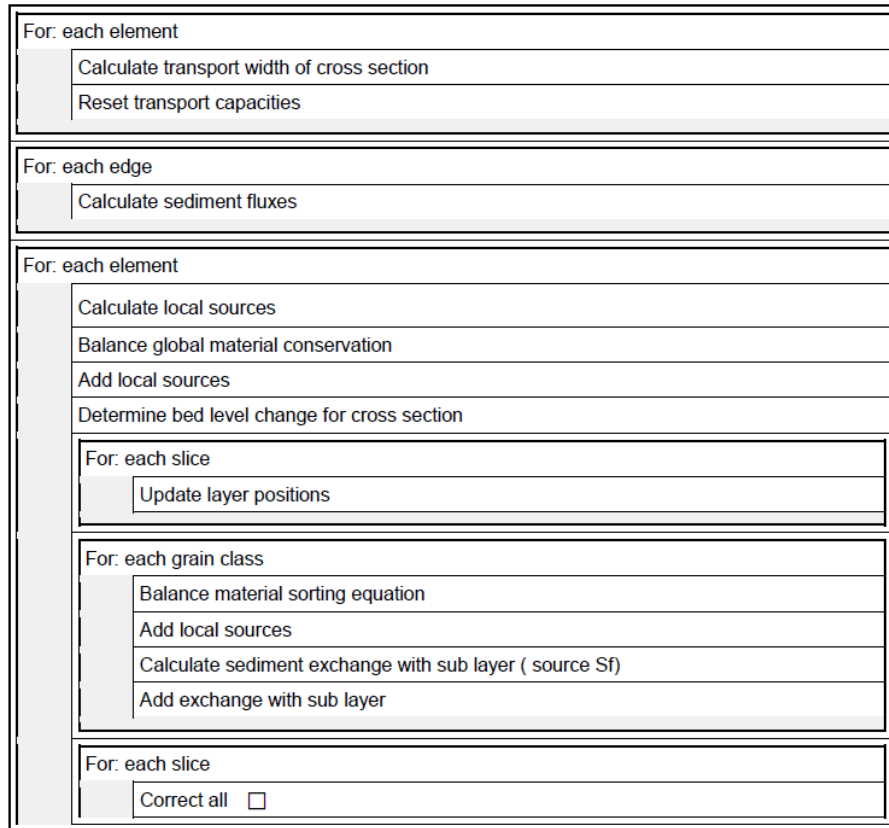
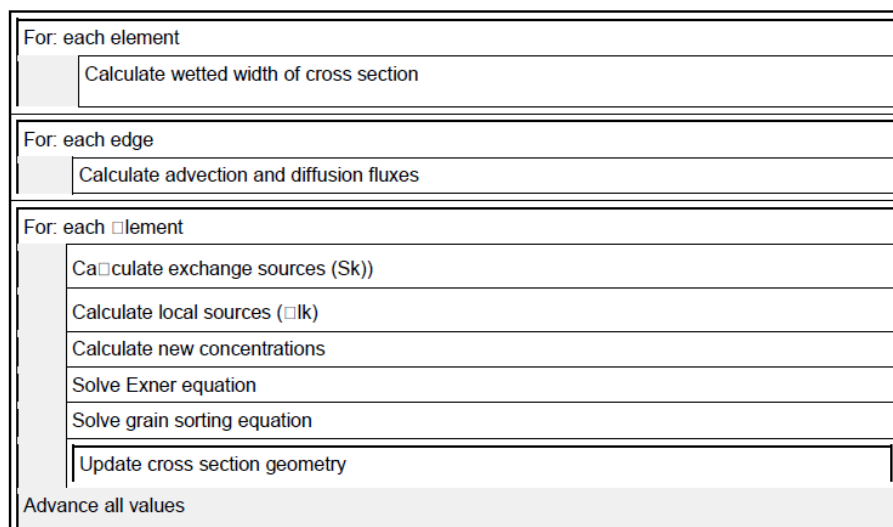


Figure 2.24 Hydrodynamic equations



*Figure 2.25 Morphodynamic equations*



*Figure 2.26 Suspended load equations*



## 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:

$$\int_{\Omega} \mathbf{U}_t \, d\Omega + \int_{\Omega} \nabla \cdot (\mathbf{F}, \mathbf{G}) \, d\Omega + \int_{\Omega} \mathbf{S} \, d\Omega = 0 \quad (2.59)$$

in which  $\Omega$  equals the area of the calculation cell (Figure 2.1) Using the Gauss' relation eq. 2.59 becomes:

$$\int_{\Omega} \mathbf{U}_t \, d\Omega + \int_{\partial\Omega} (\mathbf{F}, \mathbf{G}) \cdot \mathbf{n}_s \, dl + \int_{\Omega} \mathbf{S} \, d\Omega = 0 \quad (2.60)$$

Assuming  $U_t$  and  $S$  are constant over the domain for first order accuracy, it can be written:

$$\mathbf{U}_t + \frac{1}{\Omega} \int_{\partial\Omega} (\mathbf{F}, \mathbf{G}) \cdot \mathbf{n}_s \, dl + \mathbf{S} = 0 \quad (2.61)$$

The eq. 2.60 can be discretized by a two-phase scheme namely predictor corrector scheme as follows:

$$\mathbf{U}_i^{n+1} = \mathbf{U}_i^n - \frac{\Delta t}{\Omega} \sum_{j=1}^3 (\mathbf{F}, \mathbf{G})_{i,j}^n \times \mathbf{n}_j l_j - \Delta t \mathbf{S}_i \quad (2.62)$$

where

- $m$  = number of cell or element sides
- $(\mathbf{F}, \mathbf{G})_{i,j}$  = numerical flux through the side of cell
- $\mathbf{n}_{s,i}$  = 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  $((\mathbf{F}, \mathbf{G}) \cdot \mathbf{n}_s)$ . 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. 2.61, it can be written:

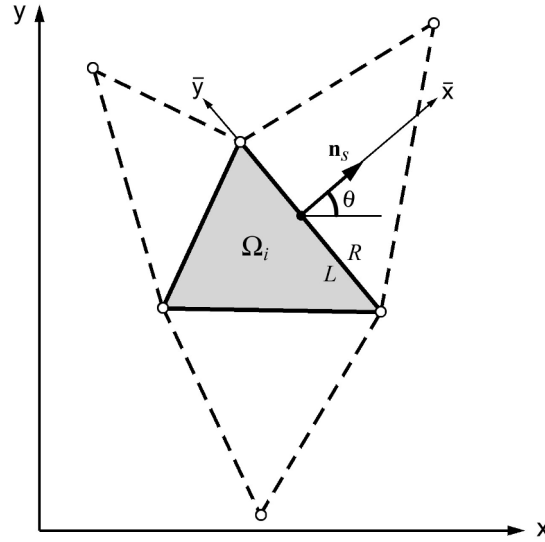


Figure 2.27 Geometry of a Computational Cell  $\Omega_i$  in FV

$$\int_{\partial\Omega} (\mathbf{F}(\mathbf{U}), \mathbf{G}(\mathbf{U})) \cdot \mathbf{n}_s \, dl = \int_{\partial\Omega} (\mathbf{F}(\mathbf{U}) \cos \theta + \mathbf{G}(\mathbf{U}) \sin \theta) \, dl \quad (2.63)$$

in which  $\mathbf{n}_s = (\cos \theta, \sin \theta)$  is the outward unit vector to the boundary of domain  $\Omega_i$  (see Figure 2.27). Based on the rotational invariance property for  $\mathbf{F}(\mathbf{U})$  and  $\mathbf{G}(\mathbf{U})$  on the boundary of the domain, it can be written according to Toro (1997):

$$(\mathbf{F}(\mathbf{U}), \mathbf{G}(\mathbf{U})) = \mathbf{T}^{-1}(\theta) \mathbf{F}(\mathbf{T}(\theta) \mathbf{U}) \quad (2.64)$$

where  $\theta$  is the angle between the vector  $\mathbf{n}_s$  and x-axis, measured counter clockwise from the x-axis (see Figure 2.27).

$$\mathbf{T}(\theta) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & \sin \theta \\ 0 & -\sin \theta & \cos \theta \end{pmatrix} \quad (2.65)$$

$\mathbf{T}^{-1}(\theta)$  = inverse of  $\mathbf{T}(\theta)$

Using eq. 2.64, eq. 2.61 can be rewritten as:

$$\mathbf{U}_t + \frac{1}{\Omega} \int_{\partial\Omega} \mathbf{T}^{-1}(\theta) \mathbf{F}(\mathbf{T}(\theta) \mathbf{U}) \, dl + \mathbf{S} = 0 \quad (2.66)$$

The quantity  $\mathbf{T}(\theta) \mathbf{U}$  is transformed of  $\mathbf{U}$ , with velocity components in the normal and tangential direction. For each cell in the computational domain, the quantity  $\mathbf{U}$ , thus  $\mathbf{T}(\theta) \mathbf{U}$  may have different values, which results in a discontinuity across the interface between cells. Therefore, the two-dimensional problem in eq. 2.59 or eq. 2.61 can be handled as a series of local Riemann problems in the normal direction to the cell interface ( $\bar{x}$ ) by the eq. 2.66.

Applying the foregoing, the flux computations over the edges are preformed in three successive steps:

- First, the vector of conserved variables  $\mathbf{U}$  is transformed into the local coordinate system at the edge with the operation  $\mathbf{T}(\theta)\mathbf{U}$ .
- 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  $\mathbf{F}[\mathbf{T}(\theta)\mathbf{U}]$ .
- The flux vector, formulated in the local coordinate system at the edge, is transformed back to Cartesian coordinates with  $\mathbf{T}^{-1}\mathbf{F}[\mathbf{T}(\theta)\mathbf{U}]$ . 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

$$\begin{aligned} h_k^{corr} &= h_k + \omega_k h_i \frac{A_i}{A_k} \\ Flux_k^{corr} &= Flux_k + \omega_k h_i \frac{A_i}{\Delta t} \end{aligned} \quad (2.67)$$

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  $\omega_k$  is a weighting factor which is obtained by weighting the volumetric fluxes of all corrected neighbouring elements  $k$ .

$$\omega_k = \frac{Flux_k}{\sum_k Flux_k} \quad (2.68)$$

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. 2.62 there are different possibilities for the evaluation of the source term  $\mathbf{S}_i$ . It can be evaluated either with the variables of the old time step as  $\mathbf{S}_i(\mathbf{U}_i)$ , which is often

referred to as unsplitted scheme, or it can be evaluated with the advanced values  $\mathbf{U}_i^{n+1/2}$ , which already include changes due to the numerical fluxes computed during this time step as  $\mathbf{S}_i(\mathbf{U}_i^{n+1/2})$ . The use of the advanced values for the source term calculation is chosen here because it gives better results (Toro, 2001). Therefore eq. 2.62 is split in following way

$$\mathbf{U}_i^{n+1/2} = \mathbf{U}_i^n - \frac{\Delta t}{\Omega} \sum_{j=1}^3 (\mathbf{F}, \mathbf{G})_{i,j}^n \cdot \mathbf{n}_j dl_j \quad (2.69)$$

$$\mathbf{U}_i^{n+1} = \mathbf{U}_i^{n+1/2} + \Delta t \mathbf{S}_i(\mathbf{U}_i^{n+1/2})$$

But, as explained in the following, the friction source term  $\mathbf{S}_{i,fr}$  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  $\mathbf{U}_i^{n+1}$  at the new time level as

$$\mathbf{U}_i^{n+1} = \mathbf{U}_i^{n+1/2} + \Delta t \mathbf{S}_{i,fr}(\mathbf{U}_i^{n+1}) \quad (2.70)$$

Considering the generalized  $c_f$  friction coefficient and after some algebraic manipulations, one obtains:

$$\mathbf{U}_i^{n+1} = \frac{\mathbf{U}_i^{n+1/2}}{1 + \Delta t \frac{\sqrt{(u_i^n)^2 + (v_i^n)^2}}{c_{f,i}^2 R_i}} \quad (2.71)$$

### Hydraulic Radius

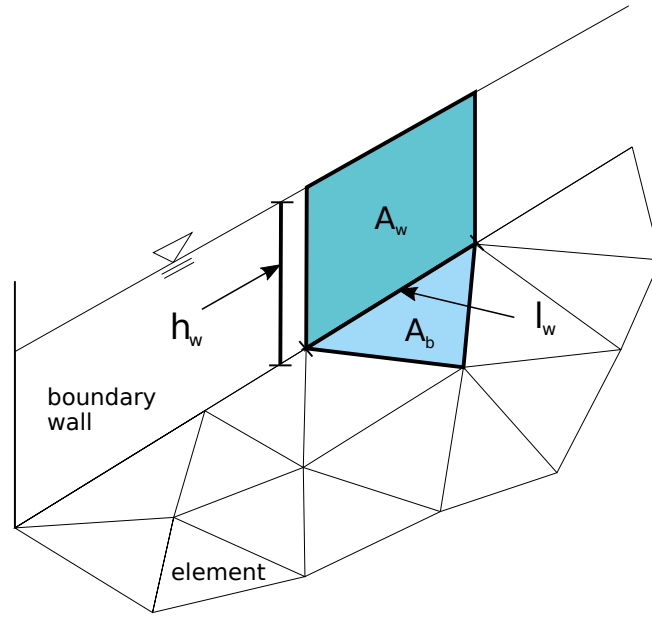
The calculation of the friction source term requires a definition of the hydraulic radius  $\mathbf{R}_i$  in the element  $i$ . The hydraulic radius is defined here as water depth in the element ( $\mathbf{R}_i = \mathbf{h}_i$ ).

### 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 2.28. 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

$$S_{i,fr,x} = \frac{u\sqrt{u^2 + v^2}}{g} \left( \frac{1}{\bar{c}_f^2 R_w} \right) \quad (2.72)$$



**Figure 2.28** Wetted perimeter of a boundary element with wall friction

with  $R_w$  being the hydraulic radius of the wall friction at the boundary edge  $w$ . The first term in eq. 2.72 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

$$R_w = \frac{V_{water}}{A_w} = \frac{A_b h}{A_b + \sum_i l_{w,i} h} \quad (2.73)$$

where  $l_w$  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:

$$\iint_{\Omega} \mathbf{S}_d \, d\Omega = \iint_{\Omega} \frac{\partial \mathbf{F}^d}{\partial x} + \frac{\partial \mathbf{G}^d}{\partial y} \, d\Omega \quad (2.74)$$

By using the divergence theorem, the diffusive flux integrals are becoming boundary integrals

$$\iint_{\Omega} \frac{\partial \mathbf{F}^d}{\partial x} + \frac{\partial \mathbf{G}^d}{\partial y} \, d\Omega = \oint_{\partial\Omega} \mathbf{F}^d \cdot \mathbf{n} + \mathbf{G}^d \cdot \mathbf{n} \, ds \quad (2.75)$$

The boundary integral is discretized by a summation over the element edges (index e)

$$\oint_{\partial\Omega} \mathbf{F}^d \cdot \mathbf{n} + \mathbf{G}^d \cdot \mathbf{n} \, ds = \sum_e (\mathbf{F}_e^d \cdot \mathbf{n}_e + \mathbf{G}_e^d \cdot \mathbf{n}_e) \, ds_e \quad (2.76)$$

The diffusive fluxes  $\mathbf{F}_e^d$  and  $\mathbf{G}_e^d$  on the element edges are calculated by a centred scheme

$$\mathbf{F}_e^d = \frac{1}{2}(\mathbf{F}_R^d + \mathbf{F}_L^d), \quad \mathbf{G}_e^d = \frac{1}{2}(\mathbf{G}_R^d + \mathbf{G}_L^d) \quad (2.77)$$

where R and L stand for a value right and left of the edge. The diffusive fluxes on the edges read then as

$$\mathbf{F}_e^d = \frac{1}{2} \begin{pmatrix} 0 \\ \left( \nu h \frac{\partial u}{\partial x} \right)_R + \left( \nu h \frac{\partial u}{\partial x} \right)_L \\ \left( \nu h \frac{\partial v}{\partial x} \right)_R + \left( \nu h \frac{\partial v}{\partial x} \right)_L \end{pmatrix} \quad (2.78)$$

and

$$\mathbf{G}_e^d = \frac{1}{2} \begin{pmatrix} 0 \\ \left( \nu h \frac{\partial u}{\partial y} \right)_R + \left( \nu h \frac{\partial u}{\partial y} \right)_L \\ \left( \nu h \frac{\partial v}{\partial y} \right)_R + \left( \nu h \frac{\partial v}{\partial y} \right)_L \end{pmatrix} \quad (2.79)$$

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  $\nu_t$  can be either set to a constant value or calculated dynamically

for each element. Using the dynamic case, the values for  $\nu_t$  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  $\varphi$  on an arbitrary element is given by

$$\left(\frac{\partial\varphi}{\partial x}\right)_{Elem} = \frac{1}{\Omega} \int_{\Omega} \frac{\partial\varphi}{\partial x} d\Omega \approx \frac{\sum_e \varphi_e \Delta y_e}{\Omega} \quad (2.80)$$

$$\left(\frac{\partial\varphi}{\partial y}\right)_{Elem} = \frac{1}{\Omega} \int_{\Omega} \frac{\partial\varphi}{\partial y} d\Omega \approx \frac{\sum_e \varphi_e \Delta x_e}{\Omega} \quad (2.81)$$

where  $\Omega$  is the area of the element and  $e$  stands for an edge.  $\varphi_e$  is a value on the edge. As in finite volume methods, most variables are defined on an element,  $\varphi_e$  has to be calculated as average of the neighbouring elements:

$$\varphi_e = \frac{1}{2}(\varphi_R + \varphi_L) \quad (2.82)$$

The spatial differences  $\Delta y_e$  and  $\Delta x_e$  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. 2.78 is computed as

$$\left(\nu h \frac{\partial u}{\partial x}\right)_L \approx \nu_L h_L \frac{\sum_e u_e \Delta y_e}{\Omega_L} \quad (2.83)$$

with  $u_e = 0.5(u_{eL} + u_{eR})$ .

The depth-averaged turbulent viscosity  $\nu_t$  can either be set to a constant value or it is calculated for every element using the formula

$$\nu_t = \frac{\kappa}{6} u_* h \quad (2.84)$$

Where  $\kappa = 0.4$  is the von Karman constant and  $u_*$  is the shear velocity which is defined as

$$u_* = \sqrt{c_f(u^2 + v^2)} \quad (2.85)$$

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. 2.61, it is required to compute the integral of the bed slope source term over a element  $\Omega_i$ .

$$\iint_{\Omega} \mathbf{S}_B \, d\Omega = \iint_{\Omega} \begin{pmatrix} 0 \\ ghS_{Bx} \\ ghS_{By} \end{pmatrix} d\Omega = g \iint_{\Omega} h \begin{pmatrix} 0 \\ -\frac{\partial z_B(x,y)}{\partial x} \\ -\frac{\partial z_B(x,y)}{\partial y} \end{pmatrix} d\Omega \quad (2.86)$$

Assuming that the bed slope values are constant over a cell, eq. 2.86 can be simplified to:

$$\iint_{\Omega} \mathbf{S}_B \, d\Omega = g \left( \iint_{\Omega} h \, d\Omega \right) \begin{pmatrix} 0 \\ S_{Bx} \\ S_{By} \end{pmatrix} = gVol_{water} \begin{pmatrix} 0 \\ S_{Bx} \\ S_{By} \end{pmatrix} \quad (2.87)$$

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 2.29):

$$z_B(x, y) = \alpha_1 + \alpha_2 x + \alpha_3 y \quad (2.88)$$

in which  $\alpha_2 = \frac{\partial z_B}{\partial x}$ ;  $\alpha_3 = \frac{\partial z_B}{\partial y}$

The constants  $\alpha_1$ ,  $\alpha_2$  and  $\alpha_3$  can be determined by inserting the nodal coordinates and equating to the corresponding nodal values of  $z_B$ . Solving for  $\alpha_1$ ,  $\alpha_2$  and  $\alpha_3$  finally gives

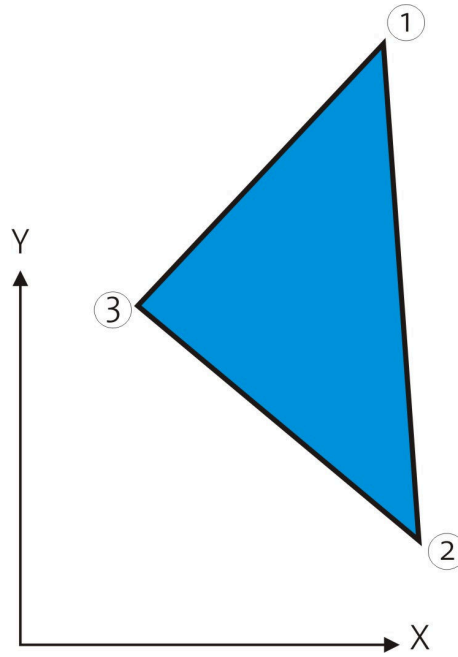
$$z_B(x, y) = \frac{1}{2\Omega} [(a_1 + b_1 x + c_1 y)z_{b,1} + (a_2 + b_2 x + c_2 y)z_{b,2} + (a_3 + b_3 x + c_3 y)z_{b,3}] \quad (2.89)$$

where

$$\left. \begin{aligned} a_1 &= x_2 y_3 - x_3 y_2 \\ b_1 &= y_2 - y_3 \\ c_1 &= x_3 - x_2 \end{aligned} \right\} \quad (2.90)$$

With the other coefficients given by cyclic permutation of the subscripts in the order 1,2,3. The area  $\Omega$  of the triangular element is given by





**Figure 2.29** A Triangular Cell

$$2\Omega = \begin{vmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{vmatrix} \quad (2.91)$$

One can compute the bed slopes in x- and y-direction in each cell as

$$\left. \begin{aligned} S_{Bx} &= -\frac{\partial z_B(x, y)}{\partial x} = -\frac{1}{2\Omega} (b_1 z_{B,1} + b_2 z_{B,2} + b_3 z_{B,3}) \\ S_{By} &= -\frac{\partial z_B(x, y)}{\partial y} = -\frac{1}{2\Omega} (c_1 z_{B,1} + c_2 z_{B,2} + c_3 z_{B,3}) \end{aligned} \right\} \quad (2.92)$$

The water volume over a cell can also be computed by using the parametric coordinates of the Finite Element Method.

$$Vol_{water} = \left( \frac{h_1 + h_2 + h_3}{3} \right) \Omega \quad (2.93)$$

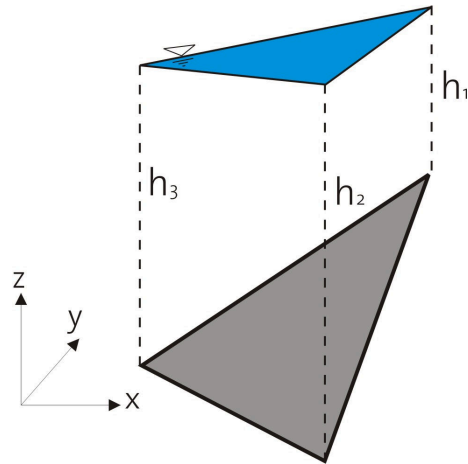
where  $h_1$ ,  $h_2$  and  $h_3$  are the water depths at the nodes 1, 2 and 3 respectively (Figure 2.30).

In the case of partially wet cells (Figure 2.31) the location of the wet-dry line (a and b) has to be determined, where the water surface plane intersects the cell surface.

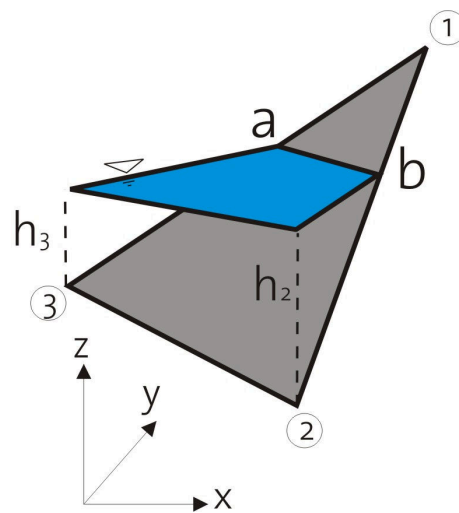
Using the coordinates of a and b, the water volume over the cell can be calculated as:

$$Vol_{water} = \Omega_{a32} \frac{h_2 + h_3}{3} + \Omega_{2ba} \frac{h_2}{3} \quad (2.94)$$

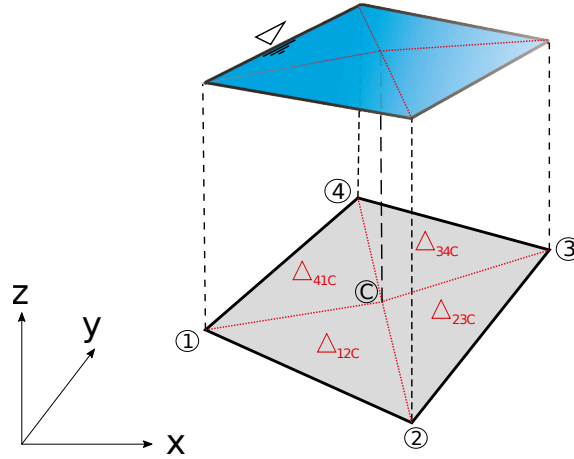
In the computation of the fluxes through the edges (1-2) and (1-3), the modified lengths are used. The modified length is computed under the assumption that the water elevation is constant over a cell.



**Figure 2.30** Water volume over a cell



**Figure 2.31** Partially wet cell



**Figure 2.32** Quadrilateral element and its division into four triangles ( $\Delta_{12C}$ ,  $\Delta_{23C}$ ,  $\Delta_{34C}$ ,  $\Delta_{41C}$ )

### 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):

$$z_C = \frac{\sum_{k=1}^4 z_k \sqrt{(x_k - x_C)^2 + (y_k - y_C)^2}}{\sum_{k=1}^4 \sqrt{(x_k - x_C)^2 + (y_k - y_C)^2}} \quad (2.95)$$

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.

$$\iint_{\Omega} \mathbf{S}_B \, d\Omega = g \left( \iint_{\Omega} h \, d\Omega \right) \begin{pmatrix} 0 \\ S_{Bx} \\ S_{By} \end{pmatrix} = g \sum_{k=1}^4 \left[ Vol_{water, \Delta_k} \begin{pmatrix} 0 \\ S_{Bx,k} \\ S_{By,k} \end{pmatrix} \right] \quad (2.96)$$

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  $\phi$  with the x- or y-coordinate, one obtains the bed slopes as follows:

$$\frac{\partial z_B(x, y)}{\partial \phi} \approx \frac{1}{A_{Quad}} \sum_{k=1}^4 \left( \frac{\partial z_{B,k}(x, y)}{\partial \phi} A_{\Delta,k} \right) , \quad \phi = x, y \quad (2.97)$$

### 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:

$$\oint_{d\Omega} \left( \frac{1}{2} g h^2 \right) n_x dl = g \iint_{\Omega} h S_{Bx} d\Omega|_{\zeta=const} \quad (2.98)$$

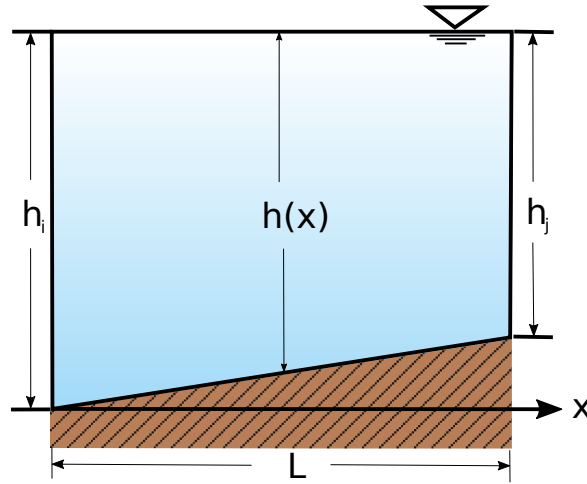
$$\oint_{d\Omega} \left( \frac{1}{2} g h^2 \right) n_y dl = g \iint_{\Omega} h S_{By} d\Omega|_{\zeta=const}$$

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  $0.5gh^2_{mod}$  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.

$$h^2_{mod} = \frac{1}{L} \int_0^L h^2(x) dx = \frac{h_i^2 + h_i h_j + h_j^2}{3} \quad (2.99)$$

$h_i$  and  $h_j$  are the water depths at the edge's left and right nodes, as shown in Figure 2.33.

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 2.33** Edge with linearly varying water depth

$$\begin{aligned}
 g \iint_{\Omega} h S_{Bx} \, d\Omega|_{\zeta=\text{const}} &= g \text{Vol}_{\text{water}} S_{Bx} = g \frac{1}{3} (h_1 + h_2 + h_3) \frac{1}{2} [b_1 z_{B,1} + b_2 z_{B,2} + b_3 z_{B,3}] \\
 &= \frac{g}{6} (h_1 + h_2 + h_3) [b_1 (\zeta - h_1) + b_2 (\zeta - h_2) + b_3 (\zeta - h_3)] \\
 &= \frac{g}{6} (h_1 + h_2 + h_3) \left[ \underbrace{\zeta (b_1 + b_2 + b_3)}_{=0} - b_1 h_1 - b_2 h_2 - b_3 h_3 \right] \\
 &= -\frac{g}{6} (h_1 + h_2 + h_3) [b_1 h_1 + b_2 h_2 + b_3 h_3] \\
 \\ 
 - \oint_{d\Omega} \left( \frac{1}{2} g h^2 \right) n_x \, dl &= -\frac{1}{2} g \sum_{k=1}^n \left[ \int_{x=0}^l h(x) \, dx \right] \\
 &= -g \frac{1}{2} \left[ \frac{1}{3} (h_1^2 + h_1 h_2 + h_2^2) b_3 + \frac{1}{3} (h_2^2 + h_2 h_3 + h_3^2) b_1 + \frac{1}{3} (h_3^2 + h_3 h_1 + h_1^2) b_2 \right] \quad (2.100) \\
 &= -\frac{g}{6} (h_1 + h_2 + h_3) [b_1 h_1 + b_2 h_2 + b_3 h_3]
 \end{aligned}$$

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 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:

$$\begin{array}{ll} t^1 & Q^1 \\ \dots & \\ t^n & Q^n \\ \dots & \\ t^{end} & Q^{end} \end{array}$$

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

$$Q_i = K_i Q_{in} \quad (2.101)$$

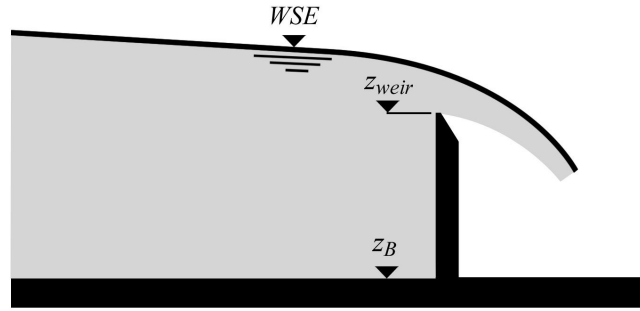
This weighting factor  $K_i$  can be calculated based on its local conveyance as

$$K_i = \frac{c_{fi} \sqrt{R_i} \sqrt{g} A_i}{\sum_{j=1}^n (c_{fj} \sqrt{R_j} \sqrt{g} A_j)} \quad (2.102)$$

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.

$$K_i = \frac{A_i}{\sum_{j=1}^n A_j} \quad (2.103)$$



*Figure 2.34* 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:

$$\frac{\partial}{\partial \mathbf{n}} = 0 \quad (2.104)$$

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 2.34), the discharge at the outlet is computed based on the weir function (Chanson, 1999)

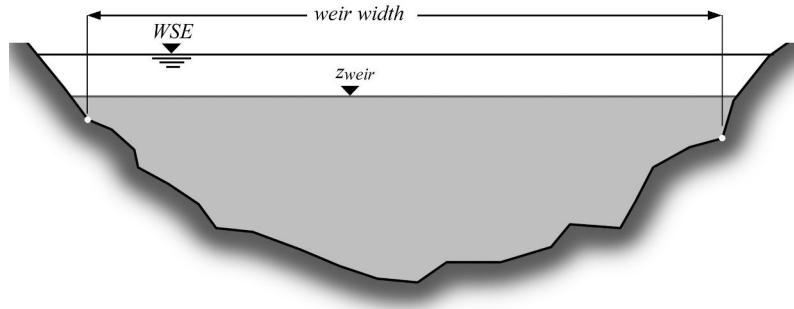
$$q = \frac{2}{3} C \sqrt{2g(h_{up} - h_{weir})^3}$$

where

$$C = 0.611 + 0.08 \frac{h_{up} - h_{weir}}{h_{weir}} \quad ; \quad h_{up} = WSE - z_B \quad \text{and} \quad h_{weir} = z_{weir} - z_B$$

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 2.35) some edges of the outlet are considered as a weir and the others have free surface elevation boundary condition. In order to avoid



**Figure 2.35** 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 2.35 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

$$q = \mu h_{gate} \sqrt{2gh_0} \quad (2.105)$$

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  $\mu$  can be defined by the user (constant value or dynamical). According to Bernoulli's equation and assuming the upstream water is at rest, the discharge coefficient is

$$\mu = \frac{\delta}{\sqrt{1 + \frac{\delta h_{gate}}{h_0}}} \quad (2.106)$$

where  $\delta$  is the contraction coefficient of the outflow jet. Assuming a sharp-edged sluice gate the contraction coefficient is calculated by Voigt (1971).

$$\delta = \frac{1}{1 + 0.64 \sqrt{1 - (h_{gate}/h_0)^2}} \quad (2.107)$$

The value of  $\mu$  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 elevation/discharge 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

$$\begin{array}{l} h^1 \quad Q^1 \\ \dots \\ h^n \quad Q^n \end{array}$$

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

$$\begin{array}{l} t^1 \quad WSE^1 \\ \dots \\ t^n \quad WSE^n \end{array}$$

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 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

$$Q = \frac{2}{3} \mu \sigma_{uw} w \sqrt{2g(z_{upstream} - z_{weir})^3} \quad (2.108)$$

is used. This is the classical Poleni formula for a sharp crested weir with an additional factor  $\sigma_{uw}$  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  $\sigma_{uw} = 1$ .  $w$  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  $\sigma$  is calculated according to the Diagram Figure 2.17.

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.

- Inner Gate:

Similar to the gate boundary condition, the inner gate has three modes. Either the gate level is equal or less than the soil elevation. The gate is then closed and acts as a wall.

If the gate level is above the local soil elevation, the gate is considered as open. As long as the water surface elevations near the gate are below the gate level, the exact Riemann solver is used. The gate is active as soon as one of the neighbouring water surface elevations is above the gate level. Similar to the gate boundary condition, the calculation of the specific discharge is

$$q = \mu h_{gate} \sqrt{2gh_0} \quad (2.109)$$

The discharge coefficient  $\mu$  is usually between 0.5 and 0.6 and is either user-defined or calculated dynamically. In the latter case the backwater effect is considered as (Bollrich 2000)

$$\mu = \delta \sqrt{\left[1 - 2 \frac{\delta h_{gate}}{h_0} \left(1 - \frac{\delta h_{gate}}{h_d}\right)\right] - \sqrt{\left[1 - 2 \frac{\delta h_{gate}}{h_0} \left(1 - \frac{\delta h_{gate}}{h_d}\right)\right]^2 + \left(\frac{h_d}{h_0}\right)^2} - 1} \quad (2.110)$$

where  $h_d$  is the water depth downstream of the gate. The definition of the other variables can be found under the description of the gate boundary condition.

As for the momentum, the velocity through the gate is taken into account in both, downstream and upstream direction.

- 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

$$\begin{array}{cc} h^1 & Q^1 \\ \dots & \\ h^n & Q^n \end{array}$$

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).

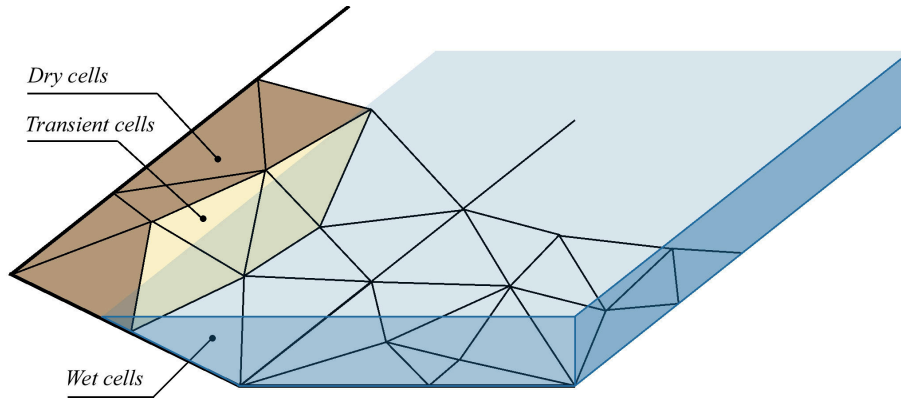
#### 2.2.3.4.4 Moving Boundaries

##### Dry, partial wet and fully wet elements

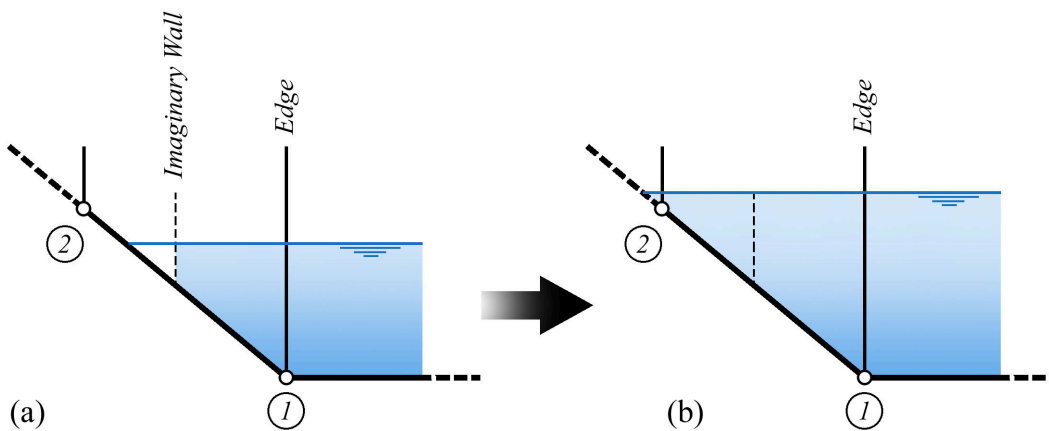
Natural rivers and streams are highly irregular in both plan form and topography. Their boundaries change with the time varying water level. The FV-based model with moving boundary treatment is capable of handling these complex and dynamic flow problems conveniently. The computational domain expands 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):



**Figure 2.36** Schematic representation of a mesh with dry, partial wet and wet elements



**Figure 2.37** Wetting process of an element

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.

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.

$$\nu_x = \frac{(\nu_x h)}{h}$$

$$\nu_y = \frac{(\nu_y h)}{h}$$
(2.111)

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, 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.

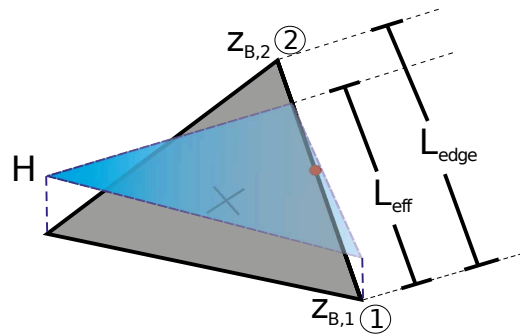
$$L_{eff} = \left( \frac{H - z_{B,1}}{z_{B,2} - z_{B,1}} \right) L_{edge}$$
(2.112)

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 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



**Figure 2.38** Definition of effective length at partially wetted edge

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, 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)

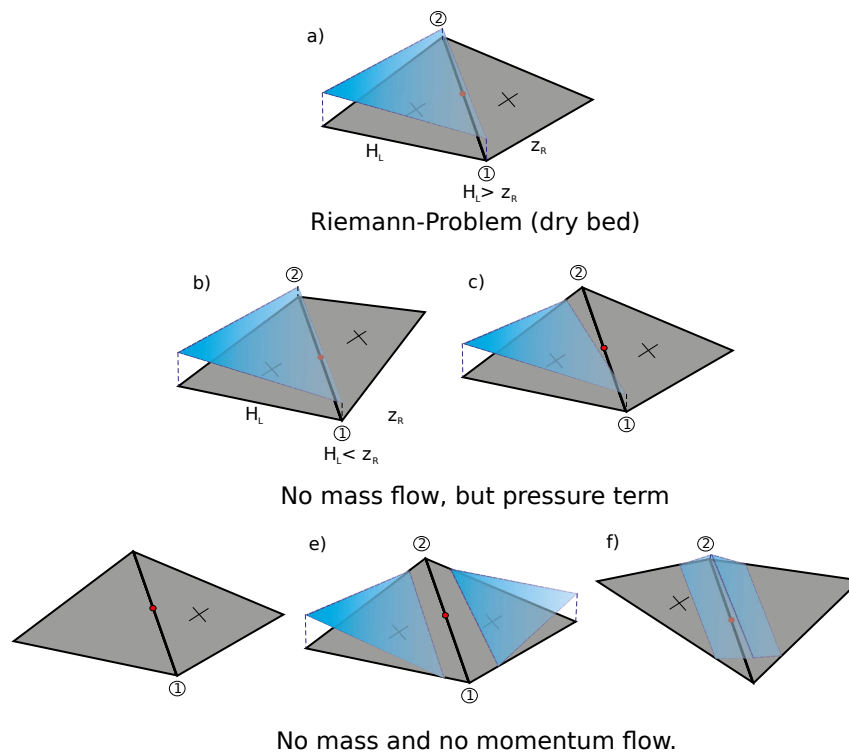


Figure 2.39 Different cases for flux computations over an edge at wet-dry interfaces

## 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

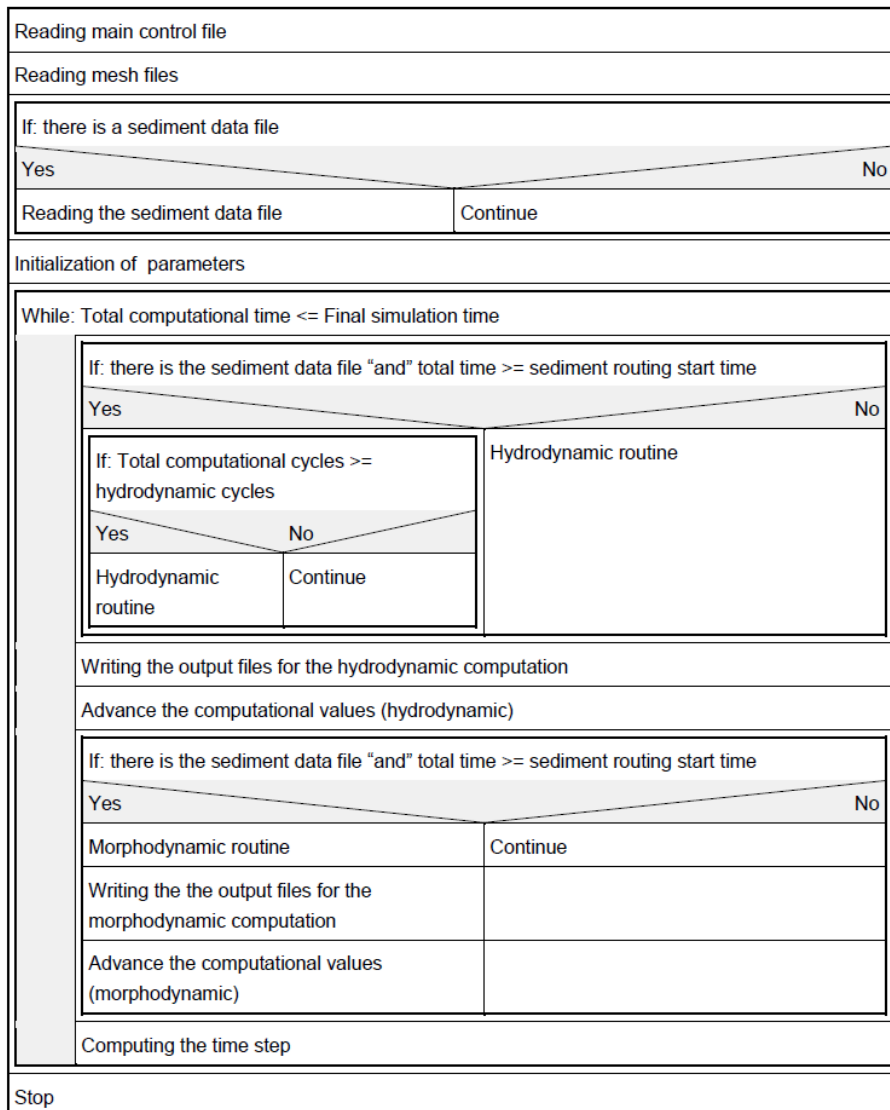
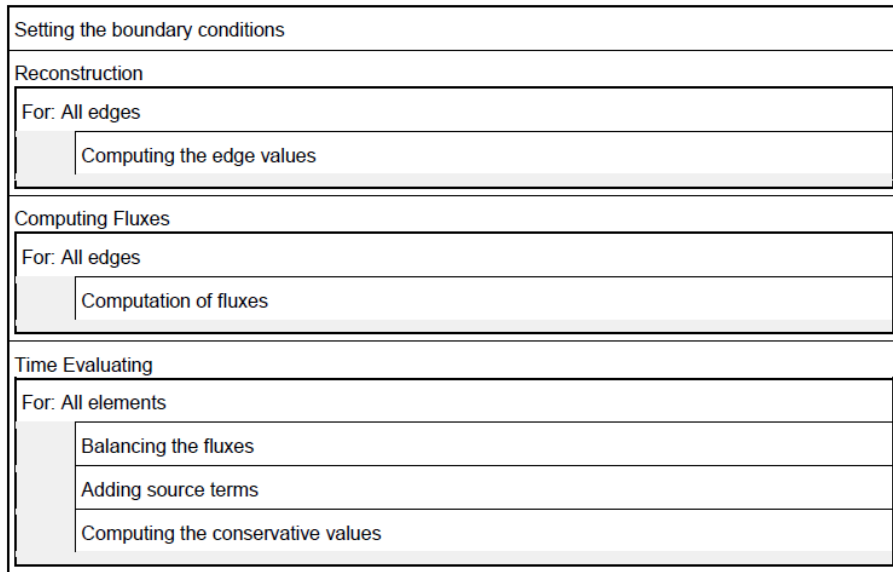
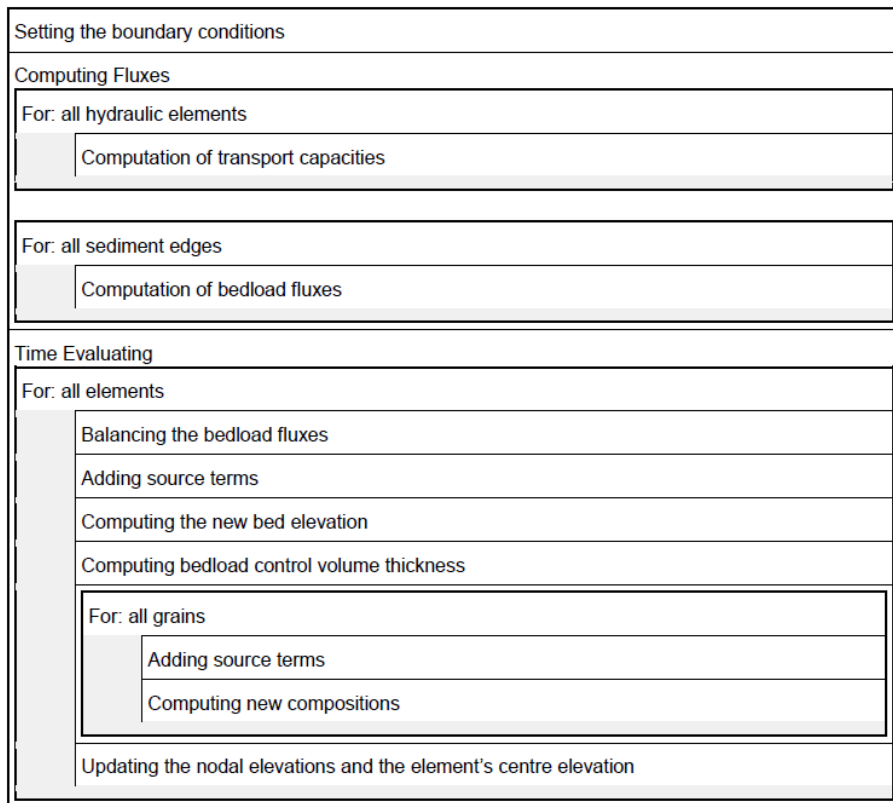


Figure 2.40 The logical flow of data through BASEplane





*Figure 2.41 Data flow through the hydrodynamic routine*



*Figure 2.42 Data flow through the morphodynamic routine*

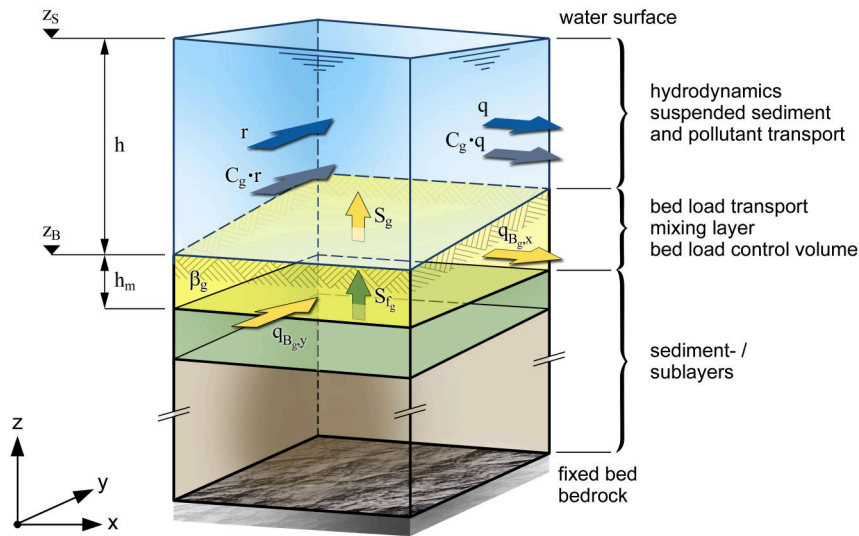


Figure 2.43 Vertical Discretisation of a computational cell

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 decrease with a limitation to account for the situation of an armoured bed (Borah et al. (1982)).

If the bed level increases ( $\Delta z_B > 0$ ):

$$h_m^{n+1} = h_m^n + \Delta z_B \quad (2.113)$$

If the bed level decreases ( $\Delta z_B < 0$ ):

$$h_m = 20\Delta z_B + \frac{d_l}{\sum \beta_{nm}(1-p)} \quad (2.114)$$

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.

$$h_m^{n+1} = factor \cdot d_{90} \quad (2.115)$$

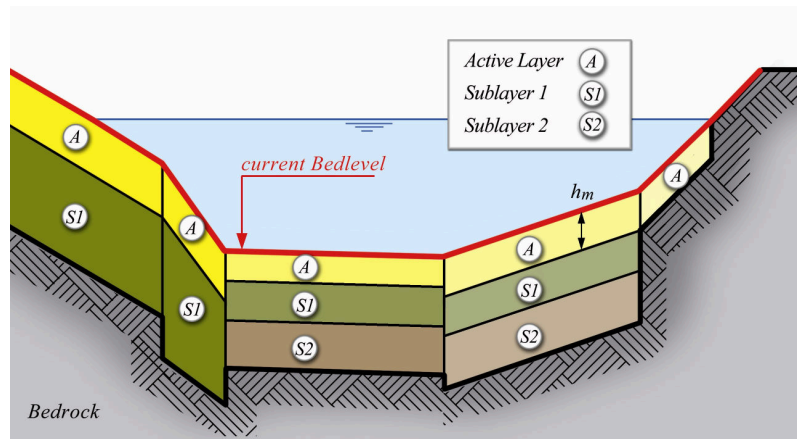


Figure 2.44 Soil Discretisation in a cross section

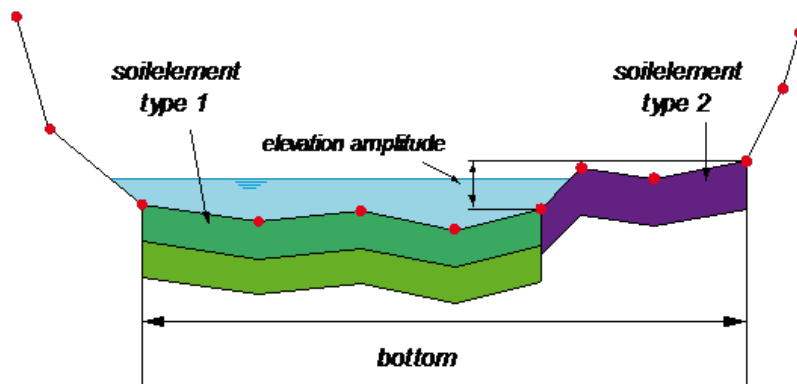


Figure 2.45 Soil Discretisation in a cross section

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 which fraction of the elevation amplitude of the soil has to be below the water level by a

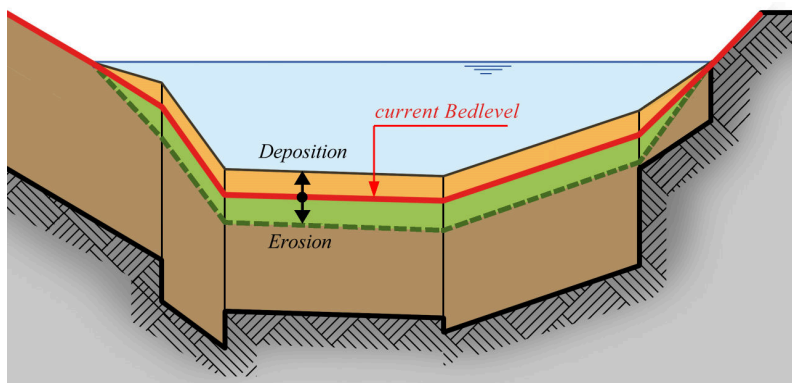


Figure 2.46 Effect of bed load on cross section geometry

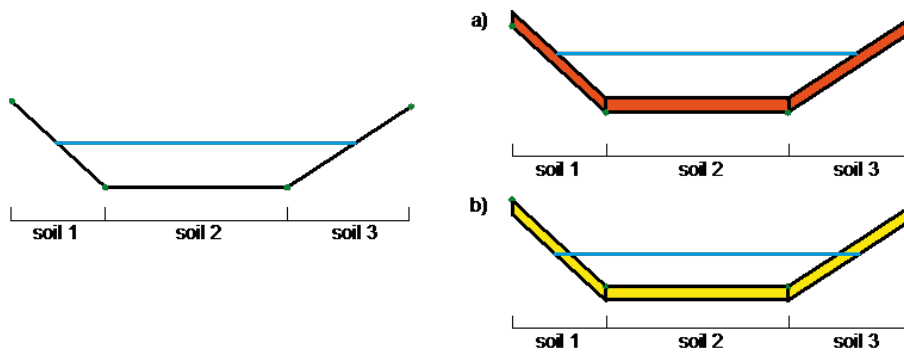


Figure 2.47 Deposition a) and erosion b) due to suspended load with a wetting fraction of 0.9.

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.

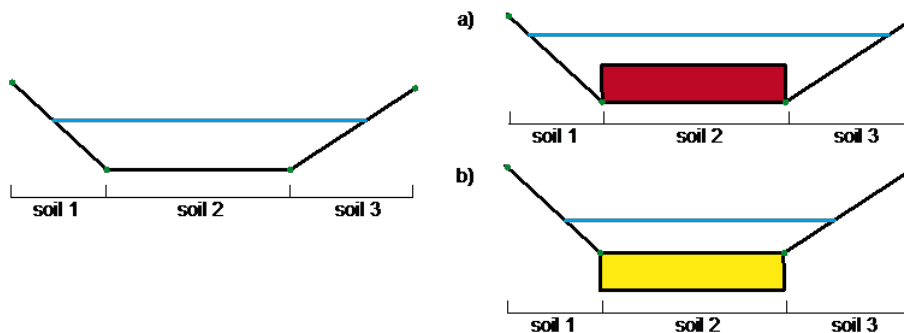


Figure 2.48 Deposition a) and erosion b) due to suspended load with a wetting fraction of 0.1.

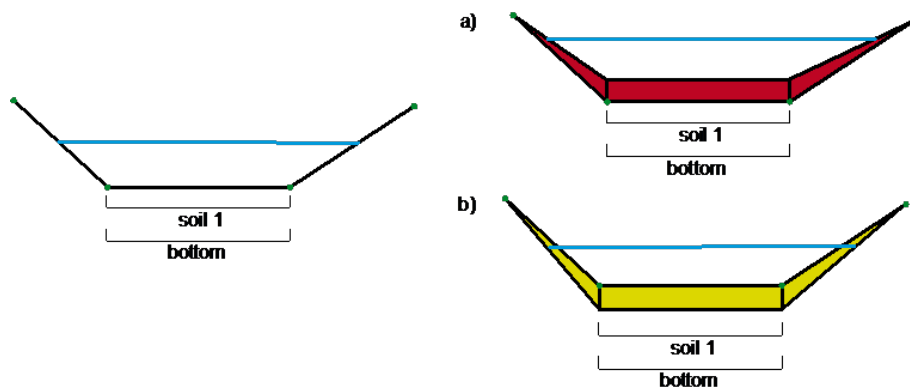


Figure 2.49 Deposition a) and erosion b) due to bed load without cross section points on embankments.

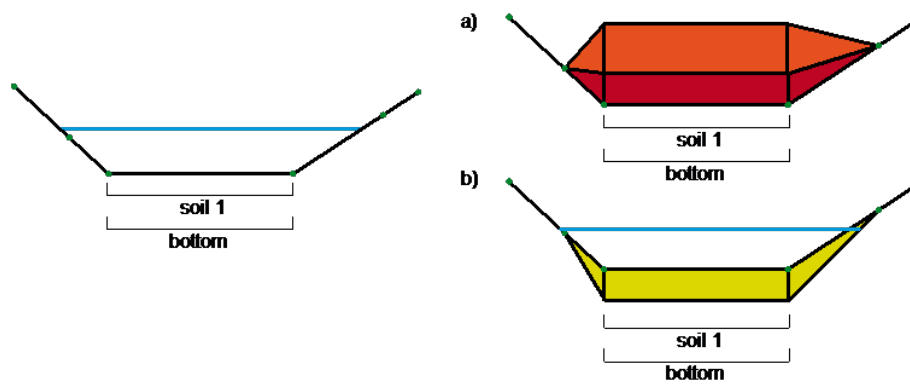


Figure 2.50 Deposition a) and erosion b) due to bed load with cross section points on embankments.

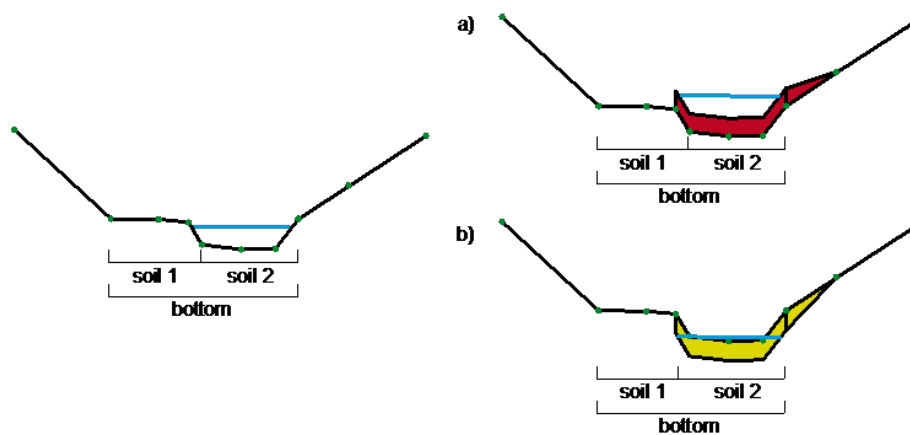


Figure 2.51 Deposition a) and erosion b) due to bed load with 2 soils defined on the bottom.

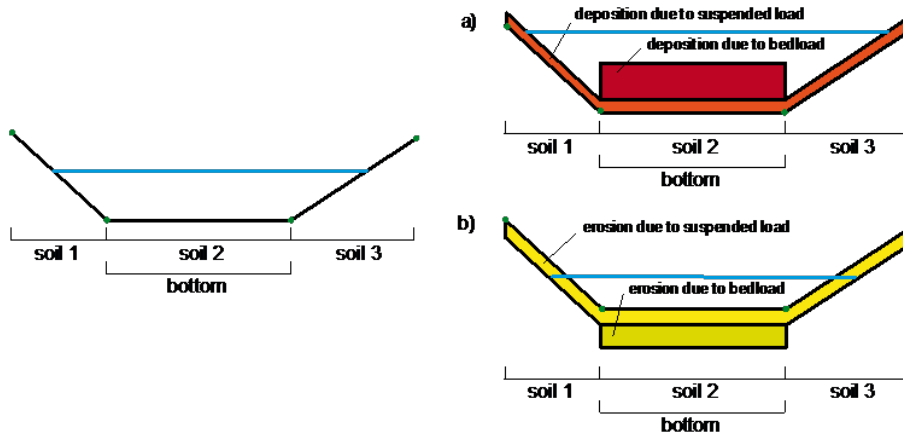


Figure 2.52 Deposition a) and erosion b) due to bed load and suspended load.

### 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:

$$A \frac{\partial C}{\partial t} + Q \frac{\partial C}{\partial x} - \frac{\partial}{\partial x} \left( A \Gamma \frac{\partial C}{\partial x} \right) - S = 0 \quad (2.116)$$

$C$  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. 2.116 is integrated over the element (see Figure 2.7)

$$\int_{x_{iL}}^{x_{iR}} \left( A \frac{\partial C}{\partial t} + Q \frac{\partial C}{\partial x} - \frac{\partial}{\partial x} \left( A \Gamma \frac{\partial C}{\partial x} \right) \right) dx = 0 \quad (2.117)$$

and the different parts are calculated as follows:

$$\int_{x_{iL}}^{x_{iR}} A \frac{\partial C_i}{\partial t} dx = A_i \int_{x_{iL}}^{x_{iR}} \frac{\partial C}{\partial t} dx \approx A_i \frac{\partial C_i}{\partial t} \Delta x_i \approx A_i \frac{C_i^{n+1} - C_i^n}{\Delta t} \Delta x_i \quad (2.118)$$

$$\int_{x_{iL}}^{x_{iR}} Q \frac{\partial C}{\partial x} dx = Q_i \int_{x_{iL}}^{x_{iR}} \frac{\partial C}{\partial x} dx = Q_i (C(x_{iR}) - C(x_{iL})) = (\Phi_{a,iR} - \Phi_{a,iL}) \quad (2.119)$$

$$\int_{x_{iL}}^{x_{iR}} \frac{\partial}{\partial x} \left( A \Gamma \frac{\partial C}{\partial x} \right) dx = \left( A \Gamma \frac{\partial C}{\partial x} \Big|_{x_{iR}} - A \Gamma \frac{\partial C}{\partial x} \Big|_{x_{iL}} \right) = \Phi_{d,iR} - \Phi_{d,iL} \quad (2.120)$$

The concentration at the new time is:

$$C_i^{n+1} = C_i^n - \frac{\Delta t}{\Delta x_i A_i} (\Phi_{a,iR} - \Phi_{a,iL} - \Phi_{d,iR} + \Phi_{d,iL}) \quad (2.121)$$

### 2.3.2.2.2 Computation of Diffusive Flux

The diffusive flux is calculated by finite differences.

$$\Phi_{d,iR} = A_{iR} \Gamma \frac{C_{i+1} - C_i}{x_{i+1} - x_i} \quad (2.122)$$

For the interpolation on the edge of the wetted area, known only in the cross sections, the geometric mean is used:

$$A_{iR} = \sqrt{A_{i+1} A_i} \quad (2.123)$$

If  $\Gamma$  is not given by the user it is calculated as follows:

$$\Gamma = \frac{\sqrt{\nu_L \nu_R}}{\sigma} \quad (2.124)$$

$\sigma$  is generally assumed to be 0.5 (Celik and Rodi 1984).

The eddy viscosity averaged over the depth can be calculated by (Fäh, 1997):

$$\nu = uh\kappa/6 \quad (2.125)$$

### 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 the 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:

$$C_{a,iL} = \frac{(x_{iL} - x_{i-1})(x_{iL} - x_{i-2})}{(x_i - x_{i-1})(x_i - x_{i-2})} C_i + \frac{(x_{iL} - x_{i-2})(x_{iL} - x_i)}{(x_{i-1} - x_{i-2})(x_{i-1} - x_i)} C_{i-1} + \frac{(x_{iR} - x_{i-1})(x_{iR} - x_i)}{(x_{i-2} - x_{i-1})(x_{i-2} - x_i)} C_{i-2} \quad (2.126)$$

More in general the concentration is:

$$C_{a,iL} = \begin{cases} C_i + g_1(C_{i+1} - C_i) + g_2(C_i - C_{i-1}) \rightarrow u > 0 \\ C_{i+1} + g_3(C_i - C_{i+1}) + g_4(C_{i+1} - C_{i+2}) \rightarrow u < 0 \end{cases} \quad (2.127)$$

Between the velocities  $u_i$  and  $u_{i+1}$  the one with the larger absolute value is determinant.

$$g_1 = \frac{(x_{iR} - x_i)(x_{iR} - x_{i-1})}{(x_{i+1} - x_i)(x_{i+1} - x_{i-1})} \quad (2.128)$$

$$g_2 = \frac{(x_{iR} - x_i)(x_{i+1} - x_{iR})}{(x_i - x_{i-1})(x_{i+1} - x_{i-1})} \quad (2.129)$$

$$g_3 = \frac{(x_{iR} - x_{i+1})(x_{iR} - x_{i+2})}{(x_i - x_{i+1})(x_i - x_{i+2})} \quad (2.130)$$

$$g_4 = \frac{(x_{iR} - x_{i+1})(x_i - x_{iR})}{(x_{i+1} - x_{i+2})(x_i - x_{i+2})} \quad (2.131)$$

The QUICK scheme tends to get unstable, 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

$$C_{iR,QUICKEST} = C_{iR,QUICK} - \frac{1}{2}Cr_{iR}(C_{i+1} - C_i) + \frac{1}{8}Cr_{iR}(C_{i+1} - 2C_i + C_{i-1}) \quad (2.132)$$

with

$$Cr_{iR} = \frac{u_{i+1} + u_i}{2} \frac{\Delta t}{\Delta x} \quad (2.133)$$

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. 2.119 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. 2.116 and dividing by the cross section area  $A$ :

$$\frac{\partial C}{\partial t} + \frac{u \partial C}{\partial x} = 0 \quad (2.134)$$

and

$$\frac{C_j^{n+1} - C_j^n}{\Delta t} = -u \frac{C_j^n - C_{j-1}^n}{x_j - x_{j-1}} \quad (2.135)$$

Thus the new concentration is



$$C_j^{n+1} = \left(1 - u \frac{\Delta t}{x_j - x_{j-1}}\right) C_j^n + u \frac{\Delta t}{x_j - x_{j-1}} C_{j-1}^n \quad (2.136)$$

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.

$$Y(Cr) = ACr^3 + BCr^2 + DCr + E \quad (2.137)$$

$$Y(0) = C_j^n ; Y(1) = C_{j-1}^n$$

$$\dot{Y}(0) = \frac{\partial C_j^n}{\partial x} ; \dot{Y}(1) = \frac{\partial C_{j-1}^n}{\partial x}$$

$$C_j^{n+1} = a_1 C_{j-1}^n + a_2 C_j^n + a_3 \frac{\partial C_{j-1}^n}{\partial x} + a_4 \frac{\partial C_j^n}{\partial x} \quad (2.138)$$

$$a_1 = Cr^2(3 - 2Cr) \quad (2.139)$$

$$a_2 = 1 - a_1 \quad (2.140)$$

$$a_3 = Cr^2(1 - Cr)\Delta x \quad (2.141)$$

$$a_4 = -Cr(1 - Cr)^2 \quad (2.142)$$

and

$$\frac{\partial C_j^{n+1}}{\partial x} = b_1 C_{j-1}^n + b_2 C_j^n + b_3 \frac{\partial C_{j-1}^n}{\partial x} + b_4 \frac{\partial C_j^n}{\partial x} \quad (2.143)$$

$$b_1 = 6Cr(Cr - 1)/\Delta x \quad (2.144)$$

$$b_2 = -b_1 \quad (2.145)$$

$$b_3 = Cr(3Cr - 2) \quad (2.146)$$

$$b_4 = (Cr - 1)(3Cr - 1) \quad (2.147)$$

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  $u_{j-1}^n, u_j^n, u_j^{n+1}$ .

$$\bar{u}_j = \frac{1}{2}(u_j^n + u_j^{n+1}) \quad (2.148)$$

$$\bar{u}^n = (u_{j-1}^n \theta) + (1 - \theta)u_{j+1}^n \quad (2.149)$$

with

$$\theta = u_i^n \frac{\Delta t}{x_{j-1} - x_j} \quad (2.150)$$

$$\hat{u} = \frac{1}{2}(\bar{u}_j + \bar{u}^n) \quad (2.151)$$

$$Cr = \hat{u} \frac{\Delta t}{\Delta x} = \frac{u_j^{n+1} + u_j^n}{2 \frac{x_{j-1} - x_j}{\Delta t} - u_{j-1}^n + u_j^n} \quad (2.152)$$

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:

$$\Phi_{iR} = \frac{1}{\Delta t} A_i \int_{t^n}^{t^{n+1}} u(t) C(x_{iR}, t) dt \quad (2.153)$$

Using the invariance property along a characteristic line (Figure 2.53) this equation can be transformed to

$$\Phi_{iR} = \frac{1}{\Delta t} A_i \int_{x_{iL}}^{x_{iR}} C(x, t^n) dx \quad (2.154)$$

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.

$$\tilde{C}_i^n(x) = \begin{cases} U_{iL}^n & \text{if } x \leq x_{i-1/2} + \alpha_i \Delta x_i \\ U_{iR}^n & \text{if } x > x_{i-1/2} + \alpha_i \Delta x_i \end{cases} \quad (2.155)$$

with

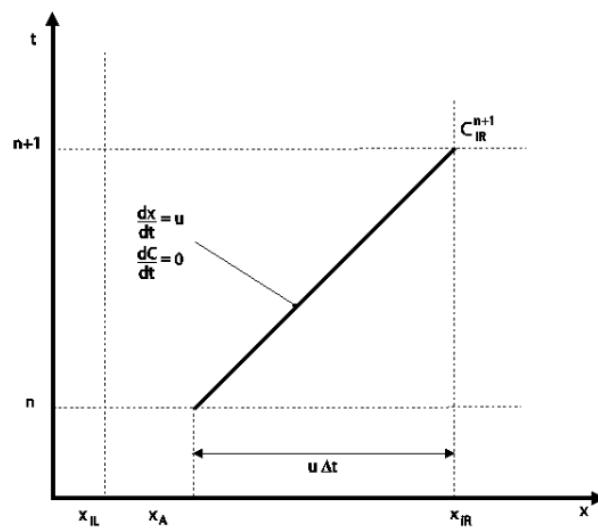


Figure 2.53 Invariance of  $C$  along the characteristic line  $y$

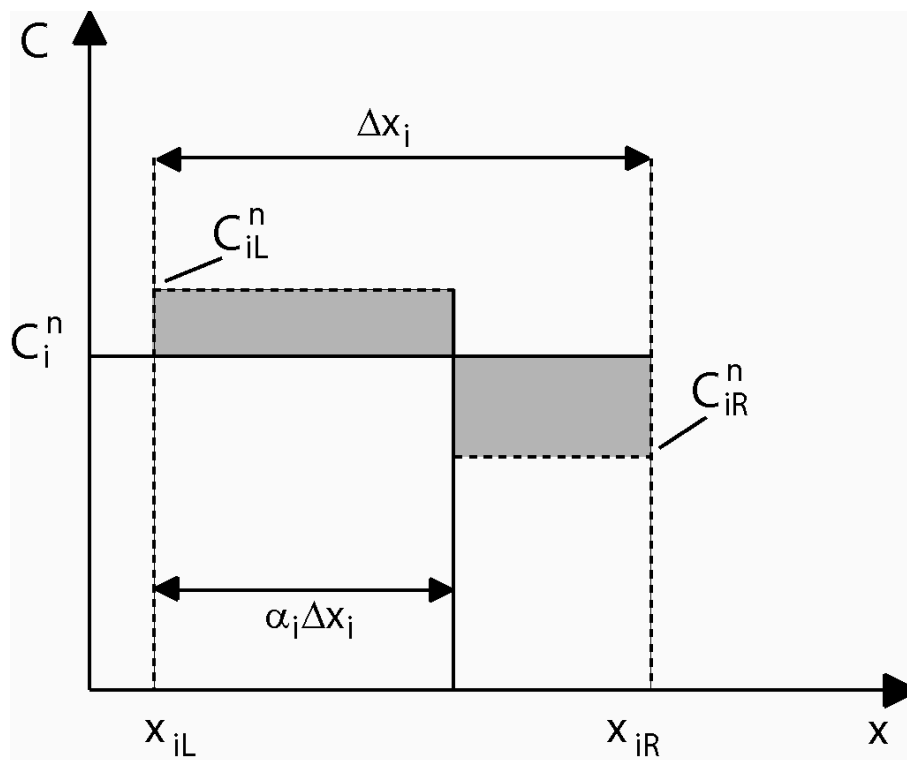


Figure 2.54 Function  $\tilde{C}_i^n(x)$

$$\alpha_i = \frac{C_i^n - C_{iR}^n}{C_{iL}^n - C_{iR}^n} \quad (2.156)$$

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:

$$\left. \begin{array}{l} C_{iL}^n = C_i^n \\ C_{iR}^n = C_i^n \\ \alpha_i = \varepsilon \end{array} \right\}$$

$\varepsilon$  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:

$$h_i = C_{iL}^n \max(x_{i+1/2} - x - (1 - \alpha_i)\Delta x_i, 0) + C_{iR}^n \min(x_{i+1/2} - x, (1 - \alpha_i)\Delta x_i) \quad (2.157)$$

If the velocity is negative respectively:

$$g_i = -C_{iL}^n \min(x - x_{i+1/2}, \alpha_{i+1}\Delta x_{i+1}) - C_{iR}^n \max(x - x_{i+1/2} - \alpha_{i+1}\Delta x_{i+1}, 0) \quad (2.158)$$

The abscissa of the foot of the characteristic is given by  $x_A = x_{i+1/2} - u_{x_{i+1/2}}^n(t)\Delta t$ .

Finally the advective flux is:

$$\Phi_{i+1/2} = \begin{cases} \frac{1}{\Delta t} h_i(x_A) & \text{if } u_{x_{i+1/2}}^n > 0 \\ \frac{1}{\Delta t} g_{i+1}(x_A) & \text{if } u_{x_{i+1/2}}^n < 0 \end{cases} \quad (2.159)$$

Furthermore the new concentrations on the edges have to be prepared for the computations of the next time step:

$$C_{i+1/2}^{n+1} = \begin{cases} C_{iL}^n & \text{if } u_{x_{i+1/2}}^n(t) > 0 \text{ and } Cr_x \geq 1 - \alpha_i \\ C_{iR}^n & \text{if } u_{x_{i+1/2}}^n(t) > 0 \text{ and } Cr_x < 1 - \alpha_i \\ C_{i+1L}^n & \text{if } u_{x_{i+1/2}}^n(t) < 0 \text{ and } Cr_x \geq -\alpha_i \\ C_{i+1R}^n & \text{if } u_{x_{i+1/2}}^n(t) < 0 \text{ and } Cr_x < -\alpha_i \end{cases} \quad (2.160)$$

The required Courant number is

$$Cr_x = \begin{cases} u_x(t) \frac{\Delta t}{\Delta x_i} & u_x(t) > 0 \\ u_x(t) \frac{\Delta t}{\Delta x_{i+1}} & u_x(t) < 0 \end{cases} \quad (2.161)$$

#### 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:

$$(1-p) \frac{\partial z_B}{\partial t} + \left( \sum_{k=1}^{ng} \frac{\partial q_B}{\partial x} + s_g - sl_{B_g} \right) = 0 \quad (2.162)$$

$q_B$  is the sediment flux per unit channel width. Integrating eq. 2.162 over the channel width, hence multiplying everything by the channel width, the following equation is obtained:

$$(1-p) \frac{\partial A_{Sed}}{\partial t} + \left( \sum_{k=1}^{ng} \frac{\partial Q_B}{\partial x} + S_g - Sl_{B_g} \right) = 0 \quad (2.163)$$

The discretisation is effected exactly in the same way as for the hydraulic mass conservation. The eq. 2.163 is integrated over the element (Figure 2.7):

$$\int_{x_{iL}}^{x_{iR}} \left( (1-p) \frac{\partial A_{Sed}}{\partial t} + \left( \sum_{k=1}^{ng} \frac{\partial Q_B}{\partial x} + S_g - Sl_{B_g} \right) \right) dx = 0 \quad (2.164)$$

The parts of the eq. 2.164 are discretized as follows:

$$(1-p) \int_{x_{iL}}^{x_{iR}} \frac{\partial A_{Sed,i}}{\partial t} dx = (1-p) \frac{A_{Sed,i}^{n+1} - A_{Sed,i}^n}{\Delta t} \Delta x \quad (2.165)$$

$$\int_{x_{iL}}^{x_{iR}} \sum_{k=1}^{ng} \frac{Q_{B,i}}{\partial x} dx = \sum_{k=1}^{ng} Q_B(x_{iR}) - \sum_{k=1}^{ng} Q_B(x_{iL}) = \Phi_{B,iR} - \Phi_{B,iL} \quad (2.166)$$

$$\int_{x_{iL}}^{x_{iR}} \sum_{k=1}^{ng} (S_g - Sl_{B_g}) dx = \sum_{k=1}^{ng} S_g - \sum_{k=1}^{ng} Sl_{B_g} \quad (2.167)$$

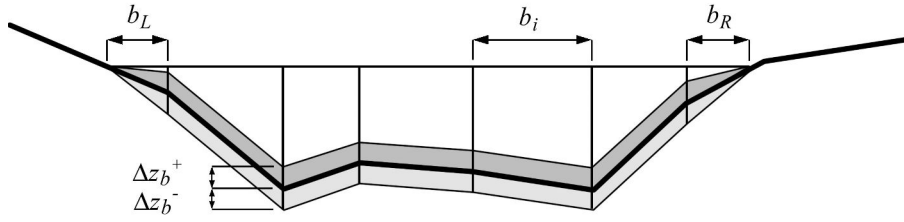
$\Phi_{B,iL}$  and  $\Phi_{B,iR}$  are the bed load fluxes through the west and east side of the cell. Their determination will be discussed later (Section 2.3.2.3.1).

The change of the sediment area is thus calculated by:

$$\Delta A_{Sed} = A_{Sedi}^{n+1} - A_{Sedi}^n = \frac{\Delta t}{\Delta x_i} (\Phi_{B,iR} - \Phi_{B,iL}) - \frac{\Delta t}{\Delta x_i} \left( \sum_{k=1}^{ng} S_g - \sum_{k=1}^{ng} Sl_{B_g} \right) \quad (2.168)$$

As the result of the sediment balance is an area, the deposition or erosion height  $\Delta z_b$  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

$$\Delta z_B = \frac{\Delta A_{Sed}}{\frac{b_r}{2} + \frac{b_l}{2} + \sum b_i} \quad (2.169)$$

As the sediments transported by bed load can be deposited only on the bed bottom, whilst the sediments transported by suspended load can be deposited on the whole wetted section, two separate values of  $\Delta A_{Sed}$  and  $\Delta z_B$  are calculated for the two processes by splitting eq. 2.168 in two parts.

The change of bed level due to bed load is:

$$\Delta A_{Sed,bl} = \frac{\Delta t}{\Delta x_i} (\Phi_{B,iR} - \Phi_{B,iL}) - \frac{\Delta t}{\Delta x_i} \left( \sum_{k=1}^{ng} Sl_{B_g} \right) \quad (2.170)$$

The change of bed level due to suspended load accordingly:

$$\Delta A_{Sed,bl} = -\frac{\Delta t}{\Delta x_i} \left( \sum_{k=1}^{ng} S_g - \sum_{k=1}^{ng} Sl_g \right) \quad (2.171)$$

### 2.3.2.2.5 Bed material sorting equation

For the 1-D computation the bed material sorting equation computation is:

$$(1-p) \frac{\partial}{\partial t} (\beta_g h_B) + \frac{\partial q_{B_g}}{\partial x} + s_g - s f_g - sl_{B_g} = 0 \quad (2.172)$$

Considering the whole width of the cross section and introducing an active layer area  $A_B$ , eq. 2.172 becomes:

$$(1-p) \frac{\partial}{\partial t} (\beta_g A_B) + \frac{\partial Q_{B_g}}{\partial x} + S_g - S f_g - Sl_{B_g} = 0 \quad (2.173)$$

The eq. 2.173 is integrated over the length of the control volume:

$$\int_{x_{iL}}^{x_{iR}} \left( (1-p) \frac{\partial}{\partial t} (\beta_g A_B) + \frac{\partial Q_{B_g}}{\partial x} + S_g - S f_g - Sl_{B_g} \right) dx = 0 \quad (2.174)$$

The different parts are discretized as follows:

$$(1-p) \int_{x_{iL}}^{x_{iR}} \frac{\partial}{\partial t} (\beta_g A_B) dx = (1-p) \frac{\beta_g^{n+1} A_B^{n+1} - \beta_g^n A_B^n}{\Delta t} \Delta x \quad (2.175)$$

$$\int_{x_{iL}}^{x_{iR}} \frac{\partial Q_{B_g}}{\partial x} dx = Q_B(x_{iR}) - Q_B(x_{iL}) = \Phi_{g,iR} - \Phi_{g,iL} \quad (2.176)$$

Then the new  $\beta_g$  at time  $n+1$  can be calculated for every element  $i$  by:

$$\beta_{gi}^{n+1} = \left( (1-p) \beta_{gi}^n A_{B_i}^n - \frac{\Delta t}{\Delta x_i} (\Phi_{iR}^k - \Phi_{iL}^k) - (S_g - S_{f_g} - S_{l_{B_g}}) \frac{\Delta t}{\Delta x_i} \right) / ((1-p) A_{B_i}^{n+1}) \quad (2.177)$$

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.

$$\beta_{gi,bl}^{n+1} = \left( (1-p) \beta_{gi}^n A_{B_i}^n - \frac{\Delta t}{\Delta x_i} (\Phi_{iR}^k - \Phi_{iL}^k) - (-S_{f_g} - S_{l_{B_g}}) \frac{\Delta t}{\Delta x_i} \right) / ((1-p) A_{B_i}^{n+1}) \quad (2.178)$$

and

$$\beta_{gi,susp}^{n+1} = \left( (1-p) \beta_{gi}^n A_{B_i}^n - (-S_{f_g} - S_{l_g}) \frac{\Delta t}{\Delta x_i} \right) / ((1-p) A_{B_i}^{n+1}) \quad (2.179)$$

### 2.3.2.2.6 Interpolation

To solve the eq. 2.166 and eq. 2.176 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 ( $\theta$ ):

$$\Phi_{B,iL} = (\theta) Q_{B,i-1} + (1-\theta) Q_{B,i} \quad (2.180)$$

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 :

$$\begin{aligned} S_{g,L} &= f(C_{iL}^n) \\ S_{g,R} &= f(C_{iR}^n) \end{aligned}$$

The volume exchanged with the bottom during  $\Delta t$  is given by the sum of the exchange on the left and the exchange on the right:

$$E_{L1} = \alpha S_{g,L} B \Delta x \Delta t \quad (2.181)$$

$$E_{L2} = \alpha_2 S_{g,L} B \Delta x \Delta t / 2 \quad (2.182)$$

$$E_L = E_{L1} + E_{L2} \quad (2.183)$$

$$E_{R1} = \alpha_2 S_{g,R} B \Delta x \Delta t / 2 \quad (2.184)$$

$$E_{R2} = \alpha_3 S_{g,R} B \Delta x \Delta t \quad (2.185)$$

$$E_R = E_{R1} + E_{R2} \quad (2.186)$$

Where  $\alpha$  is defined like in eq. 2.156,  $\alpha_2 = \min(Cr_x, 1-\alpha)$  and  $\alpha_3 = \max(1-\alpha-Cr_x, 0)$

The final mean source term is then:

$$S_g = \frac{(E_L + E_R)}{B \Delta t \Delta x} \quad (2.187)$$

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:



$$C_{iL}^{n+1} = \frac{C_{iL}^n A_i \alpha \Delta x + E_{L1}}{A_i \alpha \Delta x} \quad (2.188)$$

$$C_{iR}^{n+1} = \frac{C_{iR}^n A_i \alpha_3 \Delta x + E_{R2}}{A_i \alpha_3 \Delta x} \quad (2.189)$$

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.

$$\begin{aligned} \varphi_k &= 0 & \text{if} & \left(\frac{u_*}{w_k}\right) < 0.4 \\ \varphi_k &= 0.25 + 0.325 \ln\left(\frac{u_*}{w_k}\right) & \text{if} & 0.4 \leq \left(\frac{u_*}{w_k}\right) \leq 10 \\ \varphi_k &= 1 & \text{if} & \left(\frac{u_*}{w_k}\right) > 10 \end{aligned} \quad (2.190)$$

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:

$$Q_{B_g} = (1 - \varphi_g) Q_{B_g} \quad (2.191)$$

$$S_g = w_g (\varphi_g \beta_g C_{e_g} - C_{d_g}) \quad (2.192)$$

#### 2.3.2.3.5 Abrasion

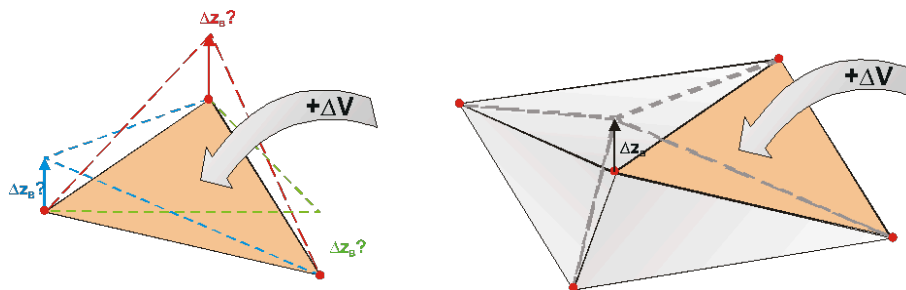
As BASEMENT always works with volumes, the abrasion after Sternberg (eq. 1.118 in the Mathematical Models section) is applied as follows:

$$V(x) = V_0 e^{-c(x-x_0)} \quad (2.193)$$

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.



**Figure 2.56** Schematic illustration of problems related to the update of bed elevations

### 2.3.2.4 Solution Procedure

The solution procedure for the one dimensional sediment transport is described in chapter Section 2.2.2.5.

## 2.3.3 Two Dimensional Sediment Transport

### 2.3.3.1 Spatial Discretisation

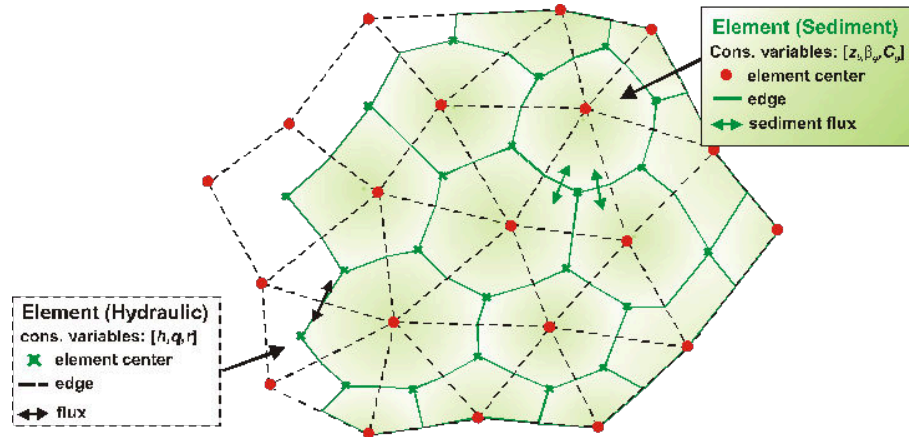
The finite volume method is applied to discretise the morphodynamic equations, slightly different from that of the hydrodynamic section. In the hydrodynamic discretisation a cell-centered approach is applied (see Figure 2.2). Thereby, the bed elevations are defined in the cell vertexes (nodes) of each 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.

Applying the same Discretisation approach for the sediment transport leads to several problematic aspects, which can be summarized briefly as follows.

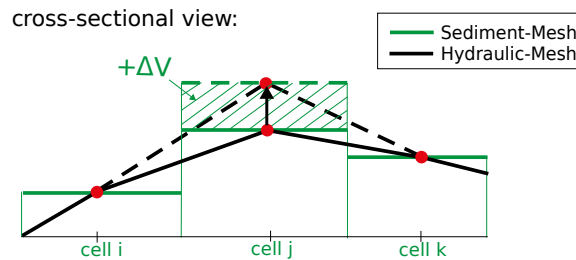
- The change of a cell's sediment volume  $\Delta V$  would have to be distributed on all nodes of the cell, where the bed elevations are defined. But it is not obvious by which criteria the sediment volume must be divided upon these nodes (see left part of Figure 2.56).
- If a nodal elevation would 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



**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  $\Delta V$  do not affect the neighboring cells' sediment volumes.

procedure results in the generation of median dual cells. Following this dual mesh approach all conservative variables of the sediment transport ( $z_b, \beta_g, C_g$ ) 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.

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. 1.87) and applying the FVM, it can be written in the following integral form

$$(1 - p) \int_{\Omega} \frac{\partial z_B}{\partial t} d\Omega + \int_{\Omega} \sum_{g=1}^{ng} \left( \frac{\partial q_{B_g,x}}{\partial x} + \frac{\partial q_{B_g,y}}{\partial y} \right) d\Omega = \int_{\Omega} \sum_{g=1}^{ng} (s_{l_g} - s_{B_g}) d\Omega \quad (2.194)$$

In which  $\Omega$  is the same computational area as defined in hydrodynamic model (Figure 2.27).

Using the Gauss' theory and assuming  $\partial z_B / \partial t$  is constant over the element, one obtains

$$(1-p) \frac{z_B^{n+1} - z_B^n}{\Delta t} + \frac{1}{\Omega} \sum_{g=1}^{ng} \int_{\partial\Omega} (q_{B_g,x} n_x + q_{B_g,y} n_y) dl = \frac{1}{\Omega} \sum_{g=1}^{ng} (Sl_g - S_{Bg}) \quad (2.195)$$

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

$$\vec{q}_{B_{res}} = \vec{q}_B + \vec{q}_{B_{lateral}} = q_B \begin{pmatrix} \cos \theta \\ \sin \theta \end{pmatrix} + q_{B_{lateral}} \begin{pmatrix} \sin \theta \\ -\cos \theta \end{pmatrix} \quad (2.196)$$

where  $\theta$  is the angle between the velocity vector and the x-axis.

#### Computation of bed load flux

The bed load transport capacity  $\vec{q}_B$  and the lateral transport  $\vec{q}_{B_{lateral}}$  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 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.

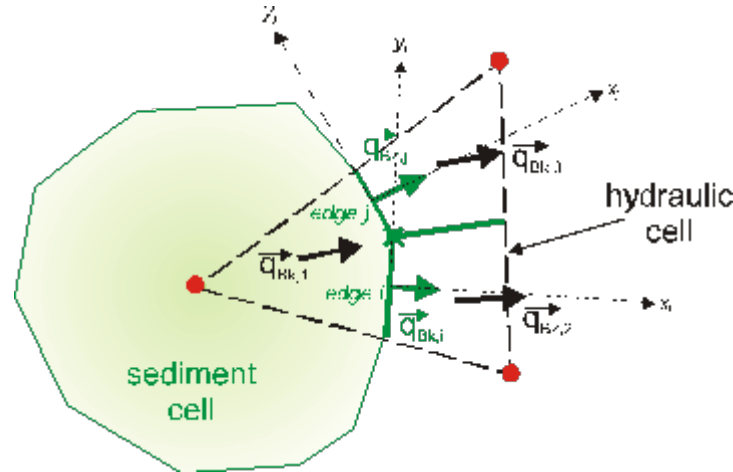


Figure 2.59 Determination of bed load flux over the sediment cell edges  $i, j$

Finally, the flux over the sediment cell edge is determined from the calculated transport capacities to its left and right sediment elements as

$$q_{B,edge} = [(\theta_{up}) \vec{q}_{B,res,L} + (1 - \theta_{up}) \vec{q}_{B,res,R}] \vec{n}_{edge} \quad (2.197)$$

where  $\theta_{up}$  is the upwind factor and  $\vec{n}_{edge}$  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  $V_{sed,k,g}$ .

$$\omega_{k,g} = \begin{cases} \frac{\sum_{j=1}^{ng} Flux_{out,j,g} - V_{sed,k,g}}{\sum_{j=1}^{ng} Flux_{out,j,g}} & \text{if } V_{sed,k,g} < \sum_{j=1}^{ng} Flux_{out,j,g} \\ 0 & \text{if } V_{sed,k,g} \geq \sum_{j=1}^{ng} Flux_{out,j,g} \end{cases} \quad (2.198)$$

$$Flux_{out,k,g}^{corr} = (1 - \omega_{k,g}) Flux_{out,k,g}$$

#### 2.3.3.2.4 Bed Material Sorting Equation

The global bed material conservation eq. 2.195 has to be solved first, because its results are needed to solve the bed material sorting equation. The bed material sorting equation (eq. 1.79) is also discretized using FVM. The discretized form is

$$(1 - p) \frac{(h_m \beta_g)^{n+1} - (h_m \beta_g)^n}{\Delta t} + \frac{1}{\Omega} \int_{\partial\Omega} (q_{B_g,x} n_x + q_{B_g,y} n_y) dl + \frac{1}{\Omega} S_g - \frac{1}{\Omega} Sl_{B_g}^n - sf_g^* = 0 \quad (2.199)$$

The bed load control volume thickness  $h_m$  has to be determined before the new values of fractions are calculated through the eq. 2.199. 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 fractions  $\beta_g^{n+1}$  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:

$$\frac{\partial}{\partial t} Ch + \frac{\partial}{\partial x} \left( Cq - h\Gamma \frac{\partial C}{\partial x} \right) + \frac{\partial}{\partial y} \left( Cr - h\Gamma \frac{\partial C}{\partial y} \right) = 0 \quad (2.200)$$

$C$  is the concentration of transported suspended material averaged over flow depth.

The eq. 2.116 also can be written as follows.

$$(Ch)_t + \nabla(Cq, Cr) - \nabla(h\Gamma C_x, h\Gamma C_y) = 0 \quad (2.201)$$

It is integrated over the area of the hydraulic element  $\Omega$  :

$$\int_{\Omega} ((Ch)_t + \nabla(Cq, Cr) - \nabla(h\Gamma C_x, h\Gamma C_y)) d\Omega = 0 \quad (2.202)$$

Applying the Gauss theorem, eq. 2.202 becomes:

$$\int_{\Omega} (Ch)_t d\Omega + \oint_{\partial\Omega} (Cq, Cr) \vec{n} dl - \oint_{\partial\Omega} (h\Gamma C_x, h\Gamma C_y) \vec{n} dl = 0 \quad (2.203)$$

the different parts are calculated as follows:

$$\int_{\Omega} (C_i h_i)_t d\Omega = h_i \int_{\Omega} \frac{\partial C_i}{\partial t} d\Omega \approx h_i \frac{\partial C_i}{\partial t} \Omega \approx h_i \frac{C_i^{n+1} - C_i^n}{\Delta t} \Omega \quad (2.204)$$

the advective flux is:

$$\oint_{\partial\Omega} (Cq, Cr) \vec{n} dl = \sum_j C_j (q_j, r_j) \vec{n}_j l_j = \sum_j \Phi_{a,j} \quad (2.205)$$

and the diffusive flux:

$$\oint_{\partial\Omega} (h\Gamma C_x, h\Gamma C_y) \vec{n} dl = \sum_j h_j \Gamma (C_x, C_y) \vec{n}_j l_j = \sum_j \Phi_{d,j} \quad (2.206)$$

The concentration at the new time  $n + 1$  is:

$$C_i^{n+1} = C_i^n - \frac{\Delta t}{\Omega_i h_i} \sum_j (\Phi_{a,j} - \Phi_{d,j}) \quad (2.207)$$

### 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:

$$\Phi_{d,j} = h_j \Gamma \begin{pmatrix} \frac{\partial C_j}{\partial x} \\ \frac{\partial C_j}{\partial y} \end{pmatrix} \vec{n}_j l_j$$

The derivatives of the concentration are given by the mean value of the derivatives in the left and the right element of the edge:

$$\begin{pmatrix} \frac{\partial C_j}{\partial x} \\ \frac{\partial C_j}{\partial y} \end{pmatrix} = \frac{1}{2} \left[ \begin{pmatrix} \frac{\partial C_L}{\partial x} \\ \frac{\partial C_L}{\partial y} \end{pmatrix} + \begin{pmatrix} \frac{\partial C_R}{\partial x} \\ \frac{\partial C_R}{\partial y} \end{pmatrix} \right] \quad (2.208)$$

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:

$$\left( \frac{\partial C}{\partial x} \right)_i = \frac{1}{\Omega} \int_{\Omega} \frac{\partial C}{\partial x} d\Omega \approx \frac{1}{\Omega} \sum_j C_j \Delta y_j \quad (2.209)$$

$$\left( \frac{\partial C}{\partial y} \right)_i = \frac{1}{\Omega} \int_{\Omega} \frac{\partial C}{\partial y} d\Omega \approx \frac{1}{\Omega} \sum_j C_j \Delta x_j \quad (2.210)$$

### 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. 2.205. The hydraulic discharges  $q$  and  $r$  over the edge are known from the hydraulic computation (eq. 2.59). 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:

$$\Phi_{a,j} = C_j \begin{pmatrix} q_j \\ r_j \end{pmatrix} \vec{n}_j l_j$$

The discharge of water over the edge  $\begin{pmatrix} q_j \\ r_j \end{pmatrix} \vec{n}_j$  is known from the hydraulic computation (eq. 2.59). 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:

$$\Phi_j = \frac{1}{\Delta t} h_j l_j \int_{t^n}^{t^{n+1}} u_i(t) C_j(t) dt \quad (2.211)$$

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:

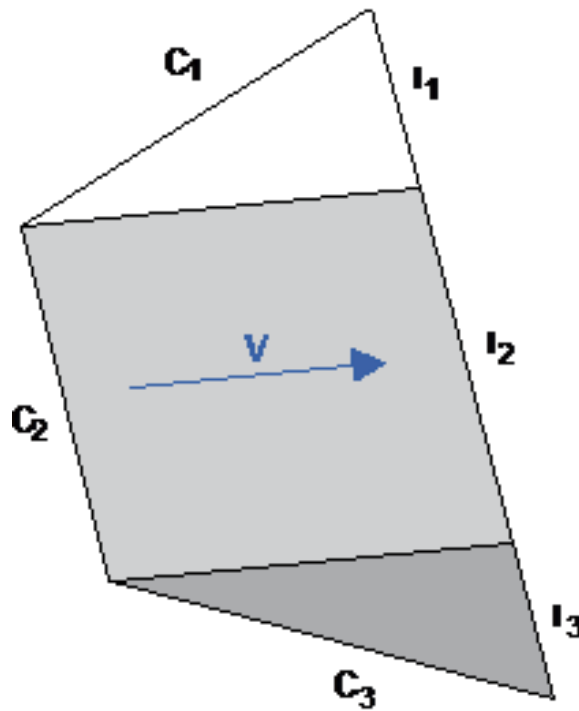
$$\Phi_j = \frac{1}{\Delta t} q_j l_j \int_{t^n}^{t^{n+1}} C_j(t) dt \quad (2.212)$$

or:

$$\Phi_j = q_j l_j C_j(t) \quad (2.213)$$

Now the concentration  $C_i(t)$  has 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





**Figure 2.60** Division of element and edge  $j$  in segments which receive the water from different upstream edges.

can be up to 3 different edges. For this reason the edge  $j$  is divided in  $k$  segments for which the fluxes  $\Phi_k$  are computed separately and then summed up:

$$\Phi_j = \sum_1^k \Phi_k \quad (2.214)$$

This procedure is illustrated in figure Figure 2.60.

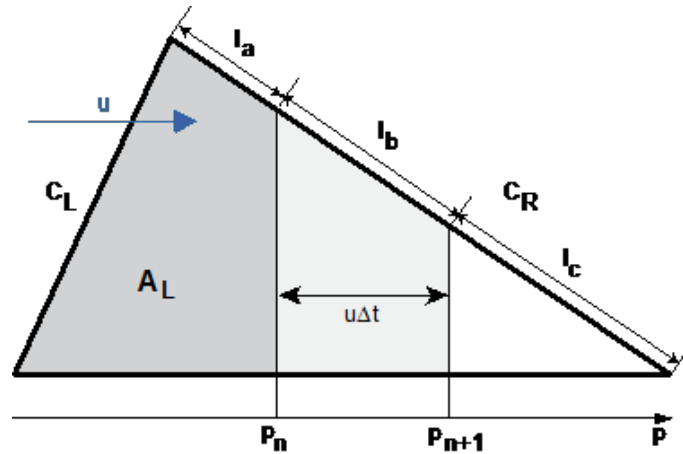
From now on only a segment concerned by one upstream edge will be considered. The concentration on the upstream edge is  $C_L$  and the one on the downstream edge  $C_R$ . The part of the segment which is over flown with concentration  $C_L$  or  $C_R$  changes while the concentration front between the two concentrations advances.

The position of the front is determined in an analogous way to 1D with eq. 2.156. But instead of representing the fraction of the distance behind the front,  $\alpha$  now represents the fraction of the area. The area which is behind the front therefore becomes:

$$A_L = \alpha_i A_i \quad (2.215)$$

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 holds:



**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.

$$\Phi_k = (C_L l_a + C_R l_c + (0.5C_L + 0.5C_R)l_b)q \quad (2.216)$$

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

$$\beta_g^{n+1/2} = \frac{1}{h_m^{n+1}} \left( (h_m \beta_g)^n - \frac{\Delta t}{(1-p)\Omega} \int_{\partial\Omega} (q_{B_g,x} n_x + q_{B_g,y} n_y)^n dl + \frac{\Delta t}{(1-p)\Omega} Sl_{B_g}^n \right) \quad (2.217)$$

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  $\beta_g^{n+1}$  by adding the  $sf_g(\beta_g^{n+1/2})$  source term as

$$\beta_g^{n+1} = \beta_g^{n+1/2} + \frac{\Delta t}{h_m^{n+1}(1-p)} sf_g \quad (2.218)$$

And the  $sf_g$  source term, which describes the material which enters or leaves the bedload control volume, is finally discretized as given below.

$$sf_g = -(1-p) \frac{\beta_g^{n+1/2}(z_F^{n+1} - z_{sub}) - \beta_g^n(z_F^n - z_{sub})}{\Delta t} \quad (2.219)$$

In the calculation of this expression for the  $sf_g$  term, cases of erosion and deposition must be considered separately.

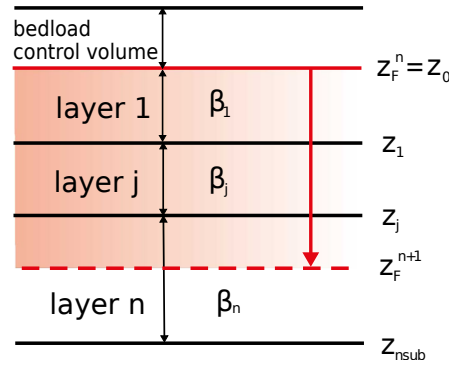
#### Erosion

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:

$$sf_g = -(1-p)\beta_g \frac{(z_F^{n+1} - z_F^n)}{\Delta t} \quad (2.220)$$

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



**Figure 2.62** Definition of mixed composition  $\beta_{g_{mix}}$  for the eroded volume (red)

$$\beta_{g_{mix}} = \frac{1}{z_F^n - z_F^{n+1}} \sum_{j=1}^{nsub} \left[ \beta_{g_j} (z_{j-1} - \max(z_j, z_F^{n+1})) \right] \quad (2.221)$$

### 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:

$$sf_g = -(1-p)\beta_g^{n+1/2} \frac{(z_F^{n+1} - z_F^n)}{\Delta t} \quad (2.222)$$

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  $\gamma_{dry}$ , for partially saturated material at the bank which is not over flown. This angle may largely exceed the material's angle of repose  $\gamma_{rep}$  in small-grained materials due to stabilizing effects of negative pore pressures.
2. A critical failure angle  $\gamma_{wet}$ , 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,  $\gamma_{dep}$ , 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  $\gamma_{wet}$  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:

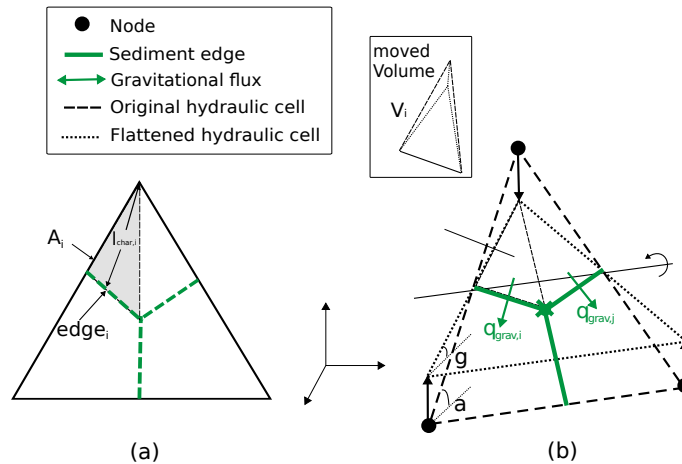
- (1) In a first step the steepness of a hydraulic cell's slope is used as an indicator if a slope failure has to be assumed. The appropriate critical failure angle is selected depending upon the water elevation. It is checked if the cell is wetted or dry and if the present sediment in the control volume 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 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:

$$V_i = \frac{1}{3} A_i l_{char,i} (\mathbf{n}_i \mathbf{S}_i - \mathbf{n}_i \mathbf{S}_{crit,i})$$

with  $A_i$  = area above the sediment edge,  $l_{char,I}$  = characteristic length,  $\mathbf{S}_i$  = slope vector of the cell,  $\mathbf{S}_{crit,i}$  = critical slope vector,  $\mathbf{n}_i$  = normal vector of sediment edge and  $h_i$  = height of pyramidal volume. If  $\gamma_{dep}$  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  $\Delta t$ . This flux leads to mass transport from the upper to the lower sediment cell which results in a flattening of the cell's slope.

$$q_{B_g, ngrav} = \frac{V_i}{\Delta t}$$



**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)

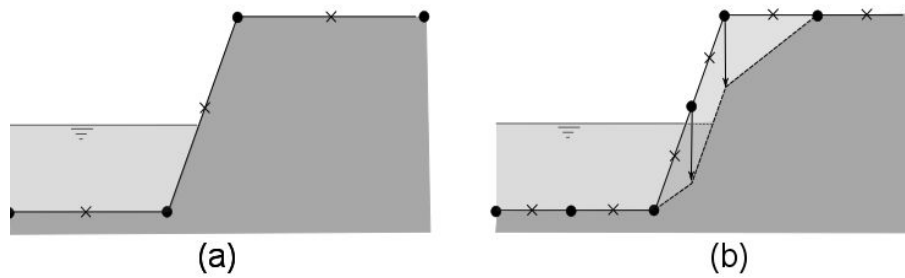
- (4) The balancing of the gravitational fluxes and the determination of the new soil elevations  $z_B$  is achieved by applying and solving the Exner equation using the same numerical approaches as outlined for the bed load transport. This procedure ensures that fixed bed elevations or surface armouring 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 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



**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.

angle  $\gamma_{wet}$  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  $\gamma_{wet}$  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. 1.85 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  $\Delta t_{seq}$  (overall time step) consists of one hydraulic time step  $\Delta t_h$  and a resulting number of time steps  $\Delta t_s$  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 determination of the time step for the mobile bed equations depends on the model being used. For suspension transport, a time step size  $\Delta t_s$  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

$$U_i^{n+1} = U_i^n + RES(U)$$



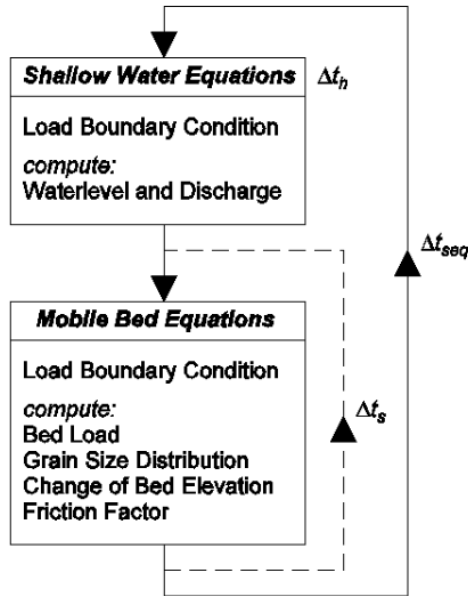


Figure 2.65 Uncoupled asynchronous solution procedure consisting of sequential steps

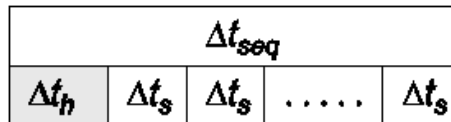


Figure 2.66 Composition of the given overall calculation time step  $\Delta t_{seq}$

$$z_B^{n+1} = z_B^n + RES(U, d)$$

$$\beta_g^{n+1} = \beta_g^n + RES(U, d_g, hm)$$

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:

$$CFL = \frac{(\sqrt{u^2 + v^2} + c)\Delta t}{L} \leq 1 \tag{2.223}$$

Where  $L$  is the length of an edge with corresponding velocities of the element  $u, v$  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  $c \gg c_3$  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  $\Delta t_s$  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  $\theta$ -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  $\theta$  is between 0.5 and 1.

### 2.4.3.2.1 Continuity Equation

The integration of the continuity equation

$$\int_{x_{iw}}^{x_{ie}} \left( \frac{\partial A_{si}}{\partial t} + \frac{\partial Q}{\partial x} - q \right) dx = F_i^n + F_i^\Phi + F_i^q = 0$$

gives the following integral terms.

$$F_i^n = \int_{x_{iw}}^{x_{ie}} \frac{\partial A_{si}}{\partial t} dx$$

$$F_i^\Phi = \int_{x_{iw}}^{x_{ie}} \frac{\partial Q}{\partial x} dx$$

$$F_i^q = - \int_{x_{iw}}^{x_{ie}} q dx$$

Applying the  $\theta$ -method:

$$F_i^{n+1} = F_i^n + \theta(F_i^\Phi + F_i^q) + (1 - \theta)(F_i^{\Phi 0} + F_i^{q0}) \quad (2.224)$$

### 2.4.3.2.2 Momentum Equation

The integration of the momentum equation:

$$\int_{x_{iw}}^{x_{ie}} \left( \frac{\partial Q}{\partial t} + \frac{\partial}{\partial x} \left( \beta \frac{Q^2}{A_{red}} \right) + g A_{red} \frac{\partial z}{\partial x} + g A_{red} S_f \right) dx = G_i^n + G_i^\Phi + G_i^{sz} + G_i^{sf} = 0$$

leads to the following integral terms:

$$G_i^n = \int_{x_{iw}}^{x_{ie}} \frac{\partial Q}{\partial t} dx$$

$$G_i^\Phi = \int_{x_{iw}}^{x_{ie}} \frac{\partial}{\partial x} \left( \beta \frac{Q^2}{A_{red}} \right) dx$$

$$G_i^{sz} = \int_{x_{iw}}^{x_{ie}} g A_{red} \frac{\partial z}{\partial x} dx$$

$$G_i^{sf} = \int_{x_{iw}}^{x_{ie}} g A_{red} S_f dx$$

Applying the  $\theta$ -method:

$$G_i^{n+1} = G_i^n + \theta(G_i^\Phi + G_i^{sz} + G_i^{sf}) + (1 - \theta)(G_i^{\Phi 0} + G_i^{sz 0} + G_i^{sf 0}) \quad (2.225)$$

### 2.4.3.3 Solution

The equation system is

$$F(x) = 0$$

with

$$F = (F_1, G_1, F_2, G_2, \dots, F_{n-1}, G_{n-1}, F_n, G_n)$$

and

$$x = (A_1, Q_1, A_2, Q_2, \dots, A_{n-1}, Q_{n-1}, A_n, Q_n)$$

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.

$$Ax^{k+1} - c = 0 \quad (2.226)$$

with

$$c = Ax^k - F$$

and

$A =$

$$\left[ \begin{array}{c}
\frac{\partial F_1}{\partial A_{red,1}} \frac{\partial F_1}{\partial G_1} \frac{\partial F_1}{\partial Q_1} \frac{\partial F_1}{\partial G_1} \\
\frac{\partial A_{red,1}}{\partial F_2} \frac{\partial Q_1}{\partial F_2} \frac{\partial A_{red,2}}{\partial F_2} \frac{\partial Q_2}{\partial F_2} \frac{\partial F_2}{\partial F_2} \frac{\partial F_2}{\partial F_2} \\
\frac{\partial A_{red,1}}{\partial G_2} \frac{\partial Q_1}{\partial G_2} \frac{\partial A_{red,2}}{\partial G_2} \frac{\partial Q_2}{\partial G_2} \frac{\partial A_{red,3}}{\partial G_2} \frac{\partial Q_3}{\partial G_2} \\
\frac{\partial A_{red,1}}{\partial F_3} \frac{\partial Q_1}{\partial F_3} \frac{\partial A_{red,2}}{\partial F_3} \frac{\partial Q_2}{\partial F_3} \frac{\partial A_{red,3}}{\partial F_3} \frac{\partial Q_3}{\partial F_3} \frac{\partial F_3}{\partial F_3} \frac{\partial F_3}{\partial F_3} \\
\frac{\partial A_{red,2}}{\partial G_3} \frac{\partial Q_2}{\partial G_3} \frac{\partial A_{red,3}}{\partial G_3} \frac{\partial Q_3}{\partial G_3} \frac{\partial A_{red,4}}{\partial G_3} \frac{\partial Q_4}{\partial G_3} \\
\frac{\partial A_{red,2}}{\partial F_4} \frac{\partial Q_2}{\partial F_4} \frac{\partial A_{red,3}}{\partial F_4} \frac{\partial Q_3}{\partial F_4} \frac{\partial A_{red,4}}{\partial F_4} \frac{\partial Q_4}{\partial F_4} \\
\vdots \\
\frac{\partial F_{n-2}}{\partial A_{red,n-3}} \frac{\partial F_{n-2}}{\partial G_{n-2}} \frac{\partial F_{n-2}}{\partial G_{n-2}} \frac{\partial F_{n-2}}{\partial G_{n-2}} \frac{\partial F_{n-2}}{\partial G_{n-2}} \frac{\partial F_{n-2}}{\partial G_{n-2}} \\
\frac{\partial A_{red,n-3}}{\partial F_{n-1}} \frac{\partial Q_{n-3}}{\partial F_{n-1}} \frac{\partial A_{red,n-2}}{\partial F_{n-1}} \frac{\partial Q_{n-2}}{\partial F_{n-1}} \frac{\partial A_{red,n-1}}{\partial F_{n-1}} \frac{\partial Q_{n-1}}{\partial F_{n-1}} \frac{\partial F_{n-1}}{\partial F_{n-1}} \frac{\partial F_{n-1}}{\partial F_{n-1}} \\
\frac{\partial A_{red,n-2}}{\partial G_{n-1}} \frac{\partial Q_{n-2}}{\partial G_{n-1}} \frac{\partial A_{red,n-1}}{\partial G_{n-1}} \frac{\partial Q_{n-1}}{\partial G_{n-1}} \frac{\partial A_{red,n}}{\partial G_{n-1}} \frac{\partial Q_n}{\partial G_{n-1}} \frac{\partial F_{n-1}}{\partial G_{n-1}} \frac{\partial F_{n-1}}{\partial G_{n-1}} \\
\frac{\partial A_{red,n-2}}{\partial F_n} \frac{\partial Q_{n-2}}{\partial F_n} \frac{\partial A_{red,n-1}}{\partial F_n} \frac{\partial Q_{n-1}}{\partial F_n} \frac{\partial A_{red,n}}{\partial F_n} \frac{\partial Q_n}{\partial F_n} \frac{\partial F_{n-1}}{\partial F_n} \frac{\partial F_{n-1}}{\partial F_n} \\
\frac{\partial A_{red,n-1}}{\partial G_n} \frac{\partial Q_{n-1}}{\partial G_n} \frac{\partial A_{red,n}}{\partial G_n} \frac{\partial Q_n}{\partial G_n} \frac{\partial A_{red,n}}{\partial G_n} \frac{\partial Q_n}{\partial G_n} \frac{\partial F_{n-1}}{\partial G_n} \frac{\partial F_{n-1}}{\partial G_n} \\
\frac{\partial A_{red,n-1}}{\partial F_n} \frac{\partial Q_{n-1}}{\partial F_n} \frac{\partial A_{red,n}}{\partial F_n} \frac{\partial Q_n}{\partial F_n} \frac{\partial A_{red,n}}{\partial F_n} \frac{\partial Q_n}{\partial F_n} \frac{\partial F_{n-1}}{\partial F_n} \frac{\partial F_{n-1}}{\partial F_n}
\end{array} \right]$$

#### 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  $i$ :

$$G_i^m = \frac{Q_i - Q_i^0}{\Delta t} \Delta x_i$$

$$G_i^\Phi = \Phi_{ie}^m - \Phi_{iw}^m$$

$$G_i^{sz} = g A_{red,i} \frac{z_{i+1} - z_{i-1}}{2}$$

$$G_i^{sf} = g A_{red,i} S_{fi} \Delta x_i \quad S_{fi} = \frac{Q_i |Q_i|}{K_i^2}$$

For the first cross section  $i = 1$ :

$$G_1^m = \frac{Q_{in} - Q_{in}^0}{\Delta t} \Delta x_1$$

$$G_1^\Phi = \Phi_{1e}^m - \frac{\beta_1 Q_{in}^2}{A_{red,in}}$$

$$G_1^{sz} = g A_{red,1} \frac{z_2 - z_1}{2}$$

$$G_1^{sf} = g A_{red,1} S_{f1} \Delta x_1 \quad S_{f1} = \frac{Q_1 |Q_1|}{K_1^2}$$

For the last cross section  $i = n$

$$G_n^m = \frac{Q_{out} - Q_{out}^0}{\Delta t} \Delta x_n$$

$$G_n^\Phi = \frac{\beta_n Q_{out}^2}{A_{red,out}} - \Phi_{nw}^m$$

$$G_n^{sz} = gA_n \frac{z_n - z_{n-1}}{2}$$

$$G_n^{sf} = gA_n S_{fn} \Delta x_n \quad S_{fn} = \frac{Q_n |Q_n|}{K_n^2}$$

### 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:

$$\frac{\partial F_n}{\partial A_{red,n-1}} = \frac{\partial Q_{out}}{\partial A_{red,n-1}} - \frac{\partial \Phi_{nw}^c}{\partial A_{red,n-1}}$$

$$\begin{aligned}\frac{\partial F_n}{\partial A_{red,n}} &= \frac{\Delta x_n}{\Delta t} \frac{dA_n}{dA_{red,n}} + \frac{\partial Q_{out}}{\partial A_{red,n}} - \frac{\partial \Phi_{nw}^c}{\partial A_{red,n}} \\ \frac{\partial F_n}{\partial Q_{n-1}} &= \frac{\partial Q_{out}}{\partial Q_{n-1}} - \frac{\partial \Phi_{nw}^c}{\partial Q_{n-1}} \\ \frac{\partial F_n}{\partial Q_n} &= \frac{\partial Q_{out}}{\partial Q_{n=out}} - \frac{\partial \Phi_{nw}^c}{\partial Q_{n=out}}\end{aligned}$$

### 2.4.3.5.2 Derivatives of the Momentum Equation

For a general cross section i:

$$\begin{aligned}\frac{\partial G_i}{\partial A_{red,i-1}} &= \frac{\partial \Phi_{ie}^m}{\partial A_{red,i-1}} - \frac{\partial \Phi_{iw}^m}{\partial A_{red,i-1}} + \frac{gA_{red,i}}{2} \frac{dz_{i-1}}{dA_{red,i-1}} \\ \frac{\partial G_i}{\partial A_{red,i}} &= \frac{\partial \Phi_{ie}^m}{\partial A_{red,i}} - \frac{\partial \Phi_{iw}^m}{\partial A_{red,i}} + \frac{g}{2}(z_{i+1} - z_{i-1}) + g\Delta x_i S_{fi} \left(1 - 2\frac{A_{red,i}}{K_i} \frac{dK_i}{dA_i}\right) \\ \frac{\partial G_i}{\partial A_{red,i+1}} &= \frac{\partial \Phi_{ie}^m}{\partial A_{red,i+1}} - \frac{\partial \Phi_{iw}^m}{\partial A_{red,i+1}} + g\frac{A_{red,i}}{2} \frac{dz_{i+1}}{dA_{red,i+1}} \\ \frac{\partial G_i}{\partial Q_{i-1}} &= \frac{\partial \Phi_{ie}^m}{\partial Q_{i-1}} - \frac{\partial \Phi_{iw}^m}{\partial Q_{i-1}} \\ \frac{\partial G_i}{\partial Q_i} &= \frac{\Delta x_i}{\Delta t} + \frac{\partial \Phi_{ie}^m}{\partial Q_i} - \frac{\partial \Phi_{iw}^m}{\partial Q_i} + 2gA_{red,i}\Delta x_i \frac{S_{fi}}{Q_i} \\ \frac{\partial G_i}{\partial Q_{i+1}} &= \frac{\partial \Phi_{ie}^m}{\partial Q_{i+1}} - \frac{\partial \Phi_{iw}^m}{\partial Q_{i+1}}\end{aligned}$$

For the first cross section i = 1:

$$\begin{aligned}\frac{\partial G_1}{\partial A_{red,1}} &= \frac{\partial \Phi_{1e}^m}{\partial A_{red,1}} - \frac{\partial \Phi_{1in}^m}{\partial A_{red,1}} + \frac{g}{2}(z_2 - z_1) + g\Delta x_1 S_{f1} \left(1 - 2\frac{A_1}{K_1} \frac{dK_1}{dA_{red,1}}\right) \\ \frac{\partial G_1}{\partial A_{red,2}} &= \frac{\partial \Phi_{1e}^m}{\partial A_{red,2}} - \frac{\partial \Phi_{1in}^m}{\partial A_{red,2}} + g\frac{A_1}{2} \frac{dz_2}{dA_2} \\ \frac{\partial G_1}{\partial Q_1} &= \frac{\Delta x_1}{\Delta t} + \frac{\partial \Phi_{1e}^m}{\partial Q_{in}} - \frac{\partial \frac{\beta_1 Q_{in}^2}{A_{red,in}}}{\partial Q_{in}} + 2gA_{red,1}\Delta x_1 \frac{S_{f1}}{Q_{in}} \\ \frac{\partial G_1}{\partial Q_2} &= \frac{\partial \Phi_{1e}^m}{\partial Q_2} - \frac{\partial \frac{\beta_1 Q_{in}^2}{A_{red,in}}}{\partial Q_2}\end{aligned}$$

For the last cross section i = n:



$$\begin{aligned}\frac{\partial G_n}{\partial A_{red,n-1}} &= \frac{\partial \Phi_{out}^m}{\partial A_{red,n-1}} - \frac{\partial \Phi_{nw}^m}{\partial A_{red,n-1}} + g \frac{A_{red,n}}{2} \frac{dz_{n-1}}{dA_{red,n-1}} \\ \frac{\partial G_n}{\partial A_{red,n}} &= \frac{\partial \Phi_{out}^m}{\partial A_{red,n}} - \frac{\partial \Phi_{nw}^m}{\partial A_{red,n}} + \frac{g}{2}(z_n - z_{n-1}) + g \Delta x_n S_{fn} \left( 1 - 2 \frac{A_n}{K_n} \frac{dK_n}{dA_{red,n}} \right) \\ \frac{\partial G_n}{\partial Q_{n-1}} &= \frac{\partial \Phi_{out}^m}{\partial Q_{n-1}} - \frac{\partial \Phi_{nw}^m}{\partial Q_{n-1}} \\ \frac{\partial G_n}{\partial Q_n} &= \frac{\Delta x_n}{\Delta t} + \frac{\partial \Phi_{out}^m}{\partial Q_n} - \frac{\partial \Phi_{nw}^m}{\partial Q_n} + 2g A_{red,n} \Delta x_n \frac{S_{fn}}{Q_n}\end{aligned}$$

### 2.4.3.6 Determination of the Derivatives with Upwind Flux Determination

With the upwind method the flux is defined as follows:

$$f(x_{ie}) = \Gamma_i^{ie} f_i + \Gamma_{i+1}^{ie} f_{i+1} \quad (2.227)$$

with

$$\begin{aligned}\Gamma_i^{ie} &= 1 \quad \text{and} \quad \Gamma_{i+1}^{ie} = 0 \quad \text{if} \quad Q_i + Q_{i+1} \geq 0 \\ \Gamma_i^{ie} &= 0 \quad \text{and} \quad \Gamma_{i+1}^{ie} = 1 \quad \text{if} \quad Q_i + Q_{i+1} < 0\end{aligned}$$

#### 2.4.3.6.1 Derivatives of the Continuity Equation

For a general cross section:

$$\begin{aligned}\frac{\partial F_i}{\partial A_{red,i-1}} &= 0 \\ \frac{\partial F_i}{\partial A_{red,i}} &= \frac{\Delta x_i}{\Delta t} \frac{\partial A_i}{\partial A_{red,i}} \\ \frac{\partial F_i}{\partial A_{red,i+1}} &= 0 \\ \frac{\partial F_i}{\partial Q_{i-1}} &= -\theta \Gamma_{i-1}^{i-0.5} \\ \frac{\partial F_i}{\partial Q_i} &= \theta (\Gamma_i^w - \Gamma_i^e) \\ \frac{\partial F_i}{\partial Q_{i+1}} &= \theta \Gamma_{i+1}^e\end{aligned}$$

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  $i$ :

$$\frac{\partial G_i}{\partial A_{red,i-1}} = \theta \left( \Gamma_{i-1}^w \left[ \beta \frac{Q^2}{A_{red}} \right] \left( \frac{1}{A_{red,i-1}} - \frac{1}{\beta_{i-1}} \frac{d\beta_{i-1}}{dA_{red,i-1}} \right) \right) - g \frac{A_{red,i}}{2} \frac{dz_{i-1}}{dA_{red,i-1}}$$

$$\begin{aligned} \frac{\partial G_i}{\partial A_{red,i}} &= \theta \left( (\Gamma_i^w - \Gamma_i^e) \left[ \beta \frac{Q^2}{A_{red}} \right]_i \left( \frac{1}{A_{red,i}} - \frac{1}{\beta_i} \frac{d\beta_i}{dA_{red,i}} \right) + \frac{g}{2} (z_{i+1} - z_{i-1}) \right) + \\ &\theta \left( g \Delta x_i S_{f_i} \left( 1 - 2 \frac{A_{red,i}}{K_i} \frac{dK_i}{dA_{red,i}} \right) \right) \end{aligned}$$

$$\frac{\partial G_i}{\partial A_{red,i+1}} = \theta \left( -\Gamma_{i+1}^e \left[ \beta \frac{Q^2}{A_{red}} \right]_{i+1} \left( \frac{1}{A_{red,i+1}} - \frac{1}{\beta_{i+1}} \frac{d\beta_{i+1}}{dA_{red,i+1}} \right) + g \frac{A_{red,i}}{2} \frac{dz_{i+1}}{dA_{red,i+1}} \right)$$

$$\frac{\partial G_i}{\partial Q_{i-1}} = -2\theta\Gamma_{i-1}^w \frac{1}{Q_{i-1}} \left[ \beta \frac{Q^2}{A_{red}} \right]_{i-1}$$

$$\frac{\partial G_i}{\partial Q_i} = \frac{\Delta x_i}{\Delta t} + 2\theta \left( (\Gamma_i^e - \Gamma_i^w) \frac{1}{Q_i} \left[ \beta \frac{Q^2}{A_{red}} \right]_i + 2gA_{red,i} \Delta x_i \frac{S_{f_i}}{Q_i} \right)$$

$$\frac{\partial G_i}{\partial Q_{i+1}} = 2\theta\Gamma_{i+1}^e \frac{1}{Q_{i+1}} \left[ \beta \frac{Q^2}{A_{red}} \right]_{i+1}$$

For cross section  $i = 1$ :

$$\frac{\partial G_1}{\partial A_{red,1}} = \theta \left( (1 - \Gamma_1^e) \left[ \beta \frac{Q^2}{A_{red}} \right]_1 \left( \frac{1}{A_{red,1}} - \frac{1}{\beta_1} \frac{d\beta_1}{dA_{red,1}} \right) + \frac{g}{2} \left( z_2 - z_1 - A_{red,1} \frac{dz_1}{dA_{red,1}} \right) \right) + \theta \left( g \Delta x_1 S_{f1} \left( 1 - 2 \frac{A_{red,1}}{K_1} \frac{dK_1}{dA_{red,1}} \right) \right)$$

$$\frac{\partial G_1}{\partial A_{red,2}} = \theta \left( -\Gamma_2^e \left[ \beta \frac{Q^2}{A_{red}} \right]_2 \left( \frac{1}{A_{red,2}} - \frac{1}{\beta_2} \frac{d\beta_2}{dA_{red,2}} \right) + g \frac{A_{red,2}}{2} \frac{dz_2}{dA_{red,2}} \right)$$

$$\frac{\partial G_1}{\partial Q_1} = \frac{\Delta x_1}{\Delta t} + 2\theta \left( (\Gamma_1^e - 1) \frac{1}{Q_1} \left[ \beta \frac{Q^2}{A_{red}} \right]_1 + 2g A_{red,1} \Delta x_1 \frac{S_{f1}}{Q_1} \right)$$

$$\frac{\partial G_1}{\partial Q_2} = 2\theta \Gamma_2^e \frac{1}{Q_2} \left[ \beta \frac{Q^2}{A_{red}} \right]_2$$

For cross section  $i = n$ :

$$\frac{\partial G_n}{\partial A_{red,n-1}} = \theta \left( \Gamma_{n-1}^w \left[ \beta \frac{Q^2}{A_{red}} \right]_{n-1} \left( \frac{1}{A_{n-1}} - \frac{1}{\beta_{n-1}} \frac{d\beta_{n-1}}{dA_{n-1}} \right) \right) - g \frac{A_n}{2} \frac{dz_{n-1}}{dA_{red,n-1}}$$

$$\frac{\partial G_n}{\partial A_{red,n}} = \theta \left( (\Gamma_n^w - 1) \left[ \beta \frac{Q^2}{A_{red}} \right]_n \left( \frac{1}{A_n} - \frac{1}{\beta_n} \frac{d\beta_n}{dA_n} \right) + \frac{g}{2} \left( z_n - z_{n-1} + A_n \frac{dz_n}{dA_n} \right) \right) + \theta \left( g \Delta x_n S_{fn} \left( 1 - 2 \frac{A_n}{K_n} \frac{dK_n}{dA_n} \right) \right)$$

$$\frac{\partial G_n}{\partial Q_{n-1}} = -2\theta \Gamma_{n-1}^w \frac{1}{Q_{n-1}} \left[ \beta \frac{Q^2}{A} \right]_{n-1}$$

$$\frac{\partial G_n}{\partial Q_n} = \frac{\Delta x_n}{\Delta t} + 2\theta \left( (1 - \Gamma_n^w) \frac{1}{Q_n} \left[ \beta \frac{Q^2}{A} \right]_n + g A_n \Delta x_n \frac{S_{fn}}{Q_n} \right)$$

### 2.4.3.7 Determination of the Derivatives with Roe Flux Determination

Fluxes and derivatives of the fluxes of the continuity equation:

$$\Phi_{ie}^c = Q(ie) = \frac{1}{2}(Q_i + Q_{i+1}) - R_{cA}(A_{i+1} - A_i) - R_{cQ}(Q_{i+1} - Q_i) \quad (2.228)$$

$$\begin{aligned} \frac{\partial \Phi_{ie}^c}{\partial A_{i-1}} &= 0 & \frac{\partial \Phi_{ie}^c}{\partial A_i} &= R_{cA}^{ie} & \frac{\partial \Phi_{ie}^c}{\partial A_{i+1}} &= -R_{cA}^{ie} \\ \frac{\partial \Phi_{ie}^c}{\partial Q_{i-1}} &= 0 & \frac{\partial \Phi_{ie}^c}{\partial Q_i} &= 0.5 + R_{cQ}^{ie} & \frac{\partial \Phi_{ie}^c}{\partial Q_{i+1}} &= 0.5 - R_{cQ}^{ie} \end{aligned}$$

$$\Phi_{iw}^c = Q(iw) = 0.5(Q_{i-1} + Q_i) - R_{cA}(A_i - A_{i-1}) - R_{cQ}(Q_i - Q_{i-1}) \quad (2.229)$$

$$\begin{aligned} \frac{\partial \Phi_{iw}^c}{\partial A_{i-1}} &= R_{cA}^{iw} & \frac{\partial \Phi_{iw}^c}{\partial A_i} &= -R_{cA}^{iw} & \frac{\partial \Phi_{iw}^c}{\partial A_{i+1}} &= 0 \\ \frac{\partial \Phi_{iw}^c}{\partial Q_{i-1}} &= 0.5 + R_{cQ}^{iw} & \frac{\partial \Phi_{iw}^c}{\partial Q_i} &= 0.5 - R_{cQ}^{iw} & \frac{\partial \Phi_{iw}^c}{\partial Q_{i+1}} &= 0 \end{aligned}$$

Fluxes and derivatives of the fluxes of the momentum equation:

$$\Phi_{ie}^m = \beta \frac{Q^2}{A} \Big|_{ie} = 0.5 \left( \beta_i \frac{Q_i^2}{A_i} + \beta_{i+1} \frac{Q_{i+1}^2}{A_{i+1}} \right) - R_{mA}(A_{i+1} - A_i) - R_{mQ}(Q_{i+1} - Q_i) \quad (2.230)$$

$$\begin{aligned} \frac{\partial \Phi_{ie}^m}{\partial A_{i-1}} &= 0 & \frac{\partial \Phi_{ie}^m}{\partial A_i} &= -0.5\beta_i \frac{Q_i^2}{A_i^2} + R_{mA}^{ie} & \frac{\partial \Phi_{ie}^m}{\partial A_{i+1}} &= -0.5\beta_{i+1} \frac{Q_{i+1}^2}{A_{i+1}^2} - R_{mA}^{ie} \\ \frac{\partial \Phi_{ie}^m}{\partial Q_{i-1}} &= 0 & \frac{\partial \Phi_{ie}^m}{\partial Q_i} &= \beta_i \frac{Q_i^2}{A_i^2} + R_{mQ}^{ie} & \frac{\partial \Phi_{ie}^m}{\partial Q_{i+1}} &= \beta_{i+1} \frac{Q_{i+1}^2}{A_{i+1}^2} - R_{mQ}^{ie} \end{aligned}$$

$$\Phi_{iw}^m = \beta \frac{Q^2}{A} \Big|_{iw} = 0.5 \left( \beta_i \frac{Q_{i-1}^2}{A_{i-1}} + \beta_i \frac{Q_i^2}{A_i} \right) - R_{mA}(A_i - A_{i-1}) - R_{mQ}(Q_i - Q_{i-1}) \quad (2.231)$$

$$\begin{aligned} \frac{\partial \Phi_{iw}^m}{\partial A_{i-1}} &= -0.5\beta_{i-1} \frac{Q_{i-1}^2}{A_{i-1}^2} + R_{mA}^{iw} & \frac{\partial \Phi_{iw}^m}{\partial A_i} &= -0.5\beta_i \frac{Q_i^2}{A_i^2} - R_{mA}^{iw} & \frac{\partial \Phi_{iw}^m}{\partial A_{i+1}} &= 0 \\ \frac{\partial \Phi_{iw}^m}{\partial Q_{i-1}} &= \beta_{i-1} \frac{Q_{i-1}}{A_{i-1}} + R_{mQ}^{iw} & \frac{\partial \Phi_{iw}^m}{\partial Q_i} &= \beta_i \frac{Q_i}{A_i} - R_{mQ}^{iw} & \frac{\partial \Phi_{iw}^m}{\partial Q_{i+1}} &= 0 \end{aligned}$$

#### 2.4.3.7.1 Derivatives of the Continuity Equation:

For a general cross section i:

$$\begin{aligned} \frac{\partial F_i}{\partial A_{red,i-1}} &= -R_{cA}^{iw} \\ \frac{\partial F_i}{\partial A_{red,i}} &= \frac{\Delta x_i}{\Delta t} \frac{\partial A_i}{\partial A_{red,i}} + R_{cA}^{ie} + R_{cA}^{iw} \\ \frac{\partial F_i}{\partial A_{red,i+1}} &= -R_{cA}^{ie} \\ \frac{\partial F_i}{\partial Q_{i-1}} &= -\frac{1}{2} - R_{cQ}^{iw} \\ \frac{\partial F_i}{\partial Q_i} &= R_{cQ}^{ie} - R_{cQ}^{iw} \\ \frac{\partial F_i}{\partial Q_{i+1}} &= \frac{1}{2} - R_{cQ}^{ie} \end{aligned}$$

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  $i$ :

$$\frac{\partial G_i}{\partial A_{red,i-1}} = \frac{1}{2} \beta_{i-1} U_{i-1}^2 - R_{mA}^{iw} + g \frac{A_{red,i}}{2} \frac{dz_{i-1}}{dA_{red,i-1}}$$

$$\frac{\partial G_i}{\partial A_{red,i}} = -\frac{1}{2} \beta_i U_i^2 + R_{mA}^{ie} + \frac{1}{2} \beta_i U_i^2 + R_{mA}^{iw} + \frac{g}{2} (z_{i+1} - z_{i-1}) + g \Delta x_i S_{fi} \left( 1 - 2 \frac{A_{red,i}}{K_i} \frac{dK_i}{dA_{red,i}} \right)$$

$$\frac{\partial G_i}{\partial A_{red,i+1}} = -\frac{1}{2} \beta_{i+1} U_{i+1}^2 - R_{mA}^{ie} + g \frac{A_{red,i}}{2} \frac{dz_{i+1}}{dA_{red,i+1}}$$

$$\frac{\partial G_i}{\partial Q_{i-1}} = \beta_{i-1} U_{i-1} - R_{mQ}^{iw}$$

$$\frac{\partial G_i}{\partial Q_i} = \frac{\Delta x_i}{\Delta t} + R_{mQ}^{ie} + R_{mQ}^{ie} + 2g A_{red,i} \Delta x_i \frac{S_{fi}}{Q_i}$$

$$\frac{\partial G_i}{\partial Q_{i+1}} = \beta_{i+1}U_{i+1} - R_{mQ}^{ie}$$

For the first cross section  $i=1$ :

$$\frac{\partial G_1}{\partial A_{red,1}} = -\frac{1}{2}\beta_1U_1^2 + R_{mA}^{1e} - \frac{\partial \Phi_{in}^m}{\partial A_{red,1}} + \frac{g}{2}(z_2 - z_1) + g\Delta x_1 S_{f1} \left( 1 - 2\frac{A_{red,1}}{K_1} \frac{dK_1}{dA_{red,1}} \right)$$

$$\frac{\partial G_1}{\partial A_{red,2}} = -\frac{1}{2}\beta_2U_2^2 + R_{mA}^{1e} + g\frac{A_{red,1}}{2} \frac{dz_2}{dA_{red,2}}$$

$$\frac{\partial G_1}{\partial Q_1} = \frac{\Delta x_1}{\Delta t} + \beta_1U_1 + R_{mQ}^{1e} - \frac{\partial \Phi_{in}^m}{\partial Q_{1=in}} + 2gA_1\Delta x_1 \frac{S_{f1}}{Q_1}$$

$$\frac{\partial G_1}{\partial Q_2} = \beta_2U_2 - R_{mQ}^{1e}$$

For the last cross section  $i=n$

$$\frac{\partial G_n}{\partial A_{red,n-1}} = \frac{1}{2}\beta_{n-1}U_{n-1}^2 - R_{mA}^{nw} + g\frac{A_{red,n}}{2} \frac{dz_{n-1}}{dA_{red,n-1}}$$

$$\frac{\partial G_n}{\partial A_{red,n}} = -\frac{1}{2}\beta_nU_n^2 + R_{mA}^{ne} + \frac{1}{2}\beta_nU_n^2 + R_{mA}^{nw} + \frac{g}{2}(z_n - z_{n-1}) + g\Delta x_n S_{fn} \left( 1 - 2\frac{A_{red,n}}{K_n} \frac{dK_n}{dA_{red,n}} \right)$$

$$\frac{\partial G_n}{\partial Q_{n-1}} = \beta_{n-1}U_{n-1} - R_{mQ}^{nw}$$

$$\frac{\partial G_n}{\partial Q_n} = \frac{\Delta x_n}{\Delta t} + \beta_nU_n + R_{mQ}^{ne} - \beta_nU_n + R_{mQ}^{nw} + 2gA_n\Delta x_n \frac{S_{fn}}{Q_n} = \frac{\Delta x_n}{\Delta t} + R_{mQ}^{ne} + R_{mQ}^{nw} + 2gA_n\Delta x_n \frac{S_{fn}}{Q_n}$$

#### 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

$$\begin{aligned} \frac{\partial \Phi_{ie}^c}{\partial A_i} &= wp\sqrt{2g(z_i - z_{weir})} \frac{dz_i}{dA_i} & \frac{\partial \Phi_{iw}^c}{\partial A_i} &= 0 \\ \frac{\partial \Phi_{ie}^c}{\partial A_{i-1}} &= 0 & \frac{\partial \Phi_{iw}^c}{\partial A_{i-1}} &= wp\sqrt{2g(z_{i-1} - z_{weir})} \frac{dz_{i-1}}{dA_{i-1}} \\ \frac{\partial \Phi_{ie}^c}{\partial A_{i+1}} &= 0 & \frac{\partial \Phi_{iw}^c}{\partial A_{i+1}} &= 0 \end{aligned}$$

$$\begin{aligned} \frac{\partial \Phi_{ie}^c}{\partial Q_{i-1}} &= 0 & \frac{\partial \Phi_{iw}^c}{\partial Q_{i-1}} &= 1 \\ \frac{\partial \Phi_{ie}^c}{\partial Q_i} &= 1 & \frac{\partial \Phi_{iw}^c}{\partial Q_i} &= 0 \\ \frac{\partial \Phi_{ie}^c}{\partial Q_{i+1}} &= 0 & \frac{\partial \Phi_{iw}^c}{\partial Q_{i+1}} &= 0 \end{aligned}$$

#### 2.4.3.8.2 Derivatives of the momentum flux

$$\begin{aligned} \frac{\partial \Phi_{ie}^m}{\partial A_{i-1}} &= 0 & \frac{\partial \Phi_{iw}^m}{\partial A_{i-1}} &= \frac{8gp^2w^2(z_{i-1} - z_{weir})^2}{3A_{weir}} \frac{dz_{i-1}}{dA_{i-1}} \\ \frac{\partial \Phi_{ie}^m}{\partial A_i} &= \frac{8gp^2w^2(z_{i-1} - z_{weir})^2}{3A_{weir}} \frac{dz_i}{dA_i} & \frac{\partial \Phi_{iw}^m}{\partial A_i} &= 0 \\ \frac{\partial \Phi_{ie}^m}{\partial A_{i+1}} &= 0 & \frac{\partial \Phi_{iw}^m}{\partial A_{i+1}} &= 0 \\ \frac{\partial \Phi_{ie}^m}{\partial Q_{i-1}} &= 0 & \frac{\partial \Phi_{iw}^m}{\partial Q_{i-1}} &= \frac{2Q_{i-1}}{A_{i-1}} \\ \frac{\partial \Phi_{ie}^m}{\partial Q_i} &= \frac{2Q_i}{A_i} & \frac{\partial \Phi_{iw}^m}{\partial Q_i} &= 0 \\ \frac{\partial \Phi_{ie}^m}{\partial Q_{i+1}} &= 0 & \frac{\partial \Phi_{iw}^m}{\partial Q_{i+1}} &= 0 \end{aligned}$$

## 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 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  $\theta$  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.

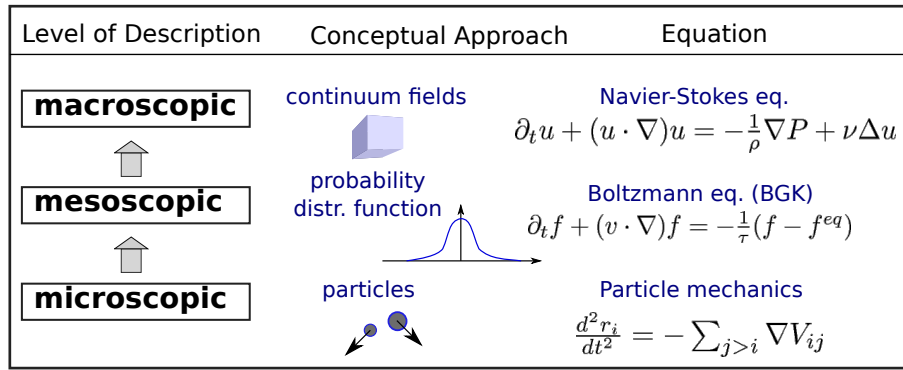


Figure 2.67 Comparison of different levels of description and conceptual approaches applied to determine fluid motion.

## 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  $f(\vec{r}, t)$  of particles in the 6-D phase-space  $\vec{r}(\vec{x}, \vec{v})$ . 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  $\vec{r}$  and  $\vec{r} + \Delta \vec{r}$ . 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  $f(\vec{r}, t)$  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  $f(\vec{r}, t)$  can be written in 1D as (Mohamad, 2011)

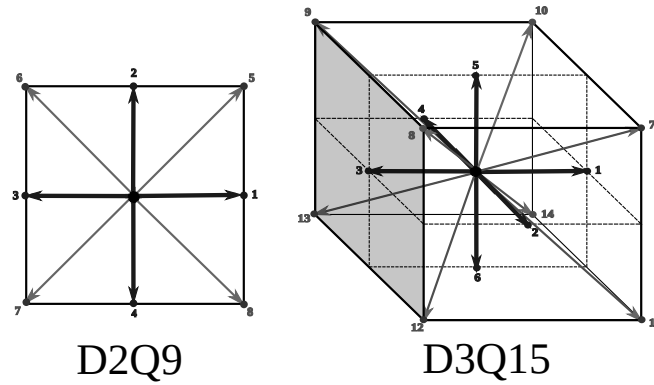
$$\frac{\partial f(\vec{r}, t)}{\partial t} + v \frac{\partial f(\vec{r}, t)}{\partial x} = \Omega \quad (2.232)$$

with  $\Omega$  as the collision operator, which describes the mutual influences of distribution functions  $f(\vec{r}, t)$  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  $\omega$ .

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

$$\underbrace{f_q(\vec{r} + \Delta t \vec{c}_q, t + \Delta t)}_{\text{advection step}} = \underbrace{f_q(\vec{r}, t) + \omega[f_q(\vec{r}, t) - f_q^{eq}(\vec{r}, t)]}_{\text{collision step}} + \underbrace{Q_q/c_m}_{\text{source}} \quad q = 1 \dots n_q \quad (2.233)$$





**Figure 2.68** 2D lattice with 9 directions (left), 3D lattice with 15 directions (right).

with a single time relaxation parameter  $\omega$  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  $\theta$  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  $c/\vartheta$  with the free adjustable constant  $\vartheta$ .

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

$$c_q = c_m \begin{cases} (0, 0, 0), (1, 0, 0), (0, 1, 0), (-1, 0, 0), (0, -1, 0), (0, 0, 1), (0, 0, -1) & q = 0 \dots 6 \\ (1, 1, 1), (-1, 1, 1), (-1, -1, 1), (1, -1, 1) & q = 7 \dots 10 \\ (1, 1, -1), (-1, 1, -1), (-1, -1, -1), (1, -1, -1) & q = 11 \dots 14 \end{cases} \quad (2.234)$$

### 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  $\theta$  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:

$$\theta = \sum_{q=0}^n f_q$$

$$h = f(\theta)$$

$$\vec{v}_f = (\theta_s - \theta_r) \cdot \left[ \sum_{q=0}^n \vec{c} f_q^{eq} + \begin{pmatrix} c_{x0} & \cdots & c_{xn} \\ c_{y0} & \cdots & c_{yn} \end{pmatrix} \cdot \begin{pmatrix} f_0^{eq} - f_0 \\ \vdots \\ f_n^{eq} - f_n \end{pmatrix} \right] \quad (2.235)$$

### 2.5.4 Equilibrium functions

The equilibrium distribution function for the  $\theta$  formulation is given in first order accuracy in all mesh directions by

$$f_q^{eq} = \begin{cases} \left(1.0 - \frac{7}{3}c_s^2\right) \cdot \theta & q = 0 \\ t_q \theta c_s^2 & q = 1, 2, 3, 4 \\ t_q \cdot (\theta c_s^2 + \vec{I} \cdot \vec{c}) & q = 5, 7, 8, 9, 10 \\ t_q \cdot (\theta c_s^2 - \vec{I} \cdot \vec{c}) & q = 6, 11, 12, 13, 14 \end{cases} \quad (2.236)$$

where  $c_s^2 = c/\vartheta$  with  $\vartheta$  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

$$f_q^{eq} = \begin{cases} \theta - 7/3c_s^2 \cdot h & q = 0 \\ t_q \cdot (hc_s^2) & q = 1, 2, 3, 4 \\ t_q \cdot (hc_s^2 + \vec{I} \cdot \vec{e}) & q = 5, 7, 8, 9, 10 \\ t_q \cdot (hc_s^2 - \vec{I} \cdot \vec{e}) & q = 6, 11, 12, 13, 14 \end{cases} \quad (2.237)$$

In contrast to the  $\theta$  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  $\theta$  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  $\vec{I}$ , which acts in vertical downward direction, is needed and is evaluated as

$$\vec{I} = -k_r(\theta)k_s \vec{e}_z \quad (2.238)$$

### 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  $\theta$  formulation, the water saturation  $\theta$  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  $0^\circ$ ,  $45^\circ$  or  $90^\circ$ . 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.

---

## 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.J. (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.F. (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.P. and Nordin, C.F. (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). Fließwiderstand 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.I., Skeels, C.P. and Ryrie, S.C. (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.E. (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.P. 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.D. 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.H. (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 Kolkiefenberechnung [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, G. (1990). Surface-based bedload transport relation for gravel rivers. *Journal of Hydraulic Research*, 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.
- Parker, G. and Sutherland, A.J. (1990). Fluvial armor. *Journal of Hydraulic Research*,.
- 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.M. and Jaeggi, M.N.R. (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. (pp. 483. 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.M., Struiksmā, N. and van Mierlo, M.C.L.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. and Crowe, 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 MÅ¼ller 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.J. (1961). River Dynamics. *Industry Press*, Beijing, China.
- Zyserman, J.A. and Fredsøe, J. (1994). Data Analysis of Bed Concentration of Suspended Sediment. *Journal of Hydraulic Engineering, ASCE*, 120(9): 1021–1042.





**BASIC SIMULATION ENVIRONMENT  
FOR MODELLING OF ENVIRONMENTAL  
FLOWS AND NATURAL HAZARDS**

---

# **REFERENCE MANUAL BASEHPC**

**VERSION 4.1.0  
JUNE 2024**



**BASEMENT**



# Contents

<b>1</b>	<b>Mathematical Models</b>	<b>5</b>
1.1	Hydrodynamics	5
1.1.1	Introduction	5
1.1.2	Governing Equations	6
1.1.3	Closure Relations	6
1.1.3.1	Friction Terms	6
1.1.3.2	Turbulent Terms	7
1.1.3.3	Lateral Inflow/Outflow	9
1.1.4	Boundary Conditions	9
1.1.4.1	Standard Boundary Conditions	10
1.1.4.2	Linked Boundary Conditions	12
1.1.4.3	Internal Boundary Conditions	13
1.1.5	Flood Tracking	14
1.2	Morphodynamics	14
1.2.1	Introduction	14
1.2.2	Bedload Sediment Transport	14
1.2.2.1	Governing Equations for Uniform Sediment Transport	14
1.2.2.2	Governing Equations for Mixed-Size Sediment Transport	15
1.2.2.3	Threshold Conditions for Sediment Movement	17
1.2.2.4	Closure Relations for Bedload Transport	19
1.2.2.5	Wilcock and Crowe (2003)	22
1.2.2.6	Correction of Bedload Direction	24
1.2.2.7	Bed Material and Fixed Bed Concept	25
1.2.2.8	Gravitational Transport	28
1.2.2.9	External Sources Terms	33
1.2.2.10	Boundary Conditions	33
1.2.2.11	Upstream Boundary Condition	33
1.2.2.12	Downstream Boundary Condition	34
1.2.2.13	Linked Boundary Condition	35
1.3	Suspended Sediment Transport	35
1.3.1	Governing Equations for Uniform Suspended Sediment Transport	35
1.3.2	Closures for Suspended Transport	35
1.3.2.1	Deposition rate	35
1.3.2.2	Erosion rate	36
1.3.2.3	Settling Velocities of Particles	36
1.3.3	External Source Terms	37
1.3.4	Boundary Conditions	38
1.3.4.1	Upstream Boundary Condition	38
1.3.4.2	Downstream Boundary Condition	38

1.4	Passive tracers . . . . .	38
1.4.1	Introduction . . . . .	38
1.4.2	Transport of passive species . . . . .	39
1.4.2.1	Governing Equations for passive specie transport . . . . .	39
1.4.2.2	External Sources Terms . . . . .	39
1.4.2.3	Boundary Conditions . . . . .	40
1.4.2.4	Upstream Boundary Condition . . . . .	40
1.4.2.5	Downstream Boundary Condition . . . . .	40
1.5	Riparian vegetation . . . . .	41
1.5.1	Introduction . . . . .	41
1.5.2	Description of Vegetation . . . . .	41
1.5.3	Effect of Vegetation on Water Flow . . . . .	41
1.5.4	Effects of Vegetation on Bedload Transport . . . . .	42
1.5.4.1	Bottom Shear Stress . . . . .	42
1.5.4.2	Critical Shear Stress . . . . .	42
1.5.5	Effects of Bed Changes on Vegetation . . . . .	42
1.5.5.1	Biomass Redistribution . . . . .	43
1.5.5.2	Plant Burial . . . . .	43
1.5.5.3	Plant Uprooting . . . . .	43
1.6	Water temperature . . . . .	43
1.6.1	Introduction . . . . .	43
1.6.2	Thermodynamics: governing equations . . . . .	44
1.6.3	Temperature closure relationships . . . . .	45
1.6.4	Initial conditions . . . . .	47
1.6.5	Boundary conditions . . . . .	47
1.6.6	Setup parameters . . . . .	47
<b>2</b>	<b>Numerical Models</b>	<b>49</b>
2.1	General View . . . . .	49
2.2	Discretization . . . . .	49
2.3	Numerical solution of Hydrodynamics . . . . .	51
2.3.1	Vectorial Form of the Governing Equations . . . . .	51
2.3.2	Spatial Discretisation . . . . .	51
2.3.3	Flux Estimation . . . . .	52
2.3.3.1	Rotational Invariance of the Shallow Water Equations . . . . .	52
2.3.3.2	Computation of the Flux . . . . .	52
2.3.3.3	The HLLC approximated Rieman Solver . . . . .	52
2.3.4	Numerical Stability . . . . .	53
2.3.5	Discretisation of Source terms . . . . .	54
2.3.5.1	Bed Slope Source Term . . . . .	54
2.3.5.2	Friction Source Term . . . . .	54
2.3.5.3	External Source Term . . . . .	55
2.3.6	Solution Procedure . . . . .	55
2.4	Numerical solution of Morphodynamics . . . . .	55
2.4.1	Numerical solution of the Exner equation . . . . .	55
2.4.1.1	Fundamentals . . . . .	55
2.4.1.2	Spatial discretization . . . . .	56
2.4.1.3	Discretization of External Source Term . . . . .	58
2.4.2	Solution procedure . . . . .	58

2.5	Numerical solution of Advection-Diffusion equation . . . . .	58
2.5.1	Numerical solution of the advective part . . . . .	58
2.5.2	Numerical solution of the diffusive part . . . . .	59
2.5.3	Discretization of external source terms . . . . .	60
2.5.4	Solution procedure . . . . .	61
2.6	Numerical solution of the thermodynamics equation . . . . .	61
2.6.1	Discretization of external source terms . . . . .	62
2.6.2	Solution procedure . . . . .	62
<b>3</b>	<b>References</b>	<b>63</b>





---

## Mathematical Models

### 1.1 Hydrodynamics

#### 1.1.1 Introduction

Mathematical models of the so-called *shallow water* type govern a wide variety of physical phenomena. Especially the one-dimensional (1D) de Saint-Venant equations (SVE) or two-dimensional (2D) shallow water equations (SWE) are of practical interest with regard to water flows with a free surface under the influence of gravity. Applications of the models include e.g.:

- River hydrodynamics
- Propagation of flood waves
- Dam break waves
- Flooding and inundation
- Ecological assessment based on flow quantities

The 2D SWE are based on the following set of hypotheses:

- the water is assumed to be incompressible; i.e. the water density  $\rho$  is constant
- the vertical acceleration of the water particles are assumed to be small compared to the longitudinal component of the acceleration. As a consequence the pressure distribution is hydrostatic;
- the bottom slope is small enough for the longitudinal coordinate to coincide with the horizontal axis;
- the flow regime is turbulent. As a consequence the head loss, mainly due to friction against the bottom, is proportional to the square of the flow velocity.

### 1.1.2 Governing Equations

The governing equations are obtained under shallow water conditions imposing mass conservation for the fluid and solid phases and the momentum principle to a flow in an open channel with a fixed bottom.

Introducing a Cartesian reference system  $(x, y, z)$  in which the  $z$  axis is vertical and the  $x - y$  plane is horizontal with respect to gravity  $g$ , the system of governing equations can be written as

$$\begin{cases} \frac{\partial h}{\partial t} + \frac{\partial q_x}{\partial x} + \frac{\partial q_y}{\partial y} = S_h \\ \frac{\partial q_x}{\partial t} + \frac{\partial}{\partial x} \left( \frac{q_x^2}{h} + \frac{1}{2}gh^2 \right) + \frac{\partial}{\partial y} \left( \frac{q_x q_y}{h} \right) + gh(S_{bx} + S_{fx}) + T_x = 0 \\ \frac{\partial q_y}{\partial t} + \frac{\partial}{\partial x} \left( \frac{q_y q_x}{h} \right) + \frac{\partial}{\partial y} \left( \frac{q_y^2}{h} + \frac{1}{2}gh^2 \right) + gh(S_{by} + S_{fy}) + T_y = 0, \end{cases} \quad (1.1)$$

where:

$h$	$[m]$	water depth
$g$	$[m/s^2]$	gravity acceleration
$u$ ( $v$ )	$[m/s]$	depth averaged velocity in $x$ ( $y$ ) direction
$q_x$ ( $q_y$ )	$[m^2/s]$	discharge per unit width in $x$ ( $y$ ) direction
$S_h$	$[m/s]$	lateral inflow/outflow discharge per unit width
$S_{fx}$ ( $S_{fy}$ )	$[-]$	friction terms in $x$ ( $y$ ) direction
$T_x$ ( $T_y$ )	$[-]$	turbulent terms in $x$ ( $y$ ) direction .

The bed slope source terms

$$S_{bx}, S_{by}$$

are evaluated as follows:

$$S_{bx} = -\frac{\partial z_B}{\partial x} \quad ; \quad S_{by} = -\frac{\partial z_B}{\partial y} \quad (1.2)$$

### 1.1.3 Closure Relations

In order to solve system (eq. 1.1) we need to specify the closure relations for the friction terms  $S_{fx}, S_{fy}$  and the value of lateral inflow/outflow discharge per unit width  $S_h$ .

#### 1.1.3.1 Friction Terms

The governing equations (eq. 1.1) have been derived under the hypothesis (H3) of turbulent flow, hence the friction term  $S_f$  can be assumed proportional to the square of the depth-averaged velocity and can be written as:

$$S_{fx} = \frac{u|\vec{u}|}{gc_f^2 h} \quad ; \quad S_{fy} = \frac{v|\vec{u}|}{gc_f^2 h} \quad (1.3)$$

where  $g$  is the gravity acceleration,  $u$  and  $v$  are the depth averaged velocities in  $x$  and  $y$  direction,  $|\vec{u}| = \sqrt{u^2 + v^2}$  is the magnitude of the velocity vector and  $c_f$  is the dimensionless friction coefficient.

Several formulae are available for the dimensionless friction coefficient  $c_f$ . Here it is quantified using both a power or a logarithmic for which are described in the next sections.

#### 1.1.3.1.1 Power Law

The Manning-Strickler power law is widely used in practice and it requires that either the Strickler's  $k_{str}$  [ $m^{1/3}/s$ ] or the Manning's  $n$  coefficients ( $k_{str} = n^{-1}$ ) is specified.

In this case the dimensionless friction coefficient  $c_f$  is calculated as

$$c_f = \frac{k_{str} h^{1/6}}{\sqrt{g}} \quad (1.4)$$

#### 1.1.3.1.2 Logarithmic Law

The following approaches are implemented to determine the friction coefficient  $c_f$ :

Chézy:

$$\begin{aligned} c_f &= 5.75 \log\left(12 \frac{R}{K_s}\right) & \text{for } R > K_s \\ c_f &= 5.75 \log(12) & \text{for } R < K_s, \end{aligned} \quad (1.5)$$

where  $K_s$  [ $m$ ] is the bed roughness height which is commonly taken to be proportional to a representative sediment size  $d_x$ . For rivers,  $K_s$  can be assumed  $K_s = n_k d_{90}$  where  $n_k = 2 \div 3$ .

Bezzola:

In this closure relation, proposed by Bezzola (2002),  $c_f$  is given as a function of the roughness sublayer height  $y_R$  [ $m$ ] (usually for rivers  $y_R \approx 1.0 d_{90}$  is a good approximation). This approach is also valid for small values of the relative submergence  $h/y_r$  Bezzola (2002).

$$\begin{cases} c_f = 2.5 \sqrt{1 - \frac{y_R}{h}} \ln\left(10.9 \frac{R}{y_R}\right), & \text{for } \frac{h}{y_R} > 2 \\ c_f = 1.25 \sqrt{\frac{h}{y_R}} \ln\left(10.9 \frac{R}{y_R}\right), & \text{for } 0.5 \leq \frac{h}{y_R} \leq 2 \\ c_f = 1.5, & \text{for } \frac{h}{y_R} < 0.5 \end{cases} \quad (1.6)$$

#### 1.1.3.2 Turbulent Terms

The turbulent and viscous stresses are considered under the eddy viscosity model, following the Boussinesq hypothesis, and can be written as

$$T_{x_i} = \frac{\partial}{\partial x_j} \left( (\nu + \nu_t) \frac{\partial}{\partial x_j} h u_{x_i} \right) \quad (1.7)$$

where  $x_i$  stands for the  $x$  or  $y$  direction,  $\nu$  and  $\nu_t$  are the laminar and turbulent kinematic viscosities, respectively,  $u_{x_i}$  is the flow velocity and  $T_{x_i}$  are the commonly denominated Reynolds stresses. The turbulent viscosity is the defining parameter when employing this

type of turbulence closure, with several closures being available in fluid mechanics literature. For SWE systems, the most common options for computing the eddy viscosity are the constant, mixing-length and  $\kappa$ - $\epsilon$  models.

### 1.1.3.2.1 Mixing-length model

The mixing-length model was derived by Ludwig Prandtl and is based on a characteristic length concept where a fluid retains some of its original characteristics (namely turbulence) before dispersing them into the surrounding fluid. In this model, the turbulent viscosity is locally derived by a length scale, the strain rate and the friction velocity as follows:

$$\nu_t = l_s^2 \sqrt{2S_{ij}S_{ij} + \frac{2.34\kappa}{u_f}} \quad (1.8)$$

where  $l_s$  is the characteristic length scale,  $S_{ij} = (\partial_{x_j}u_i + \partial_{x_i}u_j)/2$  is the strain rate,  $\kappa$  is the von Karman constant and  $u_f = c_f|\bar{u}|$  is the friction velocity. The length scale can be defined as a constant or by using  $l_s = 0.267\kappa h$  (ref).

### 1.1.3.2.2 $\kappa$ - $\epsilon$ model

The  $\kappa$ - $\epsilon$  model is one of the most widely adopted models in hydraulics and general fluid mechanics. The turbulent kinetic energy (TKE)  $\kappa$  is introduced by means of an additional conservation equation, given by:

$$\frac{\partial(kh)}{\partial t} + \frac{\partial(khu_i)}{\partial x_i} - \frac{\partial}{\partial x_j} \frac{\nu_t}{\sigma_k} \frac{\partial kh}{\partial x_j} = hP_k - h\epsilon \quad (1.9)$$

where  $P_k$  is the TKE production rate and  $\epsilon$  is its dissipation rate. The  $P_k$  term is locally defined, whereas the  $\epsilon$  term is solved through another conservation equation, as:

$$\frac{\partial(\epsilon h)}{\partial t} + \frac{\partial(\epsilon hu_i)}{\partial x_i} - \frac{\partial}{\partial x_j} \frac{\nu_t}{\sigma_\epsilon} \frac{\partial \epsilon h}{\partial x_j} = hS_\epsilon \quad (1.10)$$

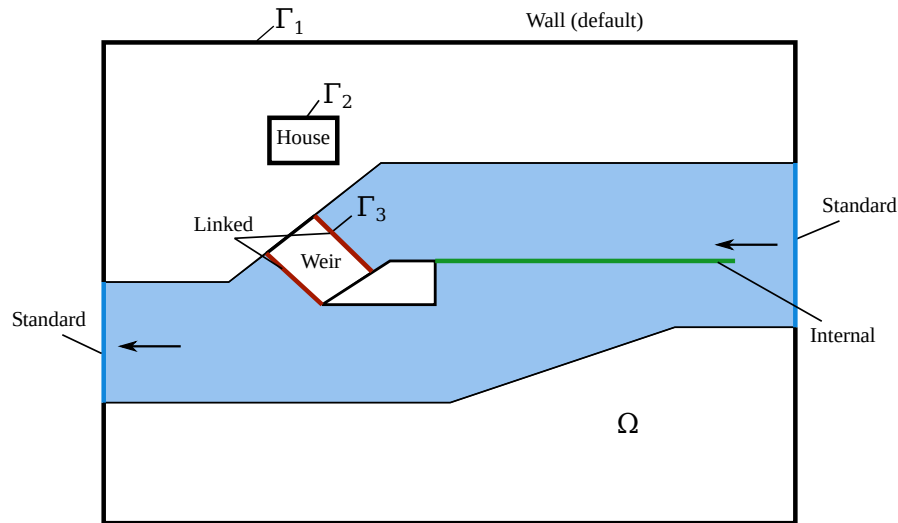
where  $S_\epsilon$  is an aggregate net source of  $\epsilon$ . In the Standard  $\kappa$ - $\epsilon$  model, the source terms for both  $\kappa$  and  $\epsilon$  are defined as:

$$P_k = 2\nu_t S_{ij} S_{ij} + \frac{u_f^3}{c_f h} \quad (1.11)$$

$$S_\epsilon = 2c_{\epsilon_1} \frac{\epsilon}{\kappa} \nu_t S_{ij} S_{ij} + 3.6 \frac{c_{\epsilon_2} \sqrt{c_\mu} u_f^4}{h^2 c_f^{3/4}} - c_{\epsilon_2} \frac{\epsilon^2}{\kappa} \quad (1.12)$$

and the constants of the standard model are  $c_\mu = 0.09$ ,  $\sigma_k = 1.00$ ,  $\sigma_\epsilon = 1.30$ ,  $c_{\epsilon_1} = 1.44$  and  $c_{\epsilon_2} = 1.92$ . The turbulent viscosity is finally obtained, locally, as

$$\nu_t = c_\mu \frac{\kappa^2}{\epsilon} \quad (1.13)$$



**Figure 1.1** Modeling domain and types of boundary conditions available. The flow is from right to left and a side weir (green line) divides the channel into a lower and an upper channel through the weir. **Standard** or **linked** boundary conditions must be provided at  $\Gamma_1$ ,  $\Gamma_2$  and  $\Gamma_3$  while **internal** boundary conditions can be specified in any place within  $\Omega$

### 1.1.3.3 Lateral Inflow/Outflow

$S_h$  is used to represent additional sources of water like rainfall and springs or water abstraction (sink) and are allocated on a set of elements defined by regions. The external source can be specified as total discharge [ $m^3/s$ ] or distributed over time [ $mm/h$ ]. Different approaches are used to manage the behaviour of the external sources:

- Exact: The specified water volume is added or extracted (non conservative)
- Available: The specified water volume to extract is limited by the available water volume in the elements (conservative)
- Infinity: All available water will be abstracted (conservative)

Addition of water always follows the “Exact” behaviour as there is no upper limit. The abstraction of water could also follow the “Exact” behaviour but the simulation might end abruptly if the available water volume is smaller than the volume prescribed. Therefore, the “Available” behaviour aims to avoid this situation. The “Infinity” behaviour abstracts all available water volume.

### 1.1.4 Boundary Conditions

After the specification of the *closure relations* there are now three equations and three unknowns, namely  $h$ ,  $q_x$  and  $q_y$ . In principle, given initial and boundary conditions, one should be able to solve system (eq. 1.1) for  $h$ ,  $q_x$  and  $q_y$  as functions of space  $x$ ,  $y$ , and time  $t$ . Given the modeling domain described in Figure 1.1, boundary conditions are required at the domain boundary  $\Gamma$  and optionally can be specified within the interior domain  $\Omega$ .

Therefore, three different types of boundary conditions can be defined:

- Standard boundary conditions: located at the domain boundary  $\Gamma_i$
- Linked boundary conditions: located at the domain boundary  $\Gamma_i$  or inside the domain  $\Omega$
- Internal boundary conditions: located inside the domain  $\Omega$

*Standard boundaries* (at  $\Gamma$ ) represent the limits of the computational domain possibly including also buildings, weirs or structures for water intake (see Figure 1.1).

#### 1.1.4.1 Standard Boundary Conditions

At the standard boundaries, two different types of boundary conditions can be specified: wall or flow boundaries. Flow boundary conditions allow the flow to enter or leave the domain while wall boundary conditions express no mass flux over the boundary. By default, the external boundaries of the domain are set as wall boundaries.

##### 1.1.4.1.1 Wall Boundaries

The *Wall* or *reflective* boundary consider the boundary at  $\Gamma_i$  and suppose it physically consists of a fixed, reflective impermeable wall. Then the physical situation is modelled imposing that:

$$\rho \vec{u} \cdot \vec{n} = 0 \quad ; \quad \frac{\partial \vec{u}}{\partial \vec{n}} = 0 \quad (1.14)$$

Where  $\vec{n}$  is the outward directed unit vector perpendicular to the wall and  $\vec{u} = (u, v)^T$  is the velocity vector. The static pressure is assumed to be zero.

##### 1.1.4.1.2 Flow Boundaries

The *Flow* boundary conditions are defined as *inflow* if they let water entering or as *outflow* if they let water leaving the domain. Flow boundaries are further distinguished into *Standard* and *Linked*. The former are applied on the boundary domain  $\Gamma$ , while the latter establish a *link* between two portions of the domain.

##### Standard

*Inflow boundaries:*

This boundary requires the specification of a value for the total volume discharge  $Q$ , [ $m^3/s$ ], which is then divided by the length of the boundary  $\Gamma$  and projected orthogonally to the boundary to obtain the values of  $q_x$  and  $q_y$ . In case of supercritical flow the following possibilities to specify the value of the water depth  $h$  are possible:

- *uniform in*:  $h$  is calculated assuming that local uniform flow conditions. The calculation proceeds as follows:

$$h = \sqrt[3]{\frac{(Q/b)^2}{gc_f^2 s}} \quad (1.15)$$

where  $c_f$  is the Chézy coefficient,  $b$  is the entire length of the boundary  $\Gamma$  and  $s$  is the value of the local bed slope that must be specified.

- *froude\_in*: In this case the flow depth  $h$  is calculated as follows:

$$h = \sqrt[3]{\frac{(Q/b)^2}{gFr^2}} \quad (1.16)$$

where  $b$  is the entire length of the boundary and  $Fr$  is the value of the local Froude number that must be specified

- *zhydrograph*: The water surface elevation (wse) at the boundary must be specified by the user. The depth is calculated as:

$$h = wse - z_B \quad (1.17)$$

where  $z_B$  is the bottom elevation at the boundary. The flow velocity at the boundary is set to zero.

#### *Outflow boundaries:*

At the outflow boundaries a value for the water depth  $h$  must be specified. These are the possible options:

- *uniform\_out*: the water depth  $h$  is calculated using equation (eq. 1.15) specifying a value for the total discharge  $Q$  and a local bed slope  $s$ . Uniform flow is calculated based on given slope and cell state at boundary (eq. 1.15).
- *weir\_out\_constant* and *weir\_out\_dynamic*: These boundary conditions establish a relation between the approaching discharge  $q$  constant and the water depth using the Poleni weir formula:

$$q = \frac{2}{3}\mu\sqrt{2g(h_{up} - w)^3} \quad (1.18)$$

where  $h_{up}$  is the water depth of the approaching flow and  $w$  is the weir elevation. The Poleni factor  $\mu$  can be either set as constant ( $\mu = 0.75$  by default) or dynamically evaluated as:

$$\mu = \frac{0.611}{a} \frac{0.75}{b} \frac{h_{up} - z_w}{w} \quad (1.19)$$

where  $a$  and  $b$  must be specified by the user in the case of *weir\_out\_dynamic* (default values are  $a = 0.611$  and  $b = 0.075$ ).

- *hqrelation\_out*: The discharge is determined as a function of the water surface elevation, thus a stage-discharge-relation has to be specified.
- *zhydrograph*: Sets a fixed water surface elevation (wse) at the boundary. The wse [m] at the boundary must be specified by the user. The depth is calculated as:

$$h = wse - z_B \quad (1.20)$$



where  $z_B$  is the bottom elevation at the boundary. The flow velocity is calculated with the Riemann solver (HLLC).

- *zero\_gradient\_out* (scientific use only): Transmissive, or transparent boundaries allow the passage of waves without any effect on them. This is mathematically obtained imposing over the entire length of the boundary that:

$$\rho \vec{u} \cdot \vec{n} = \text{const} \quad ; \quad \frac{\partial \vec{u}}{\partial \vec{n}} = 0 \quad (1.21)$$

In this case there is no need to specify further parameters.

*Note: This is boundary condition should **not** be used for practical problems and is intended for scientific use only.*

#### 1.1.4.2 Linked Boundary Conditions

This type of boundaries establish a *link* between within a certain region of the domain where equations are not solved. Once this domain portion is identified the two boundaries, between which the link is established, must be specified. Let us call them  $\Gamma_{in}$  and  $\Gamma_{out}$ . Then, one inflow boundary condition must be specified at  $\Gamma_{in}$  and one outflow boundary condition at  $\Gamma_{out}$  while in the remaining boundaries wall conditions are automatically assigned. Not necessarily,  $\Gamma_{in}$  and  $\Gamma_{out}$  must have the same number of elements.

Linked boundaries can describe a  $h - Q$  relation, a weir, or prescribed water surface elevation, i.e.:

- *weir\_linked\_constant* and *weir\_linked\_dynamic*: Similar to the standard weir boundary, the weir height  $w$  has to be specified.
- *hqrelation\_linked*: The discharge is determined as a function of the water surface elevation, thus a stage-discharge-relation has to be specified.
- *2way\_hqrelation\_linked*: The internal boundary works as dynamic wall that is controlled by water surface elevation thresholds. If the upper water surface elevation threshold is reached, the internal boundary is removed until the water level reaches the lower water surface elevation, where the wall is re-established.
- *hydrograph\_linked*: Sets a prescribed water surface elevation (WSE) at the upstream boundary. The flux over the upstream boundary is calculated with the Riemann solver and used as inflow in the downstream boundary.

The inflow conditions at the downstream boundary can be controlled via the tag *type\_downstream*. Four options are available as downstream inflow condition. For all options, the mass inflow flux is calculated based on outflow at upstream boundary. The difference between the options lies in the calculation of the momentum flux over the boundary:

- *no\_momentum*: No momentum flux over the boundary.
- *froude\_downstream*: Momentum flux calculated based on Froude number in the elements of the downstream nodestring.

- *froude*: Momentum flux calculated based on the Froude number specified via the parameter *froude\_number*.
- *uniform*: Momentum flux calculated based on uniform flow conditions for the slope specified via the parameter *slope*.

In BASEMENT versions without the option *type\_downstream*, the various linked boundary conditions differ in how the momentum flux at the downstream boundary was calculated. The following combinations correspond to the implementation of the boundary conditions in previous versions:

type	type_downstream
weir_linked_constant	froude_downstream
weir_linked_dynamic	froude_downstream
hqrelation_linked	no_momentum
2way_hqrelation_linked	no_momentum
zhydrograph_linked	no_momentum
zhydrograph_linked_kinE	froude_downstream

#### 1.1.4.3 Internal Boundary Conditions

The internal boundary condition allows a direct cell-cell relation due to the exact same number of elements on the left and on right side of the boundary. Internal boundary conditions can be used to specify internal walls, dynamic walls or an h-Q relation.

- *wall\_internal*: The wall conditions (eq. 1.14) are applied on both sides of the internal boundary.
- *dynamic\_wall\_internal*: The wall conditions are applied on the internal boundary until reaching a threshold value (time or water depth) after which the wall is removed.
- *hqrelation\_internal*: A stage-discharge relation is applied on one side of the internal boundary, while on the other side, wall conditions apply (unidirectional flow).

For the *hqrelation\_internal* boundary condition, the inflow conditions at the downstream boundary can be controlled via the tag *type\_downstream*. Four options are available as downstream inflow condition. For all options, the mass inflow flux is calculated based on outflow at upstream boundary. The difference between the options lies in the calculation of the momentum flux over the boundary:

- *no\_momentum*: No momentum flux over the boundary.
- *froude\_downstream*: Momentum flux calculated based on Froude number in the elements of the downstream nodestring.
- *froude*: Momentum flux calculated based on the Froude number specified via the parameter *froude\_number*.
- *uniform*: Momentum flux calculated based on uniform flow conditions for the slope specified via the parameter *slope*.

In BASEMENT versions without the option *type\_downstream*, the various linked boundary conditions differ in how the momentum flux at the downstream boundary was calculated. The following combinations correspond to the implementation of the boundary conditions in previous versions:

type	type_downstream
hqrelation_internal	no_momentum

### 1.1.5 Flood Tracking

The flood tracking aims at extracting the flood arrival time, the maximum water depth, flow velocity, specific discharge and bed shear stress along the numerical simulation and over a selected domain area. The flood tracking provides outputs within a tracking time step defined by the user.

## 1.2 Morphodynamics

### 1.2.1 Introduction

Morphodynamic models provide scientific frameworks for advancing our understanding of river systems. The research on involved topics is an important and socially relevant undertaking regarding our environment. Nowadays numerical models are used for different purposes, from answering questions about basic morphodynamic research to managing complex river engineering problems. Due to increasing computer power and the development of advanced numerical techniques, morphodynamic models are now more and more used to predict the bed patterns evolution to a broad spectrum of spatial and temporal scales. The development and the success of application of such models are based upon a wide range of disciplines from applied mathematics for the numerical solution of the equations to geomorphology for the physical interpretation of the results.

Applications of morphodynamic models include:

- Damming of river basins
- Morphological changes due to width changes (e.g. River widenings)
- Effects of sediment mining
- River straightening

### 1.2.2 Bedload Sediment Transport

#### 1.2.2.1 Governing Equations for Uniform Sediment Transport

The governing equations are obtained under shallow water conditions imposing mass conservation for the fluid and solid phases and the momentum principle to a flow in an open channel with a cohesionless bottom.

Introducing a Cartesian reference system  $(x; y; z)$  in which the  $z$  axis is vertical and the  $x - y$  plane is horizontal, the system of governing equations is described by the system of equations (eq. 1.1) for hydrodynamics coupled with one equation for the conservation of the total sediment mass. The conservation of sediment mass is ensured by the Exner equation (eq. 1.22), named after the Austrian sedimentologist Felix M. Exner (Exner, 1925). The Exner equation allows to describe the bed evolution due to erosion or deposition, which results in the elevation change of the actual bed level  $z_B$ :

$$(1 - p) \frac{\partial z_B}{\partial t} + \frac{\partial q_{B_x}}{\partial x} + \frac{\partial q_{B_y}}{\partial y} - Sl_b = S_s \quad (1.22)$$

where  $p$  is the porosity,  $Sl_b$  is the source term per unit width specifying local input or output of sediment material (e.g. slope collapse or excavation), and  $\vec{q}_B = \begin{pmatrix} q_{B_x} \\ q_{B_y} \end{pmatrix}$  is the specific bedload flux. The term  $S_s$  describes the exchange per unit width between the sediment and the suspended material (see Section 1.3). The Exner equation describes the bed evolution due to erosion or deposition processes, which results in changes of the bed level  $z_B$ .

The Exner equation is solved in a decoupled way, meaning that the shallow water equations and the Exner equation are solved in sequence. This approach makes the assumption that the bedload flux is much slower than the water flow velocity (Soares-Frazão and Zech, 2011).

### 1.2.2.2 Governing Equations for Mixed-Size Sediment Transport

#### 1.2.2.2.1 Hirano-Exner Model

The mixed-size sediment transport is based on the Hirano-Exner model Hirano (1971), which extends the Exner equation to sediment mixtures consisting of  $N$  discrete grain size classes with grain size diameters  $d_g$ . The subscript  $g$  denotes the grain size class. The model is based on the assumption, that the river bed can be distinguished into different vertical layers. Only the sediment in the uppermost layer, also called active layer, is in directly exposed to the forces of the flow and therefore, only sediment from this layer is available for bedload transport. The active layer is the control volume for the mass conservation equation of each grain size class (Eq. eq. 1.23) and has a finite thickness  $L_a$  (see Fig. Figure 1.2). The sediment mixture in the active layer is determined by the active layer fractions  $F_g$  of each grain class  $g$ . The substrate below the active layer may be further distinguished into  $j$  sublayers with sublayer fractions  $f_{j,g}$ . The conservation of mass for each grain size class  $g$  in the active layer is then given as

$$(1 - p) \frac{\partial (F_g \cdot L_a)}{\partial t} + \frac{\partial q_{B,g_x}}{\partial x} + \frac{\partial q_{B,g_y}}{\partial y} + S_{s,g} - Sl_{b,g} - S f_g = 0 \quad \text{for } g = 1, \dots, N, \quad (1.23)$$

where  $p$  is the porosity,  $S_{s,g}$  is the source term for the exchange of sediment with the water column (suspended sediment transport),  $Sl_{b,g}$  is the source term per unit area specifying local input or output of sediment material (e.g. sediment replenishment or excavation), and  $\vec{q}_{B,g} = (q_{B,g_x}, q_{B,g_y})$  is the specific bedload flux of grain fraction  $g$ . The source term

$Sf_g$  for the flux between the active layer and the uppermost sublayer due to changes in the bed level  $z_B$  or in the active layer thickness  $L_a$  is given as

$$Sf_g = -(1-p)f_{I,g} \frac{\partial(z_B - L_a)}{\partial t}, \quad (1.24)$$

where the fraction  $f_{I,g}$  depends on whether the lower edge of the active layer defined at  $z = z_B - L_a$  moves up- or downwards as:

$$f_{I,g} = \begin{cases} f_{1,g} & \text{if } \frac{\partial(z_B - L_a)}{\partial t} < 0, \\ F_g & \text{if } \frac{\partial(z_B - L_a)}{\partial t} > 0. \end{cases} \quad (1.25)$$

Here,  $f_{1,g}$  denotes the fraction of grain class  $g$  of the uppermost sublayer (index 1).

The conservation of the sediment fractions in the active layer for each sublayer  $j$  are given as:

$$\sum_{g=1}^N F_g = 1 \quad \text{and} \quad \sum_{g=1}^N f_{j,g} = 1 \quad \text{for } j = 1, \dots$$

The global conservation of sediment mass is defined by the extension of the Exner equation:

$$(1-p) \frac{\partial z_B}{\partial t} + \sum_{g=1}^N \left( \frac{\partial q_{B,gx}}{\partial x} + \frac{\partial q_{B,gy}}{\partial y} + S_{s,g} - Sl_{b,g} \right) = 0 \quad \text{for } g = 1, \dots, N. \quad (1.26)$$

Finally, we require closure relations for the sediment transport rate  $\vec{q}_{B,g}$ , which are obtained from a sediment transport formula (see Section Section 1.2.2.4).

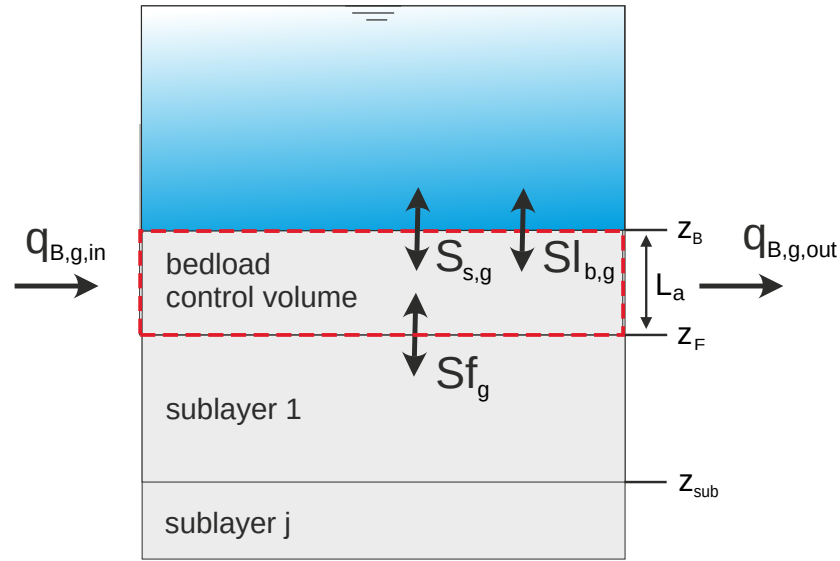
#### 1.2.2.2.2 Active layer thickness

The thickness of the active layer (bed load control volume)  $L_a$  is an important calibration parameter. This parameter influences significantly the grain sorting process. The active layer thickness can be specified in the ACTIVE\_LAYER block within the PARAMETER block. The *active\_layer\_type* can either be set to:

- constant: The active layer thickness is constant and defined via the parameter *active\_layer\_thickness*
- d90: The active layer thickness is calculated at each time step as  $L_a = kd_{90}$ , where  $k$  is the user specified *active\_layer\_factor* and  $d_{90}$  is the characteristic grain diameter for which 90% of the sediment in the active layer is smaller than  $d_{90}$ .

#### 1.2.2.2.3 Limitations

For computational performance reasons, the maximum available grain classes is limited to 10. Similarly, the number of sublayers available is limited to 2. Coupling of bedload and suspended load transport is not yet implemented ( $S_{s,g}$  for  $g > 1$ ), i.e. suspended load transport only works for uniform sediment transport models.



**Figure 1.2** Definition sketch of overall control volume (red) of bed material sorting equation

### 1.2.2.3 Threshold Conditions for Sediment Movement

The key dimensionless parameter quantifying sediment mobility is the Shields parameter defined as:

$$\theta = \frac{|\vec{\tau}_b|}{(\rho_s - \rho)gd} \quad (1.27)$$

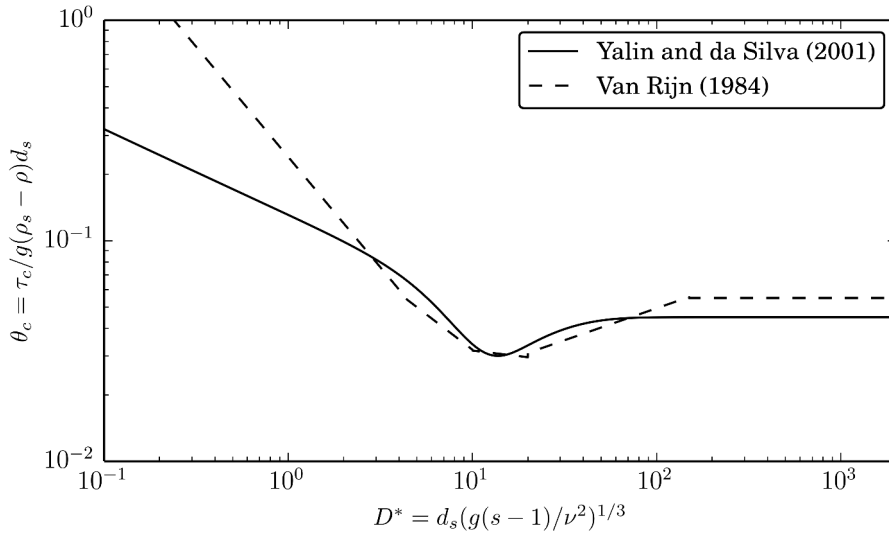
where  $vec\tau_b$  is the bed shear stress (drag force acting on the particle),  $d$  is the sediment diameter,  $\rho$  and  $\rho_s$  are the water and sediment density, respectively. The Shields parameter can be interpreted as the ratio scaling the impelling force of flow drag acting on a particle to the Coulomb force resisting motion acting on the same particle.

The bed shear stress  $\vec{\tau}_b = \begin{pmatrix} \tau_{bx} \\ \tau_{by} \end{pmatrix}$  is usually estimated by a closure condition using an empirical or semi-empirical formula. Here we use the quadratic friction law which relates the depth-averaged velocities to the bed shear stress as follows:

$$\tau_{bx} = \rho \frac{|\vec{u}|u}{c_f^2} \quad ; \quad \tau_{by} = \rho \frac{|\vec{u}|v}{c_f^2} \quad (1.28)$$

where  $\vec{u} = \begin{pmatrix} u \\ v \end{pmatrix}$  is the flow velocity vector,  $\rho$  is the density of water and  $c_f$  is the dimensionless Chézy friction coefficient as defined in Section 1.1.3.1.

When a granular bed is subjected to a turbulent flow, it is found that virtually no motion of the grains is observed below a critical value ( $\theta_{cr}$ ) of the Shields parameter. According to the Shields' theory Shields (1936),  $\theta_{cr}$  can be expressed as a function of the Reynolds number  $Re^* = \frac{du_*}{\nu}$ . Alternatively, the diagram of incipient motion (see Figure 1.3) can be



**Figure 1.3** Modified Shields diagram for initiation of sediment motion

plotted as a function of the dimensionless grain diameter  $D^*$  ( $\theta_{cr} = f(D^*)$ ), where

$$D^* = d \left[ \frac{g(s-1)}{\nu^2} \right]^{1/3}$$

The curve representing the particle incipient motion ( $\theta = \theta_{cr}$ ) can be divided into three parts in the log-log graph:

- for  $D^* \leq 3$ , can be approximated by a linear segment;
- for  $3 \leq D^* \leq 100$  this is represented by a curve with a relative minimum;
- for  $D^* > 100$  by a constant trend. “

An explicit formulation of the Shields curve was proposed by Yalin and Silva (2001). It reads

$$\theta_{cr} = 0.13D^{*-0.392} \exp(-0.015D^*) + 0.045(1 - \exp(-0.068D^*)) \quad (1.29)$$

Two approaches for determining the dimensionless critical Shields parameter are available (`theta_critical_approach`):

- *constant*: The default dimensionless critical Shields parameter is constant and equal to the user-specified value `theta_critical` (default)
- *yalin*: The critical Shields parameter is determined with the approach of Yalin and Silva (2001) (Eq. eq. 1.29).  $D^*$  is calculated based on the arithmetic mean grain diameter of the active layer.

### 1.2.2.3.1 Influence of Local Slope on Incipient Motion

The threshold condition for incipient motion of grains developed by Shields is valid for almost horizontal bed. In case of sloped bed in flow direction or transverse to it, the stability of grains is either increased or reduced due to the gravity. The critical shear stress value can be adapted consequently to account for the influence of local slopes. One approach is to multiply the critical shear stress for almost horizontal bed  $\theta_{cr}$  with the correction factors  $k_l$  and  $k_t$  for the local bed slope in the longitudinal and transversal flow direction. In the following, the critical Shields stress corrected for arbitrary bed slope  $\delta$  is referred to as  $\theta_{c,\delta}$ , defined as:

$$\frac{\theta_{c,\delta}}{\theta_{cr}} = k_l k_t = k \quad (1.30)$$

The correction factor  $k_l$  and  $k_t$  are calculated as suggested by van Rijn (1989):

$$k_l = \cos \delta_l \left( 1 - \frac{\tan \delta_l}{\tan \gamma} \right) \quad (1.31)$$

$$k_t = \cos \delta_t \sqrt{1 - \frac{\tan^2 \delta_t}{\tan^2 \gamma}} \quad (1.32)$$

where  $\delta_l$  is the angle between the horizontal and the bed along flow direction,  $\delta_t$  is the slope angle transversal to the flow direction and  $\gamma$  is the angle of repose of the sediment material.

Other formulations are also available, as for example the one proposed by Chen et al. (2010):

$$k = \frac{1}{\tan \gamma} \left( \cos^2 \left( \frac{\pi}{2} - \delta_l \right) - 1 + \frac{1 + \tan^2 \gamma}{(1 + \tan^2 \delta_l + \tan^2 \delta_t)} \right)^{0.5} + \cos \left( \frac{\pi}{2} - \delta_l \right) \quad (1.33)$$

### 1.2.2.4 Closure Relations for Bedload Transport

In order to solve system (eq. 1.1) and equation (eq. 1.22) we need to specify the closure relations. For the friction terms  $S_{fx}$ ,  $S_{fy}$  and the value of lateral inflow/outflow discharge per unit width  $S_h$  we can use the relations already introduced in the Hydrodynamic part (Section 1.1.3). For the Exner equation we need relations quantifying the bedload discharge. Let us now introduce the dimensionless bedload transport rate  $\Phi$  also known as the Einstein bedload number, first introduced by Hans Albert Einstein in 1950, and given by

$$\Phi = \frac{q_B}{\sqrt{(s-1)gd^3}} \quad (1.34)$$

where  $s = \rho_s/\rho$ .

It is common practice to quantify bedload transport empirically relating  $\Phi$  with either the Shields stress  $\theta$  or the excess of the Shields stress  $\theta$  above some appropriately defined ‘‘critical’’ Shields stress ( $\theta - \theta_{cr}$ ). The critical Shields stress  $\theta_{cr}$  is defined so as to fit experimental or field data and provide a threshold for which the bedload transport rate is too low to be of interest. The Shields parameter, takes the following form



$$\theta = \frac{h\sqrt{S_{fx}^2 + S_{fy}^2}}{(s-1)d} \quad (1.35)$$

where  $h$  is the water depth,  $S_{fx}$  and  $S_{fy}$  the friction slope in x- and y-direction respectively,  $s = \rho_s/\rho_w$ , and  $d$  is the grain size diameter. Note that Eq. 1.27 and Eq. 1.35 are equivalent.

In what follows, we describe the bedload transport formulas that are implemented to calculate the transport capacity  $q_B = |\vec{q}_B|$ , where the specific bedload flux vector  $\vec{q}_B = (q_{B_x}, q_{B_y})$  generally has the same direction as the water flow.

For practical purposes, the bedload transport formula can be calibrated by an additional pre-factor (*factor*). The bedload transport capacity is obtained from the closure relation scaled by this pre-factor.

#### 1.2.2.4.1 Meyer-Peter and Müller (1948)

The bedload transport formula of Meyer-Peter and Müller (Meyer-Peter and Müller, 1948) defines the specific bedload transport rate  $q_B$  as:

$$q_B = \alpha(\theta - \theta_{cr})^m \sqrt{(s-1)gd^3} \quad (1.36)$$

Herein,  $\alpha$  denotes the bedload coefficient (originally  $\alpha = 8$ ),  $m$  the bedload exponent (originally  $m = 1.5$ ),  $\theta$  is the dimensionless bed shear stress (Shields parameter),  $\theta_{cr}$  is the critical dimensionless bed shear stress,  $d$  is the grain diameter,  $s = \rho_s/\rho$  and  $g$  stands for the gravitational acceleration. Meyer-Peter and Müller observed in their experiments that the first 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 were moving. They proposed the value of  $\theta_{cr} = 0.047$ . The formula of Meyer-Peter and Müller is applicable in particular for coarse sand and gravel with grain diameters larger than 1 mm (Malcherek, 2001).

The bedload coefficient  $\alpha$ , the exponent  $m$  and the critical Shields parameter  $\theta_{cr}$  can be adapted by the user in the MPM-like formula.

#### 1.2.2.4.2 Meyer-Peter and Müller (1948) with hiding and exposition effects after Ashida and Michiue (1971)

This closure relation for bedload transport of sediment mixtures extends the formula of Meyer-Peter and Müller (Meyer-Peter and Müller, 1948) by consideration of hiding and exposition effects according to the approach Ashida and Michiue (Ashida and Michiue, 1971). It defines the specific bedload transport rate  $q_B$  as

$$q_{B,g} = \alpha(\theta - \theta_{cr,g})^m \sqrt{(s-1)gd_m^3}, \quad (1.37)$$

where  $d_m$  is the geometric mean diameter of the sediment mixture in the active layer, and  $\theta_{cr,g}$  is the dimensionless critical shear for grain class  $g$  calculated as:

$$\theta_{cr,g} = \theta_{cr,ref} \xi_g. \quad (1.38)$$

The reference dimensionless critical shear stress  $\theta_{cr,ref}$  is usually defined with a fixed value (e.g.  $\theta_{cr,ref} = 0.047$ ) and the hiding factor  $\xi_g$  is defined as Ashida and Michiue (1971):

$$\xi_g = \begin{cases} [\log(19)/\log(19d_g/d_m)]^2 & d_g/d_m \geq 0.4 \\ 0.843d_m/d_g & d_g/d_m < 0.4 \end{cases} \quad (1.39)$$

$d_g$  and  $d_m$  denote the grain size diameter of grain class  $g$  and the geometric mean diameter of the sediment mixture in the active layer, respectively.

The bedload coefficient  $\alpha$ , the exponent  $m$  and the critical Shields parameter  $\theta_{cr}$  can be adapted by the user in the MPM-like formula.

#### 1.2.2.4.3 Grass Formula

The Grass formula (Grass, 1981) proposes a simple bedload transport formula, where  $q_b$  is a function of the flow velocity  $u$  and a dimensional constant  $\alpha$  and does not require the evaluation of the Shields stress:

$$q_B = \alpha(u - u_c)^m \cdot \sqrt{(s-1)gd^3} \quad (1.40)$$

where  $\alpha \in [0, 1]$  is a dimensional constant that encompasses the effects of grain size and kinematic viscosity and is usually determined from experimental data,  $u_c$  is the critical velocity and the exponent  $m$  is usually set to  $m = 3$ . The threshold condition for incipient motion of grains is typically set to zero, meaning that the bedload transport and the fluid motion start simultaneously. The coefficient  $\alpha$  characterizes the interaction between the bed and the fluid. If  $\alpha = 0$ , no sediment transport occurs. If  $\alpha = 1$  the interaction between the bed and fluid is the largest.

#### 1.2.2.4.4 Engelund and Hansen (1972)

Engelund and Hansen (1972) proposed a transport formula for uniform bed material taking into account at the same time the presence of both bed- and suspended-load. This formula is commonly used as a bulk load formula and reads

$$q_B = 0.05\sqrt{(s-1)gd^3} \cdot c_f^2\theta^{2.5} \quad (1.41)$$

where  $d$  denotes the median sediment size of the bed material,  $c_f$  the dimensionless Chézy friction coefficient and  $\theta$  is the dimensionless bed shear stress (see eq. 1.35). The Engelund and Hansen formula for bedload transport does not consider the critical shear stress as threshold condition for incipient motion.

#### 1.2.2.4.5 Smart & Jäggi (1983)

Smart and Jaeggi (1983) developed a bedload transport formula for steep channels using their own experimental results and the results of Meyer-Peter and Müller (Meyer-Peter and Müller, 1948). The specific bedload transport rate  $q_B$  is defined as:

$$q_B = \frac{\alpha}{(s-1)} \left(\frac{d_{90}}{d_{30}}\right)^{0.2} J^{0.6} |\bar{q}| (J - J_{cr}) \quad (1.42)$$

where  $s$  is the sediment density coefficient ( $s = \rho_s/\rho$ ),  $|\vec{q}|$  is the magnitude of the specific discharge and  $d_{30}$  and  $d_{90}$  are the characteristic grain size diameters, i.e. 30 % resp. 90 % (by weight) of the bed material are smaller. The energy slope  $J$  and the critical slope for the initiation of the bedload transport  $J_{cr}$  calculated as

$$J = \frac{\theta(s-1)d_m}{h} \quad (1.43)$$

$$J_{cr} = \frac{\theta_{cr}(s-1)d_m}{h} \quad (1.44)$$

where  $\theta$  is the dimensionless bed shear stress (see eq. 1.35),  $\theta_{cr}$  the critical dimensionless bed shear stress,  $d_m$  the mean grain size diameter and  $h$  the water depth. Smart and Jaeggi (1983) recommend values of  $\alpha = 4$  and  $\theta_{cr} = 0.05$ . The scope of application is for bed slopes  $0.005 \leq J \leq 0.2$  (Smart and Jaeggi, 1983).

### 1.2.2.5 Wilcock and Crowe (2003)

Wilcock and Crowe developed a sediment transport model for sand/gravel mixtures (Wilcock and Crowe, 2003), similar to Parker's model (Parker, 1990), and it was developed with a large experimental results dataset. It references 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 observed among earlier hiding functions implemented in other transport models, such as the Oak Creek and the Cambridge ones (Parker and Sutherland, 1990). The Wilcock and Crowe model (Wilcock and Crowe, 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. The specific bedload transport rate  $q_{B,g}$  for grain class  $g$  is defined as:

$$q_{B,g} = \frac{F_g u_*^3}{(s-1)g} W_g^*, \quad (1.45)$$

where  $F_g$  is the active layer fraction of grain size class  $g$ ,  $u_*$  is the shear velocity ( $u_* = (\tau/\rho)^{0.5}$ ,  $s$  is the sediment density coefficient ( $s = \rho_s/\rho$ ) with the sediment density  $\rho_s$  and water density  $\rho$ , and  $g$  the gravitational acceleration. The dimensionless bedload transport rate  $W_g^*$  is defined as “

$$W_g^*(\phi_g) = \begin{cases} 0.002\phi_g^{7.5} & \phi_g < 1.35 \\ 14 \left(1 - \frac{0.894}{\phi_g^{0.5}}\right)^{4.5} & \phi_g \geq 1.35 \end{cases} \quad (1.46)$$

with  $\phi_g = \tau/\tau_{r,g}$ . The reference shear stress  $\tau_{r,g}$  of grain class  $g$  (a surrogate for a critical Shields number) is defined as:

$$\tau_{r,g} = \tau_{r,g}^*(s-1)\rho g d_g \quad (1.47)$$

where  $d_g$  is the grain diameter if grain size class  $g$ , and the dimensionless reference shear stress  $\tau_{r,g}^*$  of grain class  $g$  as:

$$\tau_{r,g}^* = \tau_{r,m}^* \left( \frac{d_g}{d_m} \right)^{b-1} ; \quad \text{with } b = \frac{0.67}{1 + \exp\left(1.5 - \frac{d_g}{d_m}\right)} \quad (1.48)$$

and the dimensionless reference shear stress of the mean grain diameter if the surface (geometric)  $\tau_{r,m}^*$  is:

$$\tau_{r,m}^* = 0.021 + 0.015 \exp(-20F_s). \quad (1.49)$$

The non-linear effect of sand content  $F_s$  on gravel transport is taken into account in  $\tau_{r,m}^*$ . Wilcock and Crowe (Wilcock and Crowe, 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  $\tau_{r,m}^*$  (a surrogate for a critical Shields number) and the fraction sand in the active layer  $F_s$ . Note that  $\tau_{r,m}^*$  decreases as  $F_s$  increases, causing an increase of  $\phi_g$  and in turn of the fraction bedload  $q_{B,g}$ .

#### 1.2.2.5.1 Hunziker and Jaeggi (2002)

Hunziker and Jaeggi (Hunziker and Jaeggi, 2002) proposed a bed load formula for fractional bed load transport of graded sediment (also in (Hunziker, 1995)):

$$q_{Bg} = 5\beta_g [\xi_g(\theta'_{dms} - \theta_{cdms})]^{3/2} \sqrt{(s-1)gd_{ms}^3} \quad (1.50)$$

where  $\theta'_{dms}$  denotes the Shields parameter of the mean grain size of the surface bed material  $d_{ms}$  according to eq. 1.51,  $\xi_g$  denotes the hiding function applied on the excess shear stress  $(\theta'_{dms} - \theta_{cdms})$ .

$$\theta'_{dms} = \frac{\tau'_b}{\rho_w (s-1) d_{ms}} \quad (1.51)$$

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  $\theta_{cdms}$  of the mean grain size diameter is determined according to

$$\theta_{cdms} = \theta_{ce} \left( \frac{d_{mo}}{d_{ms}} \right)^{0.33} \quad (1.52)$$

where  $\theta_{ce}$  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

$$\frac{d_{ms}}{d_{mo}} = 0.0163\theta'^{-1.45}_{dms} + 0.6 \quad (1.53)$$

Finally, the hiding function is determined as

$$\xi_g = \left( \frac{d_g}{d_{ms}} \right)^{-\alpha} \quad (1.54)$$

where  $\alpha$  is an empirical parameter depending on the Shields parameter (see also Hunziker and Jaeggi (2002)) according to eq. 1.55, which is limited to a range between  $-0.4$  and  $2.0$ .

$$\alpha = 0.011\theta'_{dms}{}^{-1.5} - 0.3 \quad (1.55)$$

### 1.2.2.6 Correction of Bedload Direction

The 2D projection of the solid discharge along  $x$  and  $y$  is obtained through standard procedures, that are mostly based on empirical basis and which account for the downward effect of gravity on sediment particles due to local bed slope and the presence of spiral flow motion in curved reaches.

#### 1.2.2.6.1 Lateral Bed Slope Effect

Empirical bedload formulas were originally derived for situations where bed slope equals flow direction. However, in case of lateral bed slope with respect to flow direction, the bedload direction differs from the flow direction due to gravity acting on the bed material. Figure 1.4 illustrates the deviation of the bedload transport direction due to lateral bed slope in a Cartesian coordinate system.

The bedload direction is corrected for lateral bed slope based on the following approach (e.g. see Ikeda (1982) and Talmon et al. (1995)):

$$\tan \varphi_b = -f(\theta) \cdot \vec{s} \cdot \vec{n}_q \quad \text{for} \quad \vec{s} \cdot \vec{n}_q < 0 \quad (1.56)$$

$$f(\theta) = N_l \left( \frac{\theta_{cr}}{\theta} \right)^{M_l} \quad (1.57)$$

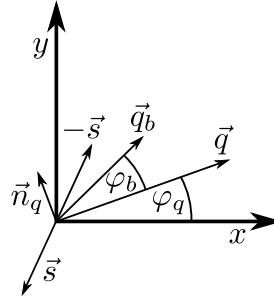
where  $\varphi_b$  = bedload direction with respect to the flow vector  $\vec{q}$ ,  $N_l$  = lateral transport factor ( $0.75 \leq N_l \leq 2.63$ ),  $M_l$  = lateral transport exponent (typically  $M_l = 0.5$ ),  $\vec{s} = \left( \frac{\partial z_B}{\partial x}, \frac{\partial z_B}{\partial y} \right)$  bed slope (positive uphill, negative downhill),  $\vec{n}_q$  = unit vector perpendicular to  $\vec{q}$  pointing in downhill direction ( $\vec{s} \cdot \vec{n}_q < 0$ ),  $\theta$  = effective dimensionless shear stress and  $\theta_{cr}$  = critical dimensionless shear stress of sediment.

The direction of the bedload transport under the influence of lateral bed slope is written as:

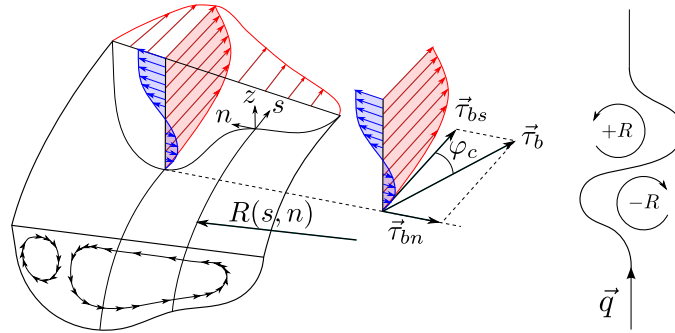
$$\frac{q_{B_y}}{q_{B_x}} = \tan(\varphi_b + \varphi_q) \quad (1.58)$$

#### 1.2.2.6.2 Curvature Effect

Curvature in rivers may cause deviation of the bedload direction from the depth averaged flow direction. Due to three dimensional spiral flow motion, the bedload direction tends to



**Figure 1.4** Bed load transport deviation angle  $\varphi_b$  from the flow direction  $\vec{q}$  due to the lateral bed slope  $\vec{s}$  (Vonwiller, 2017)



**Figure 1.5** Effect of spiral motion in river bend on bed shear stress  $\vec{\tau}_b$  with deviation angle from main flow direction  $\varphi_c$  (Vonwiller, 2017)

point towards the inner side of the curve, while the flow direction points towards the outer side (Figure 1.5). This curvature effect is taken into account according to an approach proposed by Engelund (1974), where the deviation angle  $\varphi_c$  of the bottom shear stress  $\vec{\tau}_b$  (positive counterclockwise and vice versa) from the main flow direction is determined as

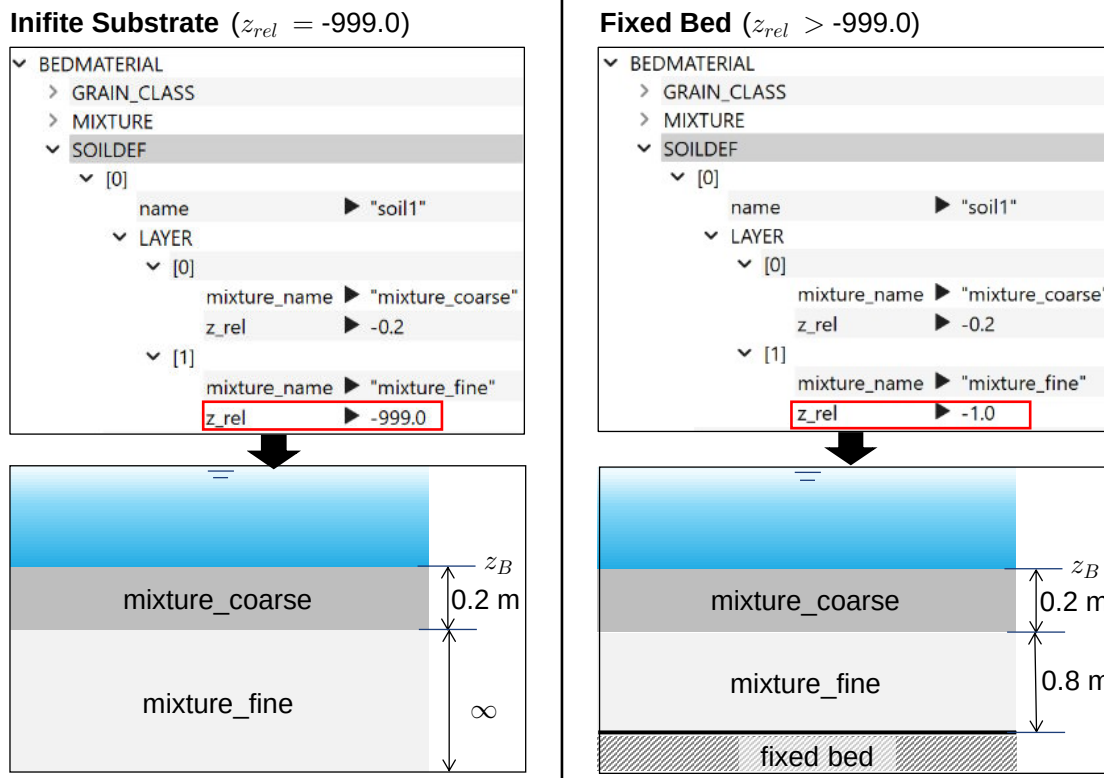
$$\tan \varphi_c = \frac{|\vec{\tau}_{bn}|}{|\vec{\tau}_{bs}|} = -N_* \frac{h}{R} \quad (1.59)$$

where  $\vec{\tau}_{bn}$  and  $\vec{\tau}_{bs}$  are the bed shear stress normal to and in the flow direction respectively,  $h$  denotes the water depth,  $N_*$  is 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, 1961).

### 1.2.2.7 Bed Material and Fixed Bed Concept

With the feature of the bedload transport for sediment mixtures, the concept of soil layers, as it is implemented in the BASEMD module, was adopted for BASEHPC. Soils consisting of different vertical layers can be defined via the SOILDEF block to characterize the bed substrate. Soils can subsequently be assigned to different regions of the computational



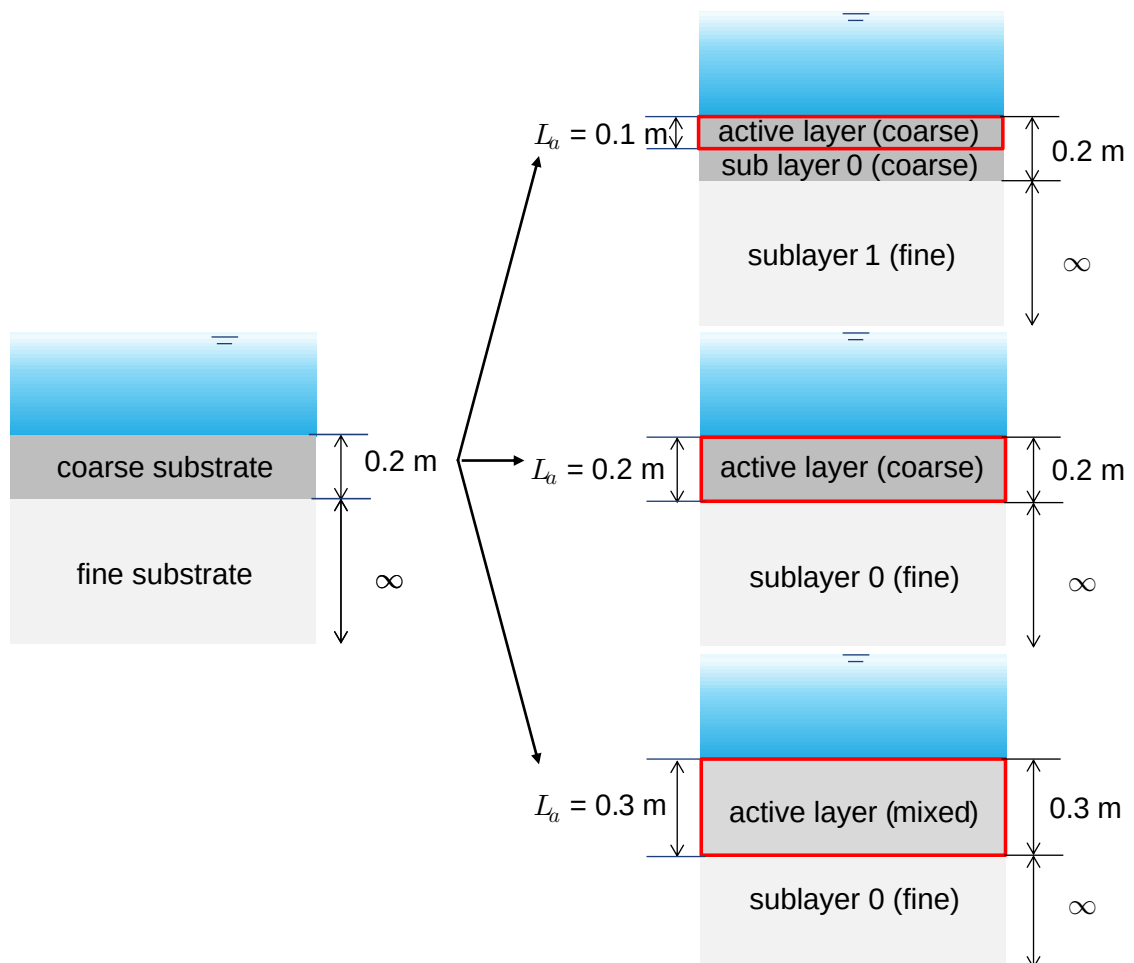
**Figure 1.6** Definition of an infinite substrate by using default value of  $z_{rel} = -999.0$  for the lowest soil layer or definition of a fixed bed by using a value  $z_{rel} > -999.0$

domain within the SOIL\_ASSIGNMENT block. Currently, the number of layers per soil is limited to 2 for reasons of computational efficiency. Each layer is defined through a sediment mixture and an elevation relative to the bed elevation ( $z_{rel} \leq 0$ ), which defines the lower edge of the layer. When a new layer is created in the graphical user interface (GUI), the default relative elevation for is -999.0 meters. Setting a  $z_{rel}$  value for the lowest layer of a soil other than the default value of -999.0, is equivalent to defining a fixed bed elevation at this relative elevation, as illustrated in Fig. Figure 1.6.

The soil definition in combination with the active layer thickness detmines the initialization of the vertical bed substrate. An example is given in Figure Figure 1.7. In the example, the soil is characterized by two layers: a upper layer with coarse sediment of 0.2 m thickness and a lower layer with finer sediment with infinite thickness. The choice of the active layer thickness  $L_a$  either smaller, equal or larger than the upper soil layer affects the number of initialized sublayers, as well as the composition of the active layer at the start of the simulation.

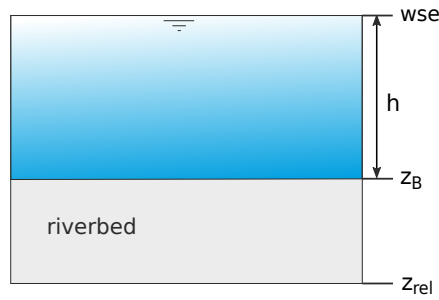
For restart or re-run simulations, the composition of bed material, including the active layer and sublayer composition can be provided from a result H5-file via the INITIAL block inside the BEDMATERIAL block.

Morphodynamic simulations generate deposition and erosion patterns of the riverbed. Erosion processes, if not limited, can proceed indefinitely in the vertical direction. This limit can be imposed by defining a non-erodible fixed bed elevation  $z_{rel}$ , below which the river bed is considered as *fixed*. This threshold also determines the amount of sediment available for transport (see Figure 1.8). There are three ways to define a fixed bed elevation:



**Figure 1.7** The initialization of the active layer and sublayers is determined by the soil layers and the active layer thickness





**Figure 1.8** Fixed bed concept and definition

(1) via regions in the FIXED\_BED block, or (2) via a separate .2dm mesh file in the FIXED\_BED block, or (3) via layers in the SOILDEF block, as mentioned above. If fixed bed elevations are defined both in the FIXED\_BED and SOILDEF blocks, definitions in FIXED\_BED block overwrite fixed bed information from the SOILDEF block.

When the fixed bed elevation is specified via regions or via layers, the fixed bed elevation must be provided relative to the initial bottom elevation  $z_B$  with  $z_{rel} \leq 0$ . When the fixed bed elevation is specified via a mesh, the elevation in the separate .2dm mesh file must correspond to the absolute fixed bed elevation  $z_{fix}$  [m]. Moreover, the fixed bed mesh file must have the exact same topology as the original computational mesh. In case the specified mesh only has elevation information on the nodes, the elevation is interpolated with the same method as specified in the INTERPOLATION block (default: mean). If the fixed bed elevation of the fixed bed mesh exceeds the bottom elevation of the computational mesh, the fixed bed elevation is defined at the elevation of the computational mesh.

The accuracy of the fixed bed correction is guaranteed by defining the maximal overshoot below the fix bed elevation and the maximal number of iterations required for the correction.

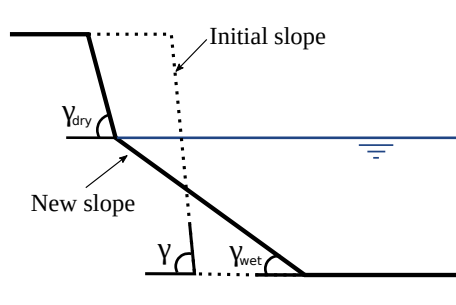
**Warning:** Defining a fixed bed elevation can slow down computations compared to simulations with a mobile bed.

### 1.2.2.8 Gravitational Transport

Gravitational induced riverbank or sidewall failures are significant aspects concerning erosion and transport modelling. Such processes may play an important role in many 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. The main idea of the implemented geometrical approach is to assume that a slope failure takes place if the local bed slope  $\gamma$  becomes steeper than a critical slope  $\gamma_{cr}$  (Figure 1.9).

$$q_{B,grav} = \begin{cases} 0 & \text{if } (\gamma \leq \gamma_{cr}) \\ f(\gamma, \gamma_{cr}) & \text{if } (\gamma > \gamma_{cr}) \end{cases} \quad (1.60)$$

The sliding material is moved from the sediment element with higher elevation to the lower



**Figure 1.9** Critical failure angles for slope collapse

situated element until the stable condition:  $\gamma \leq \gamma_{cr}$  is reached. Two 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  $\gamma_{dry}$ , 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  $\gamma_{wet}$  which is in the range of the material's angle of repose

#### 1.2.2.8.1 Calculation Procedure

The flux due to gravitational transport  $q_{B,grav}$  is calculated with the following procedure, by looping over each element of the computational grid:

1. In a first step, the local bed slope  $\gamma_i$  is calculated with respect to each neighbouring element  $i$ , where  $z$  is the bed elevation of the main element,  $z_i$  is the bed elevation of the neighbour element  $i$  and  $d_i$  is the distance between the element centers.

$$\gamma_i = \text{textarctan} \left( \frac{z - z_i}{d_i} \right) \quad (1.61)$$

2. For local bed slopes  $\gamma_i$  exceeding the critical slope  $\gamma_{cr}$ , the new bed elevations  $z_{new}$  and  $z_{i,new}$  are determined such that the stable condition  $\gamma_i = \gamma_{cr}$  is reached for all neighbouring cells  $i$ . The critical angle  $\gamma_{cr}$  is selected according to eq. 1.62, where  $h$  is the water depth in the main element and  $h_{min}$  is the user-specified minimum water depth.

$$\gamma_{cr} = \begin{cases} \gamma_{wet} & \text{if } h \geq h_{min} \\ \gamma_{dry} & \text{if } h < h_{min} \end{cases} \quad (1.62)$$

3. If the calculated bed elevation change of the main element  $\delta_z = z - z_{new}$  is smaller than the user-specified parameter  $min\_bed\_change$  (default: 0.001 m)  $\delta_{z,min}$ , no gravitational transport occurs to avoid oscillatory behaviour and to reduce the

computational effort. If the minimum bed elevation change  $\delta_{z,min}$  is exceeded, the specific gravitational flux  $q_{B,grav,i}$  [ $\text{m}^2/\text{s}$ ] to each neighbour element  $i$  is calculated from the bed elevation change in element  $i$  according to eq. 1.63, where  $A_i$  is the area of element  $i$ , and  $l_i$  is the length of the edge connecting the main element to its  $i^{\text{th}}$  neighbour element. If the calculated bed elevation change of the main element  $\delta_z = z - z_{new}$  is larger than the user-defined maximum bed elevation change  $\delta_{z,max}$ , the gravitational flux is limited, such that  $\delta_z = \delta_{z,max}$ . The maximum bed elevation change  $\delta_{z,max}$  is calculated by eq. 1.64, where  $r_{b,max}$  is the user-specified parameter `max_bed_change_rate` and  $\Delta t$  is the current update time step.

$$q_{B,grav,i} = \begin{cases} 0 & \text{if } \delta_z < \delta_{z,min} \\ \frac{(z_{i,new} - z_i) \cdot A_i}{l_i} & \text{if } \delta_z \geq \delta_{z,min} \\ \frac{(z_{i,new} - z_i) \cdot A_i}{l_i} \cdot \frac{\delta_{z,max}}{\delta_z} & \text{if } \delta_z \geq \delta_{z,max} \end{cases} \quad (1.63)$$

$$\delta_{z,max} = r_{b,max} \cdot \Delta t \quad (1.64)$$

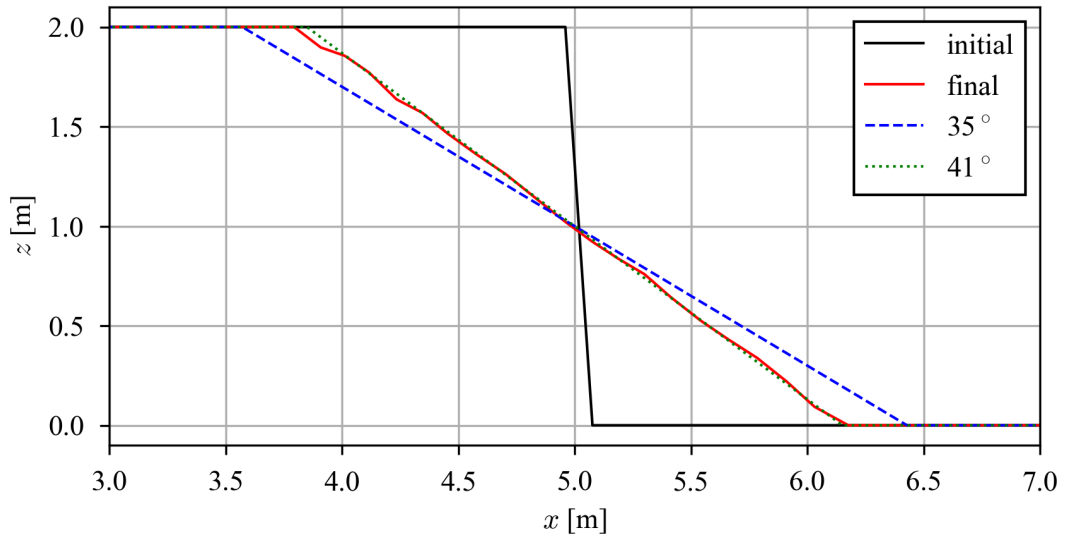
4. If a non-erodible fixed bed elevation  $z_{fix}$  is specified, the gravitational transport flux is corrected the same way as the bedload transport flux (see Section 1.2.2.7).
5. Finally, the balancing of the gravitational fluxes and the determination of the new bed elevations  $z$  is achieved by solving the Exner equation using the same numerical approaches as outlined for the bed load transport. This procedure ensures that fixed bed elevations are taken into account and the mass continuity is fulfilled.

#### 1.2.2.8.2 Time Scale of the Gravitational Transport Process

Since the sediment movement due to gravitational transport during one update time step is limited to adjacent elements, it may take many update time steps to reach a stable condition on a larger scale, e.g. a bank slope spanning over multiple elements. The time scale until a globally stable condition is reached, is influenced by the update time step (`update_time`), the maximum bed change rate (`max_bed_change_rate`) and the grid resolution. The parameter `update_time` (default: 0.0 s) determines at which frequency the gravitational transport procedure (steps 1-5 above) is executed. Generally, a smaller update time step reduces the time scale until a globally stable condition is reached. The default behaviour is to set the update time step value to 0.0 seconds, which results in the gravitational transport procedure being executed at the same time step as determined from the hydraulic CFL-criterion (see Section 2.3.4).

Furthermore, the speed of the gravitational transport process can be limited with the parameter `max_bed_change_rate` (default: 1.0 m/s). This parameter represents a maximum rate at which the bed elevation of a cell can be lowered due to gravitational transport and determines the maximum bed elevation change during one update time step. Generally, a smaller value may increase the time scale until a globally stable condition is reached.

Since sediment movement due to gravitational transport is limited to adjacent elements, also the grid resolution effects the time scale until a globally stable condition is reached. A finer grid resolution (smaller elements) increases the necessary number of update cycles the reach a stable slope over a specific length. However, a finer grid resolution may also



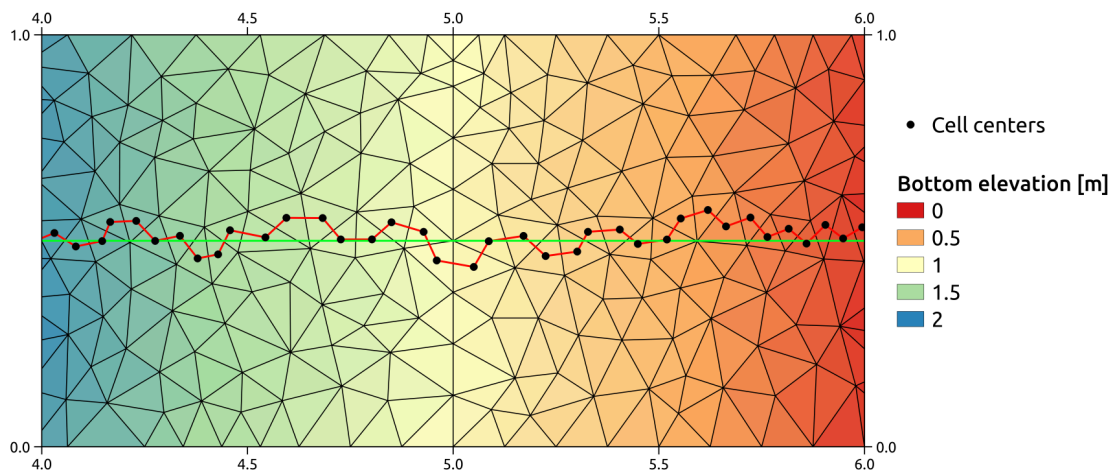
**Figure 1.10** Initial and final bed elevation profiles along the centerline are compared to profiles with angles of  $35^\circ$  and  $41^\circ$ . The slope of the bed profile at stable conditions (final) exceeds the critical angle of  $\gamma_{dry} = 35^\circ$ .

decrease the hydraulic time step and therefore may the increase frequency at which the gravitational transport procedure is executed.

### 1.2.2.8.3 Bed slope at Stable Condition

The implemented approach for modelling gravitational transport processes ensures that the local bed slope does not exceed the user-specified critical angle. Due to the spatial discretization with constant bed elevations within an element, the local bed slope is calculated between two element centers. On a larger scale (e.g. a bank slope spanning over multiple elements), the bed slope may deviate from the user-specified critical angle after reaching stable conditions. The stable bed slope over multiple elements, generally exceeds the critical bed slope.

To illustrate this effect, a simple test case was simulated. The rectangular computational domain is initially split into two parts: the left side with a bed elevation of 2 meters and the right side with a bed elevation of 0 meters. The domain is completely dry and the critical angle for dry material is set to  $\gamma_{dry} = 35^\circ$ . The simulation only considers only gravitational transport and is stopped after a stable condition is reached. The initial and final bed elevation profile along the centerline are illustrated in Figure 1.10. The initial profile exhibits the nearly vertical bed step. The final profile at stable conditions exhibits a slope of approximately  $41^\circ$ , which exceeds the critical angle of  $\gamma_{dry} = 35^\circ$ . The reason for this deviation on a larger scale is that the distance which is considered for the calculation of the slope is not a straight line, but a line connecting the element centers. This is illustrated in Figure Figure 1.11. Calculating the large scale slope considering the length of the green line results in a slope of approximately  $41^\circ$ , while considering the length of the red line results in a slope of approx.  $35^\circ$ .



**Figure 1.11** The red line connects the element centers and represents the distance, which is taken into account for the **local slope** calculation and thus for the gravitational transport. The green line represents the direct distance, which is taken into account for the **large scale slope**. The slope along the red line corresponds to  $35^\circ$ , while the slope along the green line corresponds to  $41^\circ$ .

### 1.2.2.9 External Sources Terms

The source term  $Sl_b$  represents additional sediment mass input or output (sink) that occurs locally on the computational domain on a set of elements defined by regions. The source can be specified as total volume flux including porosity [ $m^3/s$ ]. Different approaches are used to manage the behaviour of the external sources in case of a negative flux (sink):

- Exact: The specified sediment volume is added or extracted (non conservative)
- Available: The specified sediment volume to extract is limited by the defined fixed bed elevation of the elements (conservative)
- Infinity: All available sediment will be abstracted (conservative)

Addition of sediment always follows the “Exact” behaviour as there is no upper limit. The abstraction of sediment could also follow the “Exact” behaviour but the simulation might end abruptly if the available sediment volume is smaller than the volume abstracted. Therefore, the “Available” behaviour aims to avoid this situation. The “Infinity” behaviour abstracts all available sediment volume.

### 1.2.2.10 Boundary Conditions

After the specification of the *closure relations* for the sediment transport, the system of governing equations (eq. 1.1) and (eq. 1.22) can be solved within the modeling domain described in Figure 1.1, provided boundary conditions (morphologic boundary conditions) are specified at the domain boundary  $\Gamma$ . For the sediment transport only *external boundaries* that allow sediment flowing into or out of the domain can be specified. A morphologic boundary condition can be co-located with a hydraulic boundary condition. In case no hydraulic boundary condition is specified, the boundary will behave as a wall and sediment transport will not occur.

### 1.2.2.11 Upstream Boundary Condition

- *equilibrium\_in*: After erosion or deposition up to a user specified reference bed elevation (*reference\_bed\_elevation*) this upstream boundary condition grants a equilibrium condition, i.e. the same amount of sediment leaving the first computational cell in flow direction enters the cell from the upstream boundary. This leads to a constant bed elevation at the boundary condition.
- *sedimentograph*: based on a sediment hydrograph describing the bedload inflow as function of time (constant or variable). The bedload is defined as a volumetric flow rate  $Q_b = \frac{\mu_s}{\rho_s} [m^3/s]$ , where  $\mu_s$  is the sediment mass flow rate [ $kg/s$ ] and  $\rho_s$  the sediment density [ $kg/m^3$ ]. Notice that the porosity is not considered in the bedload input and is specified separately as own parameter value. The volumetric flow rate is either distributed using a geometrical weighting (*sedimentograph*), using wetted area weighting (*sedimentograph\_warea*) or using wetted conveyance weighting (*sedimentograph\_conveyance*). **Note:** When using the pre-factor (*factor*) described in Section 1.2.2.4, it is automatically applied to the volumetric flow rate at the boundary for these types of boundary conditions, i.e. the sediment hydrograph is scaled by the pre-factor.

- *transport\_capacity*: the sediment inflow is defined by calculating the equilibrium transport capacity according to the hydraulic state at the boundary. The bedload is defined as a compact volumetric flow rate (without porosity)  $Q_b$  [ $m^3/s$ ]. The volumetric flow rate is either distributed using a geometrical weighting (*transport\_capacity*), using wetted area weighting (*transport\_capacity\_warea*) or using wetted conveyance weighting (*transport\_capacity\_conveyance*). **Note:** When using the pre-factor (*factor*) described in Section 1.2.2.4, the pre-factor is implicitly included in the volumetric flow rate at the boundary for this type of boundary condition, i.e. the transport capacity at the boundary is scaled by the pre-factor. Additionally, an independent scaling factor can be specified (*boundary\_factor*), only applying to these types of the boundaries.

For the sediment discharge and transport capacity boundary condition types, the specific sediment discharge  $q_b$  is distinguished by three weighting schemes:

1. Geometrical weighting with respect to the total nodestring length  $L_n$ .

$$q_b = \frac{Q_b}{L_n} \quad \left[ \frac{m^3}{s \cdot m} \right] \quad (1.65)$$

2. Wetted area weighting

$$q_b = \frac{Q_b}{A_{w,tot}} \cdot h \quad \left[ \frac{m^3}{s \cdot m^2} \right] \quad (1.66)$$

3. Conveyance weighting

$$q_b = \frac{Q_b}{K_{tot}} h \sqrt{c_f h} \quad \left[ \frac{m^3}{s \cdot m} \right] \quad (1.67)$$

with  $K_{tot} = A_{w,tot} \sqrt{c_f h}$  the total conveyance and  $c_f$  the friction coefficient.

### 1.2.2.12 Downstream Boundary Condition

One downstream boundary condition is available:

- *equilibrium\_out*: After erosion or deposition up to a user specified reference bed elevation (*reference\_bed\_elevation*) this downstream boundary condition grants a equilibrium condition, i.e. all sediment entering the last computational cell will leave the cell over the downstream boundary. This leads to a constant bed elevation at the boundary condition.

### 1.2.2.13 Linked Boundary Condition

One linked boundary condition is available:

- *equilibrium\_linked*: At the upstream boundary, erosion or deposition is possible up to a user specified reference bed elevation (*reference\_bed\_elevation*). After reaching the reference elevation, this boundary condition grants a equilibrium condition, i.e. all sediment leaving the computational cells on the upstream side is entering at the downstream boundary with a lag of one timestep. This leads to a constant bed elevation at the upstream boundary.

## 1.3 Suspended Sediment Transport

### 1.3.1 Governing Equations for Uniform Suspended Sediment Transport

Similarly to bedload, the governing equations are also derived under shallow water conditions ensuring mass conservation extends to the suspended sediment phase. In a Cartesian reference system  $(x; y; z)$  in which the  $z$  axis is vertical and the  $x - y$  plane is horizontal, the conservation of suspended sediment mass is ensured by an advection-diffusion equation with source terms, describing the bed interaction and elevation changes arising from resuspension and deposition.

$$\frac{\partial}{\partial t} C_s h + \frac{\partial}{\partial x_i} \left( C_s h u_i - h D_{s_{ij}} \frac{\partial C_s}{\partial x_j} \right) = S_s + S_l \quad (1.68)$$

where  $C_s$  is the concentration of suspended sediment,  $D_{s_{ij}}$  is its diffusivity (tensor),  $S_s$  are local sources and  $S_l$  aggregates the source terms that parameterize vertical exchanges with the bed.

### 1.3.2 Closures for Suspended Transport

For suspended sediment transport, the source term  $S_s$  represents the exchange with the bed and is calculated through the difference between the deposition rate  $q_d$  and the resuspension rate  $q_e$ .

$$S_s = q_e - q_d \quad (1.69)$$

#### 1.3.2.1 Deposition rate

The deposition rate is expressed as the a sink flux:

$$q_d = w_s \alpha_s C_s \quad (1.70)$$

where  $w_s$  is the settling velocity and  $\alpha_s$  is an adaptation coefficient. The adaptation coefficient can be determined through two distinct approaches, the first being provided by Lin (1984) as



$$\alpha_s = m \left( 3.25 + 0.55 \ln \left( \frac{w_s}{\kappa u_*} \right) \right) \quad (1.71)$$

where  $\kappa$  is the von Karman constant and  $m$  is a tuning parameter (default value 1.0). The second is based on the critical shear stress approach of Xu (1998)

$$\alpha_s = \begin{cases} \left( 1 - \frac{\tau}{\tau_{c,d}} \right) & \tau < \tau_{c,d} \\ 0 & \tau \geq \tau_{c,d} \end{cases} \quad (1.72)$$

with

$$\tau_{c,d} = \frac{\rho_s - \rho}{\rho_s} \frac{ghw_s}{mU} \quad (1.73)$$

and where  $m$  is a calibration parameter which has been suggested to take a value of 0.0018.

### 1.3.2.2 Erosion rate

The erosion (or resuspension) rate,  $q_e$ , is given by

$$q_e = w_s \beta_s C_r \quad (1.74)$$

where  $w_s$  is the settling velocity of particles (see Section 1.3.2.3),  $\beta_s$  is a calibration constant and  $C_r$  is a reference concentration which can be determined by multiple options available in the literature. According to van Rijn (1984), the reference concentration can be calculated as

$$C_r = a \frac{d_g T_g (\theta_c)^c}{b D^{*d}} \quad (1.75)$$

where:

$T_g$  is the dimensionless characteristic number for the bottom shear stress,  $b$  is the reference height above the mean bed bottom,  $D^*$  is the dimensionless diameter and  $a$ ,  $c$  and  $d$  are calibration constants with default values  $(a, b, c, d) = (0.015, 0.05, 1.5, 0.3)$ . Another approach is the one of Zyserman and Fredsøe (1994), calculated as

$$C_r = \frac{a(\theta_s - \theta_c)^c}{1 + b(\theta_s - \theta_c)^d} \quad (1.76)$$

where  $a$ ,  $b$ ,  $c$  and  $d$  are calibration constants with default values  $(a, b, c, d) = (0.331, 0.72, 1.75, 1.75)$ .

### 1.3.2.3 Settling Velocities of Particles

The settling velocity  $w_s$  of sediment particles is an important parameter when determining suspended load. Many different empirical or semi-empirical formulas have been suggested in literature, with the following being implemented:

### 1.3.2.3.1 van Rijn

The sink rate can be determined according to the grain diameter after van Rijn (1984):

$$w_s = \frac{(s-1)gd^2}{18\nu} \quad \text{for } 0.001 < d \leq 0.1\text{mm} \quad (1.77)$$

$$w_s = \frac{10\nu}{d} \left( \sqrt{1 + \frac{0.01(s-1)gd^3}{\nu^3}} - 1 \right) \quad \text{for } 0.1 < d \leq 1\text{mm}$$

$$w_s = 1.1\sqrt{(s-1)gd} \quad \text{for } d \geq 1\text{mm}$$

where  $d$  is the diameter of the grain,  $\nu$  is the kinematic viscosity and  $s = \rho_s/\rho$  is the specific density.

### 1.3.2.3.2 Wu and Wang

Another approach for the computation of the sink velocity is the one of Wu et al. (2000):

$$w = \frac{M\nu}{Nd} \left[ \sqrt{\frac{1}{4} + \left( \frac{4N}{3M^2} (D^*)^3 \right)^{1/n}} - \frac{1}{2} \right]^n \quad (1.78)$$

where:

$$M = 53.5e^{-0.65S_p}$$

$$N = 5.65e^{-2.5S_p} \quad \{\#eq:30\_mm\_SuspLoad\_Wu2\}$$

$$n = 0.7 + 0.9S_p$$

$S_p$  is the Corey shape factor, with a value for natural sediments of about 0.7 (0.3 - 0.9).

### 1.3.2.3.3 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 regions:

$$w_s = \sqrt{\left( 13.95 \frac{\nu}{d} \right)^2 + 1.09(s-1)gd} - 13.95 \frac{\nu}{d} \quad (1.79)$$

## 1.3.3 External Source Terms

The source term  $Sl_s$  represents additional sediment mass input or output (sink) that occurs locally on the computational domain on a set of elements defined by regions. The source can be specified as total volume flux including porosity [ $m^3/s$ ] or by an imposed concentration [-]. Different approaches are used to manage the behaviour of the external sources in case of a negative flux (sink):

- Exact: The specified sediment volume is added or extracted (non conservative)

- Available: The specified sediment volume to extract is limited by the defined fixed bed elevation of the elements (conservative)
- Infinity: All available sediment will be abstracted (conservative)
- Concentration: This sink type allows to impose a specified concentration.

Addition of sediment follows the “Exact” behaviour as there is no upper limit. The abstraction of sediment could also follow the “Exact” behaviour but the simulation might end abruptly if the available sediment volume is smaller than the volume abstracted. Therefore, the “Available” behaviour aims to avoid this situation. The “Infinity” behaviour abstracts all available sediment volume.

### 1.3.4 Boundary Conditions

For the suspended sediment transport only external boundaries that allow sediment flowing into or out of the domain can be specified. Boundary conditions of suspended transport type can be co-located with hydraulic or bedload boundary conditions.

#### 1.3.4.1 Upstream Boundary Condition

Two upstream boundary conditions are available:

- *discharge\_in* and *discharge\_in\_warea*: based on a suspended sediment discharge inflow, either as a constant or a function of time, the prescribed volumetric flow rate [ $m^3/s$ ] is imposed at the boundary. The volumetric flow rate is either distributed using a geometrical weighting or a wet-area weighting.
- *concentration\_in*: the suspended sediment inflow is defined by forcing a target concentration. The suspended sediment input rate is thus given by the target concentration paired with the total hydrodynamic mass flux.

#### 1.3.4.2 Downstream Boundary Condition

- *zero\_gradient\_out*: this downstream boundary allows the free outflow of any tracer quantities in the flow. **Note:** If this condition is not prescribed, a wall condition is assumed and the tracer quantities will be retained (no outflow).

## 1.4 Passive tracers

### 1.4.1 Introduction

A multitude of dissolved species are present in environmental flows. In the context of hydraulic and environmental engineering, numerical modelling of scalar transport becomes a relevant tool mostly because of its versatility. In terms of advective phenomena, the most common applications include

- Pollutant fate and transport
- Accumulation or depletion of nutrients
- Calculation of water residence times
- Flow visualization

## 1.4.2 Transport of passive species

### 1.4.2.1 Governing Equations for passive specie transport

The governing equations are obtained under the shallow water framework and impose mass conservation for both fluid and dissolved phases.

In a Cartesian frame of reference  $(x; y; z)$  in which the  $z$  axis is vertical and the horizontal lies in the  $x - y$  plane, the system of governing equations is formed by (eq. 1.1) for hydrodynamics and is coupled with multiple equations for the conservation of the total tracer masses. The conservation of each tracer mass is ensured by the scalar continuity equation (eq. 1.80), which is tightly coupled to the shallow water equations. This equation allows to describe the evolution of the specie concentration as:

$$\frac{\partial}{\partial t} C_k h + \frac{\partial}{\partial x_i} \left( C_k h u_i - h D_{kij} \frac{\partial C_k}{\partial x_j} \right) = Sl_k \quad (1.80)$$

where  $C_k$  is the concentration of specie  $k$ ,  $D_{kij}$  is it's diffusion tensor and  $Sl_k$  is the local source term, specifying additional sources or sinks of the specie  $k$ .

### 1.4.2.2 External Sources Terms

The source term  $Sl_k$  conveys an input or output (sink) that occurs locally on the computational domain over elements limited by regions. It can be specified as total volumetric flux [ $m^3/s$ ] or by an imposed concentration [-]. Different approaches define the behaviour of external sources if it imposes negative fluxes (sink):

- Exact: The specified volume is added or extracted (non conservative)
- Available: The specified volume to extract is limited by the available tracer volume (conservative)
- Infinity: All available tracer volume will be extracted (conservative)
- Concentration: This sink type allows to impose a specified tracer concentration.

Additionally, there is the option of forcing a target concentration  $C_k^f$  homogeneously across all cells defined by the region.

### 1.4.2.3 Boundary Conditions

The system of governing equations (eq. 1.1) and (eq. 1.80) can be solved within the modeling domain described in Figure 1.1, provided that boundary conditions (hydrodynamical and tracer boundary conditions) are specified at the domain boundary  $\Gamma$ . For the tracer transport only *external boundaries* that allow tracer flowing into or out of the domain can be specified. A tracer boundary condition should be co-located with a hydraulic boundary condition. Otherwise, the boundary will behave as a wall and tracer transport will not occur.

### 1.4.2.4 Upstream Boundary Condition

Two upstream boundary conditions are available:

- *discharge\_in* and *discharge\_in\_warea*: based on a tracer discharge inflow, either as a constant or a function of time, the prescribed volumetric flow rate  $Q_k$  [ $m^3/s$ ] is imposed at the boundary. The volumetric flow rate is either distributed using a geometrical weighting or a wet-area weighting, and the specific tracer discharge  $q_k$  is thus given by:

1. Geometrical weighting with respect to the total nodestring length  $L_n$ .

$$q_k = \frac{Q_k}{L_n} \quad \left[ \frac{m^3}{s \cdot m} \right]$$

2. Wetted-area weighting

$$q_k = \frac{Q_k}{A_{w,tot}} \cdot h \quad \left[ \frac{m^3}{s \cdot m} \right]$$

- *concentration\_in*: the tracer inflow is defined by forcing a target tracer concentration  $C_k^f$ . The volumetric flow rate is thus given by the target concentration paired with the total hydrodynamic mass flux  $q$  through that boundary as  $q_k = q\phi_k^f$ .

**Note:** If the hydrodynamical mass flow at the boundary is not inward directed then no tracer flux is imposed.

### 1.4.2.5 Downstream Boundary Condition

One downstream boundary condition is available:

- *zero\_gradient\_out*: this downstream boundary allows the free outflow of any tracer quantities in the flow. **Note:** If this condition is not prescribed, a wall condition is assumed and the tracer quantities will be retained (no outflow).

## 1.5 Riparian vegetation

### 1.5.1 Introduction

Riparian vegetation is known to play an important role in mediating river morphodynamic processes at various spatial and temporal scales. Predicting river morphology has to account for the feedbacks between plants, flow, and sediment transport. On one hand, vegetation can modify the flow resistance and the shear stresses acting on the riverbed, thus influencing the overall flow pattern and rate of sediment transport. On the other hand, erosion and deposition processes may cause plant mortality through uprooting and burial during floods. Including vegetation into river morphodynamic simulations has potential applications for

- estimating sedimentation and erosion processes
- restoration projects
- riparian forests dynamics
- flood hazard mapping

### 1.5.2 Description of Vegetation

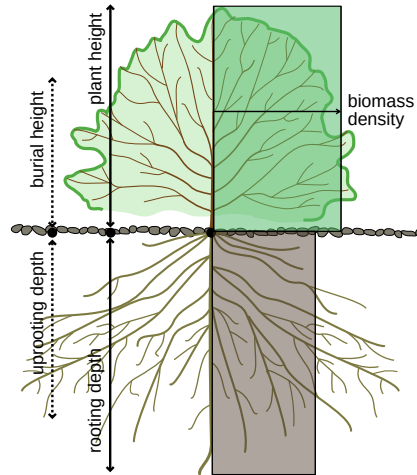
Vegetation is described by a dimensionless aboveground biomass density,  $B_c$  (subscript  $c$  stands for canopy), belowground density,  $B_r$  (subscript  $r$  stands for roots) and a rooting depth,  $D_r$  (see Figure 1.12). The sum of  $B_c$  and  $B_r$  should ideally be 1 at maximum, while there is no specific limit for the rooting depth. The model accounts also for the vegetation height,  $H$ , which is derived as  $H = aB_c^b$ , where  $a$  and  $b$  are constant parameters.  $B_c$  and  $B_r$  are considered to distribute linearly along the main vertical axis of the plant. The uprooting depth,  $D_{upr}$ , namely the depth at which uprooting occurs, is defined as  $\beta_{upr}D_r$ , with  $\beta_{upr}$  a constant input parameter that decreases the maximum erosion that vegetation can withstand. The burial height,  $H_{bur}$ , represents the effective plant height when the plant is submerged (due to bending) and is defined as  $H_{bur} = \beta_{bur}H$ , with  $\beta_{bur}$  a constant parameter (see Section 1.5.5).

### 1.5.3 Effect of Vegetation on Water Flow

The aboveground vegetation is assumed to change the bed roughness. This is included by modifying the Strickler's coefficient  $k_{str}$  [ $m^{1/3}/s$ ] (Bertoldi et al., 2014), such as

$$k_{str}(t) = k_{s,g} + (k_{s,v} - k_{s,g})B_c(t) \quad (1.81)$$

where  $k_{s,g}$  represents the roughness of the bare bed, which depends on the sediment grain size, while  $k_{s,v}$  ( $< k_{s,g}$ ) is the roughness of a completely vegetated bed assumed to vary with species-specific canopy characteristics. The Strickler's coefficient can change during the simulation as  $B_c$  changes (see Section 1.5.5) (Caponi et al., 2020).



**Figure 1.12** Schematic representation of vegetation in the model. The vegetation state is described by an aboveground, belowground biomass, and a rooting depth. The other variables (uprooting depth, burial height, and plant height) are derived through specific functions and input parameters. Adapted from Caponi et al. (2020)

## 1.5.4 Effects of Vegetation on Bedload Transport

### 1.5.4.1 Bottom Shear Stress

The presence of vegetation is also known to affect the shear stresses acting on the bed surface and responsible for sediment transport. The reduction of bed shear stress is included in the model by multiplying the total shear stress by a factor  $\gamma < 1$  and computing the sediment flux using a reduced dimensionless bed shear stress (Shields parameter),  $\gamma\theta$ . The parameter  $\gamma$  ranges between 0 and 1 and it is chosen according to  $\gamma = k_{str}(t)/k_{s,g}$ , with  $k_{str}$  evaluated as in eq. 1.81, (Caponi and Siviglia, 2018).

### 1.5.4.2 Critical Shear Stress

The role of root-enhanced riverbed cohesion is taken into account by increasing the critical Shields parameter (Bertoldi et al., 2014). Assuming an MPM-like formula (eq. 1.36), the critical Shields parameter  $\theta_{cr}$  is defined as

$$\theta_{cr}(t) = \theta_{cr,g} + (\theta_{cr,v} - \theta_{cr,g})B_r(t) \quad (1.82)$$

in which  $\theta_{cr,g}$  and  $\theta_{cr,v}$  ( $> \theta_{cr,g}$ ) represent the values used for bare bed and completely vegetated riverbed, respectively (Caponi and Siviglia, 2018).

## 1.5.5 Effects of Bed Changes on Vegetation

Bed level change causes three effects on vegetation in the model: uprooting, burial, and biomass redistribution.

### 1.5.5.1 Biomass Redistribution

As soon the bed level changes, the portion of vegetation above and below ground changes as well. The model assumes that the total biomass remains constant during a simulation, unless uprooting occurs. This means that in case of a positive bed level change,  $\Delta z_B > 0$  (deposition),  $B_c$  will decrease by  $\Delta z_B(B_c/H_{bur})$  and  $B_r$  will increase by the same amount, where  $H_{bur}$  is a fraction of the plant height. On the contrary, in case of bed level erosion,  $B_c$  will increase by  $\Delta z_B(B_r/D_r)$  and  $B_r$  will decrease by the same amount (Caponi et al., 2020). Changes in  $B_c$  and  $B_r$  during the simulation will change the effects on flow and sediment transport associated with these values following eq. 1.82 and eq. 1.81.

### 1.5.5.2 Plant Burial

Plant burial is the mechanism by which a plant get covered by sediments, that is, when the bed level increases in a cell until reaching the burial plant height  $H_{bur}$ . This value takes into account the bending of the plant when submerged, which reduces the effective height of the plant. In case the bed level change reaches  $H_{bur}$ ,  $B_c$  will go to zero, causing the  $k_{str}$  value to equal  $k_{s,g}$ . In case the bed level decreases again during the simulation,  $B_c$  can increase as well (Caponi et al., 2020). Plants can in fact resist well sedimentation processes, thanks to the high flexibility of the stems that prevents breakage.

### 1.5.5.3 Plant Uprooting

Plant uprooting occurs when the pulling forces applied on the plant by the water flow equal the resisting forces of the plant, which are given by the root system anchoring. We consider that this can occur only when part of the plant roots are already exposed to the flow, that is, when the bed level has decreased during the simulation (type II of uprooting, following the conceptual model of Edmaier et al. (2011)). Therefore, the uprooting is modelled by defining a critical depth at which the plant is removed. This depth is currently defined as a percentage of the rooting depth as  $D_{cr} = \beta_{upr}D_r$ , where  $\beta_{upr}$  is a parameter that modulates the resistance of the plant and depends on soil characteristics and plant species. When uprooting occurs,  $B_c$  and  $B_r$  are set to zero and vegetation cannot influence flow and sediment transport anymore during the simulation.

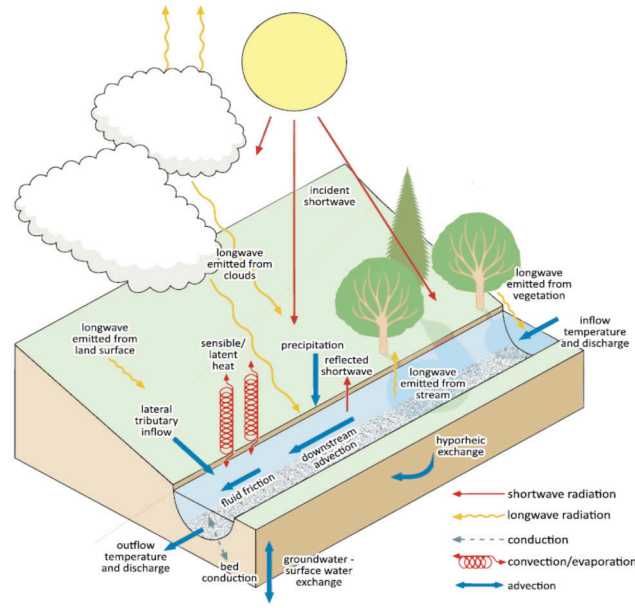
## 1.6 Water temperature

### 1.6.1 Introduction

River water temperature is a fundamental physical property of flowing water, having a key role in many ecological processes (Caissie, 2006). Its magnitude, fluctuations and seasonality have a critical impact on biota behaviour, metabolism and distribution. River temperature shows also spatial heterogeneity, i.e. thermal landscapes, resulting from complex interplay between atmospheric conditions with landscape and reach morphological characteristics.

In this context a modelling solution that account for 2D (bi-dimensional, depth averaged) hydro-thermal dynamics to quantify and simulate thermal dynamics is desirable yet challenging. It requires a good amount and quality of field data (e.g. atmospheric,





**Figure 1.13** Schematic representation of energy and hydrological exchanges determining river water temperature. From Dugdale et al. (2017)

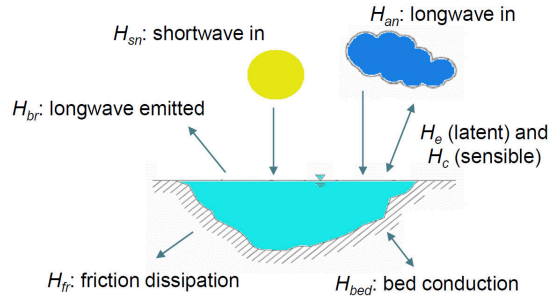
groundwater level, topography, etc.), and a solid model for hydro-thermodynamic simulation at adequate spatial and temporal scale.

### 1.6.2 Thermodynamics: governing equations

River water temperature can be modelled assuming the passive advection and diffusion of a scalar quantity, in the form of dissolved or particulated particles (Vanzo et al., 2016). The transport of the water temperature can be described by the following advection-diffusion-reaction equation:

$$\begin{aligned} & \partial_t q_T + \partial_x \left[ \frac{q_x q_T}{h} - h (K_{xx} \partial_x T + K_{xy} \partial_y T) \right] + \partial_y \left[ \frac{q_y q_T}{h} - h (K_{yx} \partial_x T + K_{yy} \partial_y T) \right] \\ & = \frac{H_{atm}}{c_w \rho_w} + \frac{H_b}{c_w \rho_w}, \end{aligned} \quad (1.83)$$

where the equation unknown is the specific thermal mass  $q_T = hT$  [ $m^\circ C$ ], with  $T$  the water temperature and  $h$  the water depth. The terms  $K_{ij}$  [ $m^2/s$ ] are the components of the 2D diffusion tensor. The right-hand side terms of eq. (eq. 1.83) are the “reaction” terms, hence represent the variation of water temperature due to external forcings, such as atmospheric ones. In particular,  $H_{atm}$  [ $W/m^2$ ] is the total net energy flux exchanged with the water column from the surface, i.e. with the atmosphere, whilst  $H_b$  collects the energy fluxes related with the river bed. Finally,  $\rho_w$  and  $c_w$  are the water density [ $Kg/m^3$ ] and specific heat ( $4186 JKg^{-1}K^{-1}$ ), respectively.



**Figure 1.14** Schematic representation of thermal fluxes that can be simulated

### 1.6.3 Temperature closure relationships

The atmospheric net heat exchange  $H_{atm}$  [ $W/m^2$ ] to or from the water column (positive for incoming fluxes) is the sum of different factors:

$$H_{atm} = H_{sn} + H_{an} + H_{br} + H_e + H_c. \quad (1.84)$$

The incoming net short-wave radiation flux  $H_{sn}$  is expressed as

$$H_{sn} = 0.97H_{si}(1.0 - SF), \quad (1.85)$$

where the user-provided total incoming radiation  $H_{si}$  [ $W/m^2$ ] is corrected with albedo (3%) and a user-provided shade factor  $SF$  [0 to 1]. Incoming long wave radiation is  $H_{an}$  and reads:

$$H_{an} = \sigma (T_a + 273.15)^4 (Ca + 0.084900481\sqrt{e_a})(1 - R_l), \quad (1.86)$$

where  $\sigma=5.67e^{-8}W.m^{-2}K^{-4}$  is the Stefan-Boltzman constant,  $T_a$  [ $degC$ ] is the air temperature,  $e_a$  [kPa] is the air vapor pressure and the two coefficients  $Ca=0.6$  (Brundt's coefficient) and  $R_l=0.03$  (reflective coefficient) are assumed constant. The air vapor pressure  $e_a$  is calculated from the user-provided relative humidity  $RH$  [%] and the saturation of vapor pressure  $e_v$  [kPa] as:

$$e_a = 0.01RH \cdot e_v. \quad (1.87)$$

The saturation of vapor pressure  $e_v$  [kPa] is evaluated via Tetens equation, as function of the air temperature:

$$e_v = \left( 0.61078 \cdot e^{\frac{17.27 \cdot T_a}{T_a + 237.3}} \right). \quad (1.88)$$

The emitted long wave radiation  $H_{br}$  is a function of the river water temperature  $T$ , and is calculated as:

$$H_{br} = -\sigma\epsilon(T + 273.15)^4, \quad (1.89)$$

with same symbols as before, and emissivity  $\epsilon=0.97$ .

The evaporation (latent) heat flux  $H_e$  is evaluated as:

$$H_e = - \left( 9.2 + 0.46v_w^2 \right) \cdot (e_v - e_a), \quad (1.90)$$

where  $v_w$  [m/s] is the 2m-above-soil wind speed. Similarly, the convective (sensible) heat flux  $H_c$  reads:

$$H_c = -0.47 \left( 9.2 + 0.46v_w^2 \right) \cdot (T - T_a). \quad (1.91)$$

Detailed references for the formulation of the atmospheric heat fluxes  $H_{atm}$  are given in (Siviglia and Toro, 2009).

The heat exchanges with the river bed  $H_b$  [ $W/m^2$ ] are evaluated in the form of two contributions,

$$H_b = H_{bed} + H_{fr}. \quad (1.92)$$

The first right-hand side term  $H_{bed}$  represents the convective heat exchange with the bed and it is (at this stage) a crude simplification of the relation proposed in (Caissie and Luce, 2017) (Eq. 3):

$$H_{bed} = -k_b \frac{\partial T}{\partial z} \approx -k_b \frac{T - T_b}{L_b}, \quad (1.93)$$

where  $T_b$  is the user-provided reference bed temperature at the sediment depth  $L_b$  [m]. The bed thermal conductivity  $k_b$  [ $W/m/K$ ] has default value of 1.5. The second term of the bed fluxes  $H_{fr}$  can be used to simulate the water heat dissipation by friction, and it is proposed similarly to (Beltaos, 2013):

$$H_{fr} = -f_{fr} \rho_w \frac{u^3}{c_f^2}, \quad (1.94)$$

where  $u$  is the magnitude of the flow,  $c_f$  the friction term (see (Vanzo et al., 2021)). The pre-factor  $f_{fr}$  [-] (default value 0, ranging 0 to 1) can be used to tune the contribution of such term. It is worth mentioning that the formulation was proposed in the framework of modelling ice formation in rivers and it should be carefully applied/tuned.

The terms  $K_{ij}$  of the diffusion tensor vary considerably with respect to the physical nature of the transported species. Diffusive transport is modelled in terms of both molecular diffusion  $K^m$  and turbulent dispersion  $K_{ij}^t$ , such that  $K_{ij} = K^m I_{ij} + K_{ij}^t$ , with  $I_{ij}$  the identity matrix. The molecular diffusion is assumed as an isotropic Fickian process with constant coefficient  $K^m$ . Turbulent dispersion is anisotropic ( $K_{ij}^t$ ) and scales with the friction velocity  $u_* = \|\mathbf{u}\|/c_f$  and water depth via a longitudinal  $\alpha_L$  and transversal  $\alpha_T$  non-dimensional coefficients. Suitable values for open channel flows in natural environments are  $\alpha_L=13$  and  $\alpha_T=1.2$  (Vanzo et al., 2016).

### 1.6.4 Initial conditions

The user is requested to define the initial conditions of the simulation. Different types of initial conditions are available, similarly to the other modules:

- **zero**: temperature is set to 0 in all the computational cells (default);
- **region defined**: user explicitly defines the initial values of the water temperature. Different values can be assigned to different region of the computational domain;
- **continue**: values are taken from the result file of previous simulations at the provided time  $t$ .
- **assigned from file** (experimental): water temperature initial conditions are read from a text file (single column of values) where the row numbers matches the computational cell ID from the mesh file.

### 1.6.5 Boundary conditions

In the current version of the water temperature module, only *standard* BCs are implemented for river temperature simulations, with three types: i) temperature inflow as a constant value or ii) as a time-series (`passive_in`) and (iii) Neumann BC outflow (`passive_out`). User can set a multiplication factor (default value=1) to scale the input timeseries.

### 1.6.6 Setup parameters

The parameters needed in the equations of section Section 1.6.3 are to be found in the sub-blocks PARAMETERS and DIFFUSION of the temperature module. They all have default values. A brief explanation follows.

Block PARAMETER \* “fluid\_specific\_heat”[J/Kg/K]: Water specific heat. Default value = 4186 J/Kg/K; \* “temperature\_start” [s]: possibility to delay the beginning of temperature calculation. Default value = 0s, i.e. starts from the beginning; \* “source\_update\_time” [s]: interval for updating the source terms. Default = 0s (i.e. at every computational timestep). Larger values lead to faster simulations, but convergence needs to be tested; \* “friction\_heating\_factor” [-]: to calibrate the heat dissipation by friction with the river bed. Default=0, i.e. deactivated; \* “bed\_thermal\_thickness”[m]: vertical thickness to reach constant soil temperature. Default = 1m; \* “bed\_thermal\_conductivity”[W/m/K]: it depends on soil type. Default value = 1.5 W/m/K. Hint: Set to 0.0 to exclude this contribution from the simulation.

Block DIFFUSION

- “type”:
  - none (equivalent to removing the block)
  - constant: fixed diffusion coefficients
  - dynamic: diffusion coefficients depend on flow conditions
- “molecular\_diffusion” [m<sup>2</sup>/s]: in water. Default value = 1e-9 m<sup>2</sup>/s,
- “longitudinal\_diffusion\_coeff”:

- constant value in [m<sup>2</sup>/s] if type=constant
- non-dimensional factor (suggested 13) if type=dynamic
- default value = 0
- “transversal\_diffusion\_coeff”:
  - constant value in [m<sup>2</sup>/s] if type=constant
  - non-dimensional factor (suggested 1.2) if type=dynamic
  - default value = 0
- “maximum\_relaxation\_parameter”[-]: related with the numerical solver (Vanzo et al., 2016). Default value = 0.1.

---

# Numerical Models

## 2.1 General View

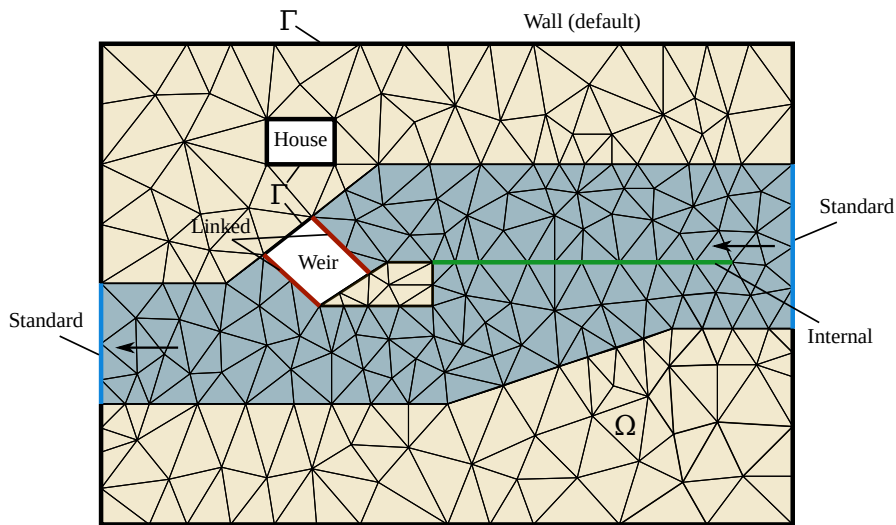
The governing equations of hydro- and morphodynamics are conservation laws expressing conservation of mass and momentum. The aim of the numerical simulation is to solve these equations over the computational domain and for a given time. The computational domain is discretized by a computational mesh consisting of elements and conservation equations are applied on each domain element. In order to numerically solve the conservation equations, the mathematical model is approximated by numerical schemes, i.e. the numerical approximation consists of the spatial and temporal discretization of the conservation equations including an algorithm that solves the discretized equations.

For BASEHPC::BASEplane, the spatial discretisation of the domain is based on an unstructured mesh made of triangular elements (Figure 2.1). For the conservation equations, the spatial discretisation follows the finite volume scheme, while for the temporal discretisation an explicit first order Euler scheme is used. The software processes the different modules (e.g. hydraulic, morphology, tracers) in a decoupled way (Figure 2.2).

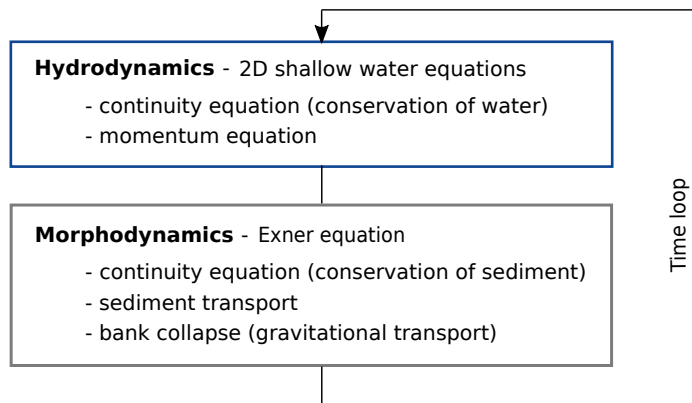
The discretization and the solution method for the hydro- and morphodynamic equations will be presented in the following sections.

## 2.2 Discretization

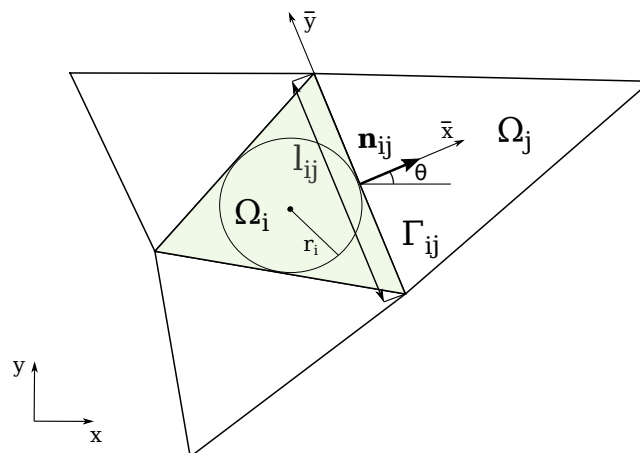
The problem is discretised adopting a finite volume approach over unstructured triangular meshes. A conforming triangulation  $T_\Omega$  of the computational domain  $\Omega \subset \mathbb{R}^2$  by elements  $\Omega_i$  such that  $T_\Omega = \bigcup \Omega_i$ , is assumed. Hereafter we will use the following notation: given a finite volume  $\Omega_i$ ,  $j = 1, 2, 3$  is the set of indexes such that  $\Omega_j$  is a neighbour of  $\Omega_i$ ;  $\Gamma_{ij}$  is the common edge of two neighbour cells  $\Omega_i$  and  $\Omega_j$ , and  $l_{ij}$  its length.  $\mathbf{n}_{ij} = (n_{ij,x}, n_{ij,y})$  is the unit vector which is normal to the edge  $\Gamma_{ij}$  and points toward the cell  $\Omega_j$  (see Figure 2.3). Data are represented by cell averages  $U_i^n$  and the numerical solution sought at time  $t^{n+1} = t^n + \Delta t$ , is denoted by  $U_i^{n+1}$ .



**Figure 2.1** Modeling domain, types of boundary conditions and computational mesh. The flow is from right to left and a side weir (green line) divides the channel into a lower and an upper channel through the weir. External boundary conditions must be provided at  $\Gamma_1$ ,  $\Gamma_2$  and  $\Gamma_3$  while internal boundary conditions can be specified in any place within  $\Omega$



**Figure 2.2** Overview of the numerical model



**Figure 2.3** Element (shaded triangle) of unstructured triangular mesh and used notation.

## 2.3 Numerical solution of Hydrodynamics

### 2.3.1 Vectorial Form of the Governing Equations

For numerical convenience, the system of governing equations (eq. 1.1) is rewritten in vectorial form in terms of the water surface elevation  $H = h + z_B$ . It now reads:

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}_x}{\partial x} + \frac{\partial \mathbf{F}_y}{\partial y} = \mathbf{S} \quad (2.1)$$

where the vector of unknowns is

$$\mathbf{U} = \begin{pmatrix} H \\ q_x \\ q_y \end{pmatrix} \quad (2.2)$$

the vector fluxes are

$$\mathbf{F}_x = \begin{pmatrix} q_x \\ uq_x + \frac{1}{2}g(H^2 - 2Hz_b) \\ uq_y \end{pmatrix} ; \quad \mathbf{F}_y = \begin{pmatrix} q_y \\ vq_x \\ vq_y + \frac{1}{2}g(H^2 - 2Hz_b) \end{pmatrix} \quad (2.3)$$

and the vector of source terms is

$$\mathbf{S} = \begin{pmatrix} S_h \\ gHS_x \\ gHS_y \end{pmatrix}. \quad (2.4)$$

The motivation of using  $H$  instead of  $h$  lies in the fact that it is easier to develop numerical schemes which preserve depth positivity and satisfy the well-balanced property.

### 2.3.2 Spatial Discretisation

In order to discretise the system of governing equations, the domain is meshed by a set of triangular elements. The spatial discretization of the conservation equations is carried out by the finite volume method, where the differential equations are integrated over the single elements, i.e. control volumes. The water surface elevation is defined at the element center and is equally distributed over the element.

By integrating the governing system of equations eq. 2.1 in the control volume  $V = [\Omega_i] \times [t^n, t^{n+1}]$ , we obtain

$$\mathbf{U}_i^{n+1} = \mathbf{U}_i^n - \frac{\Delta t}{|\Omega_i|} \sum_{j=1}^3 l_{ij} [\mathbf{F}_{ij}] + \Delta t \mathbf{S}_i. \quad (2.5)$$



### 2.3.3 Flux Estimation

#### 2.3.3.1 Rotational Invariance of the Shallow Water Equations

The flux  $\mathbf{F}_{ij}$  are evaluated taking advantage of the rotational invariance property of the shallow water equations. According to this property the two-dimensional homogeneous shallow water equations satisfy the following equality (Toro, 2009):

$$\mathbf{n}_{ij} \cdot [\mathbf{F}_x(\mathbf{U}), \mathbf{F}_y(\mathbf{U})] = \mathbf{T}^{-1}(\theta) \mathbf{F}_x[\mathbf{T}(\theta)\mathbf{U}] \quad (2.6)$$

where  $\theta$  is the angle between the vector  $\mathbf{n}_{ij}$  and x-axis, measured counter clockwise from the  $x$ -axis (see Figure 2.3) and

$$\mathbf{T}(\theta) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & \sin \theta \\ 0 & -\sin \theta & \cos \theta \end{pmatrix} \quad (2.7)$$

being

$\mathbf{T}^{-1}(\theta) = \text{inverse of } \mathbf{T}(\theta)$ .

#### 2.3.3.2 Computation of the Flux

The flux  $\mathbf{F}_{ij}$  is obtained at every edge of the finite volume mesh, as the solution of the one-dimensional projected Riemann problem along the normal direction of the two conservation laws eq. 2.1. The computational steps can be summarized as follows:

- First, the vector of conserved variables  $\mathbf{U}$  is transformed into the local coordinate system  $(\bar{x}, \bar{y})$  (see Figure 2.3) at the edge with the operation  $\mathbf{T}(\theta)\mathbf{U}$ .
- A one-dimensional, local Riemann problem is formulated and solved in the normal direction of the edge. From this calculation the new flux vector over the edge  $\mathbf{F}[\mathbf{T}(\theta)\mathbf{U}]$  is defined.
- The flux vector, formulated in the local coordinate system is transformed back to the global coordinates (Cartesian) with  $\mathbf{T}^{-1}\mathbf{F}[\mathbf{T}(\theta)\mathbf{U}]$ . The sum of the fluxes of all edges of an element gives the total fluxes in the  $x$ - and  $y$  directions.

The fluxes are calculated in the normal direction of the element edges. The normal direction of the edge is defined positive from element  $i$  (L) to element  $j$  regarding the edge direction.

#### 2.3.3.3 The HLLC approximated Riemann Solver

The HLLC approximate Riemann solver (Toro, 1994) is a modified HLL (Harten, Lax and van Leer) approximate Riemann solver that includes the shear wave.

The numerical flux at the cell interface is computed as follows:

$$\mathbf{F}_{ij}^{HLLC} = \begin{cases} \mathbf{F}_i & \text{if } 0 \leq S_i, \\ \mathbf{F}_{*i} = \mathbf{F}_i + S_i(\mathbf{U}_{*L} - \mathbf{U}_i) & \text{if } S_i \leq 0 \leq S_*, \\ \mathbf{F}_{*j} = \mathbf{F}_j + S_j(\mathbf{U}_{*R} - \mathbf{U}_j) & \text{if } S_* \leq 0 \leq S_j, \\ \mathbf{F}_j & \text{if } 0 \geq S_j. \end{cases} \quad (2.8)$$

The wave speed velocities are estimated as:

$$S_i = u_i - \sqrt{gh_i}\xi_i; \quad S_j = u_j + \sqrt{gh_j}\xi_j \quad (2.9)$$

where  $\xi_{K=(i,j)}$  is defined as:

$$\xi_K = \begin{cases} \sqrt{\frac{1}{2} \left[ \frac{(h_* + h_K)h_*}{h_K^2} \right]} & \text{if } h_* > h_K, \\ 1 & \text{if } h_* \leq h_K. \end{cases} \quad (2.10)$$

with  $h_*$ , an estimate for the exact solution of the water depth in the star region obtained using the depth positivity condition. It reads as

$$h_* = \frac{1}{2}(h_L + h_R) - \frac{1}{4}(u_R - u_L)(h_L - h_R)/(\sqrt{gh_L} + \sqrt{gh_R}) \quad (2.11)$$

In case of dry-bed conditions, the wave speeds are estimated as the exact dry front speed, i.e.:

$$S_i = \begin{cases} u_i - 2\sqrt{gh_i} & \text{if } h_i = 0, \\ \text{usual estimate} & \text{if } h_i > 0, \end{cases} \quad (2.12)$$

$$S_j = \begin{cases} u_j + 2\sqrt{gh_j} & \text{if } h_j = 0, \\ \text{usual estimate} & \text{if } h_j > 0. \end{cases}$$

And the middle estimated wave speed  $S_*$  corresponds to the front wave speed in case of dry-bed problem.

The expression of the states  $\mathbf{U}_{*i}, \mathbf{U}_{*j}$  and the middle wave speed  $S_*$  can be found in the book of Toro (2009).

### 2.3.4 Numerical Stability

Numerical stability is assured by choosing the time step  $\Delta t$  for time integration such that it obeys the Courant-Friedrichs-Lewy (CFL) condition. In 2-D the Courant number (CFL) can be defined as follows:

$$CFL = \frac{(\sqrt{u^2 + v^2} + c)\Delta t}{r_i} \quad (2.13)$$

where  $r_i$  is the radius of the inscribed circle that defines the element center (Figure 2.3),  $u, v$  are the corresponding velocities of the element and  $c = \sqrt{gh}$ . The HLLC scheme is stable for

$$0 < CFL \leq 1 \quad (2.14)$$

### 2.3.5 Discretisation of Source terms

#### 2.3.5.1 Bed Slope Source Term

The bed slope source term (eq. 1.2) is discretized using the robust modified-state approach proposed by Duran et al. (2013). The discretization presents a motionless steady states-preserving scheme:

$$\mathbf{S}_{b,i} = \sum_{j=1}^m l_{ij} \mathbf{S}_{b,ij} = \sum_{j=1}^m l_{ij} \begin{pmatrix} 0 \\ gH_{ij}^*(z_i - \bar{z}_{ij}) \vec{\mathbf{n}}_{ij} \end{pmatrix} \quad (2.15)$$

where  $\bar{z}_{ij} = \check{z}_{ij} - \Delta_{ij}$  with  $\check{z}_{ij} = \max(z_{bi}, z_{bj})$  the maximum bed elevation between cells  $i$  and  $j$  and  $\Delta_{ij} = \max(0, \check{z}_{ij} - H_i)$ .  $H_{ij}^*$  is the approximated value of the water surface elevation  $H$  at the cell interface  $\Gamma_{ij}$ .

#### 2.3.5.2 Friction Source Term

We handle the inhomogeneous character of system eq. 1.1 due to the presence of frictional source terms by adopting a robust splitting technique Toro (2001). We initially consider the initial value problem (IVP)

$$\left. \begin{array}{l} PDE : \mathcal{A}(\mathbf{U}) = \mathcal{S}(\mathbf{U}) \\ IC : \mathbf{U}(x, y, 0) = \mathbf{U}_i^n \end{array} \right\} \text{IVP} .$$

where  $\mathcal{A}$  represents the advective operator

$$\mathcal{A}(\mathbf{U}) = \frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}_x}{\partial x} + \frac{\partial \mathbf{F}_y}{\partial y} = \mathbf{0} ,$$

and  $\mathcal{S}$  represents the frictional source term operator.

The numerical solution is then obtained by subsequently integrating *two* initial value problems (IVPs):

$$\left. \begin{array}{l} ODEs : \frac{d\mathbf{U}}{dt} = \mathcal{S}(\mathbf{U}) \\ ICs : \mathbf{U}(x, y, 0) = \mathbf{U}_i^n \end{array} \right\} \xrightarrow{\Delta t} \bar{\mathbf{U}}_i \quad \text{IVP1} ,$$

$$\left. \begin{array}{l} PDEs : \mathcal{A}(\mathbf{U}) = 0 \\ ICs : \mathbf{U}(x, y, 0) = \bar{\mathbf{U}}_i \end{array} \right\} \xrightarrow{\Delta t} \mathbf{U}_i^{n+1} \quad \text{IVP2} ,$$

The initial condition (IC) for IVP1 is  $\mathbf{U}_i^n$ , corresponding to the initial condition of the full problem IVP. The solution of IVP1 is obtained solving a system of ordinary differential equations (ODEs) after integration by a time step  $\Delta t$  and is denoted by  $\bar{\mathbf{U}}_i$ . IVP2 is then integrated by a time step  $\Delta t$ , with initial condition given by the solution of IVP1  $\bar{\mathbf{U}}_i$ . The solution of IVP2  $\mathbf{U}_i^{n+1}$  is obtained solving an hyperbolic homogeneous system of partial differential equations (PDEs) and represents the approximate solution of the full problem

IVP.\ Since we adopt an implicit second-order Runge-Kutta method for solving the ODEs systems IVP1 and an explicit finite volume method for solving IVP2, the integration time step  $\Delta t$  is determined accordingly with the *CFL* stability condition for IVP2.

### 2.3.5.3 External Source Term

An external source is defined as specific mass flux  $\delta$  (m/s), uniformly distributed over a number of elements of the domain with a specific surface area. The external source can either be specified as discharge ( $m^3/s$ ) or precipitation intensity (mm/h) for a specific region of the domain. The external source value is divided among the cells composing the region and converted to cell specific mass flux  $\delta_i$ . The volume allocated is characterized by different behaviors:

$$\begin{aligned}
 \text{Exact:} \quad & S_{h,i} = \delta_i \\
 \text{Available:} \quad & S_{h,i} = \delta_i && \text{if } \delta_i \cdot \Delta t > 0 \\
 & S_{h,i} = \max(\delta_i, -h_i) && \text{if } \delta_i \cdot \Delta t < 0 \\
 \text{Infinity:} \quad & S_{h,i} = \delta_i && \text{if } \delta_i \cdot \Delta t > 0 \\
 & S_{h,i} = -h_i && \text{if } \delta_i \cdot \Delta t < 0
 \end{aligned} \tag{2.16}$$

Where  $h_i$  is the water depth of the element  $i$ . The external source volume is added to the initial water volume.

$$h_i^{t+1} = h_i^t + S_{h,i} \cdot \Delta t \tag{2.17}$$

## 2.3.6 Solution Procedure

The numerical solution procedure of BASEMENT explains how the discretised shallow water equation (eq. 1.1) is solved inside a defined time step  $\Delta t$  through a sequence of loops over the edges or cells (Figure 2.4).

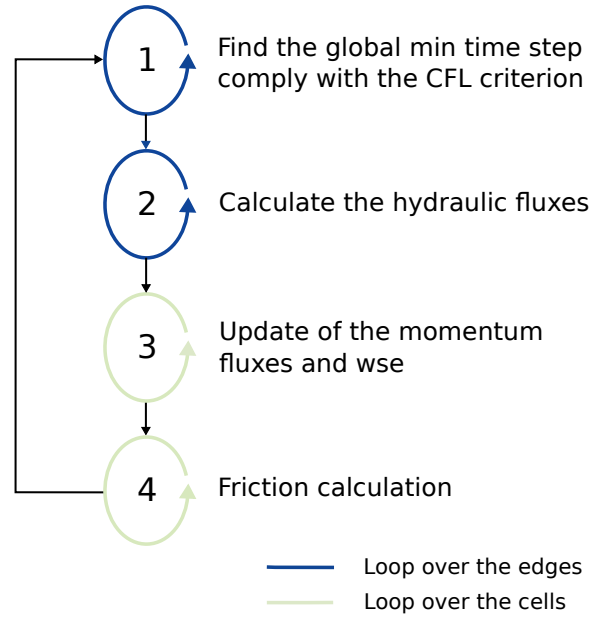
First, a global minimum time step  $\Delta t$  should be defined. Then, the hydraulic fluxes (liquid mass, x-momentum and y-momentum) are calculated with a HLLC Riemann solver at the element edges according to the initial states of the left and right cells (Section 2.3.3). Subsequently, the hydraulic state variables i.e. cell centered quantities are updated and finally, the friction (source term) is calculated using an implicit scheme, thus looping twice over the cell.

## 2.4 Numerical solution of Morphodynamics

### 2.4.1 Numerical solution of the Exner equation

#### 2.4.1.1 Fundamentals

The Exner equation assures that sediment mass is conserved in the bed and is used to model the riverbed time evolution. The rate of sediment transport is determined using a closure equation. The cell centered finite volume approach is used to discretise the Exner equation and in particular the HLL approximate Riemann solver with a wave speed



**Figure 2.4** Numerical solution procedure of hydrodynamic simulation for each time step  $\Delta t$

estimator defined in Soares-Frazão and Zech (2011) is adopted. The shallow water and the Exner equations create a system of equations that is solved in a decoupled way (Figure 2.2). This approach makes the assumption that the bed load flux is much slower than the water flow velocity (Soares-Frazão and Zech, 2011).

#### 2.4.1.2 Spatial discretization

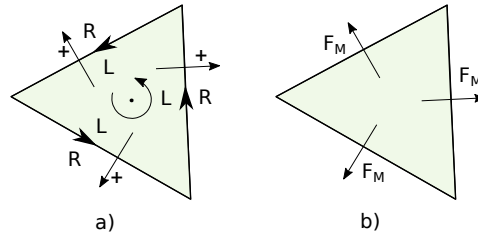
In order to discretise the the Exner equation, we use the same unstructured mesh adopted for the hydrodynamic part and the same finite volume approach. As a consequence, the bed level  $z_B$  is defined at the element center and is equally distributed over the element.

By integrating the Exner equation in the control volume  $V = [\Omega_i] \times [t^n, t^{n+1}]$ , we obtain

$$z_{Bi}^{n+1} = z_{Bi}^n - \frac{\Delta t}{|\Omega_i|} \sum_{j=1}^3 [q_{Bij} \cdot l_{ij}] + \Delta t \mathbf{S}_i . \quad (2.18)$$

The calculation of the sediment flux at the cell interface proceeds as follows:

1. loop over the cells and calculate:
  1. correction terms for the bed-load vector directions (if selected by the user), therefore:
    - calculation of the local bed slope, for the lateral-transport correction (see section Section 1.2.2.6.1)
    - calculation of the local curvature of the flow field, for the spiral flow correction (see section Section 1.2.2.6.2)



**Figure 2.5** a) Sign convention for the edge direction: counterclockwise b) Positive morphological flux direction at edges: from left (L) to right (R)

2. loop over the cell interfaces and:

1. calculate the flux projection along the normal vector ( $n_{ij,x}, n_{ij,y}$  of edge  $\Gamma_{ij}$ , i.e.:  $q_{Bi,n} = q_{Bi,x} \cdot n_{ij,x} + q_{Bi,y} \cdot n_{ij,y}$  and  $q_{Bj,n} = q_{Bj,x} \cdot n_{ij,x} + q_{Bj,y} \cdot n_{ij,y}$  with  $j=1,2,3$ )
  2. compute the flux at the interface using the approximate HLL Riemann solver at the interface
- Evaluate the wave speeds at the interface. this is obtained following the approach proposed by Soares-Frazão and Zech (2011), for which the wave speeds can be calculated as an approximation of the smallest eigenvalue of the system of governing equations, i.e. Shallow water and Exner. They read:

$$\lambda_1 = 1/2(u_n - c - \sqrt{(u_n - c)^2 + 4a_2c^2}) \quad (2.19)$$

$$\lambda_2 = 1/2(u_n - c + \sqrt{(u_n - c)^2 + 4a_2c^2}) \quad (2.20)$$

where  $u_n = u \cdot n_{ij,x} + v \cdot n_{ij,y}$ ,  $c = \sqrt{gh}$  and  $a_2 = \frac{\partial q_{b,n}}{\partial q_n}$  which is the derivative of the bed load discharge in the normal flow direction with respect to the hydraulic flux direction. Then the speeds estimate are

$$S^- = \min(\lambda_{1,L}, \lambda_{1,R}) \quad (2.21)$$

and

$$S^+ = \max(\lambda_{2,L}, \lambda_{2,R}) \quad (2.22)$$

- Flux calculation:

$$q_{Bij}^{HLL} = \begin{cases} q_{Bi,n} & \text{if } S^- \geq 0, \\ \frac{q_{Bi,n}S^+ - q_{Bj,n}S^- + S^-S^+(z_{Bj} - z_{Bi})}{S^+ - S^-} & \text{if } S^- < 0 < S^+, \\ q_{Bj,n} & \text{if } S^+ \leq 0. \end{cases} \quad (2.23)$$

The convention for the positive bed load flux direction is the same as for the hydrodynamic flux and is presented on Figure 2.5

### 2.4.1.3 Discretization of External Source Term

The source term  $S_{l_b}$  describes a local input or removal of sediment mass into a river.

An external source is defined as specific mass flux  $\delta$  (m/s), uniformly distributed over a number of elements of the domain (region) with a specific surface area. The external source can be specified as the total volume flux ( $m^3/s$ ) for a specific region of the domain. The external source value is divided among the cells composing the region and converted to cell specific mass flux  $\delta_i$ . The volume allocated is characterized by different behaviors:

$$\begin{aligned}
 \text{Exact:} & \quad S_{b,i} = \delta_i \\
 \text{Available:} & \quad S_{b,i} = \delta_i & \text{if } \delta_i \cdot \Delta t > 0 \\
 & \quad S_{b,i} = \max(\delta_i, -(z_{Fix} - z_i)) & \text{if } \delta_i \cdot \Delta t < 0 \\
 \text{Infinity:} & \quad S_{b,i} = \delta_i & \text{if } \delta_i \cdot \Delta t > 0 \\
 & \quad S_{b,i} = -(z_{Fix} - z_i) & \text{if } \delta_i \cdot \Delta t < 0
 \end{aligned} \tag{2.24}$$

Where  $z_i$  is the bottom elevation and  $z_{Fix}$  the fixed bed elevation of the element  $i$ . The external source volume is added to the initial bottom elevation of element  $i$ .

$$z_i^{t+1} = z_i^t + S_{b,i} \cdot \Delta t \tag{2.25}$$

## 2.4.2 Solution procedure

The numerical solution procedure of BASEMENT explains how the discretised Exner equation (eq. 1.22) is solved through a sequence of loops over the edges or cells (Figure 2.6).

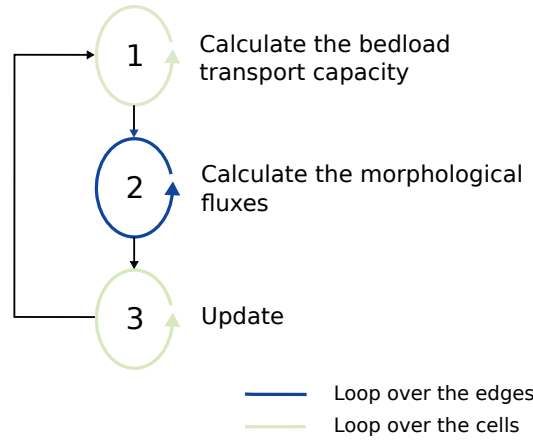
In the numerical simulation, the hydrodynamic and morphodynamic simulations are performed in a decoupled way. The morphodynamic simulation is executed after the hydrodynamic simulation, using the hydraulic fluxes to calculate the morphological fluxes. This approach assumes that the sediment transport is much slower than the water velocity, which is an accurate assumption for the numerical modelling of slow flood with morphological changes occurring over a long period (Soares-Frazão and Zech, 2011). The numerical solution procedure of Figure 2.6 is performed after the step 4 of Figure 2.4 inside the same time step  $\Delta t$ .

The numerical solution of the Exner equation starts with a loop over the cells in order to find the bedload transport capacity  $q_b$  with a potential correction due to a curvature effect or lateral bed slope. Then, the morphological fluxes  $F_M$  are calculated at the element edges and finally, the bed elevation  $z_b$  is updated over the cells.

## 2.5 Numerical solution of Advection-Diffusion equation

### 2.5.1 Numerical solution of the advective part

The scalar advection equation assures that scalar masses are conserved in the flow column when moving with the surrounding fluid. The same cell cell-centered finite volume approach used for the SWE is employed in scalar advection, namely the HLLC approximate Riemann solver. In order to discretize the scalar advection equation, the same unstructured mesh adopted for the hydrodynamic and morphodynamic parts is used, as is the same finite



**Figure 2.6** Numerical solution procedure of morphodynamic simulation for each time step  $\Delta t$

volume approach. As a consequence, the scalar concentration  $\phi_s$  is defined at the element center and is equally distributed over the element. This numerical approach is applied to turbulent kinetic energy quantities ( $\kappa, \epsilon$ ), suspended sediment and tracers.

By integrating the scalar advection equation

$$\frac{\partial}{\partial t} \phi_k h + \frac{\partial}{\partial x_i} \left( \phi_k h u_i - h D_{s_{ij}} \frac{\partial \phi_k}{\partial x_j} \right) = S_{\phi_k} \quad (2.26)$$

in the control volume  $V_i = [\Omega_i] \times [t^n, t^{n+1}]$  for a generic quantity defined by  $\phi_k$ , we obtain:

$$q_{\phi_k i}^{n+1} = q_{\phi_k i}^n - \frac{\Delta t}{|\Omega_i|} \sum_{j=1}^3 [f_{\phi_k, ij} \cdot l_{ij}] + \Delta t S_{\phi_k, i} \quad (2.27)$$

where  $f_{s_{ij}}$  is the intercell scalar advective flux. The scalar advective fluxes at the cell interface are defined by:

$$f_{\phi_k, ij} = \frac{q_{\phi_k i}}{h} q_m \quad \text{if } q_m > 0 \quad (2.28)$$

$$f_{\phi_k, ij} = \frac{q_{\phi_k j}}{h} q_m \quad \text{if } q_m < 0 \quad (2.29)$$

where  $q_m$  is the hydrodynamic mass flux. This approach significantly reduces the numerical effort involved in the computation of the numerical fluxes for each species, without sacrificing accuracy or stability.

### 2.5.2 Numerical solution of the diffusive part

The diffusion equation ensures that quantities are conserved in the flow column due when turbulent or molecular exchanges originate some form of mixing. By integrating the scalar diffusion equation



$$\frac{\partial}{\partial t} \phi_k h - \frac{\partial}{\partial x_i} \left( h D_{s_{ij}} \frac{\partial \phi_k}{\partial x_j} \right) = 0 \quad (2.30)$$

in the control volume  $V_i = [\Omega_i] \times [t^n, t^{n+1}]$  we obtain:

$$q_{\phi_k i}^{n+1} = q_{\phi_k i}^n - \frac{\Delta t}{|\Omega_i|} \sum_{j=1}^3 [d_{\phi_k, ij} \cdot l_{ij}] \quad (2.31)$$

where  $d_{\phi_k, ij}$  is the intercell scalar diffusive flux. To compute the solution of the diffusive terms, the SVT solver Vanzo et al. (2016) is employed. It involves the augmentation of the existing system with a set of new conservation quantities and equations for each diffusive specie  $\phi_k$ . The role of these new variables is to allow a relaxation of the diffusive part, such as

$$\lim_{\zeta \rightarrow 0} \psi_x^{\phi_k} = \partial_x \phi_k, \quad \lim_{\zeta \rightarrow 0} \psi_y^{\phi_k} = \partial_y \phi_k \quad (2.32)$$

where  $\psi$  denotes a relaxation variable, subject to the following relations

$$\partial_t \psi_x^{\phi_k} - \partial_x \frac{\phi_k}{\zeta} = -\frac{\psi_x^{\phi_k}}{\zeta}, \quad \partial_t \psi_y^{\phi_k} - \partial_y \frac{\phi_k}{\zeta} = -\frac{\psi_y^{\phi_k}}{\zeta} \quad (2.33)$$

which are formally denoted as the relaxation sub-system. The main purpose of this relaxation is to preserve the hyperbolicity of the numerical scheme, as demonstrated by Vanzo et al. (2016). To construct the diffusive numerical fluxes, the following Riemann problem is considered along some edge normal as

$$\begin{cases} \partial_t \Phi_k + \partial_\xi D_{n_{ij}} = \mathbf{0}, & \xi \in \mathbb{R}, t > 0, \\ \Phi_k(\xi, 0) = \begin{cases} \Phi_{k_i}^n & \text{if } \xi < 0, \\ \Phi_{k_j}^n & \text{if } \xi > 0, \end{cases} \end{cases} \quad (2.34)$$

where  $\Phi_k$  is the conserved variables vector and  $D_n$  is the diffusive numerical flux vector. The structure of this Riemann problem is composed of three unique waves (from a total of six). One of these is always a stationary contact discontinuity and the two remaining waves are symmetrical, with the same propagation speed in opposite directions. The problem thus becomes characterized by a single intermediate state  $\Phi_k^*$  where the fluxes are directly evaluated. The exact expression for the intermediate state  $\Phi_k^*$  and numerical  $D(\Phi_k^*)_n$  can be found in Vanzo et al. (2016).

### 2.5.3 Discretization of external source terms

The source term  $S_{\phi_k}$  describes a local input or removal of scalar mass into a river. An external source is defined as specific mass flux  $\delta$  (m/s), uniformly distributed over a number of elements of the domain (region) with a specific surface area. The external source can be specified as the total volume flux ( $m^3/s$ ) for a specific region of the domain. The external source value is divided among the cells composing the region and converted to cell specific mass flux  $\delta_i$ . The volume allocated is characterized by different behaviors:

$$\begin{array}{ll}
\text{Exact:} & S_{\phi_{k,i}} = \delta_i \\
\text{Available:} & S_{\phi_{k,i}} = \delta_i \quad \text{if } \delta_{\phi_k} \cdot \Delta t > 0 \\
& S_{\phi_{k,i}} = \max(\delta_i, -q_{\phi_{k,i}}) \quad \text{if } \delta_i \cdot \Delta t < 0 \\
\text{Infinity:} & S_{\phi_{k,i}} = \delta_i \quad \text{if } \delta_i \cdot \Delta t > 0 \\
& S_{\phi_{k,i}} = -q_{\phi_{k,i}} \quad \text{if } \delta_i \cdot \Delta t < 0
\end{array} \tag{2.35}$$

The external source volume is then added to the initial value of element  $i$  through a first-order Euler approach.

$$q_{\phi_{k,i}}^{t+1} = q_{\phi_{k,i}}^t + S_{\phi_{k,i}} \cdot \Delta t \tag{2.36}$$

### 2.5.4 Solution procedure

The numerical solution for all modules involving scalar advection and diffusive terms – turbulence, suspended sediment and tracers – is obtained as follows:

1. Loop over the cell interfaces and compute the advective flux at each interface:
  - Retrieve the hydrodynamic fluxes  $\mathbf{F}_{ij}$  at the interface between cells  $i$  and  $j$  and extract the first component of the flux vector (mass fluxes) to the variable  $q_m$ .
  - For each specie, perform the advective flux calculation through the simplified HLLC solver described above
2. Loop over the cell interfaces and compute the diffusive flux at each interface:
  - Retrieve the hydrodynamic and scalar quantities  $\mathbf{Q}_h$  and  $\Phi_k$  at the interface between cells  $i$  and  $j$  perform the flux calculation using the SVT solver
3. Define the diffusion specific timestep and update the global minimum time step  $\Delta t$
4. Loop over the cells and update the conserved quantities with the fluxes at each of its interfaces:
  - Retrieve and add the advective fluxes  $f_{\phi_{ij}}$  and diffusive fluxes  $d_{\phi_{ij}}$  at each of the three interfaces.
  - For each specie, perform the update as prescribed in Equations 2.27, 2.31, and 2.36.
5. Add the external source terms

## 2.6 Numerical solution of the thermodynamics equation

The numerical integration of the evolution equation for the thermodynamics (eq. eq. 1.83) is analogous to the numerical strategy of section Section 2.5. Hence, for details of the numerical integration of the advective and diffusive parts, please cf. sections Section 2.5.1 and Section 2.5.2, respectively.

### 2.6.1 Discretization of external source terms

The temperature source terms (right-hand side of eq. Section 1.6.2) describes the thermal exchanges of the column of water with the atmosphere and the river bed. The calculated source contributions (i.e. energy fluxes) are added/subtracted exactly from the thermal mass of each computational cell. Then the external source volume is then added to the initial value of element  $i$  through a first-order semi-implicit Euler approach:

$$q_T^{t+1} = q_T^t + \frac{(H_{atm} + H_b)}{(c_w \rho_w - \Delta t \partial_T (H_{atm} + H_b))} \cdot \Delta t \quad (2.37)$$

### 2.6.2 Solution procedure

The numerical solution is equivalent to the other modules involving scalar advection and diffusive terms and it is obtained as follows:

1. Loop over the cell interfaces and compute the advective flux at each interface:
  - Retrieve the hydrodynamic fluxes at the interface between cells  $i$  and  $j$  and extract the first component of the flux vector (mass fluxes), then perform the advective flux calculation as in Section 2.5.1.
1. Loop over the cell interfaces and compute the diffusive flux at each interface:
  - Retrieve the hydrodynamic and thermal quantities  $\mathbf{Q}$  and  $\mathbf{q}_T$  at the interface between cells  $i$  and  $j$  perform the flux calculation using the SVT solver (Section 2.5.2).
3. Define the diffusion specific timestep and update the global minimum time step  $\Delta t$
4. Loop over the cells and update the conserved quantities with the fluxes at each of its interfaces:
  - Retrieve and add the thermal advective and diffusive fluxes at each of the three interfaces.
  - Perform the update of the advective and diffusive thermal fluxes, equivalent to equations eq. 2.27} and eq. 2.31}, respectively.
1. Add the external source terms with eq. eq. 2.37.

---

## References

- 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.
- Beltaos, S. (2013). River ice formation. *Committee on river ice processes; the environment, Canadian geophysical . . .*.
- 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). Fließwiderstand und Sohlenstabilität natürlicher Gerinne [PhD thesis]: Eidgenössische Technische Hochschule Zürich.
- Caissie, D. (2006). The thermal regime of rivers: A review. *Freshwater biology*, 51(8): 1389–1406.
- Caissie, D. and Luce, C.H. (2017). [Quantifying streambed advection and conduction heat fluxes](#). *Water Resources Research*, 53(2): 1595–1624.
- Caponi, F. and Siviglia, A. (2018). [Numerical modeling of plant root controls on gravel bed river morphodynamics](#). *Geophysical Research Letters*, 45(17): 9013–9023.
- Caponi, F., Vetsch, D.F. and Siviglia, A. (2020). [A model study of the combined effect of above and below ground plant traits on the ecomorphodynamics of gravel bars](#). *Scientific Reports*, 10(1): 1–14.
- Chen, X., Ma, J. and Dey, S. (2010). Sediment transport on arbitrary slopes: Simplified model. *Journal of Hydraulic Engineering-ASCE*, 136(5): 311–317.
- Dugdale, S.J., Hannah, D.M. and Malcolm, I.A. (2017). River temperature modelling: A review of process-based approaches and future directions. *Earth-Science Reviews*, 175: 97–113.
- Duran, A., Liang, Q. and Marche, F. (2013). On the well-balanced numerical discretization of shallow water equations on unstructured meshes. *Journal of Computational Physics*, 235: 565–586.
- Edmaier, K., Burlando, P. and Perona, P. (2011). Mechanisms of vegetation uprooting by flow in alluvial non-cohesive sediment. *Hydrology and Earth System Sciences*, 15(5): 1615–1627.
- 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*.
- Exner, F.M. (1925). Ueber die wechselwirkung zwischen wasser und geschiebe in fluessen. Akademie der Wissenschaften, Mathematische Naturwissenschaft Abt. IIa, Wien, Austria.
- Grass, A.J. (1981). Sediment transport by waves and currents. *University College, London, Dept. of Civil Engineering, London, UK*.
- Hirano, M. (1971). *Proceedings of the japan society of civil engineers*, No. 1971: 55–65. *Japan Society of Civil Engineers*.
- 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.
- 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.
- Malcherek, A. (2001). Sedimenttransport und Morphodynamik. Vorlesungsskript der Universität der Bundeswehr München, München.
- Meyer-Peter, E. and Müller, R. (1948). Formulas for Bed-Load Transport, 2nd Meeting IAHR, Stockholm, Sweden.
- Parker, G. (1990). Surface-based bedload transport relation for gravel rivers. *Journal of Hydraulic Research*, 28(4): 417–436.
- Parker, G. and Sutherland, A.J. (1990). Fluvial armor. *Journal of Hydraulic Research*.
- Rozovskii, I.L. (1961). 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.
- Siviglia, A. and Toro, E.F. (2009). [WAF method and splitting procedure for simulating hydro-and thermal-peaking waves in open-channel flows](#). *Journal of Hydraulic Engineering*, 135(8): 651–662.
- Smart, G.M. and Jaeggi, M.N.R. (1983). Sediment Transport on Steep Slopes. *VAW-Mitteilung 64*, Versuchsanstalt für Wasserbau, Hydrologie und Glaziologie (VAW). Zürich, ETH Zürich.
- Soares-Frazão, S. and Zech, Y. (2011). HLLC scheme with novel wave-speed estimators appropriate for two-dimensional shallow-water flow on erodible bed. *International Journal for Numerical Methods in Fluids*, 66.
- Talmon, A.M., Struiksmā, N. and van Mierlo, M.C.L.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. (2009). 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*.

- van Rijn, L.C. (1989). Handbook Sediment Transport by Current and Waves. *Delft Hydraulics Laboratory*, Delft, The Netherlands.
- van Rijn, L.C. (1984). Sediment Transport, Part II: Suspended Load Transport. *Journal of Hydraulic Engineering, ASCE*, 110(11): 1613–1641.
- Vanzo, D., Peter, S., Vonwiller, L., Bürgler, M., Weberndorfer, M., Siviglia, A., Conde, D. and Vetsch, D.F. (2021). BASEMENT v3: A modular freeware for river process modelling over multiple computational backends. *Environmental Modelling & Software*, 143: 105102.
- Vanzo, D., Siviglia, A. and Toro, E.F. (2016). Pollutant transport by shallow water equations on unstructured meshes: Hyperbolization of the model and numerical solution via a novel flux splitting scheme. *Journal of Computational Physics*, 321: 1–20.
- Vonwiller, L. (2017). Numerical Modeling of Morphological Response of Gravel-Bed Rivers to Sediment Supply. *VAW-Mitteilung 246*, Versuchsanstalt für Wasserbau, Hydrologie und Glaziologie (VAW). Zürich, ETH Zürich.
- Wilcock, P.T. and Crowe, J.C. (2003). Surface-based transport model for mixed-size sediment. *Journal of Hydraulic Engineering*.
- 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*.
- Zhang, R.J. (1961). River Dynamics. *Industry Press*, Beijing, China.
- Zyserman, J.A. and Fredsøe, J. (1994). Data Analysis of Bed Concentration of Suspended Sediment. *Journal of Hydraulic Engineering, ASCE*, 120(9): 1021–1042.







**BASIC SIMULATION ENVIRONMENT  
FOR MODELLING OF ENVIRONMENTAL  
FLOWS AND NATURAL HAZARDS**

---

# **TUTORIALS**

**VERSION 4.1.0  
JUNE 2024**



**BASEMENT**



# Contents

<b>1</b>	<b>Introduction to BASEmesh</b>	<b>5</b>
1.1	Introduction to BASEmesh . . . . .	5
1.2	Mesh Generation based on Pointwise Elevation Data . . . . .	7
1.2.1	Project Settings . . . . .	7
1.2.2	Coordinate Reference System Configuration . . . . .	8
1.2.3	Loading Input Data for Elevation Model . . . . .	8
1.2.4	Converting Input Elevation Data to 3D Points . . . . .	9
1.2.5	Creating 3D Lines for Elevation Meshing . . . . .	11
1.2.6	Loading the Breaklines . . . . .	14
1.2.7	Editing the Model Boundary . . . . .	14
1.2.8	Creation of the Elevation Mesh . . . . .	16
1.2.8.1	Creation of the Elevation Mesh using 3D points . . . . .	16
1.2.8.2	Creation of the Elevation Mesh using 3D Lines . . . . .	17
1.2.9	Adaption of the Breaklines for Quality Meshing . . . . .	17
1.2.10	Creation of Region Marker Points . . . . .	20
1.2.11	Creation of Quality Mesh . . . . .	22
1.2.12	Generation of the Computational Mesh . . . . .	22
1.2.13	Viewing the computational mesh . . . . .	23
1.3	BASEmesh Tips and Tricks . . . . .	26
1.3.1	Recommended Plugins . . . . .	26
1.3.2	Tips and Tricks . . . . .	26
<b>2</b>	<b>Post-Processing of 2D simulation results</b>	<b>31</b>
2.1	Introduction . . . . .	31
2.2	2D result visualization with QGIS . . . . .	31
2.2.1	Input data . . . . .	31
2.2.2	About Crayfish . . . . .	32
2.2.3	Installation . . . . .	32
2.2.4	Load and visualize data . . . . .	33
2.2.4.1	Project Settings . . . . .	33
2.2.4.2	Loading results of unsteady flow simulation (river Flaz) . . . . .	33
2.2.4.3	Display properties . . . . .	35
2.2.4.4	Export results to raster . . . . .	37
2.2.4.5	Profile tool . . . . .	38
2.2.5	Creating maps and animations . . . . .	41
2.2.5.1	Customized layout (QGIS print Layout) . . . . .	42
2.2.5.2	Export animation . . . . .	45
2.3	3D result visualization with Paraview . . . . .	48

2.3.1	Input data . . . . .	48
2.3.2	About Paraview . . . . .	48
2.3.3	Installation . . . . .	48
2.3.4	User Interface . . . . .	49
2.3.5	Import Data . . . . .	50
2.3.6	ParaView Filters . . . . .	50
2.3.6.1	Overview . . . . .	50
2.3.6.2	Applying Filters to data . . . . .	51
2.3.7	Exporting figures and animations . . . . .	56
2.3.7.1	Save Screenshot (picture) . . . . .	56
2.3.7.2	Save Animation (movie) . . . . .	57
<b>3</b>	<b>Hydrodynamics and sediment transport at the river Flaz with BASEHPC (2D)</b>	<b>59</b>
3.1	Hydrodynamics and sediment transport at the river Flaz with BASEHPC (2D) . . . . .	59
3.1.1	Introduction . . . . .	59
3.1.2	Computational Mesh . . . . .	59
3.1.3	Hydraulics . . . . .	60
3.1.3.1	Setup the Configuration File model.json . . . . .	60
3.1.3.2	Setup the Configuration File simulation.json . . . . .	63
3.1.3.3	Set up the Configuration File results.json . . . . .	63
3.1.3.4	Steady Flow Simulation . . . . .	64
3.1.3.5	Model Calibration . . . . .	64
3.1.3.6	Unsteady Flow Simulation . . . . .	64
3.1.4	Morphology with uniform sediment transport . . . . .	66
3.1.4.1	Setup the Configuration File model.json . . . . .	66
3.1.5	Morphology with mixed-size sediment transport . . . . .	71
3.1.5.1	Setup the Configuration File model.json . . . . .	71
3.1.5.2	Setup the Configuration File simulation.json . . . . .	75
3.1.5.3	Results . . . . .	75
3.1.6	Passive tracers . . . . .	75
3.1.6.1	Setup the Configuration File model.json . . . . .	75
3.1.6.2	Setup the Configuration File simulation.json . . . . .	78
3.1.6.3	Set up the Configuration File results.json . . . . .	78
<b>4</b>	<b>Hydrodynamics and sediment transport at the river Flaz with BASEMD (2D)</b>	<b>81</b>
4.1	Hydrodynamics and sediment transport at the river Flaz with BASEMD (2D) . . . . .	81
4.1.1	Introduction . . . . .	81
4.1.1.1	Case study description . . . . .	81
4.1.1.2	Tutorial structure . . . . .	82
4.1.2	Computational grid . . . . .	82
4.1.3	Setting up the model file . . . . .	84
4.1.3.1	BASEMD . . . . .	84
4.1.3.2	Physical properties . . . . .	85
4.1.3.3	Two dimensional domain . . . . .	85
4.1.3.4	Geometry . . . . .	85

4.1.3.5	Define the hydraulics . . . . .	86
4.1.3.6	Write the setup file . . . . .	90
4.1.3.7	Perform hydraulic simulations . . . . .	90
4.1.4	Morphological simulation with single-grain bed load transport . . . .	95
4.1.4.1	Define the morphological information . . . . .	96
4.1.4.2	Bedload . . . . .	100
4.1.4.3	Gravitational transport . . . . .	102
4.1.4.4	Define the output . . . . .	102
4.1.4.5	Perform morphological simulation with single-grain bed load transport . . . . .	103
4.1.5	Morphological simulation with multi-grain bed load transport . . . .	103
4.1.5.1	Morphological parameters . . . . .	103
4.1.5.2	Grain size distribution . . . . .	106
4.1.5.3	Grain mixture . . . . .	106
4.1.5.4	Define the soil composition . . . . .	107
4.1.5.5	Bed load boundary condition . . . . .	108
4.1.5.6	Bed load formula . . . . .	108
4.1.5.7	Define the output . . . . .	109
4.1.6	Perform morphological simulation with multi-grain bed load transport	109
<b>5</b>	<b>Hydrodynamics and sediment transport at the river Thur with BASEMD (1D)</b>	<b>111</b>
5.1	Hydrodynamics and sediment transport at the river Thur with BASEMD (1D)	111
5.1.1	Introduction . . . . .	111
5.1.1.1	General description . . . . .	111
5.1.1.2	Used features . . . . .	111
5.1.1.3	Purpose . . . . .	112
5.1.2	Setting up the topography file . . . . .	112
5.1.2.1	Cross sections . . . . .	112
5.1.2.2	Definition of different cross section zones . . . . .	114
5.1.2.3	Friction values . . . . .	114
5.1.2.4	Computation of water surface elevation . . . . .	116
5.1.2.5	Characterization of the sediments . . . . .	116
5.1.2.6	Define flowing zones . . . . .	117
5.1.3	Setting up the model file . . . . .	118
5.1.3.1	Setup . . . . .	118
5.1.3.2	Domain . . . . .	118
5.1.3.3	Define the physical properties . . . . .	118
5.1.3.4	One dimensional simulation . . . . .	118
5.1.4	Perform hydraulic simulations . . . . .	121
5.1.4.1	Write the setup file . . . . .	121
5.1.4.2	Setting up the simulation file . . . . .	121
5.1.4.3	Perform steady flow simulation . . . . .	122
5.1.4.4	Results of the steady flow simulation . . . . .	122
5.1.4.5	Perform simulation of the floods (Thur2) . . . . .	123
5.1.5	Edit the model file for bed load transport . . . . .	125
5.1.5.1	Define the bed material . . . . .	126
5.1.5.2	Soil assignment . . . . .	127
5.1.5.3	Define general parameters for sediment transport . . . . .	127

5.1.5.4	Define specific parameters for bed load transport . . . . .	128
5.1.5.5	Define bed load transport formula . . . . .	128
5.1.5.6	Define boundary conditions for bed load . . . . .	128
5.1.5.7	Generate a “geometry” file . . . . .	129
5.1.6	Perform bed load simulation (Thur 3) . . . . .	129
<b>6</b>	<b>Dynamics of longitudinal bed profile due to local river widening with BASEMD (1D)</b>	<b>131</b>
6.1	Dynamics of longitudinal bed profile due to local river widening with BASEMD (1D) . . . . .	131
6.1.1	Introduction . . . . .	131
6.1.1.1	General description . . . . .	131
6.1.1.2	Purpose . . . . .	132
6.1.1.3	Used features . . . . .	133
6.1.1.4	Parameter variation . . . . .	134
6.1.2	Model setup . . . . .	134
6.1.2.1	Definition of 1D topography . . . . .	134
6.1.2.2	Determination of the upstream sediment boundary conditions	135
6.1.3	Results of numerical simulations . . . . .	136
6.1.3.1	Temporal evolution of longitudinal bed profile . . . . .	136
6.1.3.2	Width and length of the widening . . . . .	136
6.1.3.3	Discharge and grain diameter . . . . .	136
<b>7</b>	<b>Laterally coupled 1D-2D hydrodynamic simulation with BASEMD</b>	<b>141</b>
7.1	Laterally coupled 1D-2D hydrodynamic simulation with BASEMD . . . . .	141
7.1.1	Introduction . . . . .	141
7.1.2	Set-up of model file . . . . .	142
7.1.2.1	General remarks on mesh creation . . . . .	142
7.1.2.2	BASEchain (1D) river model . . . . .	144
7.1.2.3	BASEplane (2D) floodplain model . . . . .	145
7.1.2.4	Lateral coupling set-up . . . . .	146
7.1.3	Coupling connections between 1D and 2D subdomains . . . . .	146
7.1.3.1	Definition of coupling connections . . . . .	146
7.1.3.2	Automatic generation of coupling connections . . . . .	147
7.1.4	Perform coupled simulation . . . . .	149
<b>8</b>	<b>Vegetation dynamics and bedload transport with BASEHPC</b>	<b>151</b>
8.1	Vegetation dynamics and bedload transport with BASEHPC . . . . .	151
8.1.1	Introduction . . . . .	151
8.1.2	Computational Mesh . . . . .	151
8.1.3	Morphology with Vegetation . . . . .	151
8.1.3.1	Setup the Configuration File model.json . . . . .	152
8.1.3.2	Setup the Configuration File simulation.json . . . . .	154
8.1.3.3	Set up the Configuration File results.json . . . . .	155
8.1.3.4	Results . . . . .	155
<b>9</b>	<b>References</b>	<b>157</b>

---

# Introduction to BASEmesh

## 1.1 Introduction to BASEmesh

**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.

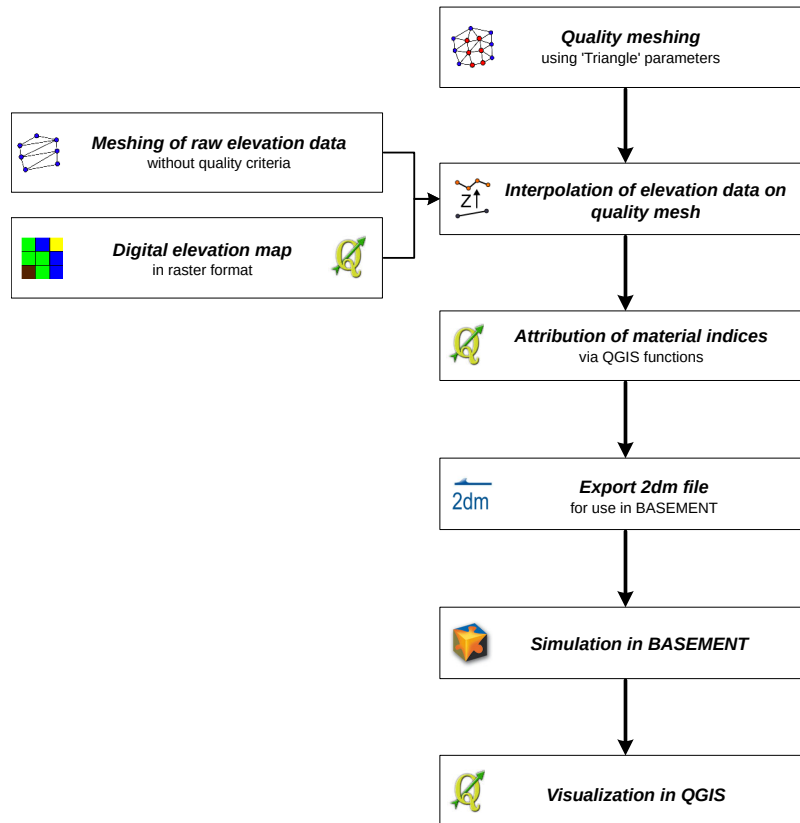
**What is the workflow of BASEmesh?** Please take a look at the figure below to see the workflow in BASEmesh. There are 3 main steps of the mesh generation process: (1) Create Elevation Mesh (2) Create Quality Mesh (3) Create Computational Mesh

For further details, please refer to Section 2.2 "2D grid generation with BASEmesh QGIS plugin" of the BASEMENT User Manual.

**This tutorial covers the following BASEmesh workflow:**

- Tutorial 1: Creation and export of a computational mesh for BASEMENT using pointwise elevation data.

**Future tutorials will cover the following workflows in BASEmesh:**



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

- 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.



## 1.2 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 .

**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 3.16 Hannover. 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 be used often 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.

Only 2 shapefiles are needed as input data for this tutorial. All other input files are created using BASEmesh and other QGIS functions. 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 2.2.1 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*.

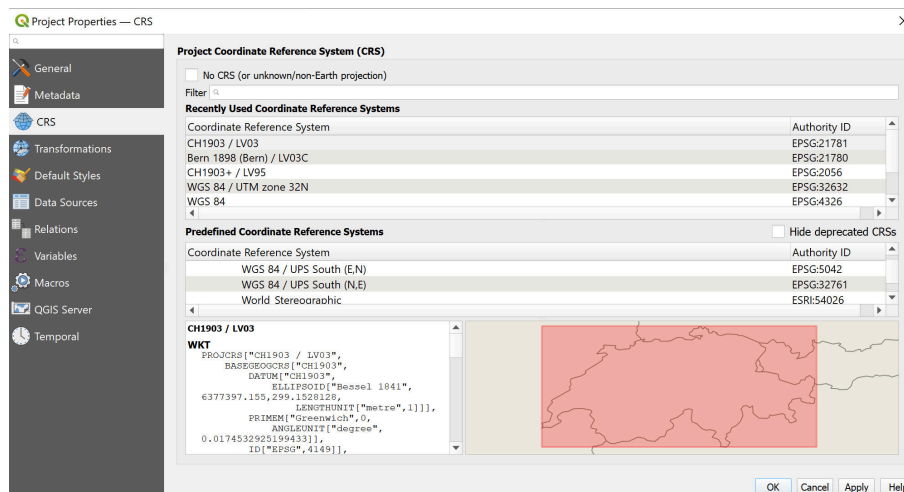


Figure 1.2 Project Properties, CRS

- (6) Select the coordinate reference system ‘CH1903 / LV03’.
- (7) Click *OK*.
- (8) Again go to *Project* → *Project Properties* → *CRS*.
- (9) 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.
- (10) Now save the project: Go to *Project* → *Save*. The project name chosen before is automatically proposed.

## 1.2.2 Coordinate Reference System Configuration

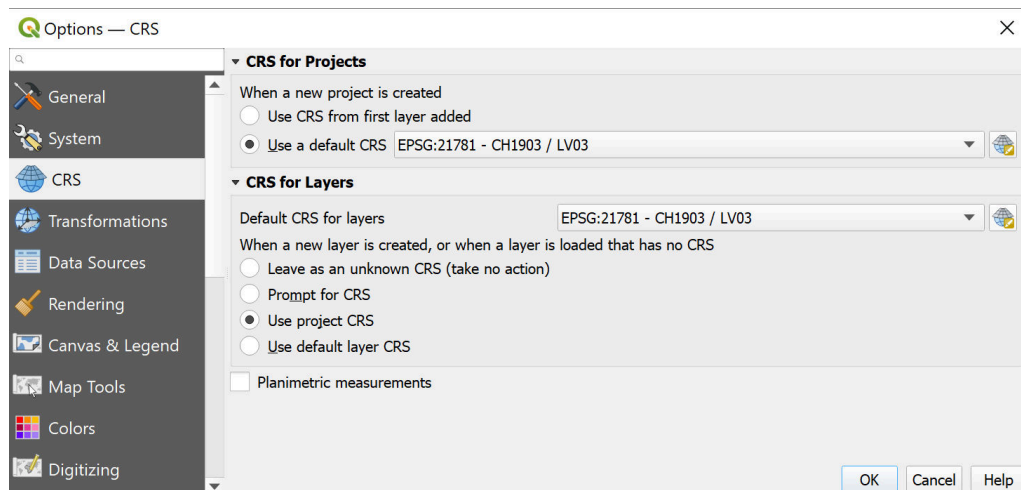
Now we need to check how QGIS determines the coordinate system of added layers. This may vary between QGIS versions and operating systems.

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

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

## 1.2.3 Loading Input Data for Elevation Model

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



*Figure 1.3 Settings, CRS*

- (1) Go to *Layer* → *Add 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, Y, and Z coordinates are set correctly.
- (3) Be sure to check *Use spatial Index*.
- (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 Converting Input Elevation Data to 3D Points

The loaded elevation data must be converted into a format that can be used by BASEmesh. We will first convert the data into a shapefile.

- (1) Right-click on layer ‘XS\_points\_straightened’ in the TOC.
- (2) Go to *Export* → *Save Features 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 Layer*. If both layers have the same symbology, you can check its properties by right-clicking and selecting *Properties*.

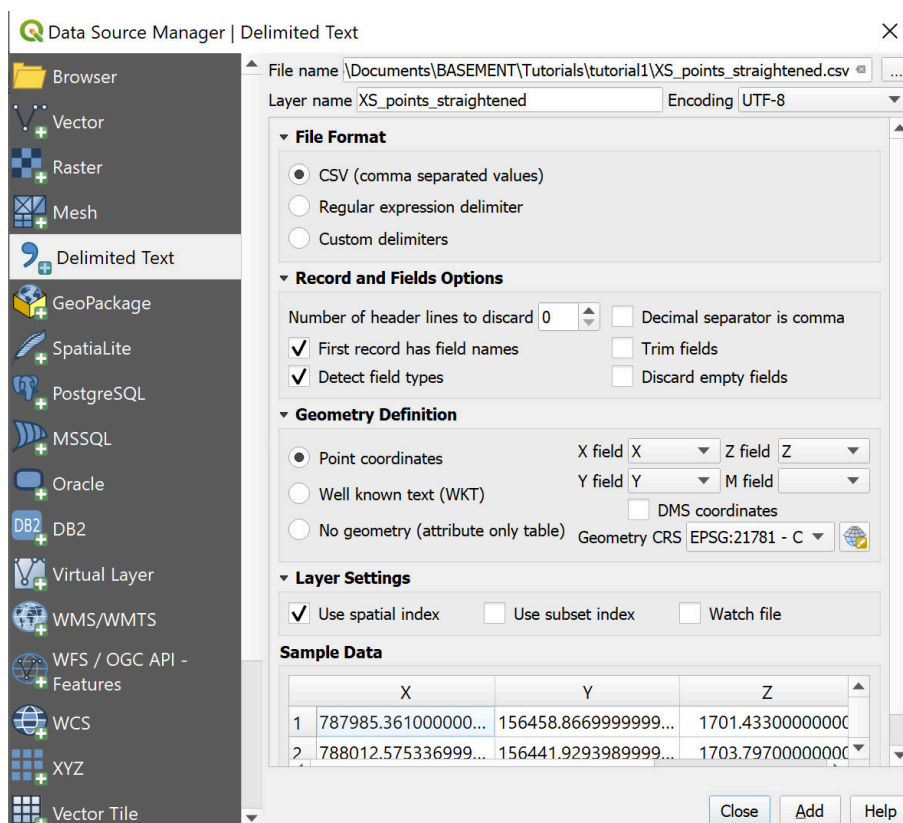


Figure 1.4 Add delimited text layer

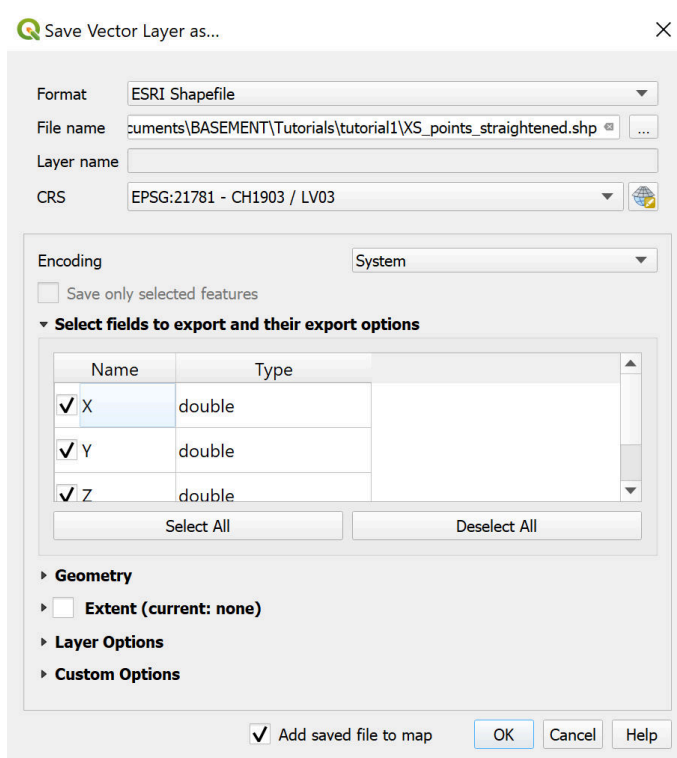
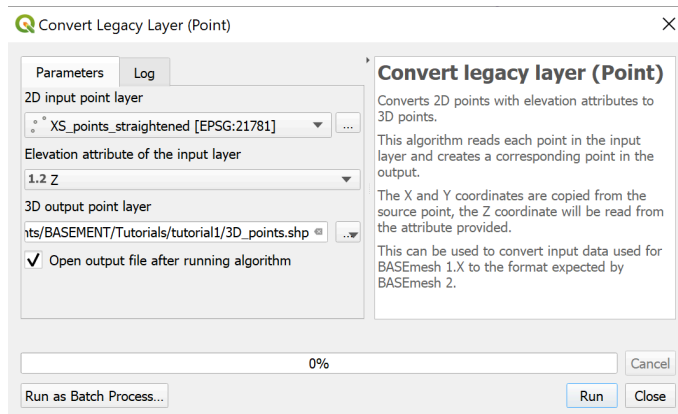


Figure 1.5 Save vector layer



**Figure 1.6** Convert points to 3D

The shapefile that we just generated contains 2D points. Now we will convert the 2D points into 3D points.

- (1) Open the BASEmesh “Convert legacy layer (Point)” tool. *Processing Toolbox* → *BASEmesh* → *Converters* → *Convert legacy layer (point)*.
- (2) Fill in the dialog according to Figure 1.6 and click run.

This 3D points file can be used in the BASEmesh *Elevation Meshing* tool.

### 1.2.5 Creating 3D Lines for Elevation Meshing

We will now use the elevation points to create 3D lines, which can also be used by the BASEmesh *Elevation Meshing* tool. First, we create a digital elevation map using the points shapefile.

- (1) Open the Natural neighbour tool from SAGA.
- (2) Fill in the dialog according to Figure 1.7 and click run (this may take a few minutes)

Now we will use the elevation points to create 2D lines, which represent each cross section of the river.

- (1) Open the QGIS points to path tool.
- (2) Fill in the dialog according to Figure 1.8 and click run (this may take a few minutes)

You will now see lines representing each cross section, plotted on top of the DEM, as in Figure 1.9. The 2D lines can now be converted to 3D lines, using the DEM.

- (1) Open the QGIS drape tool
- (2) Fill in the dialog according to Figure 1.10 and click run

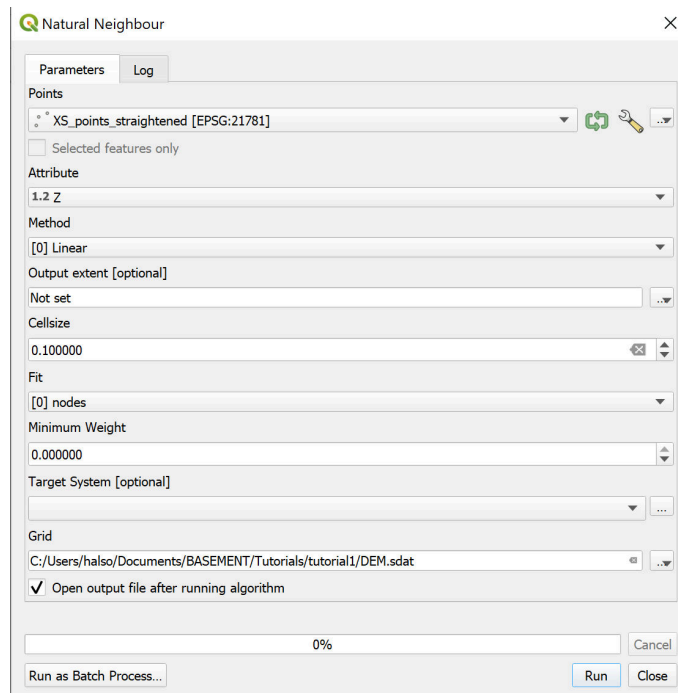


Figure 1.7 Natural Neighbor

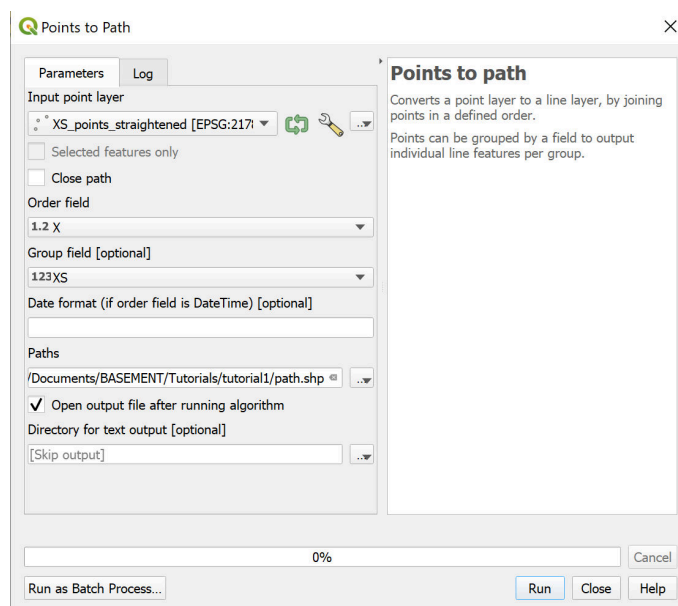
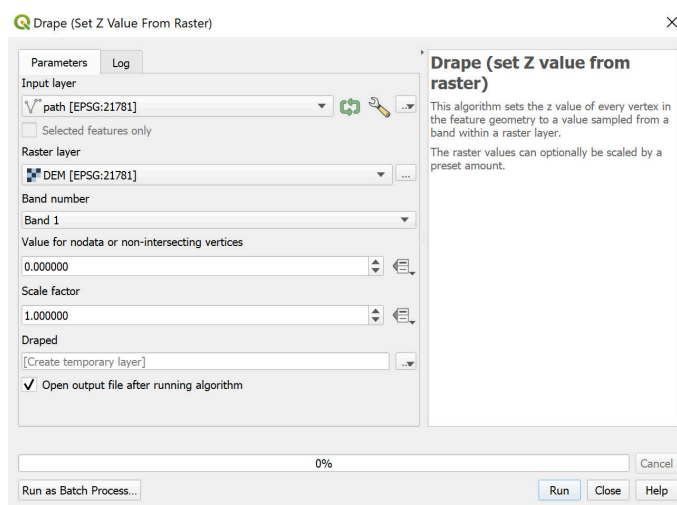


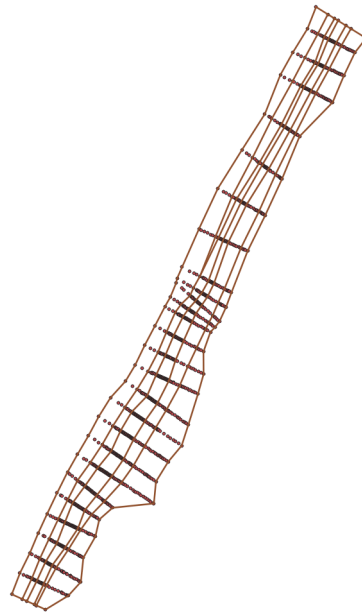
Figure 1.8 Points to Path



*Figure 1.9 2DLines on DEM*



*Figure 1.10 Tutorial 1.1 Drape*



*Figure 1.11 Breaklines*

### 1.2.6 Loading the Breaklines

- (1) Go to *Layer* → *Add Layer* → *Add Vector Layer...* (or Ctrl-Shift-V).
- (2) Browse for the shapefile provided with this tutorial ‘breaklines.shp’.
- (3) Click *Add*.

The color schemes of newly added layers are randomly chosen by QGIS. You can change them by double-clicking on a layer and selecting *Symbology*.

- (4) Pull the newly added layer ‘breaklines’ below the initial file ‘XS\_points\_straightened’. Your QGIS canvas should now look like in Figure 1.11.

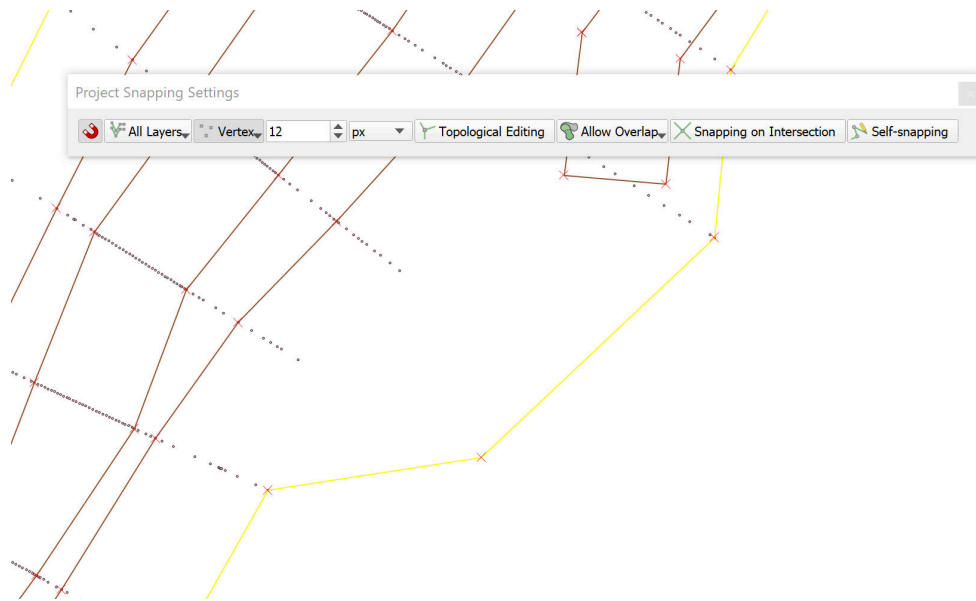
This breaklines shapefile contains several features, including one labeled ‘model\_boundary’. This feature gives the extents of the computational domain. Please note that for generating a computational mesh, the model boundary should not lie outside of the elevation points. Otherwise, there might be 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 1.3) of this Tutorial.

### 1.2.7 Editing the Model Boundary

All vertices of the model boundary polygon **must** lie within the extents of the elevation points. Otherwise there might be interpolation errors in the following steps. In our example, a portion of the model boundary does not satisfy this criteria (see lower right region of the model data). Therefore we have to move/add nodes of the boundary layer:

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





**Figure 1.12 Snapping**

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

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

- (3) Go to *Project* → *Snapping Options*. Fill in the form as shown in Figure 1.12 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.
- (4) Go to *Edit* → *Vertex 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 1.12).

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

- (7) Hover over the segment somewhere between the two boundary vertices where you want to have the new one. In the middle of the segment, a small plus sign (+) will appear. Click on that plus sign to create a new node. (Be careful with clicking: as you may move the entire segment by mistake. This 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*.

The breakline shapefile is now ready for Quality Meshing. But first, the Region Markers must be defined.

### 1.2.8 Creation of the Elevation Mesh


Using 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*. There are 3 methods for Elevation Meshing:

- (1) Creation of the Elevation Mesh with 3D points
- (2) Creation of the Elevation Mesh with 3D lines
- (3) Using a DEM as the Elevation Layer

In this tutorial, we will show how to perform the first 2 methods.

#### 1.2.8.1 Creation of the Elevation Mesh using 3D points

First, we will create an elevation mesh using the 3D lines that were created in Section 1.2.4.

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

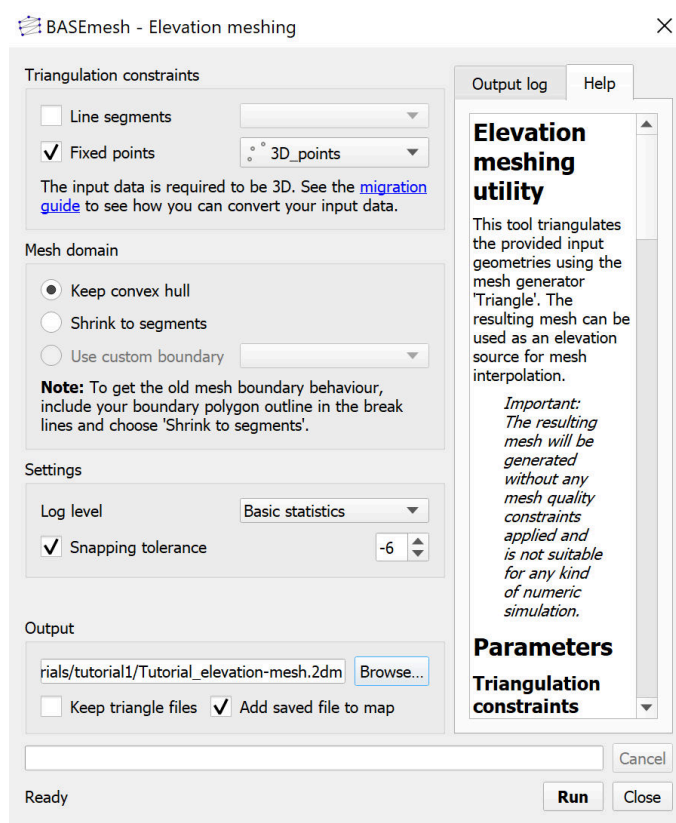
On the left side of the dialog (Figure 1.13) you can define the input layers. See Section 2.2 “2D grid generation with BASEmesh QGIS plugin” of the BASEMENT User Manual for further explanations. On the right side status messages as well as Triangle’s output messages are displayed during meshing. Tool-specific help can be found in the *Help*-tab.

- (2) Select the different fields according to Figure 1.13.
- (3) Choose an output filename. The default is the name of the project.
- (4) Click *run*.

This newly generated elevation mesh is an intermediate step and will be used as basis for the further interpolation of the elevation data. Due to its low mesh quality, it should **not** be used as computational mesh for any numerical simulations!

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

- (5) Right-click on the new layer ‘Tutorial\_Elevation\_nodes.shp’ in the TOC and go to *Open Attribute Table*.
- (6) Check if an elevation (column *Z*) has been assigned to every single node of the layer.
- (7) 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.
- (8) Check the vertices at the location of the respective points missing an elevation and make sure to use the snapping option.




**Figure 1.13** BASEmesh Elevation Meshing

- (9) If necessary, Repeat the *Elevation meshing* (steps 1 - 4).

In case the problem cannot be solved by redoing the snapping, try to vary the *Relative snapping Tolerance* in the *Elevation meshing* dialog (Figure 1.13). Increase the tolerance and try a value of -3 for example. Check the Tipps and Tricks (Section 1.3) for further explanations on this important parameter.

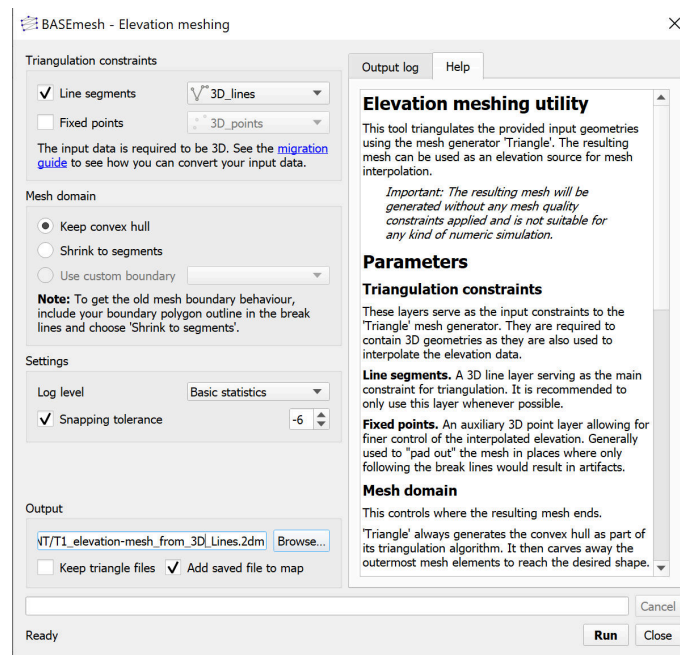
### 1.2.8.2 Creation of the Elevation Mesh using 3D Lines

Now we will create an elevation mesh using the 3d points that were created in Section 1.2.4.

- (1) Go to *Plugins* → *BASEmesh* → *Elevation meshing*, or click the respective button  in the toolbar.
- (2) Select the different fields according to Figure 1.14.
- (3) Follow the same steps as in the previous section.

### 1.2.9 Adaption of the Breaklines for Quality Meshing

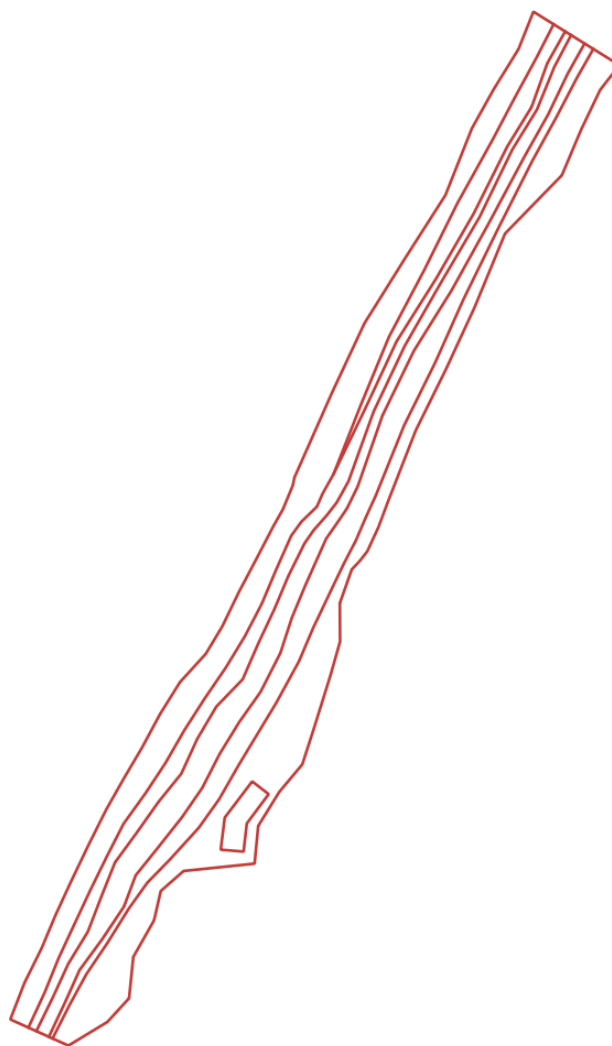
For most tasks, quality meshing requires the same basic breaklines as elevation meshing does. Nevertheless, some content might be added, e.g. building outlines or lines along which



**Figure 1.14** BASEmesh Elevation Meshing with 3D Lines

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' in the TOC and go to *Export* → *Save Feature as...*
- (2) In the next dialog choose an appropriate name and location for the new layer (e.g. 'breaklines\_w\_building.shp'). Be sure to check *Add saved file to map* at the bottom of the dialog.
- (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*.
- (11) Having activated only the 'breaklines\_w\_building' layer in the TOC, the lower part of your model should now look like Figure 1.15. If the building outline is still highlighted, select *View Select Deselect Features from all Layers*.



**Figure 1.15** Breaklines and building

### 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 1.18) afterwards: *maximum area*, *material index* and *holes*. The attributes have to be set individually for each region, which is embraced by breaklines or boundaries and are specified by placing a point into this area. Be careful, a misplaced region marker point can lead to very fine and computationally intensive meshes or wrong definition of Material Indexes respectively.

**maximum area:** The area attribute of the layer constraints the maximum area of the elements created during *Quality meshing*. If left blank, no maximum area constraints will be taken into account for this particular region (see Figure 1.17)

**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 → Create Layer → New Shapefile Layer... .
- (2) Fill in the form as shown in Figure 1.16. 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. *Length*=10 and *Precision*=3).
- (4) Add a second attribute for the *material index* with name ‘MATID’ and type ‘Whole Number’ (e.g. *Length*=3).
- (5) Add a third attribute for *holes* with name ‘hole’ and type ‘Whole Number’ (e.g. *Length*=1\*).
- (6) Optionally add another attribute with name ‘Type’ and type ‘Text data’ (e.g. *Length*=20) to assign a specific description for each region (e.g. ‘River bed’, ...)
- (7) Save the layer with name ‘regions\_points’.
- (8) Right-click on the layer ‘regions\_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 1.17
- (11) Click somewhere inside the particular region. Enter an arbitrary *id* and fill in the attributes like shown in Figure 1.17
- (12) Go to Layer → Toggle Editing and save the changes.

The value of the attributes ‘max\_area’ (left) and ‘MATID’ (right) are used as label. In this example three different region ‘Types’ have been defined: ‘river\_bed’, ‘embankment’ and ‘surrounding’.

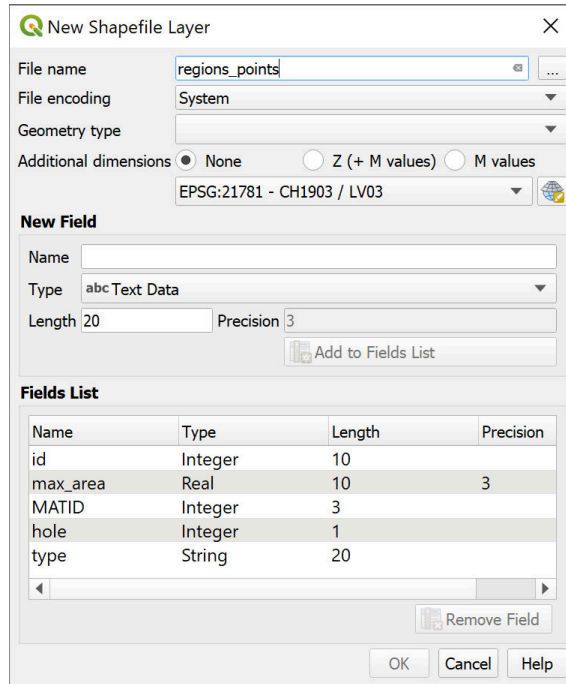


Figure 1.16 New point layer 'region\_points'

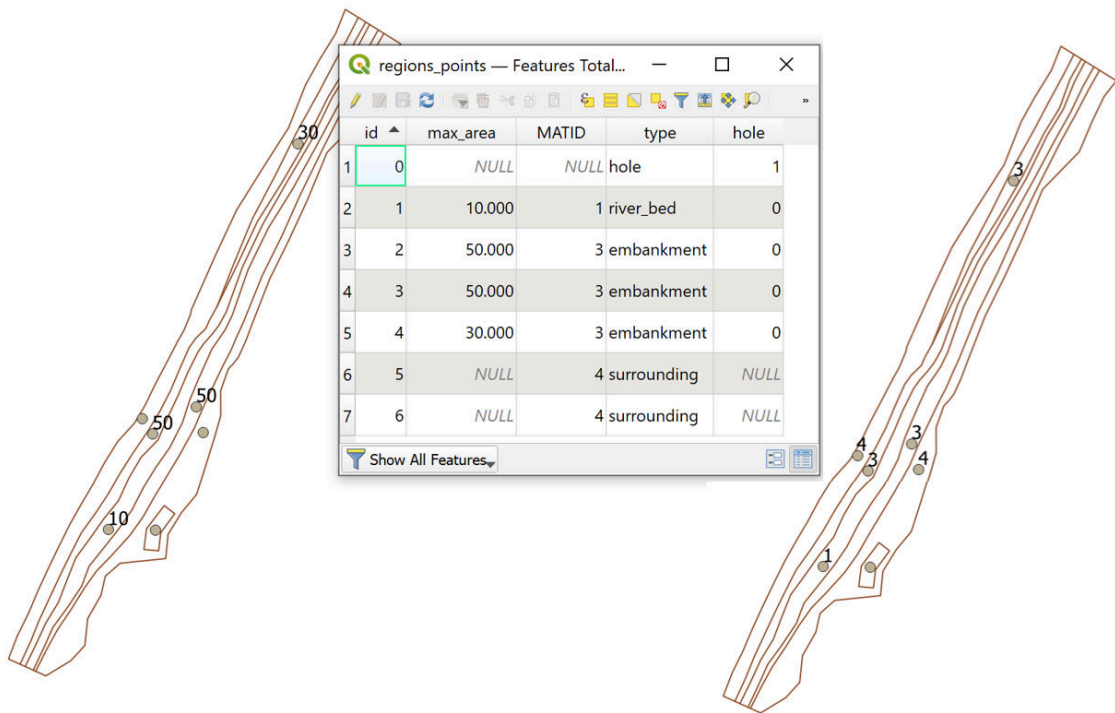



Figure 1.17 Region marker points

### 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 1.18) you can define the input layers. See Section 3.2 “2D grid generation with BASEmesh QGIS plugin” 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.

- (2) Select the different fields according to Figure 1.18
- (3) Check the optional button for *Breaklines*, and *Regions*. Within the Layer ‘region\_points’ check the attributes *maximum area*, *material index* and *holes* like shown in Figure 1.18.


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 *run* to generate the quality mesh.

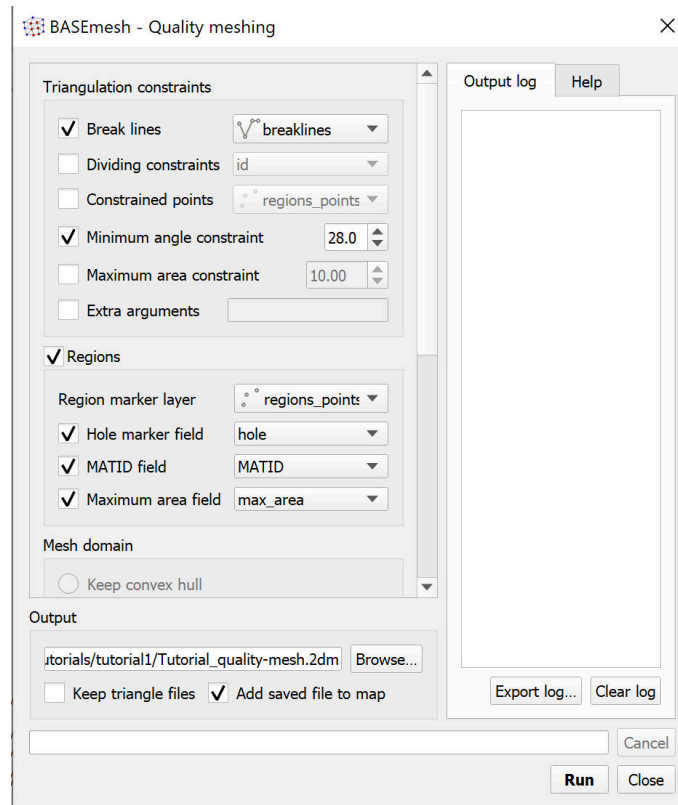
If the mesh contains regions with almost infinitely dense triangulation, again check the snapping of your model boundary and the breaklines used for quality meshing or increase the *Relative snapping tolerance* (see Tips and Tricks Section 1.3).

### 1.2.12 Generation of the Computational Mesh

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.

- (1) Go to *Plugins* → *BASEmesh* → *Interpolation*, or click the respective button  in the toolbar. A new dialog will open (Figure 1.19).
- (2) As input for *Mesh layer to interpolate* choose the quality mesh.
- (3) To make the computational mesh using an elevation mesh, check the radio button *Interpolation via elevation mesh*, and select the desired elevation mesh. If you wanted to make the computational mesh using a DEM, you could check the radio button *Interpolation via DEM (raster)* and select the DEM.





*Figure 1.18* BASEmesh Quality Meshing

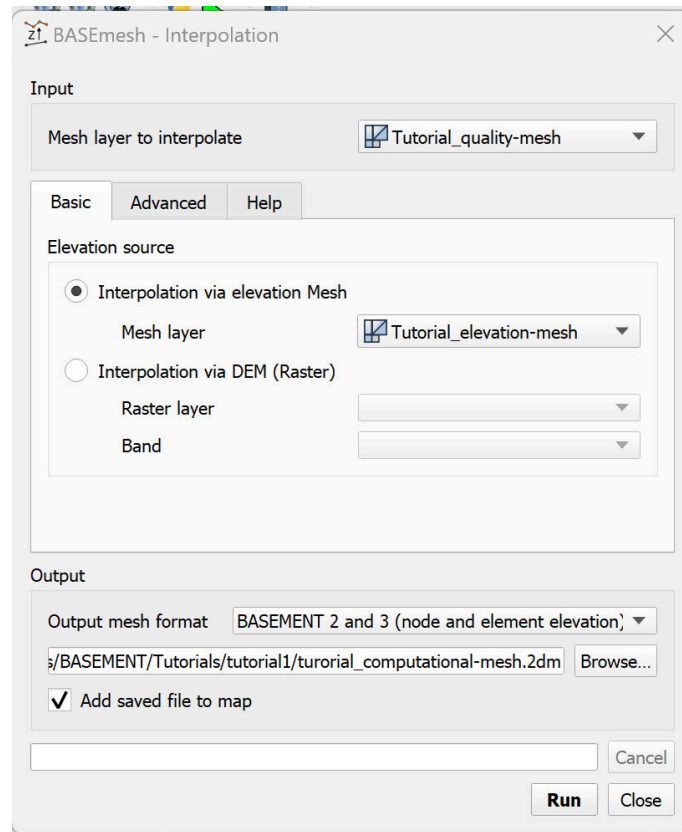
- (4) Click on *Browse* and input a name for the computational mesh.
- (5) Click on *run*. On Windows systems, the interpolation dialog might be marked as Not Responding. On Linux systems, the interpolation dialog might be darkened. This does not necessarily mean that the process stopped or had errors: 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. 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 Viewing the computational mesh

During the interpolation step, elevation data has been added to the mesh. With this additional information we are now able to view both the elevation data and the mesh itself. This can be useful to check for plausibility.

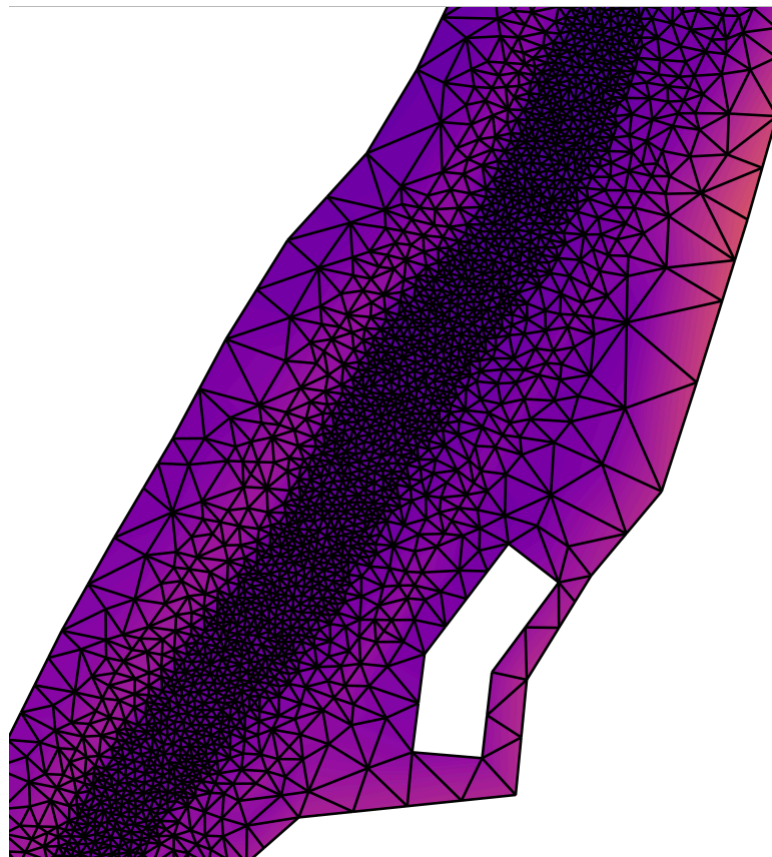
- (1) The elevation data of the computational mesh will be visible. To view the mesh, right click on the computational mesh in the layers toolbar, select *Properties* → *Symbology*



*Figure 1.19* BASEmesh Interpolation

→ select *Native Mesh Rendering* → *OK*. The computational mesh for this tutorial should look as it does in [Figure 1.20](#).

- (2) The computational mesh is now ready for use in a BASEMENT simulation.



*Figure 1.20 Computational mesh*

## 1.3 BASEmesh Tips and Tricks

### 1.3.1 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.

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

#### Python plugins:

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

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

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

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

**Dxf2ShpConverter** Conversion and import of dxf files.

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

### 1.3.2 Tips 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 geotools* → *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 1.21 .

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

- Polylines that are used for the segmentation of boundary polygons must end in vertices of the polygons. Otherwise, Triangle creates an almost infinitely dense triangulation to ensure its angle and area criteria.
- Be aware that using the *vertex editing tool* generates a new vertex point at each double-click. This can easily lead to the generation of two or more points at the same location or very near to each other. In such a case, Triangle creates a very dense triangulation due to its angle and area criteria.
- Prevent situations where breaklines (and corresponding vertices) have very short distances to each other. The resulting mesh will be very fine in these regions. Meshing sometimes requires to manually adapt the input data and to make compromises between mesh quality and accuracy in certain regions.
- Use the *Relative snapping tolerance* provided in the BASEmesh dialogues *Elevation meshing* , *Quality meshing*  and *String definition* . The mesh shown in Figure 1.22 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



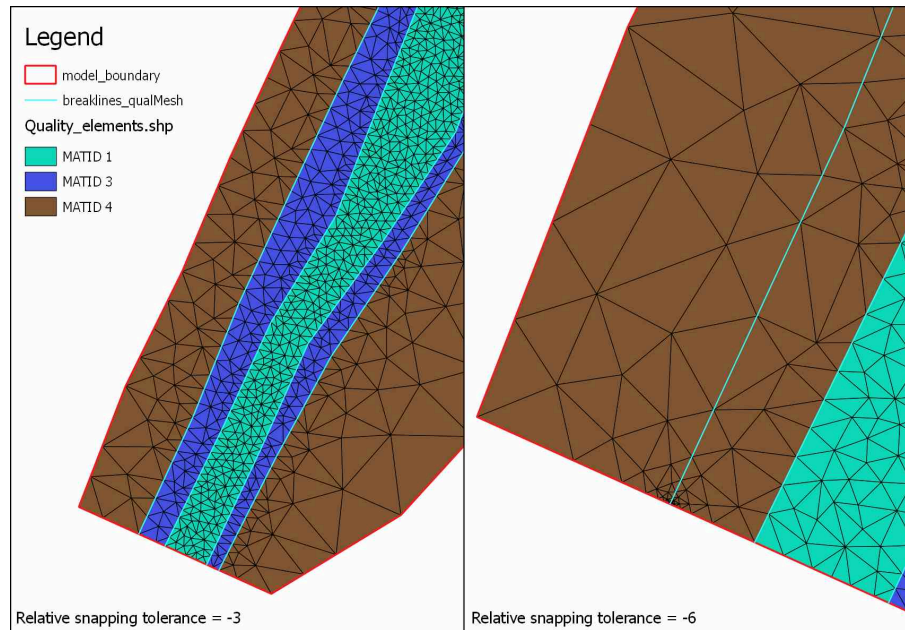
**Figure 1.21** QGIS Convex hull

model boundary since the breakline is not exactly snapped to the vertex of the model boundary. Furthermore MATID 3 was not considered for the embankment as desired. To prevent such behaviour, increase the tolerance to -3 for example. This allows you to generate a proper mesh like shown in Figure 1.22 on the left.

**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).



**Figure 1.22** Quality meshing using the Parameter ‘Relative snapping tolerance’

- 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

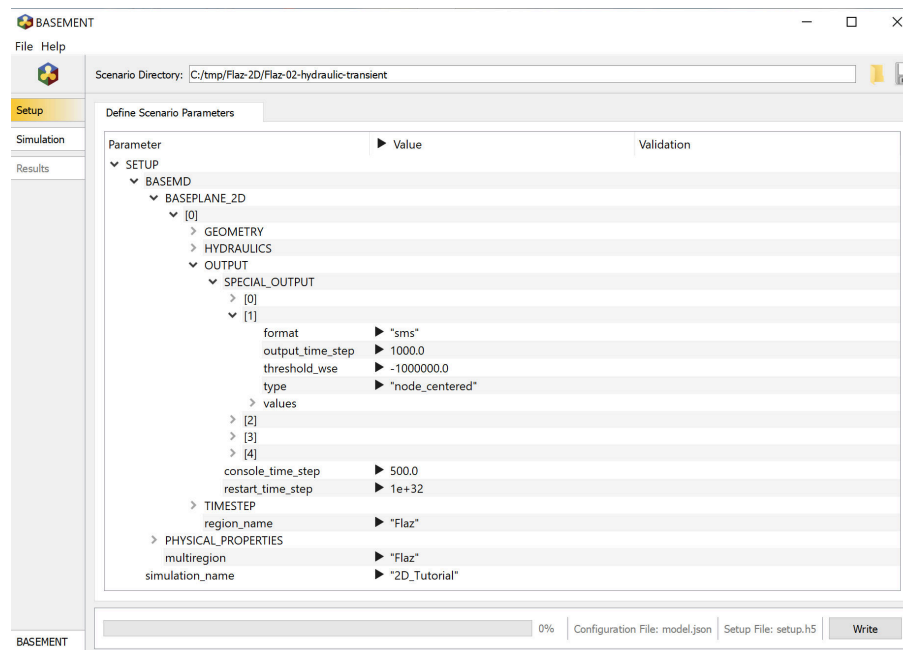
In the following tutorials it is demonstrated how to do the post-processing of 2D results generated by a BASEMENT simulation. For this purpose the free and open source GIS software [QGIS](#) is applied.

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

#### 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 3.16 ‘Hannover’. Along with 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 4.1) are used for visualization. In the *SPECIAL\_OUTPUT* Block of the BASEMENT model file select *format* ‘sms’ to generate result files that can be visualized in QGIS (see Figure 2.1). QGIS 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 choosing type *node\_centered*, these results are interpolated to the nodes by BASEMENT and written to the solution files (\*.sol).



**Figure 2.1** Settings defined in the SPECIAL\_OUTPUT Block of BASEMENT to generate time dependent 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 the functionalities of Crayfish, users can load time varying mesh into QGIS. In more recent editions of QGIS version 3, the Crayfish functionalities come incorporated automatically into the QGIS core. Currently, QGIS supports a number of hydraulic modelling software packages including BASEMENT.

The functionalities of Crayfish allow for loading and rendering of results directly, rather than converting them to other GIS formats before viewing. This allows users to flick quickly through the various output steps in the result files and to create animations.


## 2.2.3 Installation

For recent editions of QGIS version 3, the Crayfish functionalities come incorporated automatically into the QGIS core. Therefore, installation of a separate plugin is not needed.

For older versions of QGIS, 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

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

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

## 2.2.4 Load and visualize data

### 2.2.4.1 Project Settings


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

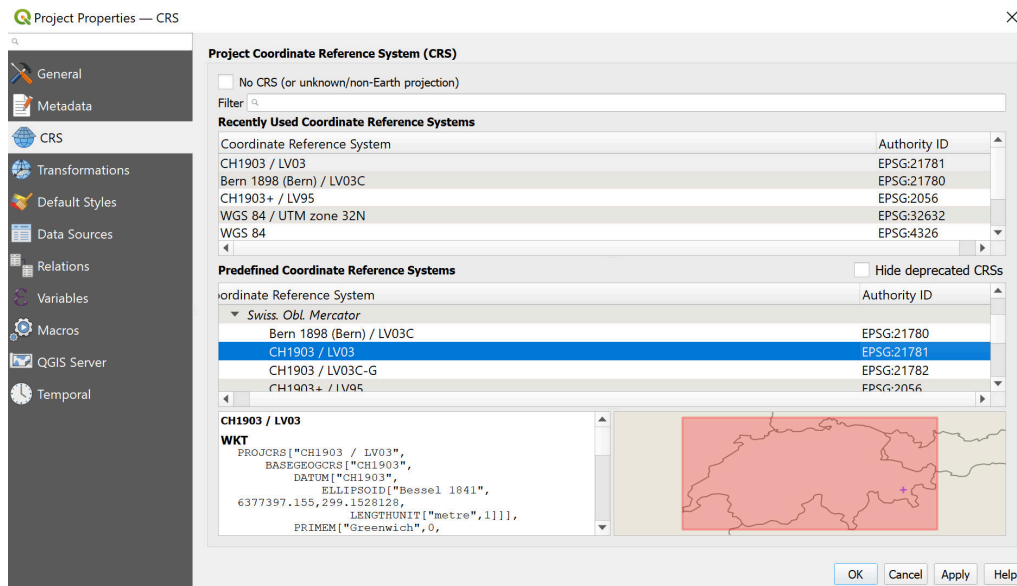
- (1) Go to *Project* → *Properties*.
- (2) Under the tab *CRS* you can see the coordinate reference system settings for the current project.
- (3) Enter EPSG code ‘21781’ into the field *Filter*.
- (4) Select the coordinate reference system ‘CH1903 / LV03’ and click *OK*.
- (5) Again go to *Project* → *Project Properties* → *CRS*.
- (6) Define a *Project title* under the tab *General* → General Settings\*.
- (7) 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

- (1) 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).
- (2) Browse to the folder ‘background\_data’ and load the areal image ‘Flaz\_si25.tif’.



**Figure 2.2** Project Properties, CRS


(3) 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).

(4) 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 right-clicking on a layer → *Style* → select a color from the palette. QGIS displays the loaded layers according to their order in the Layers control panel.

### Adding Results layers

To view 2D results, 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.

(1) Go to *Layer* → *Add Layer* → *Add Mesh Layer* or use the *Add Raster Layer* icon  in the Manage Layers toolbar).

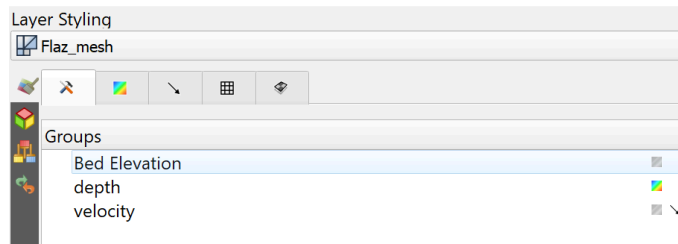
(2) Browse to the folder ‘03\_2D\_results’ and load the mesh ‘Flaz\_mesh.2dm’.

Now load the result files of the 2D simulation:

(1) Double-click on the mesh layer. Go to *Source* → *Available Datasets Layer* → *Assign extra dataset to mesh*

(2) Browse to the folder ‘03\_2D\_results’ 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



**Figure 2.3** Layer Styling Panel Groups

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.

### 2.2.4.3 Display properties

The style of the displayed quantities can be modified in the Layer Styling panel. On the 1st tab of the Layer Styling Panel, you can select which layer associated with your 2dm that you would like to edit. To select a layer, click on the icon to the right of the layer name (see Figure 2.3).

#### Contour styles

Styling of contours can be done in the 2nd tab of the Layer Styling Panel (see Figure 2.4). In this tab 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

To activate the vectors, *Go to the 1st tab of the Layer Styling Panel → Click on the arrow to the right of ‘velocity’*. Vector styling of the velocity vectors can then be done in the 3rd tab of the Layer Styling Panel (Figure 2.5).

In this tab you can:

- Set the color, length, and width of the vectors
- Filter by min/max
- Set vector head size
- set an arbitrary grid, where vectors will be displayed by activating *Display on User Grid*. This is particularly useful to view a smoother vector interpolation.

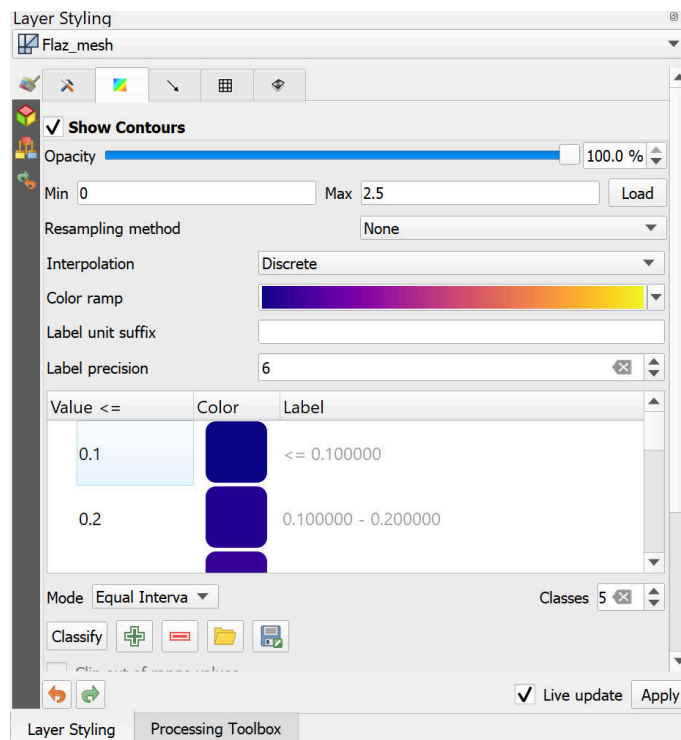


Figure 2.4 Layer Styling Panel

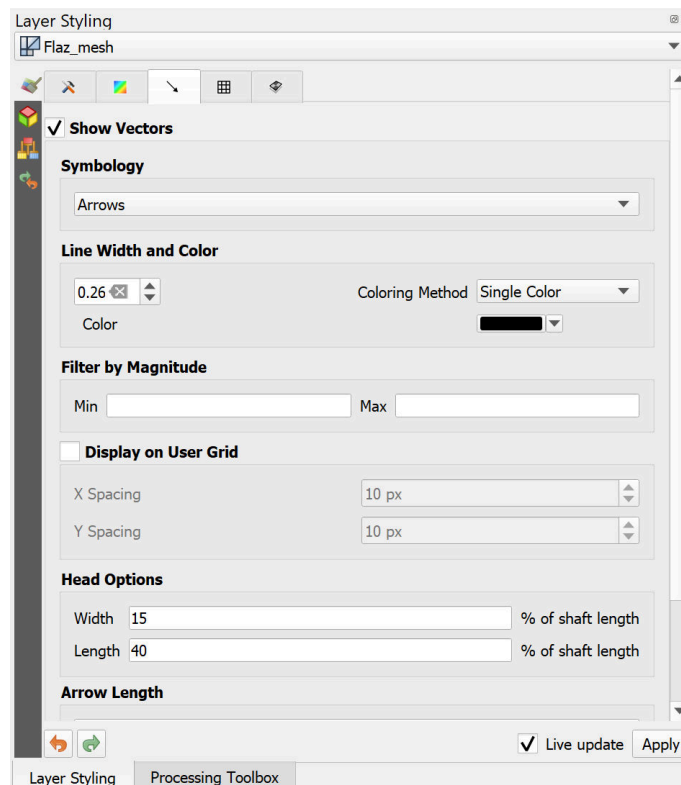
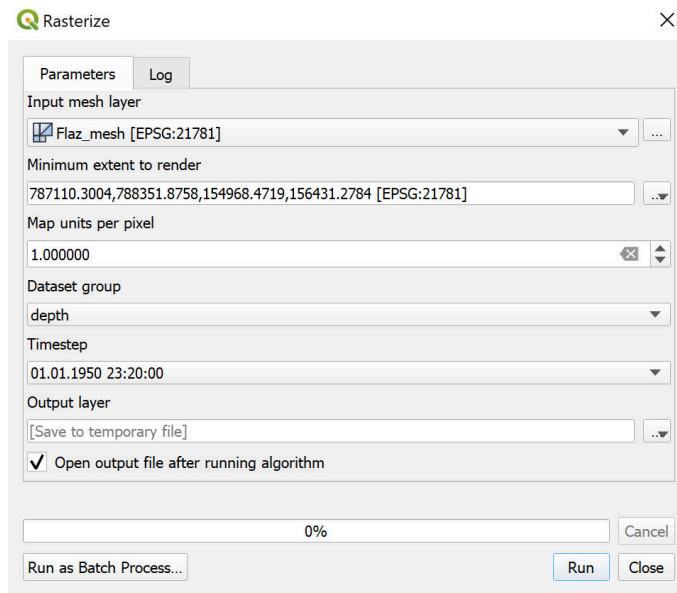


Figure 2.5 Vector Options window of the Layer Styling panel



*Figure 2.6 Rasterize*

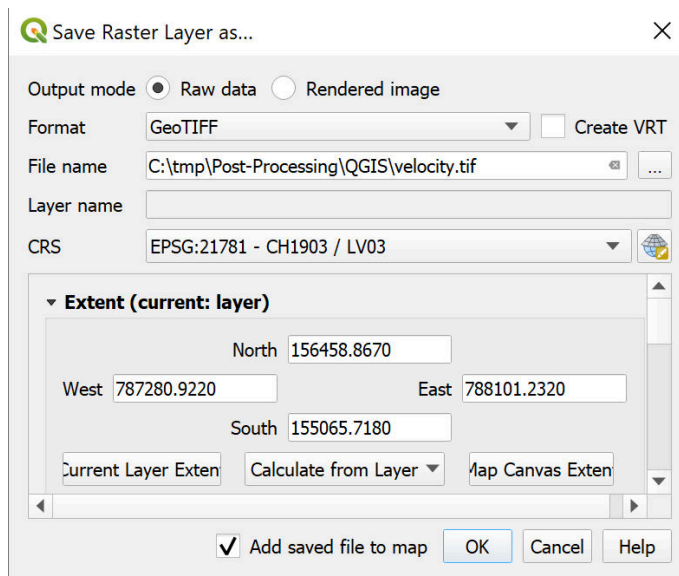
#### 2.2.4.4 Export results to raster

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

- (1) In the *Processing Toolbox*, search for ‘Rasterize.’ Select the Crayfish Rasterize tool. (Figure 2.6)
  - (2) Set the “Minimum extent to render” based on *Calculate from layer* → *Flaz\_model\_boundary* (make sure QGIS window is zoomed to show entire model domain).
  - (3) Select the mesh layer that contains the results data.
  - (4) Select the desired data, timestep, and grid resolution.
  - (5) Click OK .
- 
- (1) *Right click on the newly created layer* → *Click Export* → *Save as...*
  - (2) Save the raster layer as a GeoTIFF (Figure 2.7)

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 calculate the product depth \* velocity using the QGIS *Raster Calculator* for example. This might be of particular interest when generating flood hazard maps for example.

- (1) Go to *Raster* → *Raster Calculator*.



**Figure 2.7** Save as GeoTIFF

- (2) Specify the *Raster calculator expression* as shown in Figure 2.8 by double-clicking on the *Raster bands* depth and velocity in combination with the required *Operators*.
- (3) Define the *Output layer* and press *OK*.

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

#### 2.2.4.5 Profile tool

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

- Define a *STRINGDEF* along the desired location within your computational grid and chose a *SPECIAL\_OUTPUT* of type *stringdef\_history* with the required *stringdef\_values*.
- Use the QGIS plugin ‘Profile Tool’ and analyse the results of a Crayfish layer.

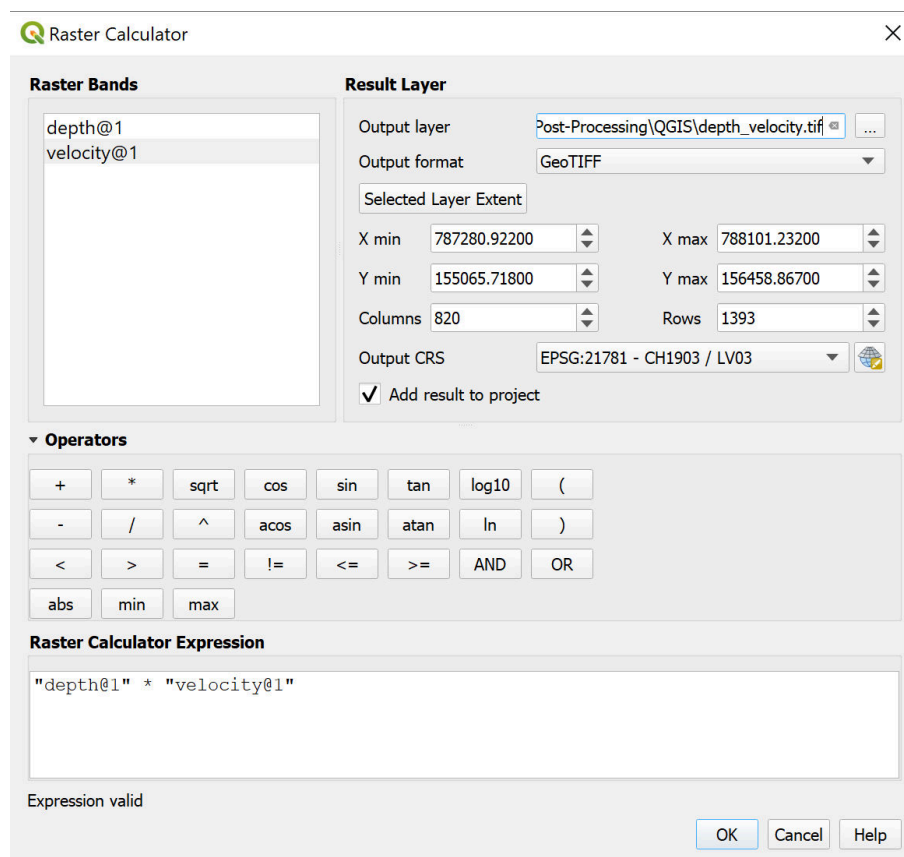
The advantage of option two is a fast and easy evaluation at any location of the modelled domain, without having defined any *stringdef\_history* outputs beforehand.

#### Installation

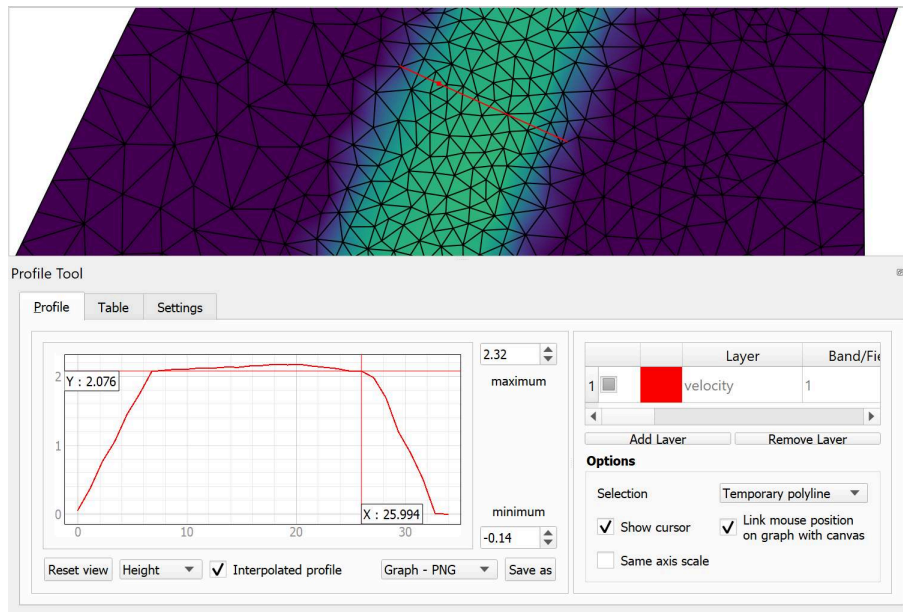
The Profile Tool plugin can be installed via the official QGIS repository:

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






*Figure 2.8 Applying QGIS Raster Calculator*



**Figure 2.9** Velocity profile along a temporary polyline generated with the ‘Profile Tool’ 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:

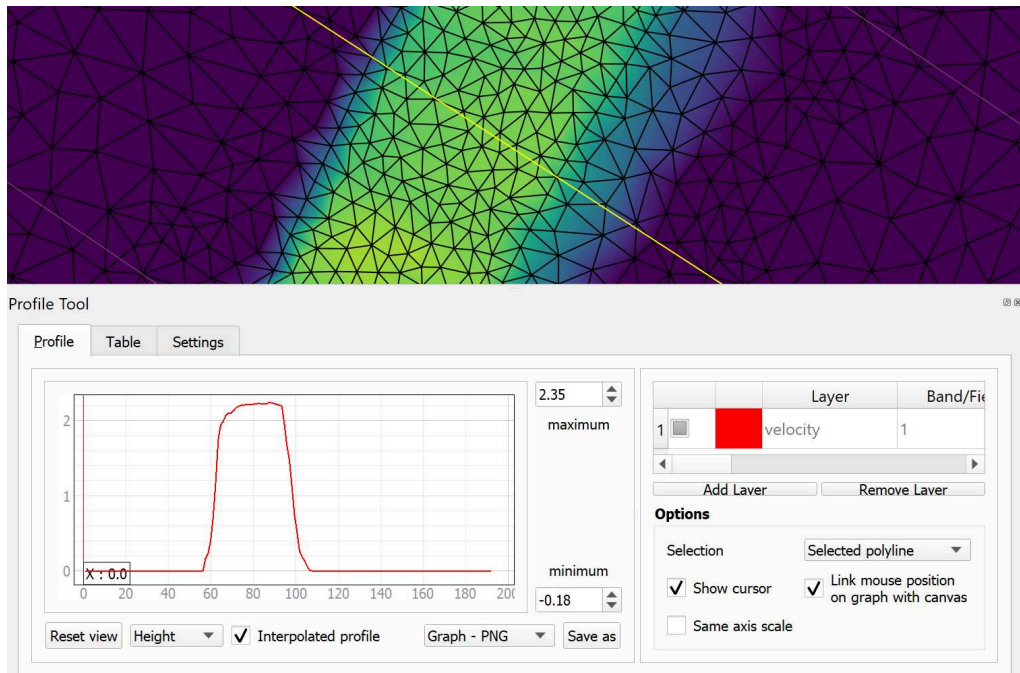
With this option, you can draw an arbitrary polyline at the location of your choice. Along this line the data of the selected *Quantity* will be extracted.

- (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 2.9).
- (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 2.9 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.



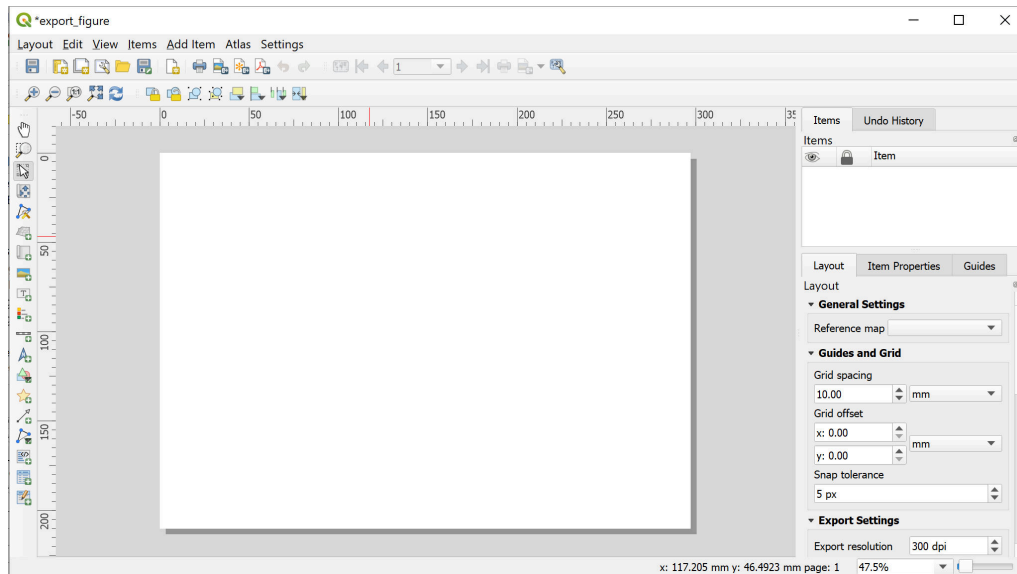
**Figure 2.10** Velocity profile along the red line drawn at the location of cross section ‘CS10’ generated with the ‘Profile Tool’ plugin

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

Again the profile of the chosen quantity should be displayed in the *Profile Tool* window (see Figure 2.10). Note that the profile also contains entries with value 0 since the extent of the cross section is wider than the wetted main channel.

## 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 Layout’ which allows you to take your GIS layers and package them to create professional maps including legend, scale bar or text boxes. It is possible to display and export BASEMENT results layers using the QGIS ‘Print Layout’. This tutorial can’t cover the whole functionality of the ‘Print Layout’, 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.



**Figure 2.11** QGIS Print Layout with some of the basic control features


### 2.2.5.1 Customized layout (QGIS print Layout)

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 Layout*.


- (1) Go to *Project* → *New Print Layout* or press the New Layout icon  in the toolbar.
- (2) You will be prompted to enter a title for the layout e.g. 'export\_figure'. If left blank, a default name will be applied.

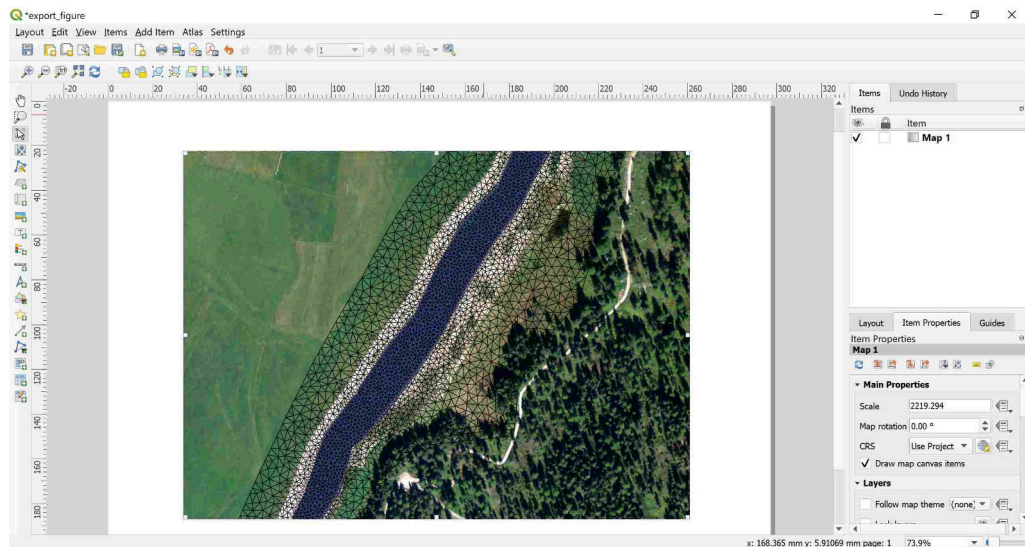
A new Print Layout opens up (see Figure 2.11). The main control buttons to add and modify maps can be found in the toolbar on the left site. You can also specify the layout and export resolution in the tab *Layout* on the right site.

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

- (1) Press the *Add 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 'depth' results shall be displayed together with the background areal image, breaklines and mesh elements (see Figure 2.12). The rendered map might not cover the desired extent of interest. To alter the displayed map extent and/or zoom:

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





**Figure 2.12** QGIS Print Layout with displayed results layer ‘depth’. Orthophoto: © 2016 swisstopo (JD100041)

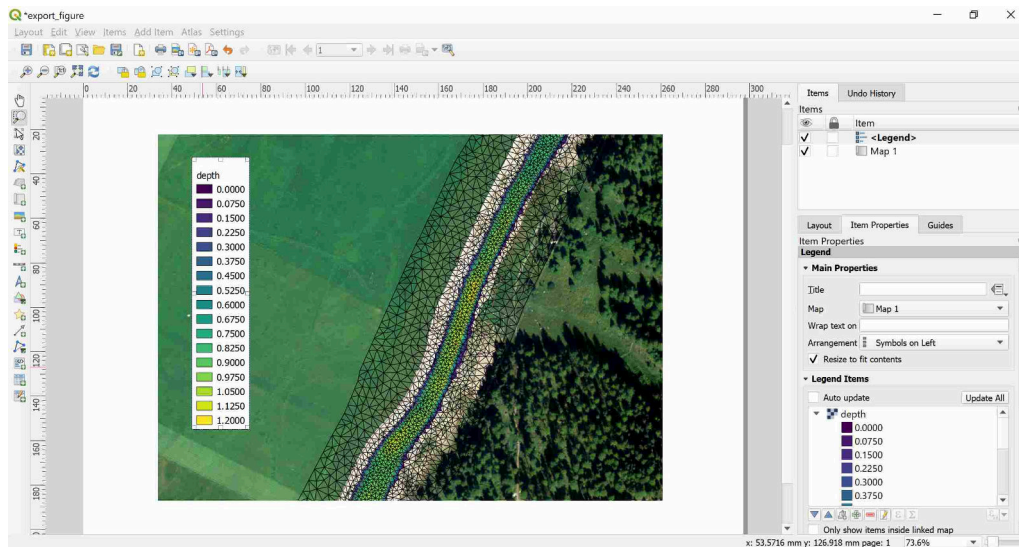
The map is now listed as ‘Map 1’ in the *Item* dialogue box. Further map properties can be defined via the tab *Item properties*. Additionally, we want to add a legend to our map:

- (1) Press *Add legend* , hold the left mouse button and drag a rectangle where you want to insert the legend.
- (2) The legend will appear as new *Item* ‘Legend’ in the *Items* dialogue box (see Figure 2.13). Go to *Item properties* in order to modify the legend entries.
- (3) 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.


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



- (1) Click on ‘depth’ in the *Legend items* dialogue box and press *edit* .
- (2) Type ‘depth [m]’ and press *OK*
- (3) 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.
- (4) To insert a scale bar press the *Add new scalebar* button  and click to the map at the desired position.
- (5) 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.





**Figure 2.13** QGIS print layout with displayed results layer ‘depth’ after inserting the legend. Orthophoto: © 2016 swisstopo (JD100041)

- (6) To insert a north arrow, press the *Add image* button  and click to the desired location.
- (7) A new *Item* ‘picture’ is listed in the Items dialogue box.
- (8) A new *Item* ‘Picture’ appears in the Items dialogue box. In the *Item properties* tab 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 Layout should now look like shown in Figure 2.14. The map is now ready to be exported or printed. You can add additional features like text or images to the map at your convenience. Just use the *Add image*  or *Add label* . You can save your map properties as template for further QGIS projects.

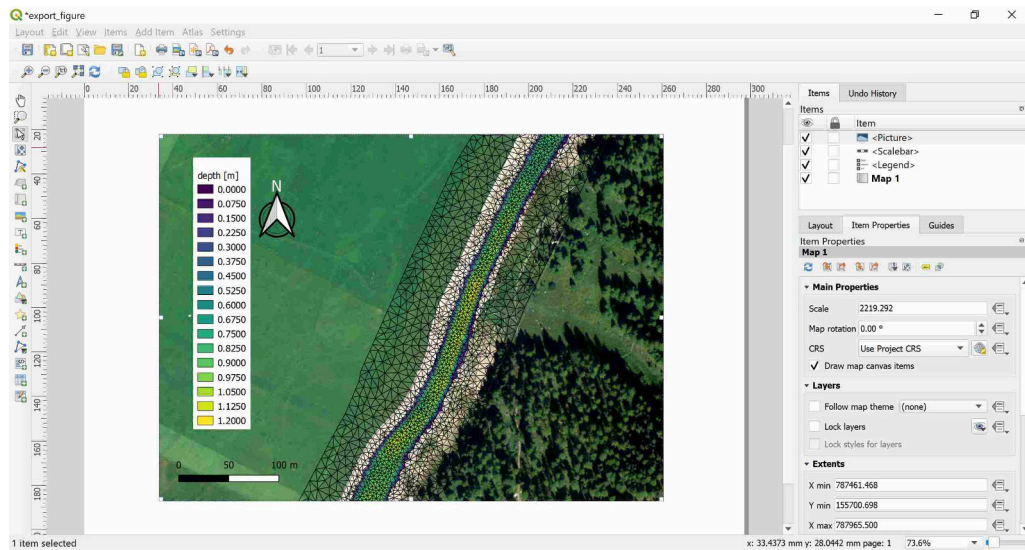
(image.png) Go to *Layout* → *Save as Template* or use the button  in the main toolbar.

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

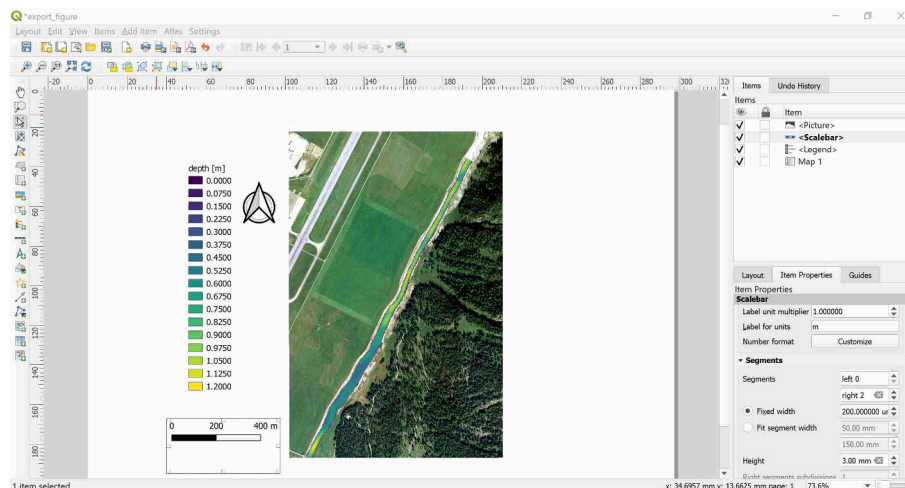
- (1) Go to *Project* → *Layout manager* where you can find a table of all print layouts defined within the main QGIS Project.

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

- (1) Go to *Laout* → *Export as Image* to export the map as raster image e.g. png, jpeg, tif, . . .
- (2) Alternatively chose export as PDF or export as SVG to save the map in vector format.




**Figure 2.14** QGIS print Layout with displayed results layer ‘depth’ after inserting legend, scalebar and north arrow. Orthophoto: © 2016 swisstopo (JD100041)




**Figure 2.15** QGIS print layout with displayed results layer ‘depth’ zoom to the full extent of the model for export as animation. Orthophoto: © 2016 swisstopo (JD100041)

### 2.2.5.2 Export animation


If your simulation results are time dependent like those provided in this tutorial, you might not just want to export selected time steps as maps, but rather create an animation over the whole run time of the simulation. Before exporting your time-dependent result, you have to define the layout of your animation in the QGIS Print Layout.

We can continue from the Print Layout of the previous section (Section 2.2.5.1). Using the *Move item content* , zoom the map out to see the entire model domain. You may want to change the size of the scale bar, which can be done in the *Item properties* tab. You may also set the paper size and orientation: *Right click on the map* → select “Page Properties”.

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

- (1) Press *Add label*  and add a text field below the Legend.
- (2) Go to *Item properties* and type ‘run time:’ into the *Main properties* field.
- (3) Define *Font size*, *Frame*, and *Background* as desired.
- (4) Again press *Add new label* and insert a second text field to the right of the first.
- (5) Don’t change the default expression ‘Lorem ipsum’ in the Main properties field.
- (6) Go down to *Item ID* and type ‘time’ into the *ID* field

Now the text field is linked to the simulation runtime, which will be displayed during the animation.

- (1) Go to *Layout* → *Save as Template* or use the button  in the main toolbar.

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

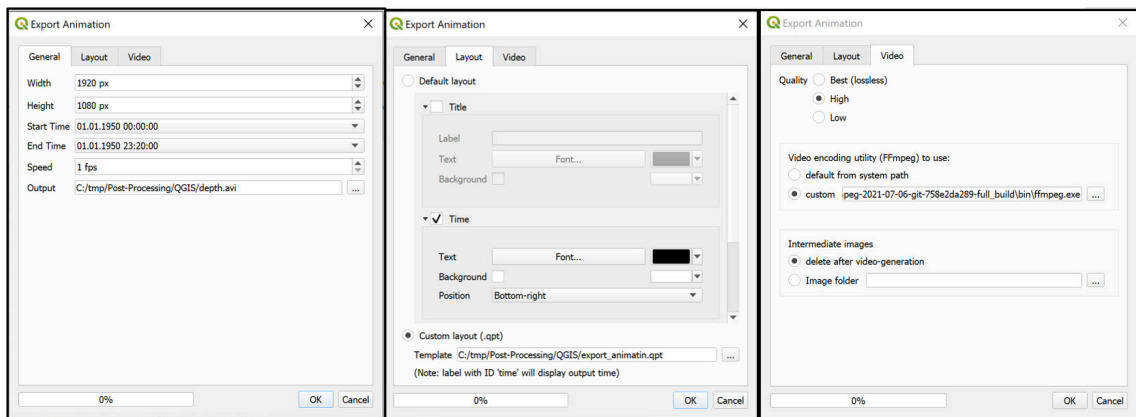
The final export of the animation is done from the QGIS main canvas:

- (1) Right click the ‘Flaz\_mesh’ layer, select *Export animation...* and a new window will appear (see Figure 2.16).
- (2) In the tab *General* you have to specify the start and end times of the animation, the frame rate in frames per second (fps), and the output directory.
- (3) In the tab *Layout* chose *Custom layout (.qpt)* and browse to the location where you saved the Print Layout template (step 56).
- (4) 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.
- (5) *Under Video encoding utility (FFmpeg) to use:* chose custom.
- (6) Press *OK* to begin generation of the animation.

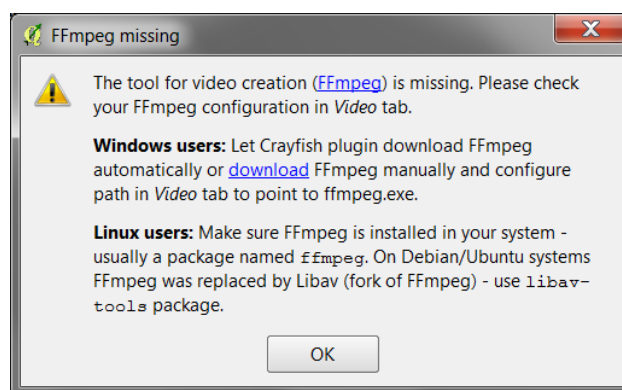
A window may pop up saying that “The tool for video creation (FFmpeg) is missing ...” (Figure 2.17). Follow the instructions depending on your system (Windows or Linux).

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

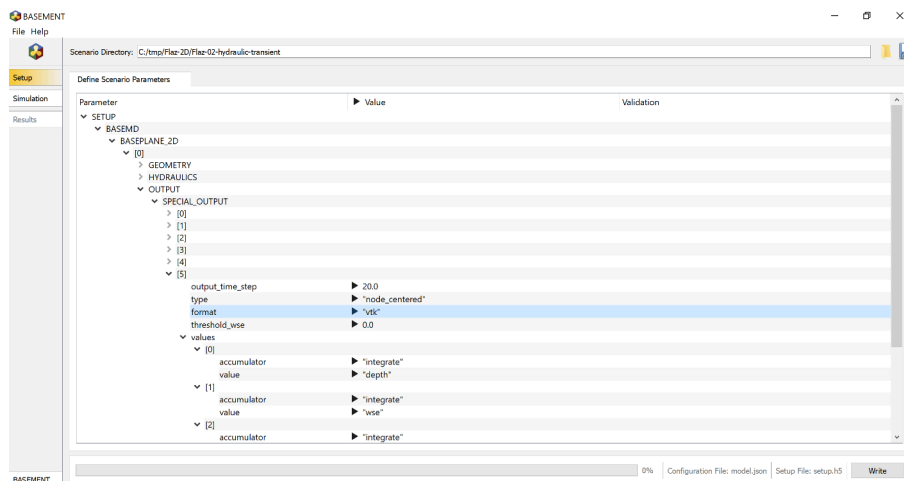




*Figure 2.16 QGIS Export Animation window*



*Figure 2.17 QGIS Export Animation FFmpeg*



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

## 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 with ParaView version 4.3.

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).

### 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.

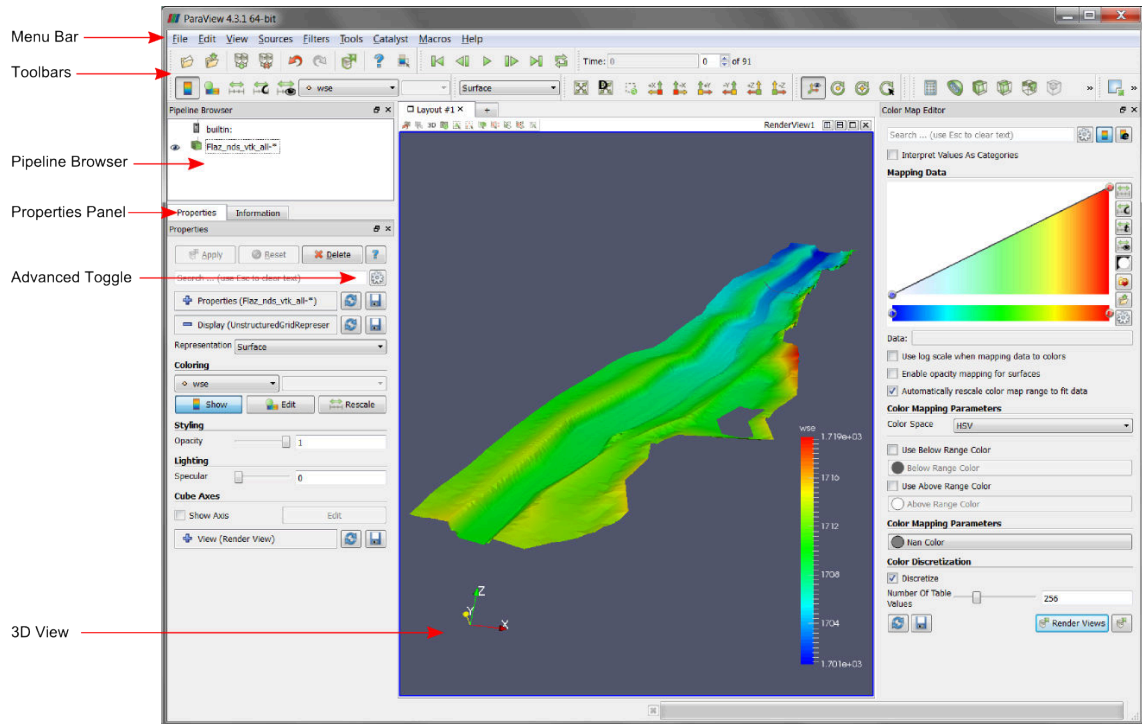


Figure 2.19 ParaView GUI

### 2.3.4 User Interface

The layout shown in Figure 2.19 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.

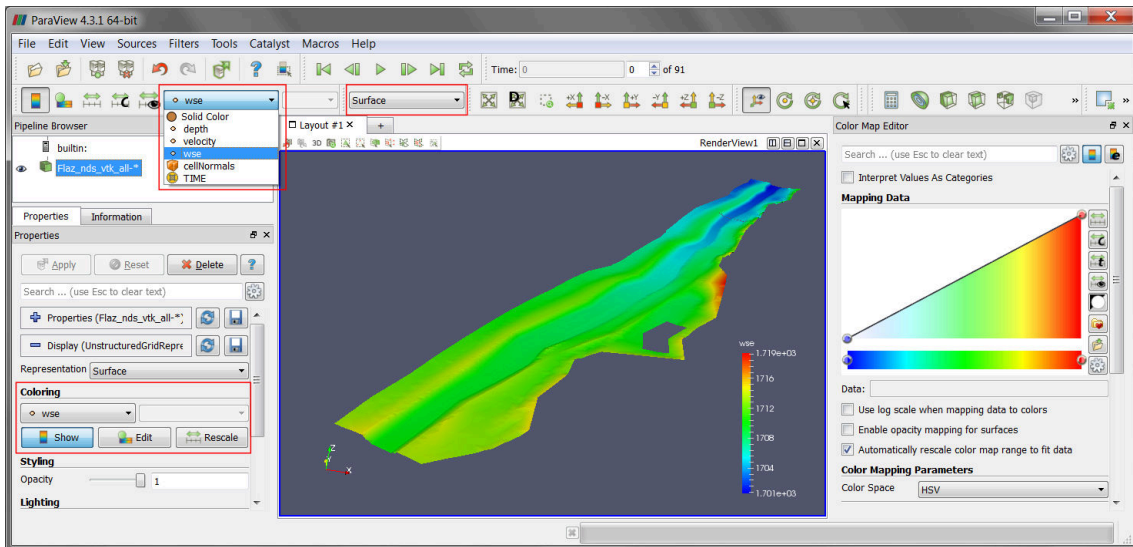



Figure 2.20 ParaView GUI with 3D view of the domain Flaz

### 2.3.5 Import Data

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

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

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

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


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


### 2.3.6 ParaView Filters

#### 2.3.6.1 Overview


In ParaView a large number of so called *Filters* can be applied. *Filters* are functional units that process the data in order to generate, extract or derive features from the data. Here


are some of the most common, which are available by clicking the respective icon in the filters toolbar.


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


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

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

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

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

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

**Glyph**  Places a glyph, a simple shape, on each point of the mesh. The glyphs may be oriented by a vector 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.

### 2.3.6.2 Applying Filters to data

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

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

#### Filter: Warp by Scalar

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

- (4) Select the data set ‘Flaz\_nds\_vtk\_all-\*’ from the *Pipeline Browser*.
- (5) In the menu bar go to *Filters* → *Alphabetical* and select *Warp By Scalar*.
- (6) In the *Properties* panel chose ‘depth’ as *Scalars* and press *Apply* (see Figure 2.22)

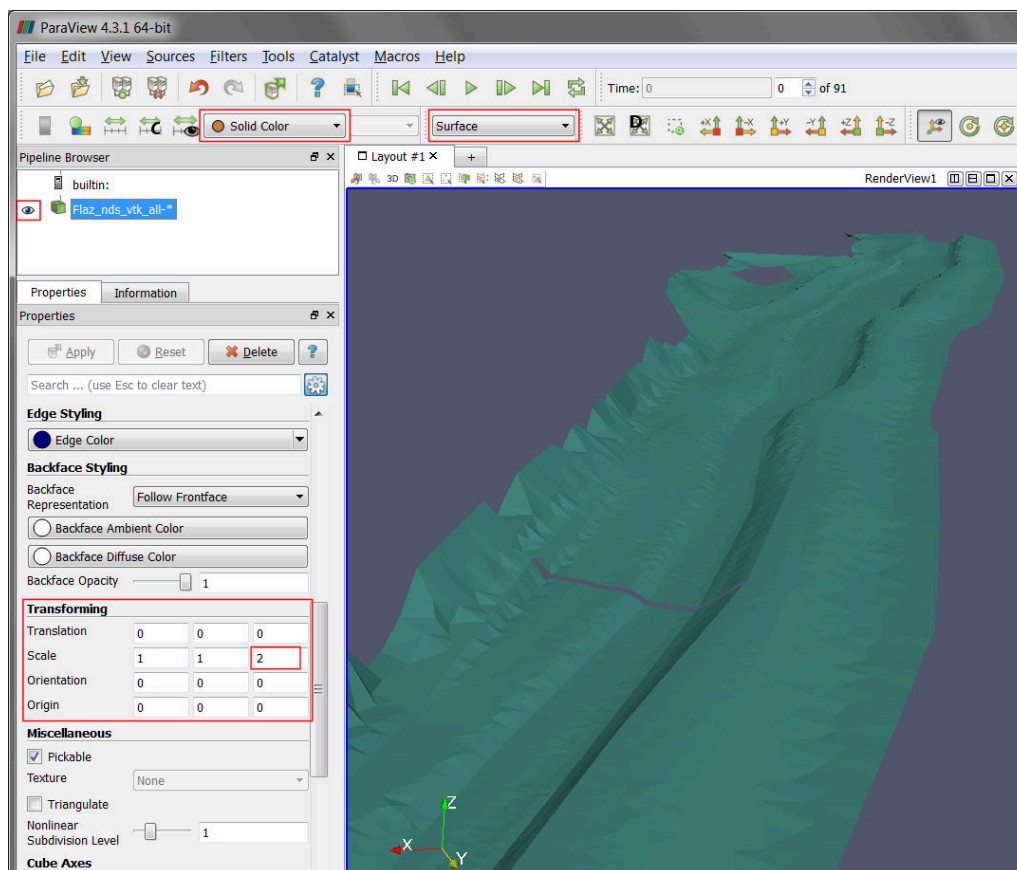


Figure 2.21 Vertical exaggeration of solid mesh

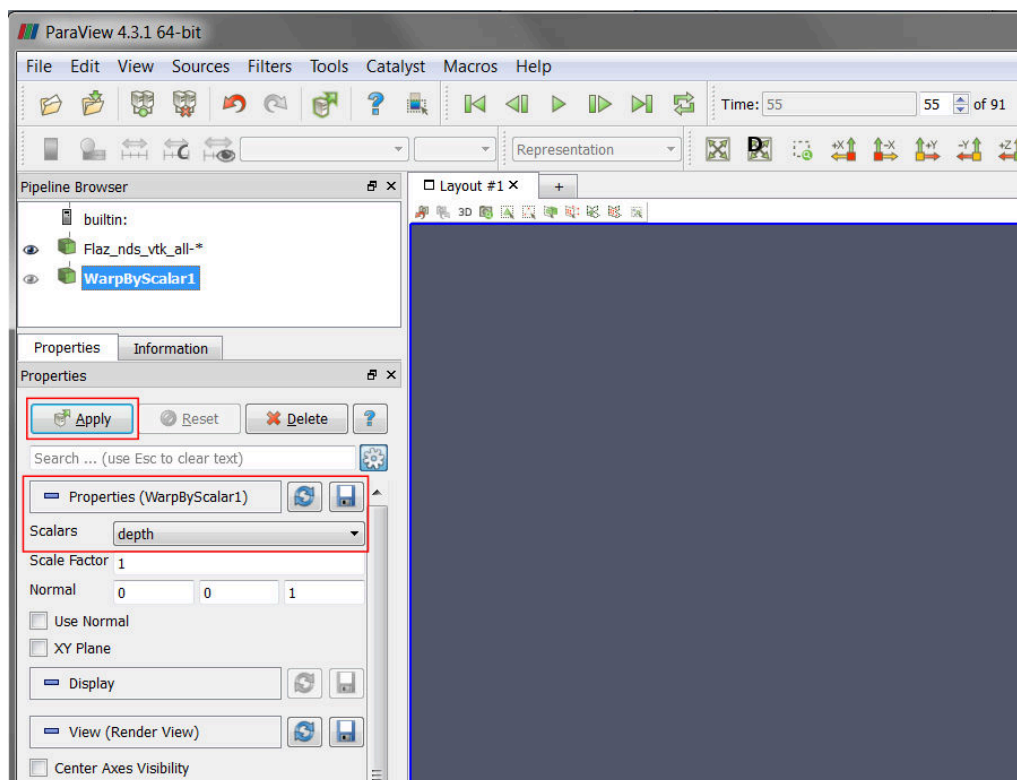


Figure 2.22 Filter, Warp by Scalar



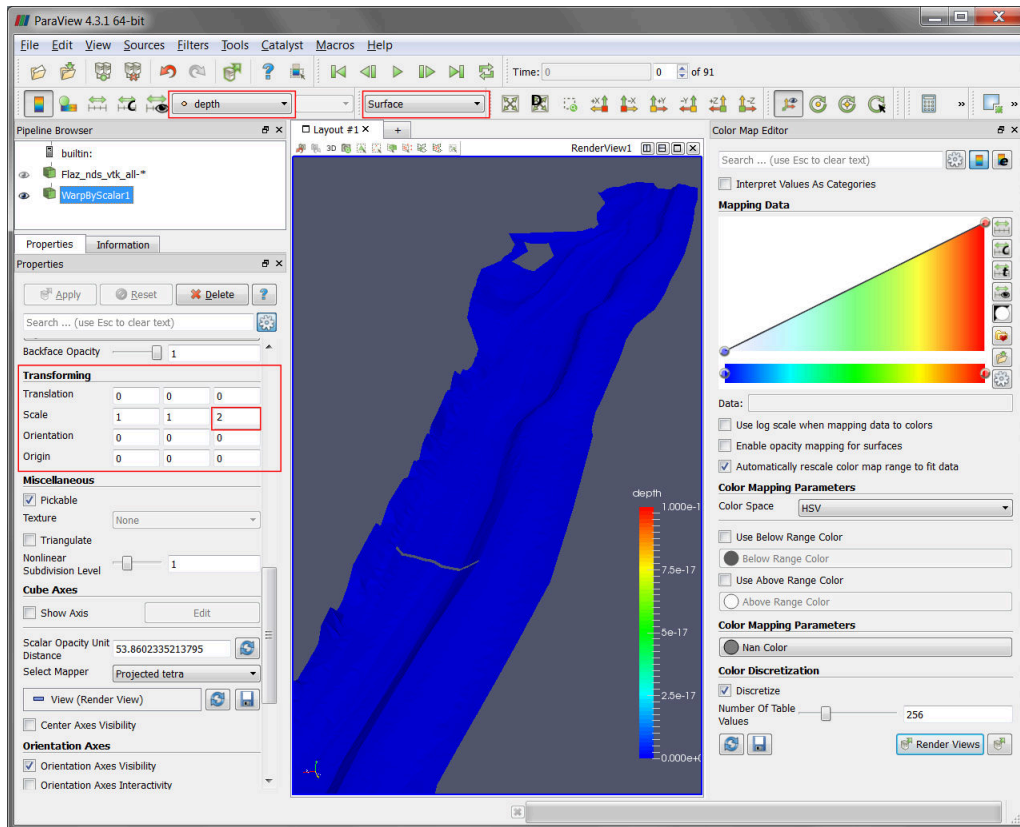



Figure 2.23 Warp by Scalar, properties

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

### Filter: Threshold

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

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

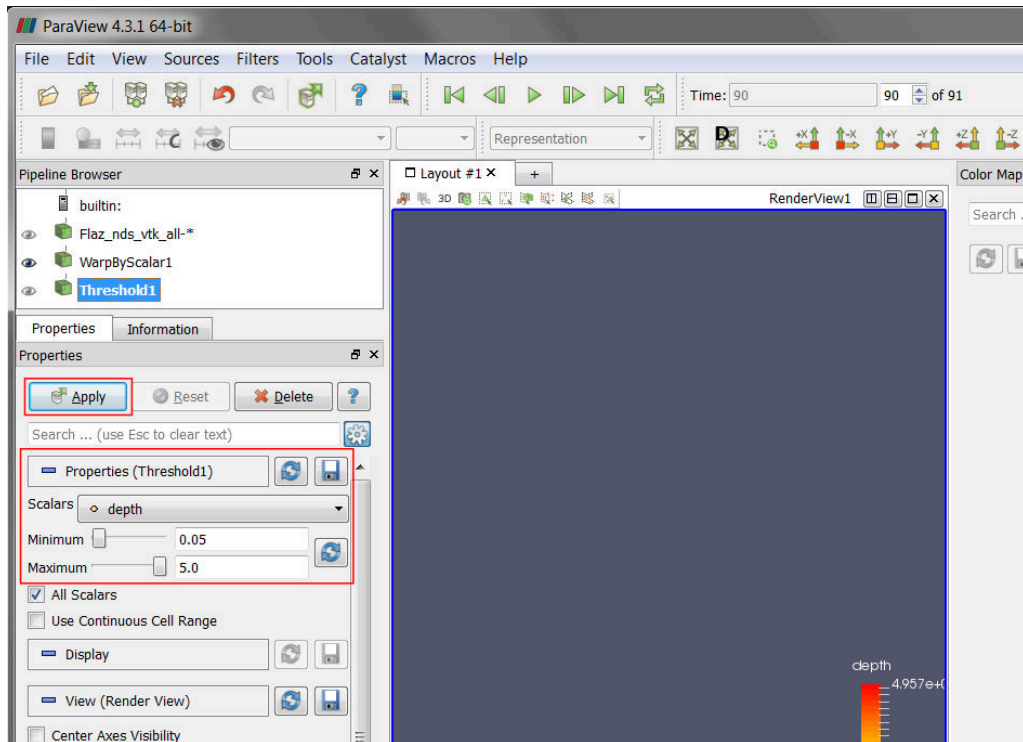


Figure 2.24 Filter, Threshold

### Annotate Time Filter

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

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

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

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

If everything went well, your 3D view should no 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



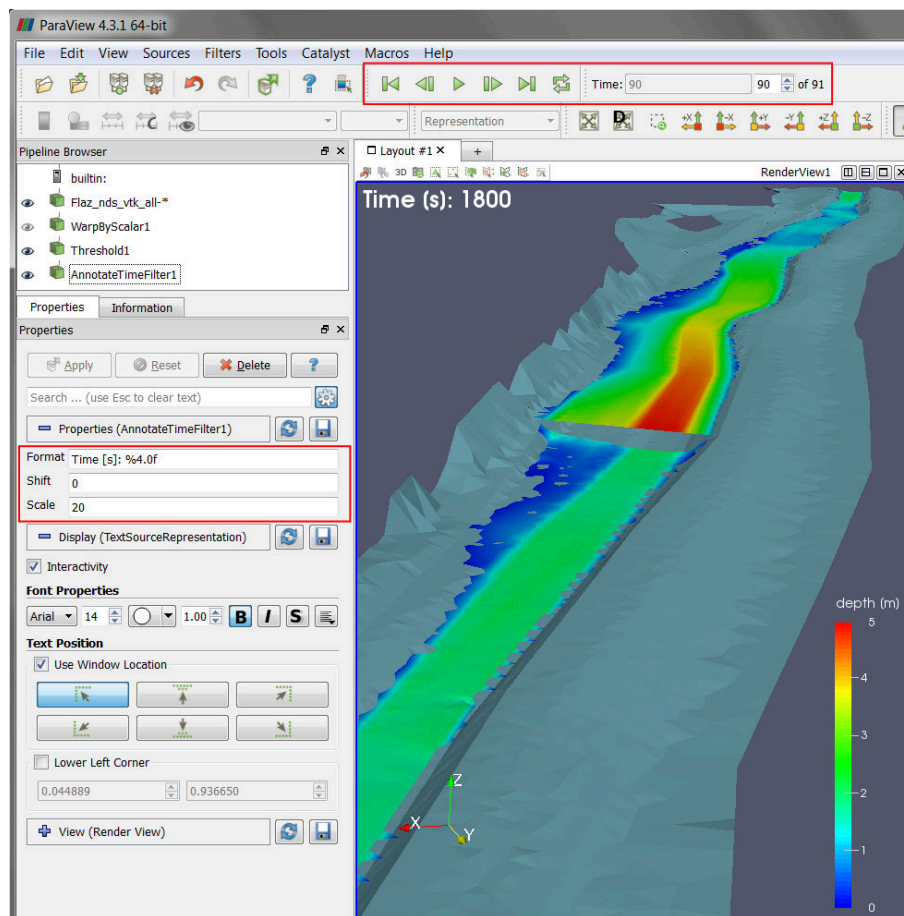
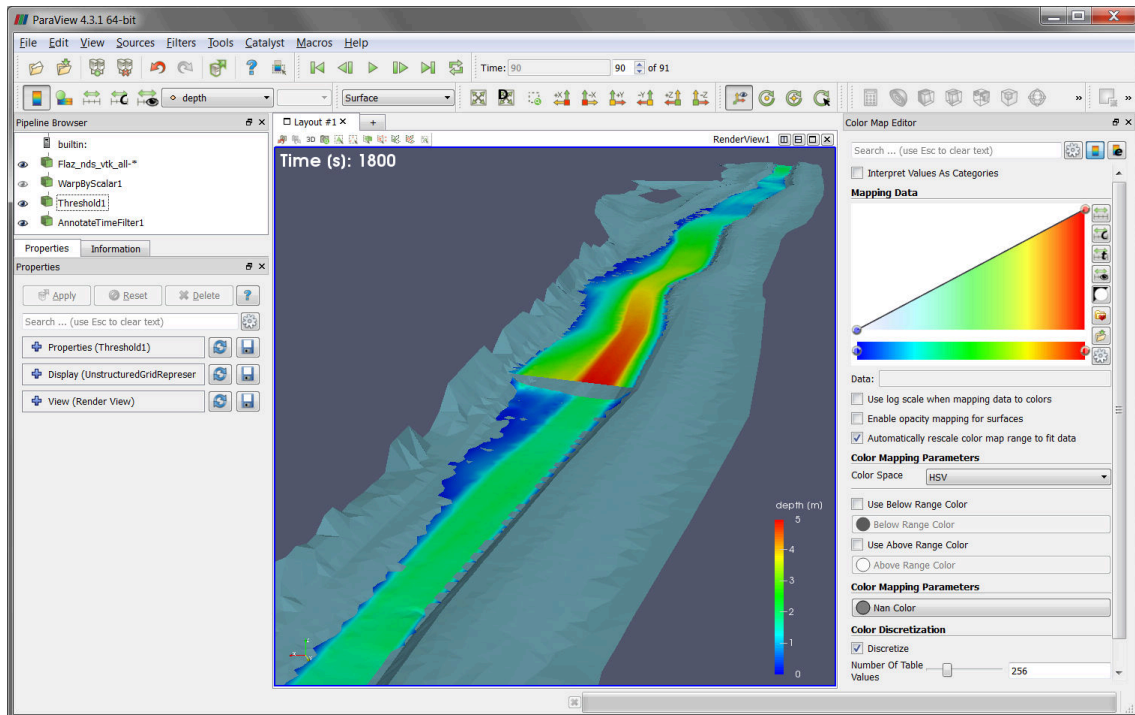


Figure 2.25 Annotation Time Filter



**Figure 2.26** Water depth at the steady state of the simulation

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)

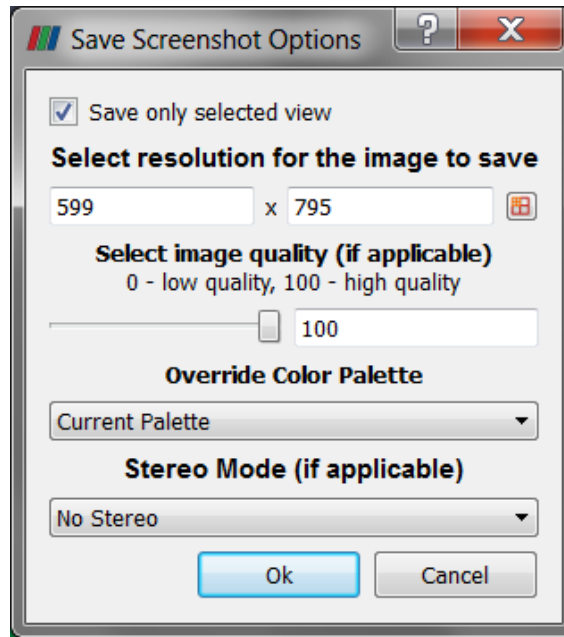
- (1) Go to *File* → *Save Screenshot* and a new window with several controls will appear. (see Figure 2.27)

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.

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

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



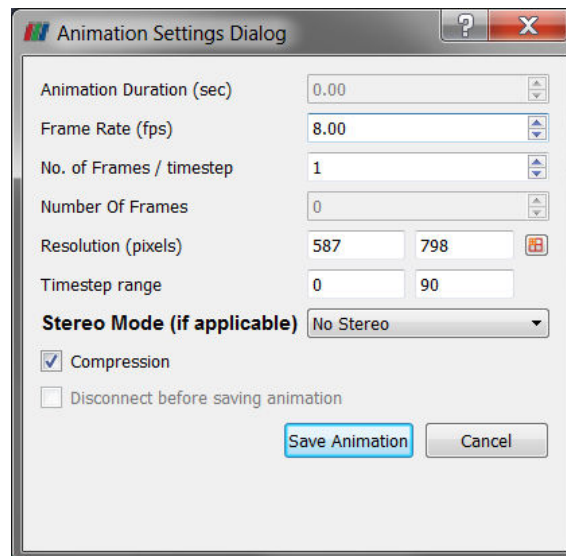
*Figure 2.27 Export Screenshot*

### 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*

---

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

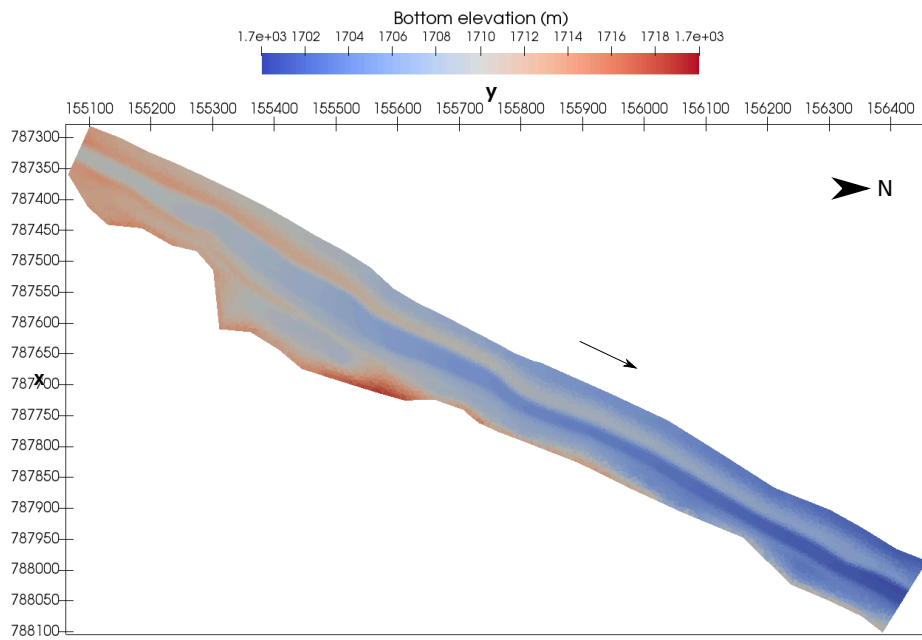
## 3.1 Hydrodynamics and sediment transport at the river Flaz with BASEHPC (2D)

### 3.1.1 Introduction

The river Flaz is located in the canton Graubünden in Switzerland. A reach of 1.5 km long is taken as example for this tutorial. The reach contains a widening section created to protect the village of Samadan from flood event. The aim of this tutorial is to show the setup of the three configuration files for the numerical simulation with BASEMENT. First, a hydraulic simulation is performed to obtain a calibrated model at steady state. Then a morphological simulation is performed adding the morphological part to the result of the hydraulic simulation.

### 3.1.2 Computational Mesh

The computational mesh of the Flaz is imported from the tutorial of BASEMENT version 2.8. The 2dm file has been modified as explain in the pre-processing tutorial for small meshes in the User manual. There are two stringdefs for the inflow and outflow boundaries. The mesh has 14'457 cells, 7'446 vertices and the interpolation method “weighted” is used to convert the mesh from version 2.8 to a 3 compatible computational mesh. Figure 3.1 shows the bottom elevation of the river Flaz used in this tutorial.



**Figure 3.1** Planar view of the initial bottom elevation of the river Flaz

### 3.1.3 Hydraulics

The configuration files (model.json, simulation.json and results.json) can be created and modified with the graphical user interface (GUI) or any text editor. The configuration files are saved in one folder and the simulation will automatically generate a new folder called “run” which contains the output.

#### 3.1.3.1 Setup the Configuration File model.json

The configuration file model.json for a hydrodynamic simulation has the following structure:

```
{
  "SETUP":{
    "simulation_name":"Flaz_steady_state",
    "BASEHPC": {
      "PHYSICAL_PROPERTIES": {...},
      "BASEPLANE_2D": {
        "GEOMETRY": {...},
        "HYDRAULICS": {
          "PARAMETER": {...},
          "FRICTION": {...},
          "TURBULENCE": {...},
          "BOUNDARY": {...},
          "INITIAL": {...}
        }
      }
    }
  }
}
```

```
}

```

The blocks `PHYSICAL_PROPERTIES` and `BASEPLANE_2D` are mandatory. The physical property is the gravity and the components of the `BASEPLANE_2D` contain information about the domain (`GEOMETRY`) and the simulation type (`HYDRAULICS`).

```
"PHYSICAL_PROPERTIES": {
  "gravity": 9.81
}
```

The `GEOMETRY` part contains the path to the mesh file and different subsections as the interpolation method, a list of `STRINGDEF` for boundary conditions and a list of `REGIONDEF` to assign the friction, external sources and different interpolation methods.

```
"GEOMETRY": {
  "mesh_file": "Flaz_mesh.2dm",
  "INTERPOLATION": {
    "method": "weighted"
  },
  "STRINGDEF": [...],
  "REGIONDEF": [...]
}
```

```
"STRINGDEF": [
  { "name": "Inflow",
    "upstream_direction": "left"},
  { "name": "Outflow",
    "upstream_direction": "left"}
],
"REGIONDEF": [
  {
    "name": "one",
    "index": [1]},
  {
    "name": "two",
    "index": [2]},
  {
    "name": "three",
    "index": [3]},
  ...
]
```

The `HYDRAULICS` block contains the subsections `PARAMETER` for the hydraulic simulation only, `FRICTION` for each region, `TURBULENCE` parameters, `BOUNDARY` for the flow conditions and `INITIAL` for the condition at time  $t=0.0$ .

```
"PARAMETER": {
  "CFL": 0.95,
```

```
"minimum_water_depth":0.002,
"fluid_density": 1000.0,
"max_time_step": 100
}

"FRICITION": {
  "type": "strickler",
  "default_friction": 30,
  "regions": [
    {"region_name": "one",
     "friction": 28.0},
    {"region_name": "two",
     "friction": 30.0},
    {"region_name": "three",
     "friction": 35.0},
    ...
  ]
}

"TURBULENCE": {
  "INITIAL": {
    "type": "zero"
  },
  "PARAMETER": {
    "turbulence_start": 0,
  },
  "REYNOLDS_STRESS": {
    "closure_model": "mix_length",
    "constants": {
      "turb_visc_max": 0.05
    }
  }
}

"BOUNDARY":{
  "STANDARD": [
    {
      "name": "Inflow",
      "string_name": "Inflow",
      "type": "uniform_in",
      "discharge_file": "Inflow_stationary.txt",
      "slope": 0.02
    },
    {
      "name": "Outflow",
      "string_name": "Outflow",
      "type": "uniform_out",
      "slope": 0.02
    }
  ]
}
```



```

    ]
  }

  "INITIAL":{
    "type": "dry"
  }

```

The hydrograph is defined constant in a file “Inflow\_stationary.txt”, where the time in seconds (left) and discharge in  $m^3/s$  (right) is indicated.

```

0.0, 50.0
3600.0, 50.0

```

### 3.1.3.2 Setup the Configuration File simulation.json

The configuration file simulation.json defines the simulation time parameters (seconds) in the block TIME and the different output types inside the OUTPUT block.

```

{
  "SIMULATION":{
    "TIME": {
      "start": 0.0,
      "end": 3600,
      "out": 600
    },
    "OUTPUT": [
      "water_surface",
      "flow_velocity_abs",
      "ns_hyd_discharge",
      "bottom_elevation",
      "spec_discharge",
      "water_depth"
    ]
  }
}

```

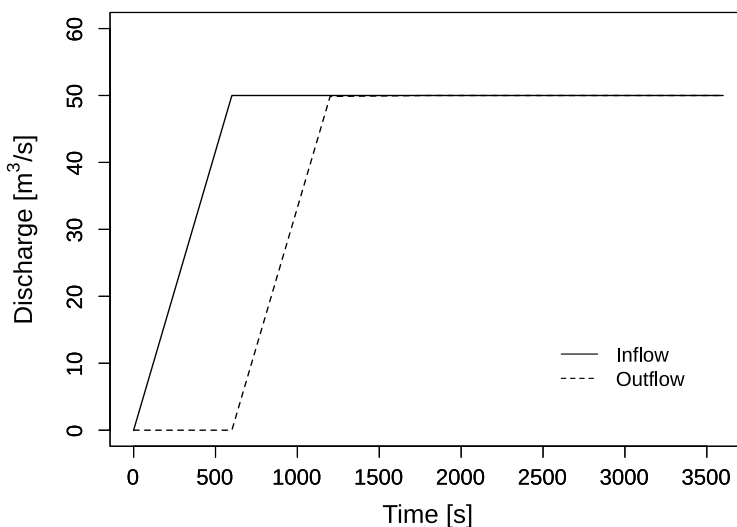
### 3.1.3.3 Set up the Configuration File results.json

The configuration file results.json defines the output format in the block EXPORT. Currently, xmdf is the only output format available.

```

{
  "RESULTS": {
    "EXPORT": [
      {"format": "xmdf"}
    ]
  }
}

```



*Figure 3.2 Inflow and outflow hydrograph at boundaries*

### 3.1.3.4 Steady Flow Simulation

The simulation results are stored inside the results.h5 binary. By calling the python script BMv3NodestringResults.py available on the [BASEMENT Website](#), the discharge values at boundaries are extracted and stored in the Discharge.csv file, where the rows are the time steps and the columns represents the stringdef in their definition order (inside the .2dm file). The steady state is reached after 1200 seconds (Figure 3.2).

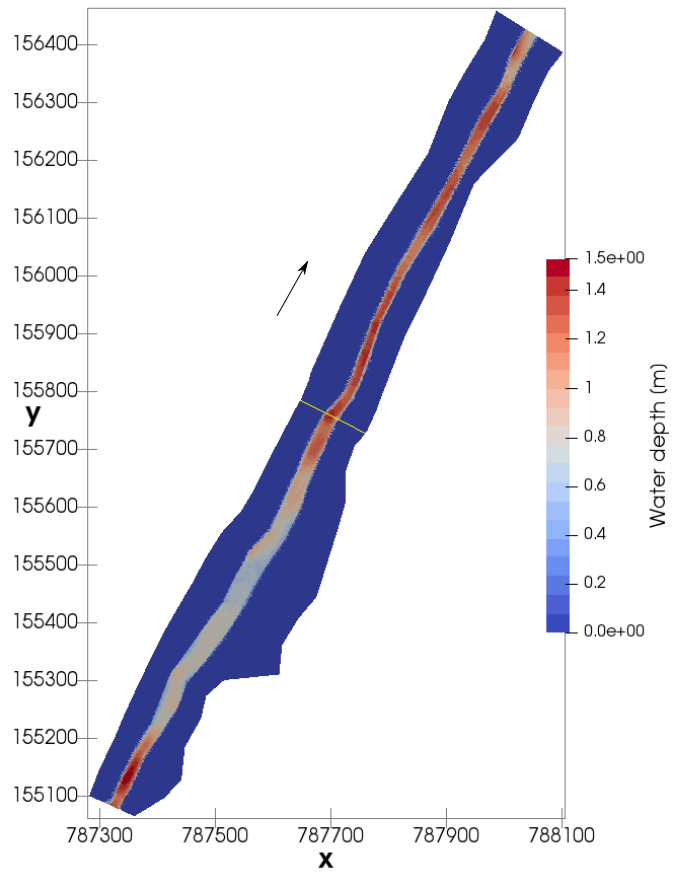
### 3.1.3.5 Model Calibration

The calibration of the friction value is done by comparing the water surface elevation between BASEMENT version 3 and version 2.8 at a cross section located in the middle of the channel (Figure 3.3). The water surface elevation and the water depth values along the cross section were obtained using the software ParaView.

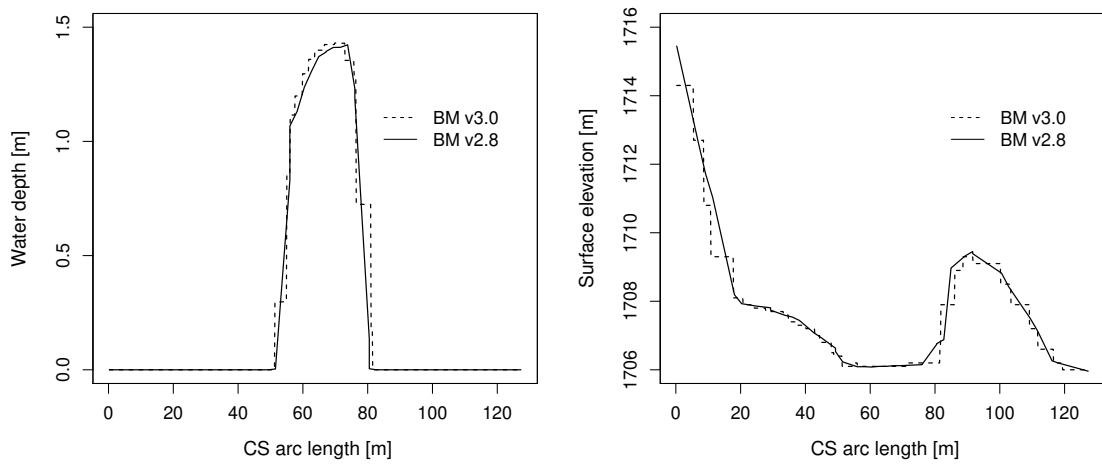
The resulting water depth and water surface elevation are compared in Figure 3.4. The steady flow simulation of BASEMENT version 3 provides similar results to those obtained with BASEMENT version 2.8. There is no need to modify the friction value defined in Section 3.1.3.1.

### 3.1.3.6 Unsteady Flow Simulation

The hydrograph based on the flood event of July 2004 provides unsteady flow conditions for the numerical simulation. The results of the steady flow simulation are stored in the binary Flaz\_steady\_state\_results.h5 inside the run/ folder and taken as initial state. The other parameters defined in Section 3.1.3.1 don't change, except for the boundary block where the new discharge file (Inflow\_transient.txt) replaces the stationary hydrograph. The initial block:



**Figure 3.3** Water depth at the end of the steady flow simulation ( $t=3600$  s) with the cross section location (yellow line)



**Figure 3.4** Cross sectional view of the water depth and surface elevation facing the opposite flow direction at the end of the steady flow simulation ( $t=3600$  s)

```
"INITIAL":{
  "type": "continue",
  "file": "../run/Flaz_steady_state_results.h5",
  "time": 3000.0
}
```

The run time in the configuration file simulation.json is increased to the last value of the discharge file (Inflow\_transient.txt). A higher “end” value will take the last discharge value written on the hydrograph to run the simulation. The starting time corresponds to the smallest time value of the discharge file Inflow\_transient.txt.

```
"TIME": {
  "start": 0.0,
  "end": 82000,
  "out": 2000
}
```

After changing the discharge file, modifying the initial block in the setup and the time of the simulation, the model is ready to proceed with the numerical simulation.

### 3.1.4 Morphology with uniform sediment transport

#### 3.1.4.1 Setup the Configuration File model.json

The unsteady flow simulation is now converted into a uniform sediment morphodynamic simulation with a morphology block and starting from the results of the steady flow simulation. The inflow hydrograph and the initial blocks are the same as described in Section 3.1.3.6. Additionally, the block morphology is defined inside the model.json file with the following structure

```
{
  "SETUP":{
    "simulation_name":"Flaz_unsteady_morph",
    "BASEHPC": {
      "PHYSICAL_PROPERTIES": {...},
      "BASEPLANE_2D": {
        "GEOMETRY": {...},
        "HYDRAULICS": {...},
        "MORPHOLOGY": {
          "INITIAL":{...},
          "PARAMETER": {...},
          "BEDMATERIAL": {...},
          "BEDLOAD": {...}
          "SUSLOAD": {...}
        }
      }
    }
  }
}
```

Inside the morphology block, the initial conditions look like:

```
"INITIAL":{
  "type": "mesh"
}
```

The morphology parameters defines the density of sediments, the porosity and the time at which the morphodynamic simulation starts. Further, the morphodynamic solver can be defined. For uniform sediment simulations, we recommend to always use the HLLC solver.

```
"PARAMETER": {
  "morphodynamic_start": 0.0,
  "sediment_porosity": 0.4,
  "sediment_density" : 2650.0,
  "morphodynamic_solver": "HLLC"
}
```

The bed material block inside the morphology block can be used to characterize the substrate. The structure of the block is described below. The parameter, soil definition and fixed bed blocks are optional, and either the soil assignment or the initial condition block is required:

```
"BEDMATERIAL": {
  "PARAMETER": {...},
  "GRAIN_CLASS": {...},
  "MIXTURE": {...},
  "SOILDEF": {...},
  "SOIL_ASSIGNMENT": {...},
  "INITIAL": {...},
  "FIXED_BED": {...}
}
```

The bed material parameters include the accuracy for fix bed correction defining the maximal overshoot below the fix bed elevation (*correction\_accuracy*) and the maximum number of iterations for fix bed correction (*max\_iteration*).

```
"BEDMATERIAL": {
  ...
  "PARAMETER": {
    "correction_accuracy": 0.001,
    "max_iteration": 10
  },
  ...
}
```

The bed material is composed of uniform grains with one diameter (m).

```
"BEDMATERIAL": {
  ...
  "GRAIN_CLASS": {
    "diameters": [0.050]
  },
  ...
}
```

Since we have only one grain size, the volume fraction of *mixture\_uniform* is equal to 100%.

```
"BEDMATERIAL": {
  ...
  "MIXTURE": [
    {
      "name": "mixture_uniform",
      "volume_fraction": [100]
    }
  ],
  ...
}
```

The soil layers and the according sediment mixture are defined in the SOILDEF-block. For a single-grain simulation it is not important how many layers are defined. The negative relative elevation defines the thickness of the layers. The default value of -999.0 results in an infinite substrate (no fixed bed). Setting a *z\_rel* value larger than -999.0 results in the definition of a fixed bed.

```
"BEDMATERIAL": {
  ...
  "SOILDEF": [
    {
      "name": "soil1",
      "LAYER": [
        {
          "z_rel": -999.0,
          "mixture_name": "mixture_uniform"
        }
      ]
    }
  ],
  ...
}
```

Soils can subsequently be assigned to different regions of the computational domain within the SOIL\_ASSIGNMENT block. Here, we define *soil1* as the default soil for the entire computational domain.

```
"BEDMATERIAL": {
  ...
  "SOIL_ASSIGNMENT": {
    "default_soil": "soil1"
  },
  ...
}
```

Note that there are three ways to define a fixed bed elevation: (1) via regions in the FIXED\_BED block, or (2) via a separate .2dm mesh file in the FIXED\_BED block, or (3) via layers in the SOILDEF block, as mentioned above. If fixed bed elevations are defined both in the FIXED\_BED and SOILDEF blocks, definitions in FIXED\_BED block overwrite fixed bed information from the SOILDEF block. In this example, we initialize the soil without a fixed bed ( $z_{rel} = -999.0$ ) and instead define the fixed bed via the FIXED\_BED block. Fixed bed elevations are assigned to different regions to prevent a high erosion. The erosion is unlimited if the fix bed is not defined.

```
"BEDMATERIAL": {
  "FIXED_BED": {
    "type": "region_defined",
    "regions": [
      {"region_name": "one",
       "z_rel": -0.8},
      {"region_name": "two",
       "z_rel": 0.0},
      {"region_name": "three",
       "z_rel": 0.0},
      ...
    ]
  }
}
```

In the bedload block, the bedload transport formula is chosen and the morphological boundary conditions are defined. The boundary condition is defined as `transport_capacity` at the inflow boundary. The value of the sediment flux is averaged over the stringdef length and equally distributed (same value) among the edges. Additionally, the mixture of the sediment entering the domain has to be specified. The outflow boundary has been reduced to the bed width composed of 6 nodes only. Therefore, the nodes located on the channel levee of the outflowing stringdefs have been removed from the .2dm file (computational mesh) manually using a text editor. The stringdef definition remains unchanged and the outflow boundary is defined as `equilibrium_out` with the reference bottom elevation set to 1700.68 m. Additionally, lateral transport due to local slope bed slope is considered with a default value of 1.5 (see Reference Manual).

```
"BEDLOAD": {
  "FORMULA": {
    "type": "MPM_like",
    "factor": 1.0,
    "coefficient": 3.2,
  }
}
```

```

    "exponent":1.6,
    "theta_critical_approach": "constant"
    "critical_value":0.047
  },
  "BOUNDARY":{
    "STANDARD": [
      {
        "name": "inflow_MOR",
        "string_name": "Inflow",
        "type": "transport_capacity",
        "boundary_factor": 0.8,
        "mixture_name": "mixture_inflow"
      },
      {
        "name": "outflow_MOR",
        "string_name": "Outflow",
        "type": "equilibrium_out",
        "reference_bed_elevation": 1700.68
      }
    ]
  },
  "DIRECTION": {
    "LATERAL_SLOPE": {
      "factor": 1.5
    }
  }
}

```

An additional suspended load block can be defined as well. The erosion and deposition laws are chosen and the morphological boundary conditions are defined. The boundary condition is defined as an input concentration at the inflow boundary. The outflow boundary is a simple zero-gradient outflow.

```

"SUSLOAD": {
  "PARAMETER": {
    "suspended_diffusivity": [0.1],
  },
  "BOUNDARY": {
    "STANDARD": [
      {
        "name": "inflow_SUS",
        "string_name": "Inflow",
        "type": "concentration_in",
        "data1": 0.0001
      },
      {
        "name": "outflow_SUS",
        "string_name": "Outflow",
        "type": "zero_gradient_out"
      }
    ]
  }
}

```



```

    }
  ]
},
"INITIAL": {
  "type": "zero"
},
"SETTLING_VELOCITY": {
  "closure_model": "van_rijn"
},
"SHEAR_PARTITION": {
  "default_friction_susp": 67.0
},
"EROSION_RATE": {
  "closure_model": "van_rijn",
  "constants": {
    "critical_shields": 0.125
  }
},
"DEPOSITION_RATE": {
  "closure_model": "lin",
  "constants": {
    "param_m": 1.0
  }
}
}
}

```

### 3.1.5 Morphology with mixed-size sediment transport

#### 3.1.5.1 Setup the Configuration File `model.json`

In order to avoid duplication, only the modifications of the `model.json` file, compared to the single-grain simulation, are detailed here.

In the PARAMETER-block important parameters for the morphological simulation are defined. For mixed-size sediment transport simulations, the *Upwind* solver must be used. We recommend the use of an upwind factor of 0.7 to 0.8. Further, the thickness of the bed load control volume is an important calibration parameter, which influences significantly the grain sorting process. Here, the constant active layer thickness is set to a constant thickness of 0.1 m.

```

"PARAMETER": {
  ...
  "morphodynamic_solver": "Upwind",
  "upwind_factor": 0.8,
  "ACTIVE_LAYER": {
    "active_layer_type": "constant",
    "active_layer_thickness": 0.1
  }
}
}

```

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

```
"BEDMATERIAL": {
  ...
  "GRAIN_CLASS": {
    "diameters": [0.001, 0.005, 0.015, 0.044, 0.082, 0.150]
  },
  ...
}
```

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.

```
"BEDMATERIAL": {
  ...
  "MIXTURE": [
    {
      "name": "mixture_inflow",
      "volume_fraction": [15, 15, 23, 27, 10, 10]
    },
    {
      "name": "mixture_element_roughness",
      "volume_fraction": [17, 11, 14, 27, 14, 17]
    },
    {
      "name": "mixture_widening",
      "volume_fraction": [21, 13, 16, 25, 11, 14]
    },
    {
      "name": "mixture_alt_bars",
      "volume_fraction": [27, 14, 14, 20, 14, 11]
    }
  ],
  ...
}
```

The soil layers with the corresponding sediment mixture are defined in the SOILDEF block. The soil can be defined with up to two layers of different material, but to keep it simple we assume a single layer. The negative bottom elevation defines the thickness of the layer. We use this especially for the river bed near the upper boundary condition to avoid uncontrolled erosion. A fully mobile bed (no fixed bed) is defined for the two soils *soil\_widerning* and *soil\_alt\_bar* by using the default value of -999.0 for *z\_rel* of the lowest layer. Furthermore the embankments are kept fix because the main focus is on set on the river bed morphology. The soils *soil\_fix\_20*, *soil\_fix\_40* and *soil\_element\_roughness* are defined to have a gradual transition from the fixed bed to the movable bed.

```
"BEDMATERIAL": {
```

```
...
"SOILDEF": [
  {
    "name": "soil_element_roughness",
    "LAYER": [
      {"z_rel": -0.8,
       "mixture_name": "mixture_element_roughness"}
    ]
  },
  {
    "name": "soil_widening",
    "LAYER": [
      {"z_rel": -2.0,
       "mixture_name": "mixture_widening"}
    ]
  },
  {
    "name": "soil_alt_bars",
    "LAYER": [
      {
        "z_rel": -2.0,
        "mixture_name": "mixture_alt_bars"
      }
    ]
  },
  {
    "name": "soil_fix",
    "LAYER": [
      {
        "z_rel": -0.0,
        "mixture_name": "mixture_element_roughness"
      }
    ]
  },
  {
    "name": "soil_fix_20",
    "LAYER": [
      {
        "z_rel": -0.2,
        "mixture_name": "mixture_element_roughness"
      }
    ]
  },
  {
    "name": "soil_fix_40",
    "LAYER": [
      {
        "z_rel": -0.4,
        "mixture_name": "mixture_element_roughness"
      }
    ]
  }
]
```

```

        }
    ]
}
],
}

```

Soils can subsequently be assigned to different regions of the computational domain within the SOIL\_ASSIGNMENT block. Here, we define *soil\_fix* as the default soil, but also individually assign the defined soils to the various regions of the mesh.

```

"BEDMATERIAL": {
    ...
    "SOIL_ASSIGNMENT": {
        "default_soil": "soil_fix",
        "regions": [
            {
                "region_name": "one",
                "soil_name": "soil_element_roughness"
            },
            {
                "region_name": "two",
                "soil_name": "soil_fix"
            },
            {
                "region_name": "three",
                "soil_name": "soil_fix"
            },
            ...
        ]
    }
}

```

Since we already defined the fixed bed elevations during the soil definition, we do not require the fixed bed block.

In the bedload block, the bedload transport formula MPM\_multi is chosen, which extends formula of Meyer-Peter and Müller (1948) for sediment mixtures by considering hiding and exposition effects with the approach of Ashida and Michiue (1971). At the inflow boundary, the sediment mixture *mixture\_inflow* is specified for any sediment entering the domain.

```

"BEDLOAD": {
    "FORMULA": {
        "type": "MPM_multi",
        ...
    },
    "BOUNDARY": {
        "STANDARD": [
            {
                "name": "inflow_MOR",

```

```

        "string_name": "Inflow",
        "type": "transport_capacity",
        "boundary_factor": 0.8,
        "mixture_name": "mixture_inflow"
    },
    ...
]
},
...
}

```

### 3.1.5.2 Setup the Configuration File simulation.json

The simulation time defined in the simulation.json file is the same as for the uniform sediment transport simulation and eleven additional outputs are defined (active layer thickness, active layer fractions, fractions of the upper and lower sublayer, elevation of the lower edge of the sublayers, the arithmetic and geometric mean grain size of the active layer, and the characteristic diameters  $d_{16}$ ,  $d_{50}$ ,  $d_{84}$ , and  $d_{90}$  of the active layer.

```

{
  "OUTPUT": [
    "...",
    "active_layer_thickness",
    "active_layer_fractions",
    "sublayer_fractions0",
    "sublayer_fractions1",
    "sublayer_elevations",
    "dm_arithmetic",
    "dm_geometric",
    "d_16",
    "d_50",
    "d_84",
    "d_90"
  ]
}

```

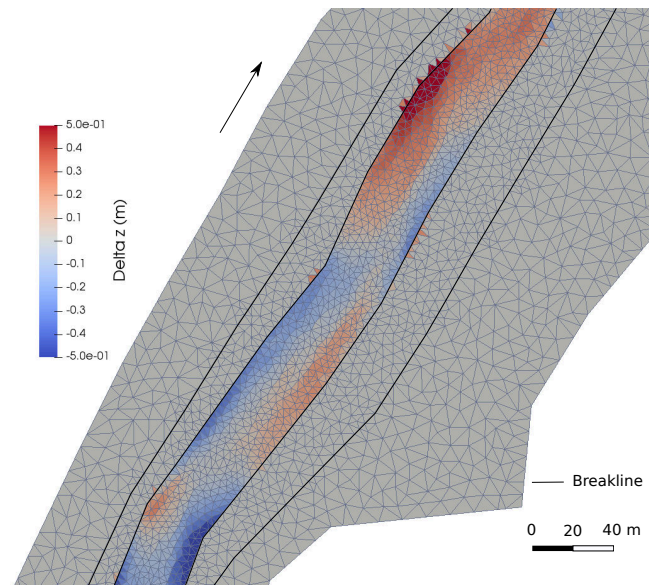
### 3.1.5.3 Results

The morphological changes of the river bed are observed on Figure 3.5. The software ParaView was used for the post-processing of the output file (.xdmf).

## 3.1.6 Passive tracers

### 3.1.6.1 Setup the Configuration File model.json

This section shows how the flow simulation could be augmented with scalar advection with a TRACERS block. A maximum of 5 tracers are presently supported. Additionally, the block tracers is defined inside the model.json file with the following structure:



**Figure 3.5** Planar view of the widening part of the river Flaz with the change in bed elevation ( $\Delta z$ ) due to the flood event at the end of the morphodynamic simulation ( $t=84'000$  s)

The configuration file model.json for a tracers simulation has the following structure:

```
{
  "SETUP":{
    "simulation_name":"Flaz_tracers",
    "BASEHPC": {
      "PHYSICAL_PROPERTIES": {...},
      "BASEPLANE_2D": {
        "GEOMETRY": {...},
        "TRACERS": {
          "PARAMETER": {...},
          "BOUNDARY": {...},
          "INITIAL": {...},
          "SOURCE": {...}
        }
      }
    }
  }
}
```

The TRACERS block contains the subsections PARAMETER, BOUNDARY and INITIAL:

```
"PARAMETER": {
  "tracers_start": 0.0,
  "num_tracers": 3,
}

"BOUNDARY":{
```

```

"STANDARD": [
  {
    "name": "tracer_inflow",
    "string_name": "Inflow",
    "type": "discharge_in",
    "tracer1_file": "Inflow_tracer1.txt",
    "tracer2_file": "Inflow_tracer2.txt",
    "tracer3_file": "Inflow_tracer3.txt"
  },
  {
    "name": "Outflow",
    "string_name": "Outflow",
    "type": "zero_gradient_out"
  }
]
}

```

The tracer discharge series are of the same format of the hydrograph: two columns with time (first col.) and imposed quantity (discharge or concentration) (second col.). To specify constant values use “tracerX” followed by the intended value, instead of “tracerX\_file”.

```

"INITIAL":{
  "type": "zero"
}

```

```

"INITIAL":{
  "type": "uniform",
  "tracers": [0.0,0.25,0.5]
}

```

```

"INITIAL":{
  "type": "region_defined",
  "regions": [
    {"region_name": "one",
     "tracers": [0.0,0.1,0.2]},
    {"region_name": "two",
     "tracers": [0.4,0.3,0.2]}
  ]
}

```

The initial conditions are prescribed in array format with the “tracers” keyword. The supported types of initial conditions with this input method are “uniform” and “region\_defined”. The remaining options are “zero”, with no additional inputs, and “continue” accepting “file” and “time” parameters.

A source block within the tracers block would look like:

```

"SOURCE": [
  {

```

```

    "name": "total_exact1",
    "type": "total",
    "data1_file": "./scalars1.dat",
    "data2_file": "./scalars2.dat",
    "data3_file": "./scalars3.dat",
    "data4_file": "./scalars4.dat",
    "data5_file": "./scalars5.dat",
    "region_name": "one",
    "sink": "exact"
  }
]

```

And, as in the boundaries case, to specify constant values use “dataX” followed by the intended value, instead of “dataX\_file”.

### 3.1.6.2 Setup the Configuration File simulation.json

The configuration file simulation.json defines the simulation time parameters (seconds) in the block TIME and the different output types inside the OUTPUT block.

```

{
  "SIMULATION":{
    "TIME": {
      "start": 0.0,
      "end": 3600,
      "out": 600
    },
    "OUTPUT": [
      "water_surface",
      "flow_velocity_abs",
      "ns_hyd_discharge",
      "bottom_elevation",
      "spec_discharge",
      "water_depth",
      "tracer1",
      "tracer2",
      "tracer3"
    ]
  }
}

```

### 3.1.6.3 Set up the Configuration File results.json

The configuration file results.json defines the output format in the block EXPORT. Currently, xmdf is the only output format available.

```

{
  "RESULTS": {

```



```
    "EXPORT": [  
      {"format": "xdmf"}  
    ]  
  }  
}
```



---

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

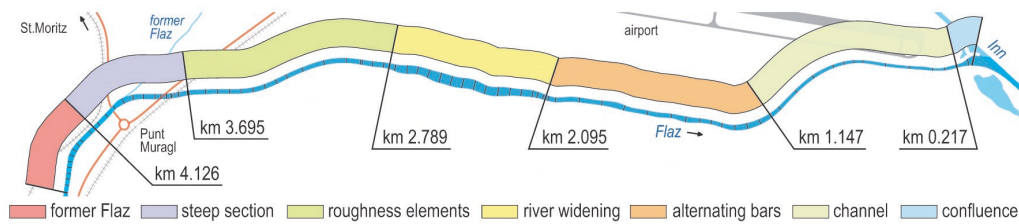
## 4.1 Hydrodynamics and sediment transport at the river Flaz with BASEMD (2D)

### 4.1.1 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.

#### 4.1.1.1 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 4.1). 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 4.1.



**Figure 4.1** Different morphological river subsections of the new section of the river Flaz

#### 4.1.1.2 Tutorial structure

In a first step the important properties of the mesh file are shown. The tutorial is designed to run `BASE_MD_` 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 `QGIS` 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 structure of this tutorial reflects the recommended steps to go from a hydraulic simulation to a morphodynamic simulation. A hydraulic simulation is initially set up. Based on this simulation the morphological part can be added with a single-grain model. An outline of outputs is given and possible visualization is shown for each step.

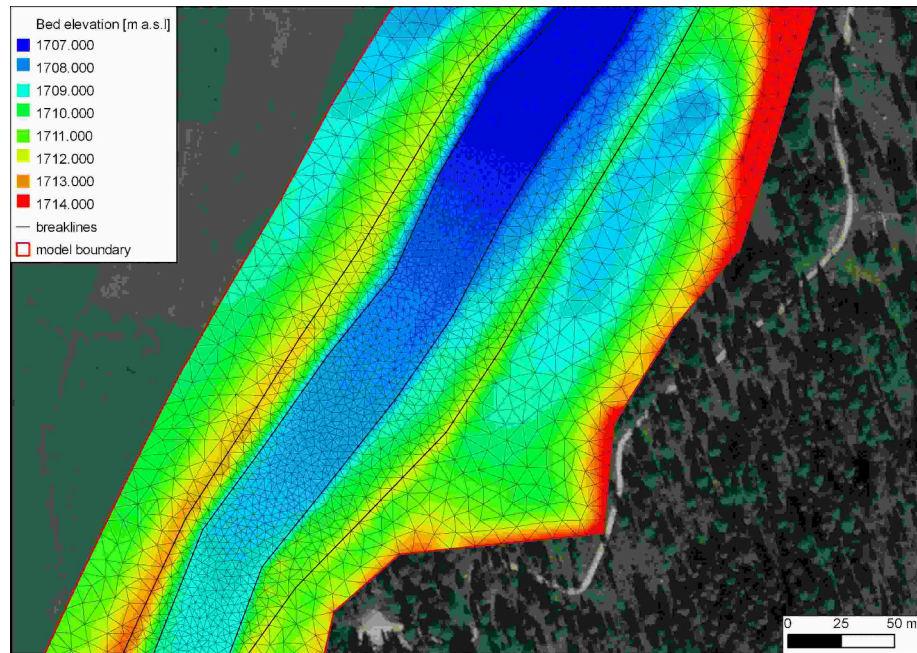
#### 4.1.2 Computational grid

The computational mesh is generated with the `QGIS` plugin `BASEmesh`. Here, only a few important features and characteristics of the computational mesh for 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 4.2).

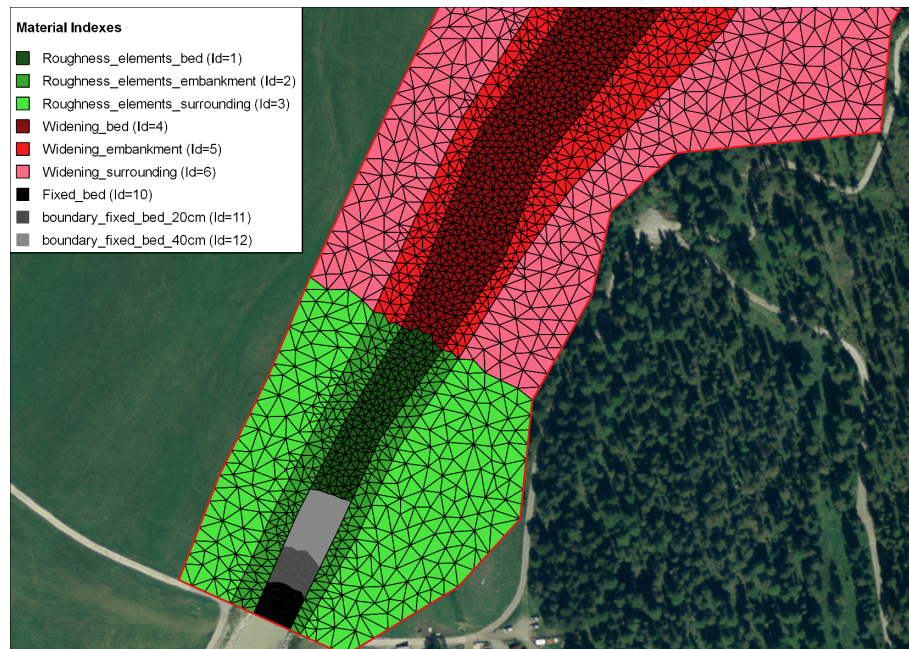
An important feature is the assignment of the material indexes to the different groups of elements. By the material index different properties such as the friction factor and the soil properties can be assigned. The material index is mainly used to assign the friction factor to the different river sections separately. For example it is usual to assign different values for the main channel, embankments, and surrounding floodplain (Figure 4.3).

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
.
```



*Figure 4.2* Computational mesh generated with BASEmesh for a section in the widening part. The bold black lines are breaklines for the mesh.



*Figure 4.3* Material indices used for assignment of the friction factor and the soil properties.

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.

### 4.1.3 Setting up the model file

The configurations files (`model.json`, `simulation.json` and `results.json`) can be created and modified with the graphical user interface (GUI) or any text editor. The configuration files must all be located in one folder.

Open the model setup file `model.json` via the menu of the BASEMENT GUI (File → Open Scenario Directory → select the folder that contains the configuration files). The structure of the model setup file looks like this:

```

SETUP
{
  "simulation_name": "2D_Tutorial"
  BASEMD
  {
    multiregion = Flaz
    PHYSICAL_PROPERTIES
    {...}
    BASEPLANE_2D
    {
      region_name = Flaz
      GEOMETRY
      {...}
      HYDRAULICS
      {...}
      TIMESTEP
      {...}
      MORPHOLOGY
      {...}
      OUTPUT
      {...}
    }
  }
}

```

#### 4.1.3.1 BASEMD

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

```

BASEMD
{
  multiregion = Flaz

```

```

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

```

#### 4.1.3.2 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]
}

```

#### 4.1.3.3 Two dimensional domain

The *BASEPLANE\_2D*-block within the BASEMD-block contains all information concerning the two dimensional model domain.

```

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

```

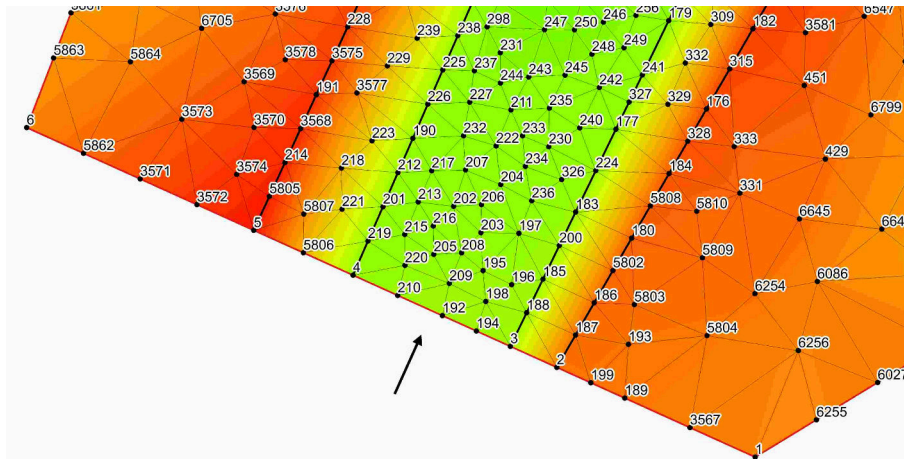
#### 4.1.3.4 Geometry

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 4.4 and Figure 4.5 respectively.

```

GEOMETRY

```



**Figure 4.4** Node id numbers for the definition in the `STRING_DEF` block for the inflow boundary.

```
{
  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
  }
}
```

#### 4.1.3.5 Define the hydraulics

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

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



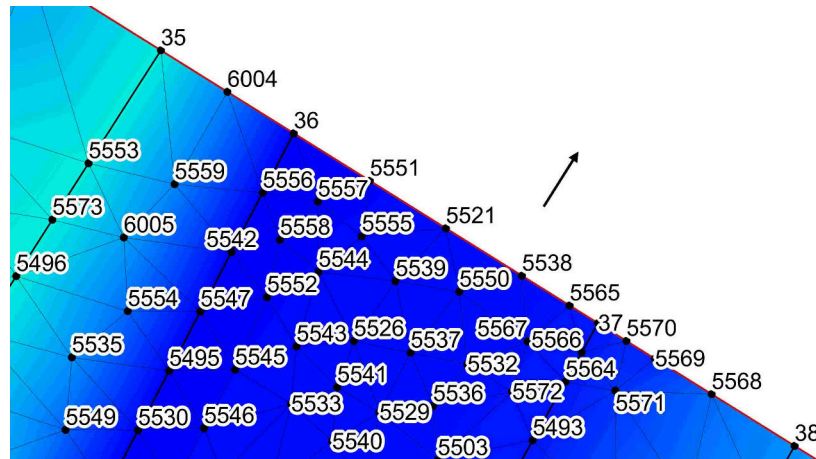


Figure 4.5 Node id numbers for the definition in the `STRING_DEF` block for the outflow boundary.

```
{...}
}
```

#### 4.1.3.5.1 Hydraulic boundary conditions

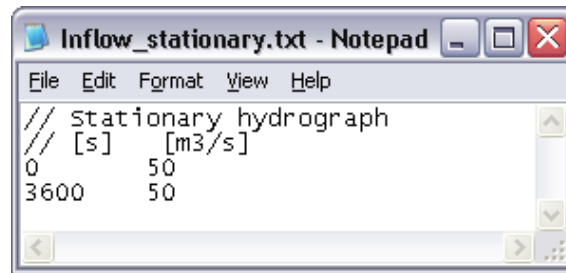
For the upper (inflow) and lower (outflow) boundary condition we have to refer to the predefined STRINGDEFs (see Section 4.1.3.4). If the boundary condition is not defined explicitly a wall boundary is considered for those edges. Except for the explicitly defined inflow and outflow boundary the model 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 4.6). As a 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 m<sup>3</sup>/s is chosen in order to continue later on with a flood hydrograph which starts in this range.



**Figure 4.6** Stationary hydrograph file saved as *Inflow\_stationary.txt*.

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 should always be done. 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]
}
```

#### 4.1.3.5.2 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
}
```

#### 4.1.3.5.3 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 4.3) 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
}
```

#### 4.1.3.5.4 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
}
```

#### 4.1.3.5.5 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 4.1.3.7).

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

#### 4.1.3.5.6 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, 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
}
}

```

#### 4.1.3.6 Write the setup file

Once all the inputs have been set in the in the *model.json* file, a setup file must be written. Click *write* in the bottom right corner of the GUI. This will combine the model file with the computational mesh and external data, and store it in a binary file named *setup.h5*.

#### 4.1.3.7 Perform hydraulic simulations

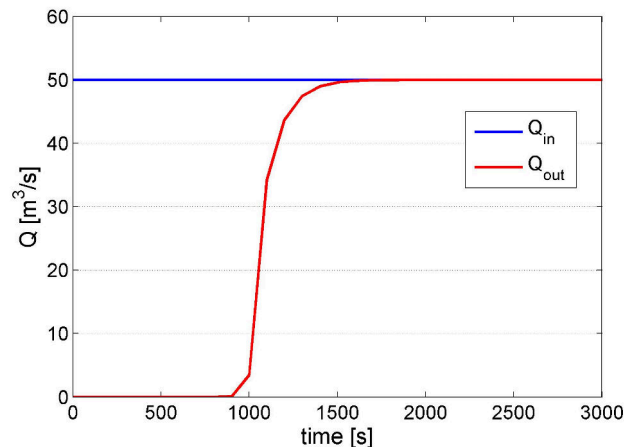
##### 4.1.3.7.1 Setting up the simulation file

To set up the simulation file, switch to the *simulation* tab on the left. Select whether the simulation should be run on a single core (select *Single-threaded on CPU*) or on multiple cores (selected *Multi-threaded on CPU*). If multiple cores are selected, you must enter the *number of CPU cores*. This selection should depend on the number of threads your computer has available.

##### 4.1.3.7.2 Perform steady flow simulation

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 configuration files, computational mesh, 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 4.7).

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



**Figure 4.7** Steady inflow hydrograph and outflow hydrograph.

of the limiting element allows for improvement of the mesh. In order to identify the element location, load the computational mesh into QGIS.

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 as shown in Figure 4.8. At the end of the simulation the flow variables of the last time step ( $t = 3000$  s) are stored in the *Flaz\_restart.h5* binary file. This file can be used later on to continue the simulation or to use it as an initial condition for a new simulation.

#### 4.1.3.7.3 Perform unsteady flow simulation

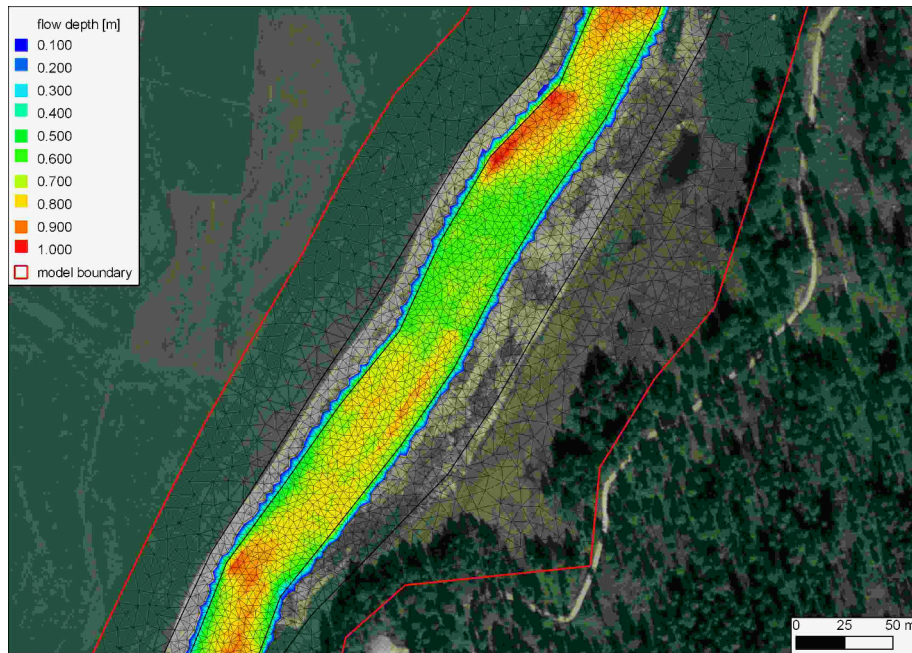
The unsteady flow simulation is based on the flood event of July 2004 (Figure 4.10). Compared to the steady flow simulation the command file needs some minor changes. First, the last time step of the steady simulation is taken as initial condition for the unsteady simulation. Therefore the file *Flaz\_restart.h5* from the steady simulation can be renamed and saved for example as *Initial\_Condition.h5*. 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.

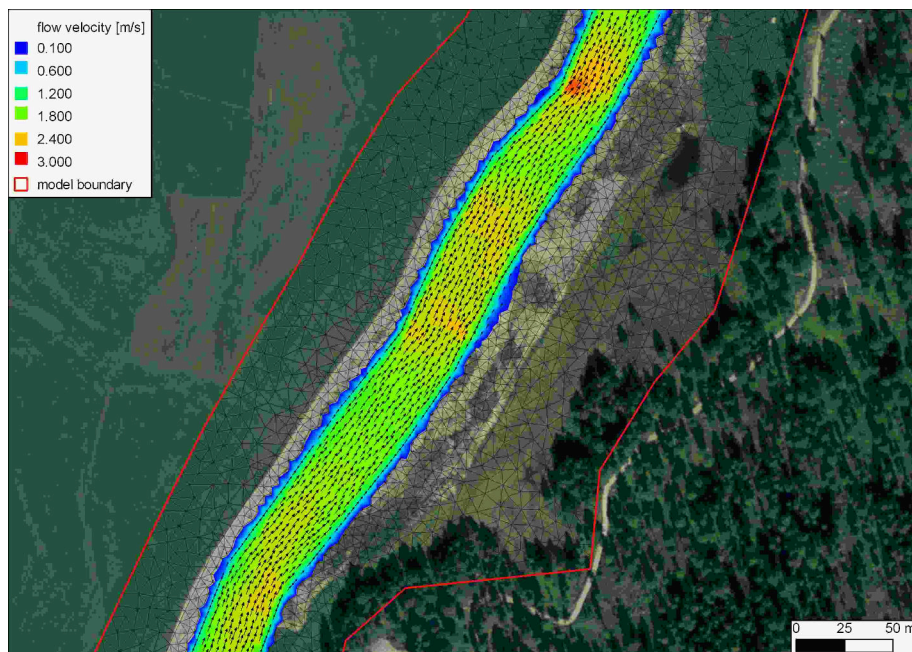
In this case we set the tag *restart\_solution\_time* equal to the solution of the last time step of the stationary simulation (*restart\_solution\_time* = 3000.027 sec). In order to start the simulation from time = 0 again, we have to set the *start\_time* in the TIMESTEP block to zero. Be aware that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

```
INITIAL
{
  type = continue
  file = Initial_Condition.h5
  restart_solution_time = 3000.027
```

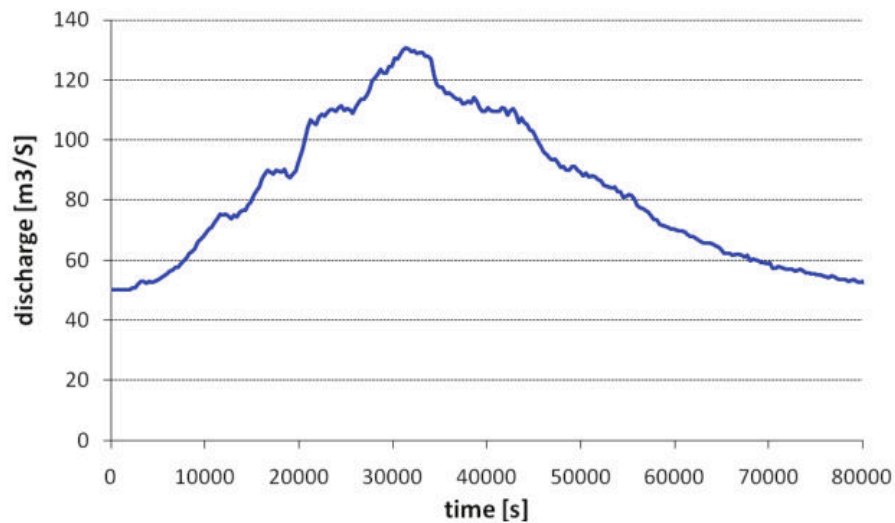




*Figure 4.8* Flow depth at the steady state of the model.



*Figure 4.9* Flow velocity and velocity vectors at the steady state of the model.



**Figure 4.10** 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 4.10 is saved in the text file *Inflow\_transient.txt* with contents as depicted in Figure 4.11. 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]
}

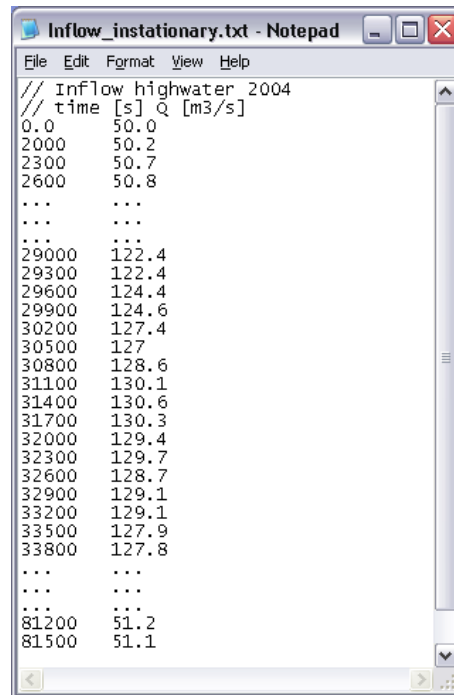
```

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 4.11** 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.

...  
}

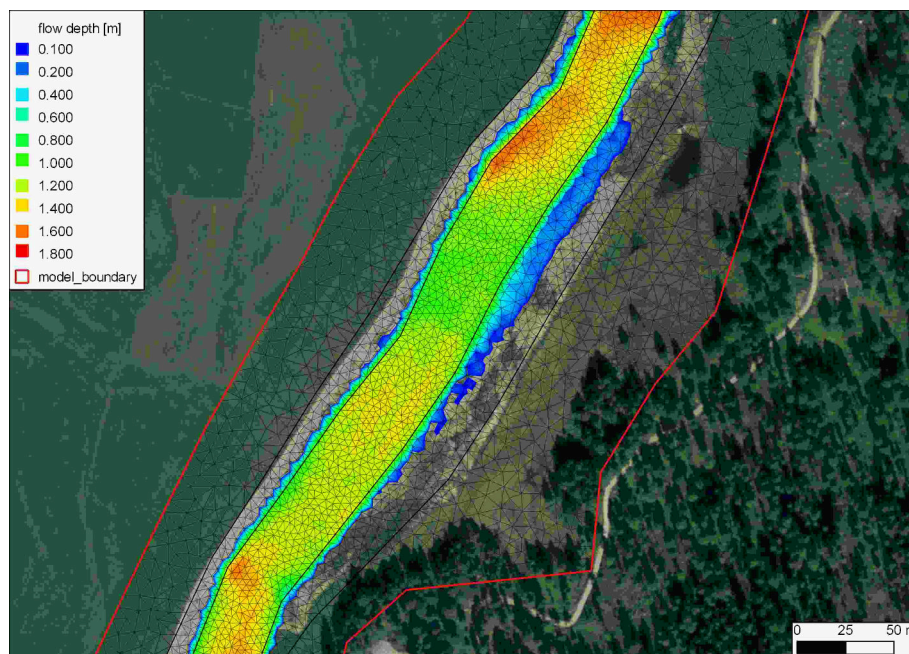
Last but not least, the OUPUT-block has to be adjusted to the needs of the simulation. For the unsteady simulation, the output time step for the SPECIAL\_OUTPUT format sms might be reduced in order to capture the maximum values of the output variables .

```

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

```

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

```

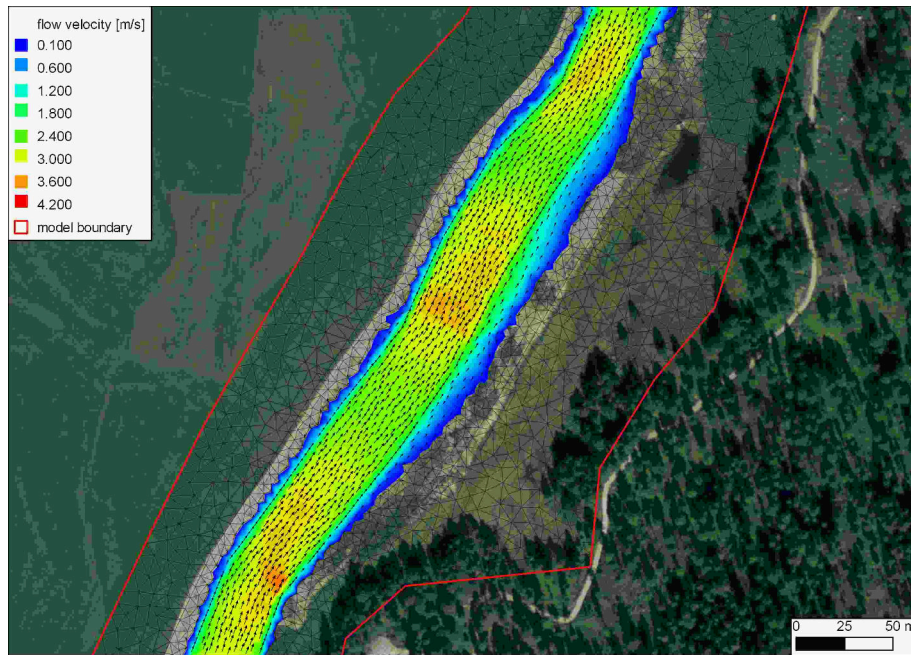
Open the model file from the BASEMENT GUI (File → Open Scenario Directory → select the folder that contains the configuration files). Run the simulation with the *Run* button of the BASEMENT window. The maxima values of the flow depths and flow velocity vectors can be visualized using QGIS as shown in Figure 4.12 and Figure 4.13.

#### 4.1.3.7.4 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.

#### 4.1.4 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.



**Figure 4.13** Maximal flow velocity and velocity vectors of the unsteady flow simulation observed at the flood peak after a run time of about 9 hours.

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 4.1.3.7.3. 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
  }
  ...
}

```

#### 4.1.4.1 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
    {...}
}

```

#### 4.1.4.1.1 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]
}

```

#### 4.1.4.1.2 Initial conditions

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

#### 4.1.4.1.3 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
    {...}
}

```

**4.1.4.1.3.1 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]
}
```

**4.1.4.1.3.2 Grain mixture**

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

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

**4.1.4.1.3.3 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
}

```

#### 4.1.4.1.3.4 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 4.1.4.1.3.3). 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 )
}
```

#### 4.1.4.1.4 Assignment of the defined soil types

The soil types defined in the SOIL\_DEF-blocks (Section 4.1.4.1.3.3) 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_altBars soil_fix soil_fix soil_fix
  soil_fix_20 soil_fix_40 )
}
```

#### 4.1.4.2 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
  {...}
}
```

##### 4.1.4.2.1 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
}
```

**4.1.4.2.2 Bed load formula**

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

```

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

```

**4.1.4.2.3 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
}

```

**4.1.4.2.4 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 )
}

```

#### 4.1.4.3 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
}
```

#### 4.1.4.4 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
    }
}

```

#### 4.1.4.5 Perform morphological simulation with single-grain bed load transport

Open the *model.json* file via the menu of the BASEMENT GUI (File → Open Scenario Directory → select the folder that contains the configuration files). Run the simulation with the Run button of the BASEMENT window.

The output files with the ending *.sol* can be visualized using QGIS . The bed elevation after the flood event is shown in Figure 4.14. 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 4.15). The morphological changes (*deltaz*) due to the flood event are shown in Figure 4.16 .The position of the simulated depositions matches quite well with the position of the banks observed in the aerial image.

#### 4.1.5 Morphological simulation with multi-grain bed load transport

In order to avoid duplication, only the modifications of the *model.json* file, compared to the single-grain simulation, are detailed here. A modification that is not considered here is the possibility to use the grain size distribution to determine the bed friction in the FRICTION-Block. To simplify this tutorial, 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.

##### 4.1.5.1 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.

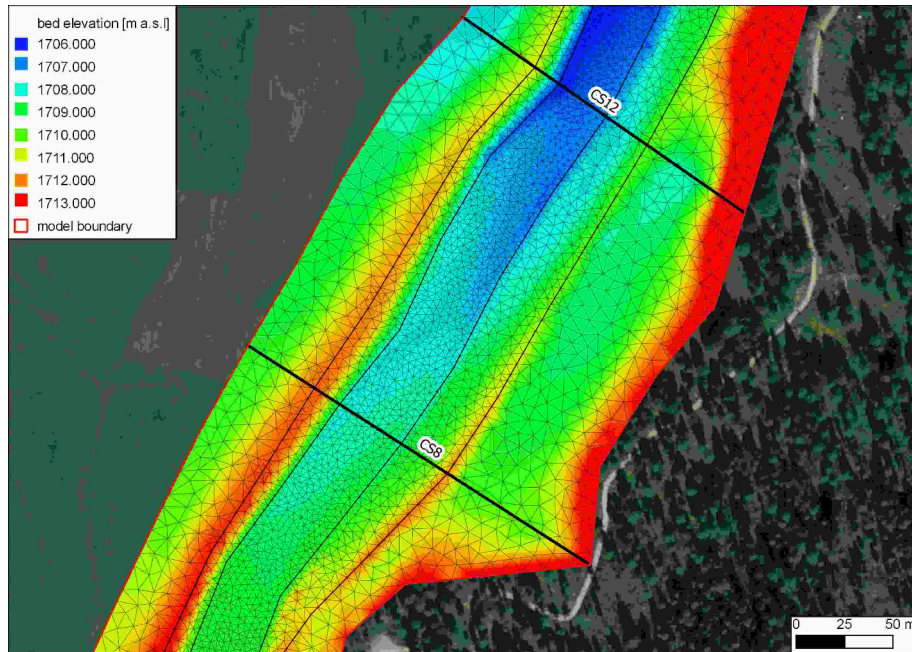


Figure 4.14 Modeled bed elevation ( $z_{bed}$ ) and two cross sections defined in the widening part.

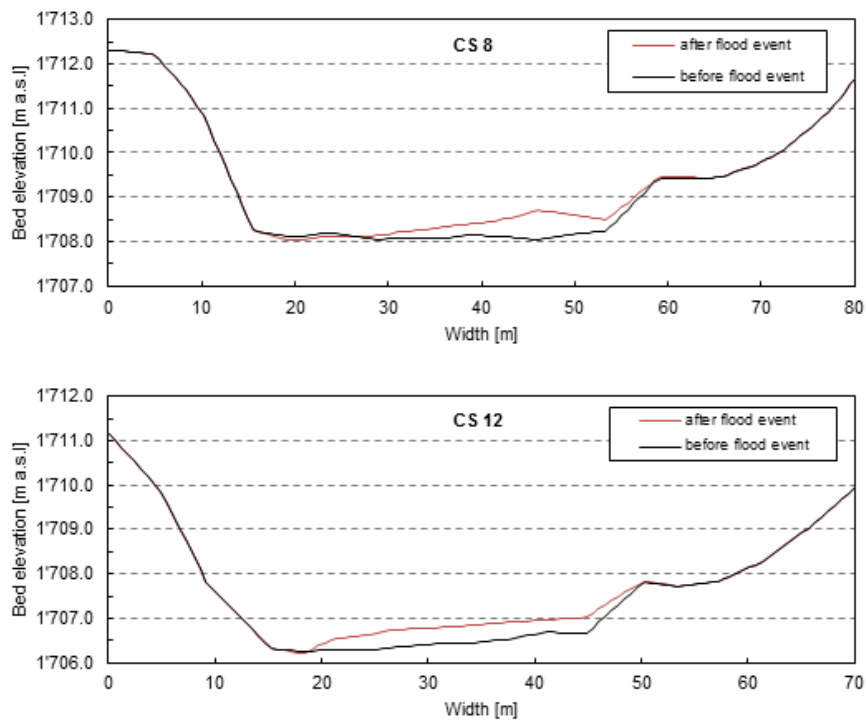
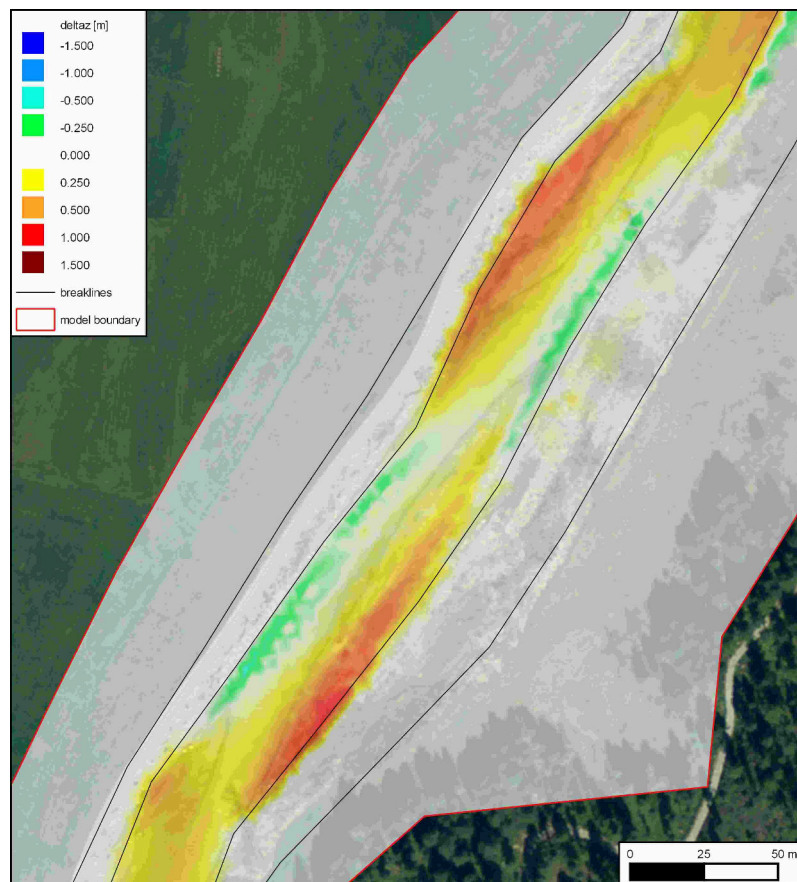


Figure 4.15 Comparison of the river bed before and after the flood event in cross-section CS 8 and CS 12.



**Figure 4.16** 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.

```

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)
}

```

#### 4.1.5.2 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]
}

```

#### 4.1.5.3 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 )
}

```

#### 4.1.5.4 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
  }
}

```

#### 4.1.5.5 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
}

```

#### 4.1.5.6 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
}

```

#### 4.1.5.7 Define the output

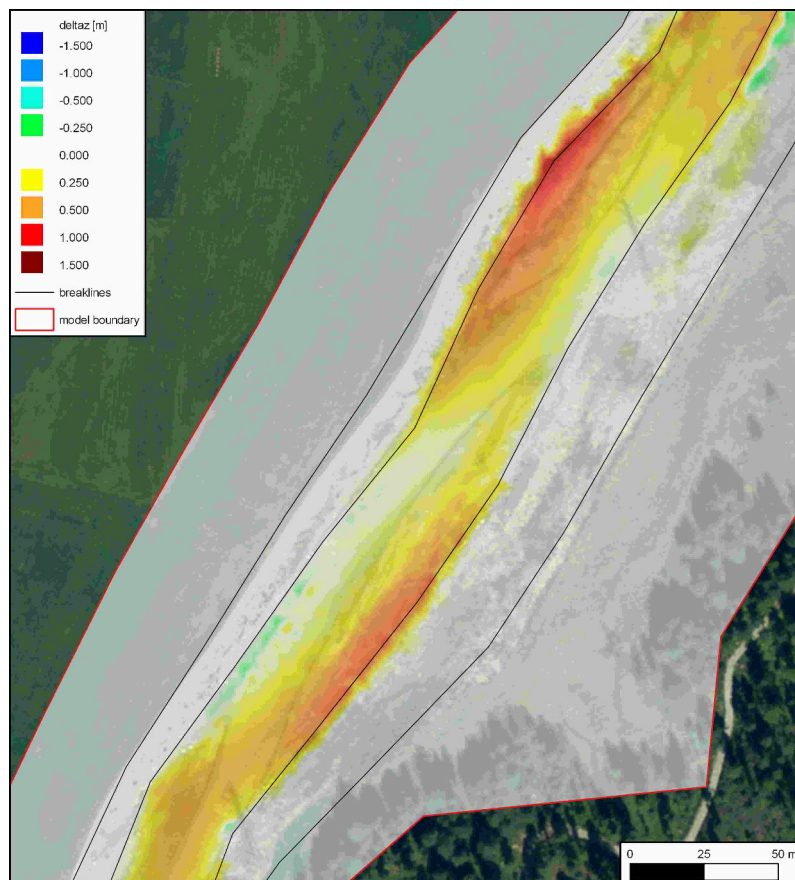
The desired output of the simulation has to be defined explicitly in the OUTPUT-block. The specific output is defined in the repeatable SPECIAL\_OUTPUT-blocks. For the multi-grain simulation some additional output may be interesting such as for example the grain size distribution in selected nodes. This 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 deltax 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
  }
}
```

#### 4.1.6 Perform morphological simulation with multi-grain bed load transport

Open the *model.json* file via the menu of the BASEMENT GUI (File → Open Scenario Directory → select the folder that contains the configuration files). Run the simulation with the Run button of the BASEMENT window.

The morphological changes *deltax* are shown in Figure 4.17. 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 4.16 and Figure 4.17). 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 4.17** Changes of the morphology ( $\text{deltaz}$ ) due to the flood event with the multi-grain model. The red colour range represents deposition and the green/blue colour range shows erosion



---

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

## 5.1 Hydrodynamics and sediment transport at the river Thur with BASEMD (1D)

### 5.1.1 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.

#### 5.1.1.1 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.

#### 5.1.1.2 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;



*Figure 5.1 View of the simulated river section*

- 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.

### 5.1.1.3 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.

## 5.1.2 Setting up the topography file

### 5.1.2.1 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

x	y	z
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
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 5.2). 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.

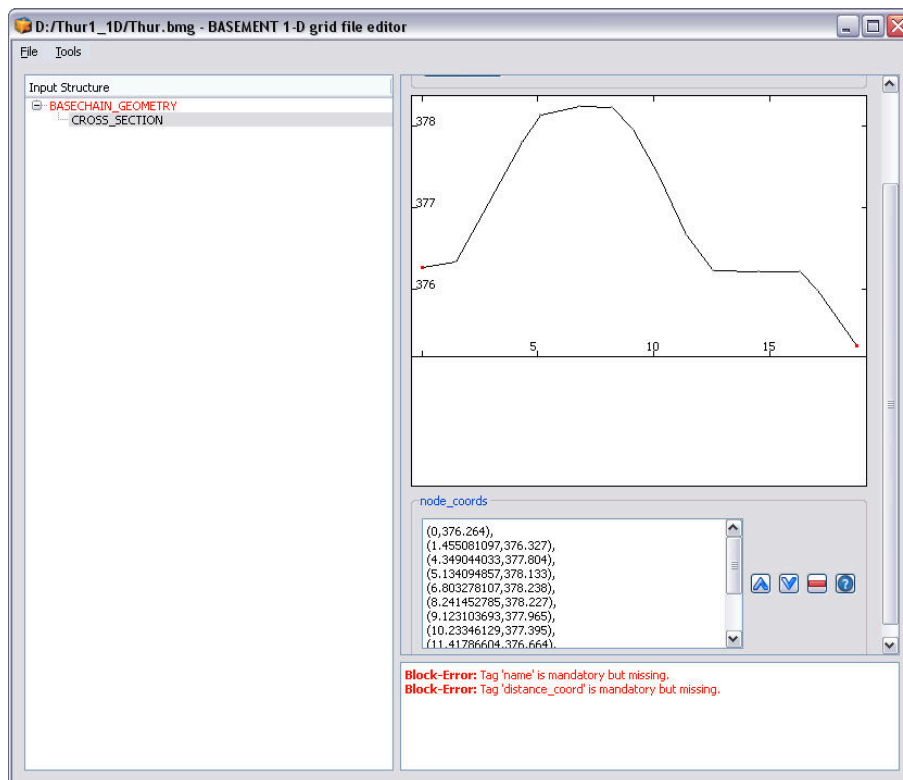


Figure 5.2 Cross section points inserted in the topography editor

### 5.1.2.2 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 5.3, 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 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 5.3. 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 5.4).

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*.

### 5.1.2.3 Friction values

For the friction determination the Strickler approach is used. This is declared in the command file by setting the *type* in the **FRICITION** 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:

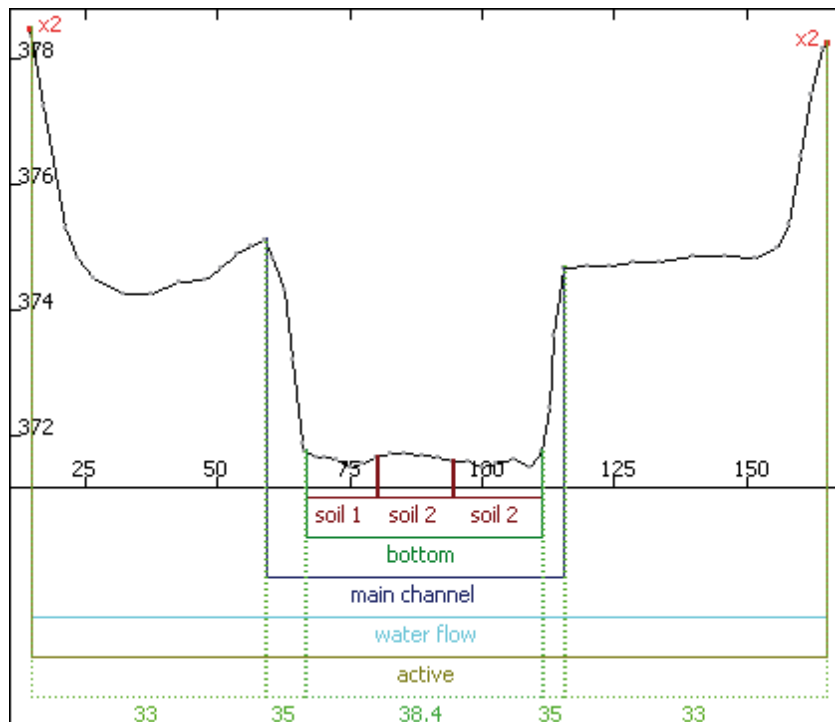


Figure 5.3 Delimitation of cross section zones

name  
CS2

distance\_coord  
0.049821

main\_channel\_range  
60.523 117.128

friction\_coefficients  
33, 35, 38.4, 35, 33

friction\_ranges  
(10.233,60.523),  
(60.523,67.274),  
(67.274,109.984),  
(109.984,117.128),  
(117.128,167.344)

node\_coords  
(10.233,377.395),  
(11.417,376.664),  
(12.570,376.221),  
(14.532,376.21),  
(16.341,376.215),  
(17.099,375.99),  
(18.745,375.296)

Figure 5.4 Definition of cross section properties

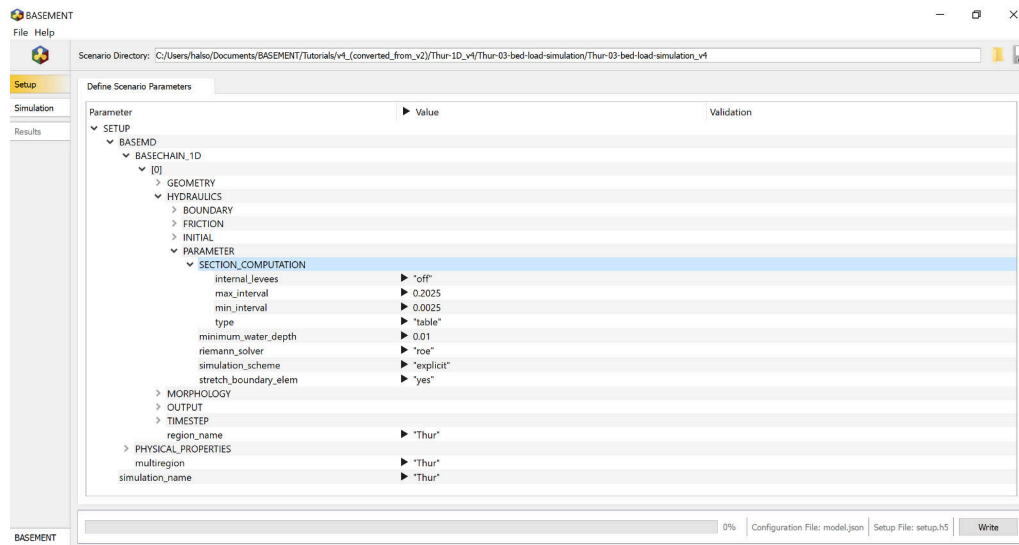


Figure 5.5 Definition of table values

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

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

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

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

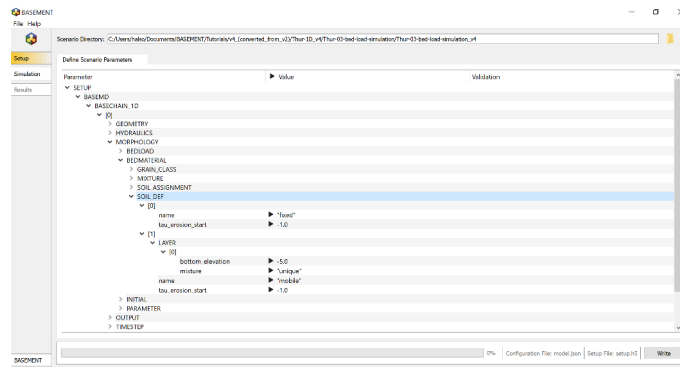
#### 5.1.2.4 Computation of water surface elevation

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.

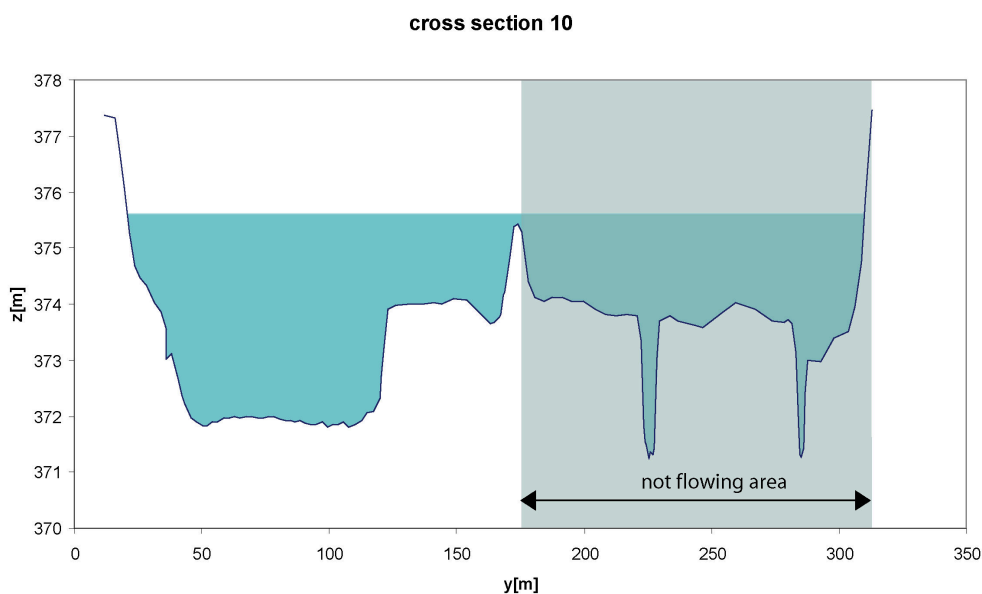
#### 5.1.2.5 Characterization of the sediments

Two types of ground will be defined:

1. the ground is not erodible;



**Figure 5.6** Declaration of soil types



**Figure 5.7** Definition of non flowing areas

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.

#### 5.1.2.6 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.

### 5.1.3 Setting up the model file

#### 5.1.3.1 Setup

The first configuration file is called “model.json”. Open the model file from the BASEMENT GUI (File → Open Scenario Directory → select the folder that contains the configuration files).

The first step is to define a project by its name:

```
SETUP
{
  "simulation_name": "Thur"
}
```

#### 5.1.3.2 Domain

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

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

#### 5.1.3.3 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
}
```

#### 5.1.3.4 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
}
```



**5.1.3.4.1 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 )
}
```

**5.1.3.4.2 Define hydraulic information**

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

```
HYDRAULICS
{
    [...]
}
```

**5.1.3.4.2.1 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
}
```

#### 5.1.3.4.2.2 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
}
```

#### 5.1.3.4.2.3 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
}
```

#### 5.1.3.4.2.4 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
}
```

#### 5.1.3.4.2.5 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
}
```

#### 5.1.3.4.3 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
}
```

#### 5.1.3.4.4 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
    }
}
```

### 5.1.4 Perform hydraulic simulations

#### 5.1.4.1 Write the setup file

Once all the inputs have been set in the in the *model.json* file, a setup file must be written. Click *write* in the bottom right corner of the GUI. This will combine the model file with the computational mesh and external data, and store it in a binary file named *setup.h5*.

#### 5.1.4.2 Setting up the simulation file

To set up the simulation file, switch to the *simulation* tab on the left. Select whether the simulation should be run on a single core (select *Single-threaded on CPU*) or on multiple cores (selected *Multi-threaded on CPU*). If multiple cores are selected, you must enter the *number of CPU cores*. This selection should depend on the number of threads your computer has available.

### 5.1.4.3 Perform steady flow simulation

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.

### 5.1.4.4 Results of the steady flow simulation

When the simulation has finished, several results files will be produced in “.dat” form. *In the Basement window, you may ignore the “Results” tab on the left side. This tab is only active for 2D simulations.* remains 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 labeled “Mass\_Flux\_[m<sup>3</sup>/s]”. If the discharges correspond to the steady inflow discharge of 30 m<sup>3</sup>/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.

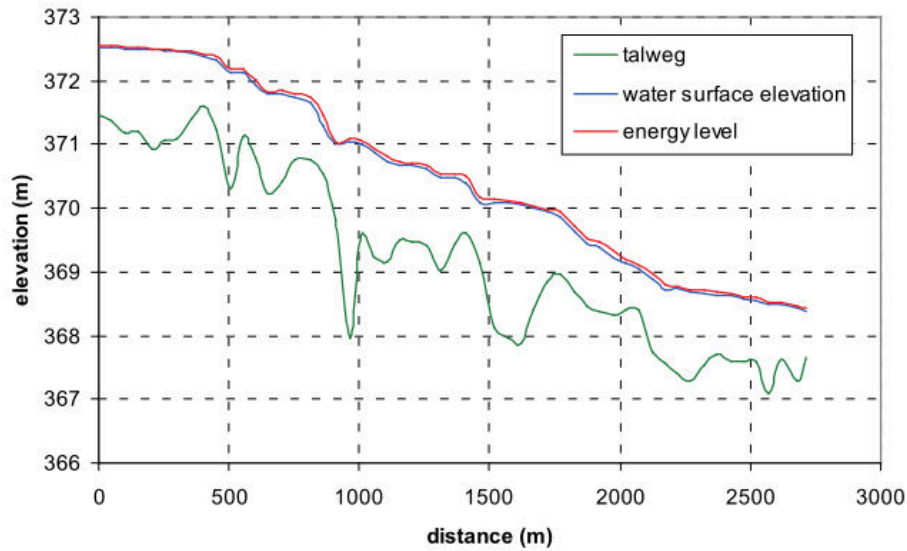
Edge_number	Mass_Flux_[m <sup>3</sup> /s]	Momentum_Flux_[m <sup>4</sup> /s <sup>2</sup> ]	...
0	30	21.0328	...
1	30	20.7677	...
2	30	20.8081	...
3	30	18.4419	...
4	30	16.5479	...
5	30	18.1862	...
6	30	19.3961	...
7	30	21.1823	...
8	30	21.8026	...
...			

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.



*Figure 5.8* Longitudinal profile

#### 5.1.4.5 Perform simulation of the floods (Thur2)

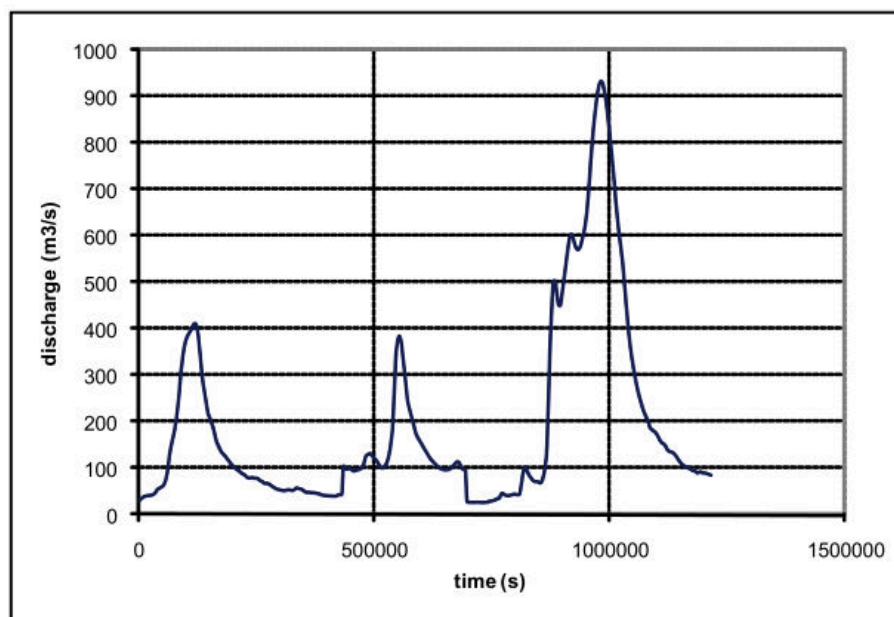
The zip file “Thur-02-flood-simulation.zip” contains the files for simulation of the flood. The “ThurInitial.dat” file contains the results of the last time step of the steady flow simulation, for use as a restart for this unsteady flow simulation.

The “ThurHydrograph.txt” 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
}
```



*Figure 5.9 Hydrograph of the flood event of 2005*

The renamed restart file “ThurInitial.dat” shall be input 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
}
```

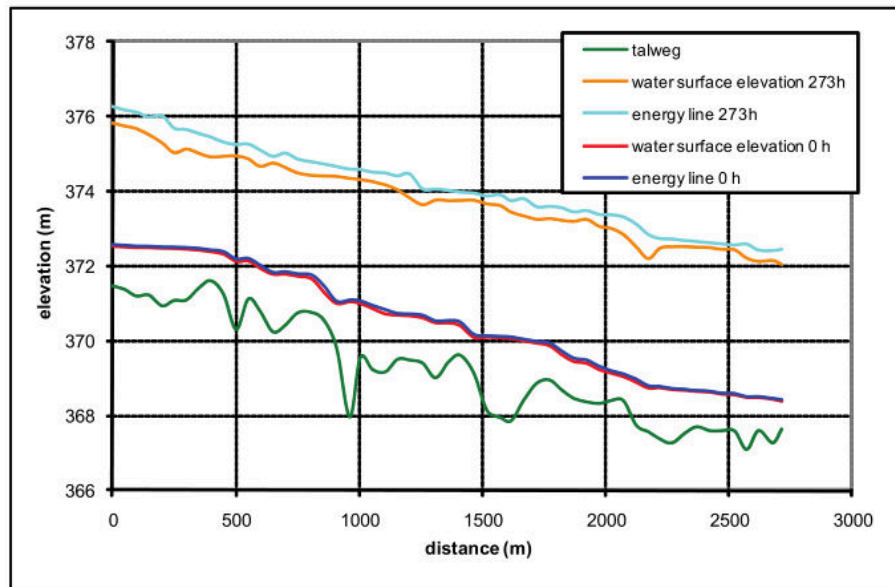


Figure 5.10 Longitudinal profile for maximum discharge

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

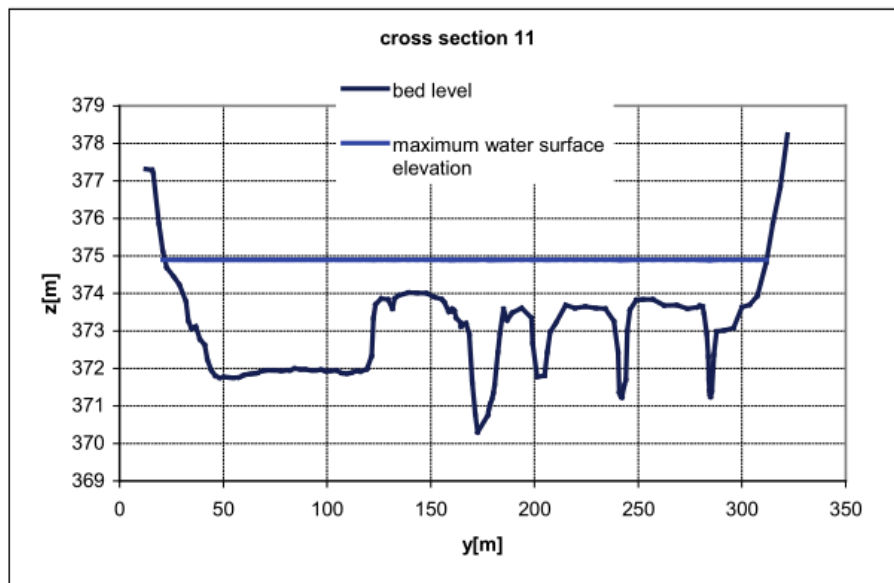
```
OUTPUT
{
  output_time_step = 1000
  console_time_step = 1000
}
```

Now we can run the simulation by writing the *setup.h5* file, and then clicking on the Run button in the simulation tab. 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<sup>3</sup>/s) at ca. 982.000 s (= 273 hours) and plot them over the “distance\_[m]” (see Figure 5.10).

Then we plot some interesting cross sections with their water surface elevation, for the same time (see Figure 5.11). 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.

### 5.1.5 Edit the model file for bed load transport

The zip file “Thur-03-bed-load-simulation.zip” contains the files for simulation of the flood with bed load transport. Information about bed load is grouped in the block MORPHOLOGY. In the block BASECHAIN-1D, chose MORPHOLOGY and press Add Block. Go in the block MORPHOLOGY.



*Figure 5.11 Example of resulting water surface elevation*

#### 5.1.5.1 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
```



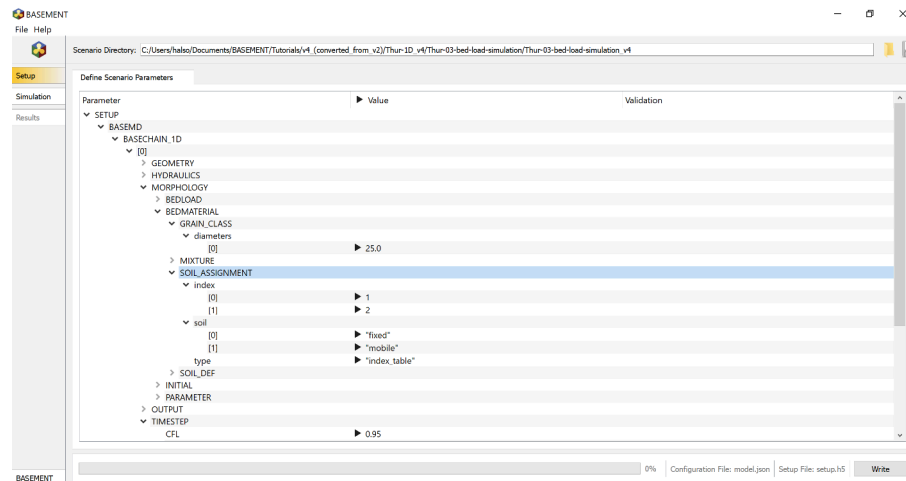


Figure 5.12 Soil assignment in the user interface

```

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

```

### 5.1.5.2 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.

### 5.1.5.3 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
}
```

#### 5.1.5.4 Define specific parameters for bed load transport

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

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

#### 5.1.5.5 Define bed load transport formula

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

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

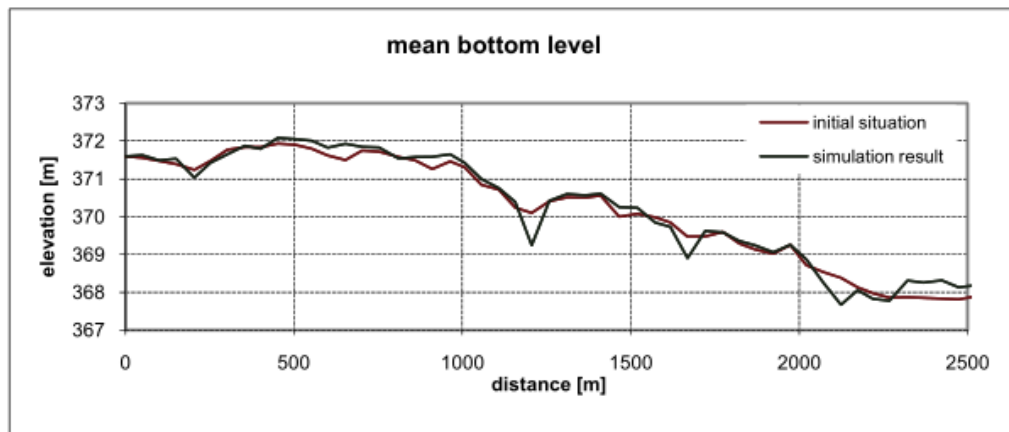
#### 5.1.5.6 Define boundary conditions for bed load

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 5.13* Longitudinal profile of mean bottom level

#### 5.1.5.7 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 )
  }
}
```

#### 5.1.6 Perform bed load simulation (Thur 3)

Run the simulation of the flood with bedload transport. 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 5.13).

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 5.14).

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.

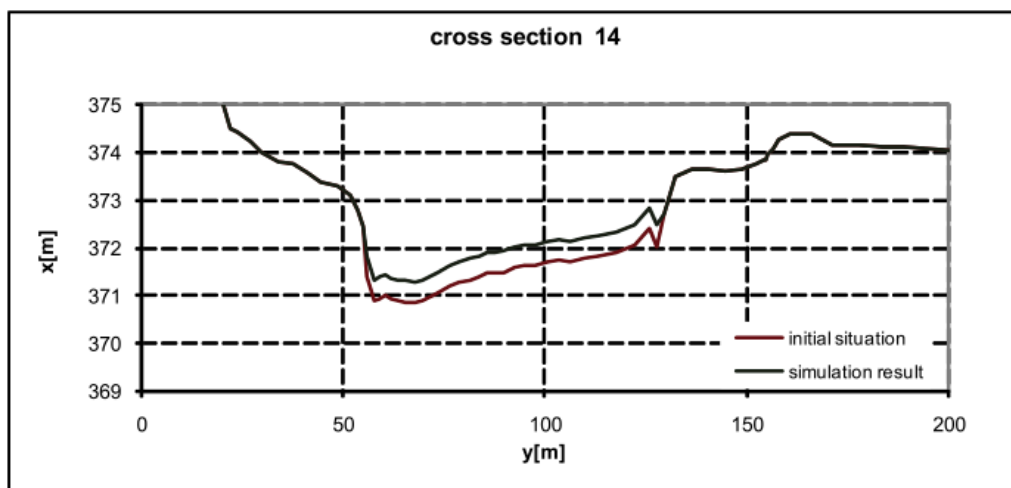


Figure 5.14 Transversal cross section profile

# 6

---

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

### 6.1 Dynamics of longitudinal bed profile due to local river widening with BASEMD (1D)

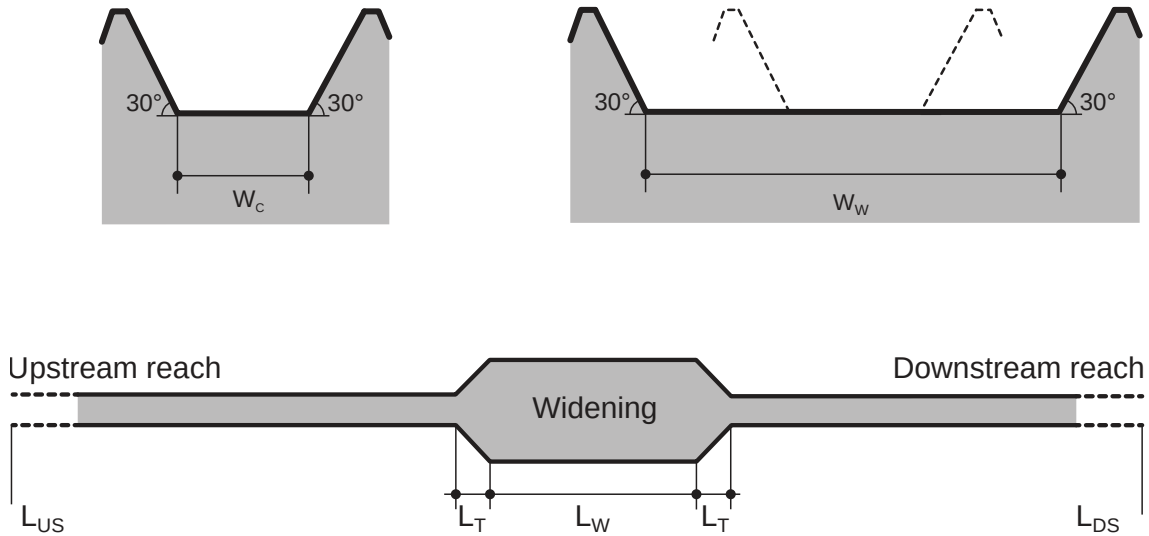
#### 6.1.1 Introduction

In this tutorial the capabilities of *BASEchain* in modelling the evolution of the longitudinal bed profile due to a local symmetric widening is shown and discussed. The general concepts of the 1D simulation with *BASEMENT* and how to set up the configuration files are demonstrated in the previous tutorial ‘Hydrodynamics and sediment transport at the river Thur’ (Section 5.1).

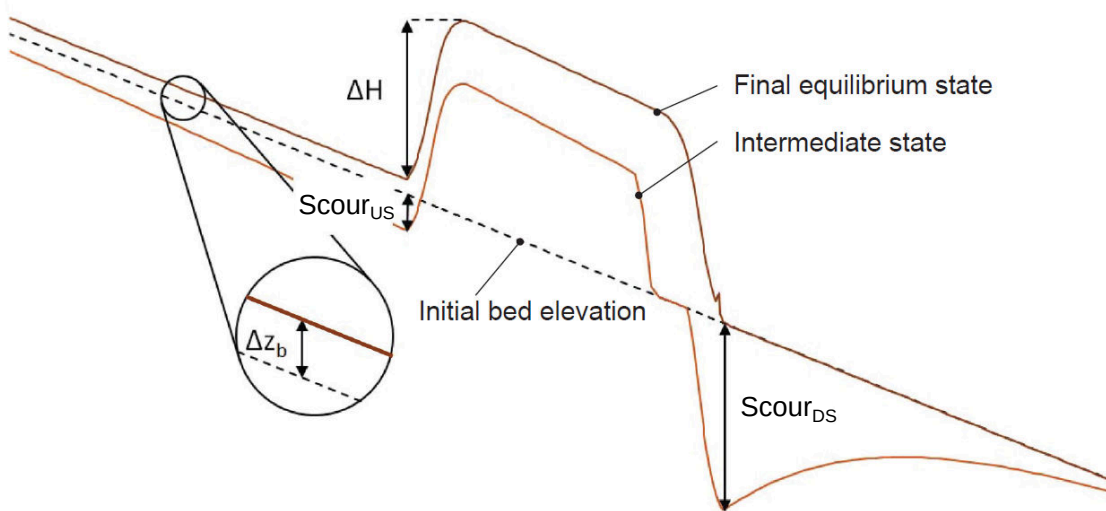
##### 6.1.1.1 General description

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 6.1): 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 6.2). 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



**Figure 6.1** Sketch of symmetric widening used for the numerical simulations in this tutorial.



**Figure 6.2** Temporal evolution of the longitudinal bed profile due to a local river widening.

the widening. This deposition ( $\Delta H$ ) 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 capacity to convey the entire bedload discharge downstream. The widening may induce an uplift ( $\Delta z_b$ ) 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.

### 6.1.1.2 Purpose

Based on the general conditions of the river widening at the River Thur near Altikon (ZH, Figure 6.3), 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



**Figure 6.3** River widening at the River Thur near Altikon (ZH) in 2014, source: C. Herrmann, BHAteam Ingenieure AG, Frauenfeld

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.

### 6.1.1.3 Used features

*BASEMENT* version 4.0 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 6.1.

**Table 6.1** *BASEMENT* parameters used for this tutorial

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

#### 6.1.1.4 Parameter variation

Starting from the default configuration of the reference geometry, the parameters are altered according to Table 6.2. The input files and *BASEMENT* command files (\*.bmc) for each run are provided and can be identified by their run-number.

**Table 6.2** Summary of the simulation runs for different parameter combinations

Parameter	run	$W_W$ [m]	$L_W$ [m <sup>2</sup> ]	$Q$ [m <sup>3</sup> /s]	$d$ [mm]
	default	100	500	800	20
Width variation	1	150	500	800	20
	2	200	500	800	20
Length 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

### 6.1.2 Model setup

#### 6.1.2.1 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 6.4). 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 6.3:

**Table 6.3** 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



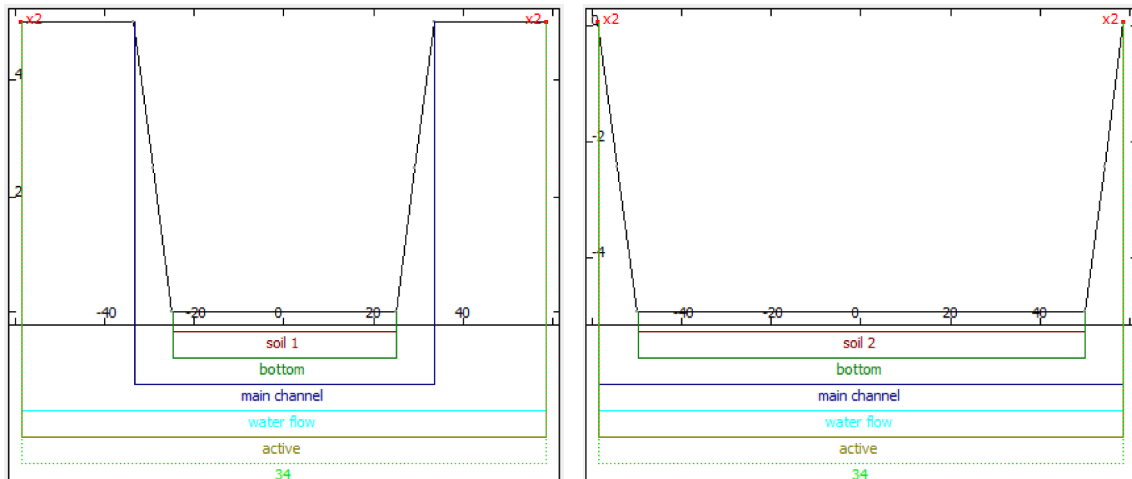
Type	Symbol	Units	Description	Default value	Variation
	$L_{DS}$	m	Length downstream channel	3000	
	$W_C$	m	Width channel	50	
	$W_W$	m	Width widening	100	100-200
	$S_C$	%	Channel slope	0.15	
	$S_W$	%	Initial slope widening	0.15	
Additional variables	$Q$	m <sup>3</sup> /s	Steady discharge	800	200-1600
	$Q_S$	m <sup>3</sup> /s	Steady sediment discharge	0.178	$f(Q, d)$
	$d$	mm	Grain diameter (single grain)	20	5-40
	$k_{St,C}$	m <sup>1/3</sup> /s	Strickler roughness channel	34	
	$k_{St,W}$	m <sup>1/3</sup> /s	Strickler roughness widening	34	

### 6.1.2.2 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 6.4) 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 6.4** Upstream sediment boundary inflow  $Q_S$  defined for different stationary discharges  $Q$  and mean grain size diameters  $d$ .

	$Q$ [m <sup>3</sup> /s]	$d$ [mm]	$Q_S$ [m <sup>3</sup> /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 6.4** Cross sections of the BASEMENT topography representing the channel (left) and the widening (right) to illustrate the geometry and the bottom/soil definition.

### 6.1.3 Results of numerical simulations

#### 6.1.3.1 Temporal evolution of longitudinal bed profile

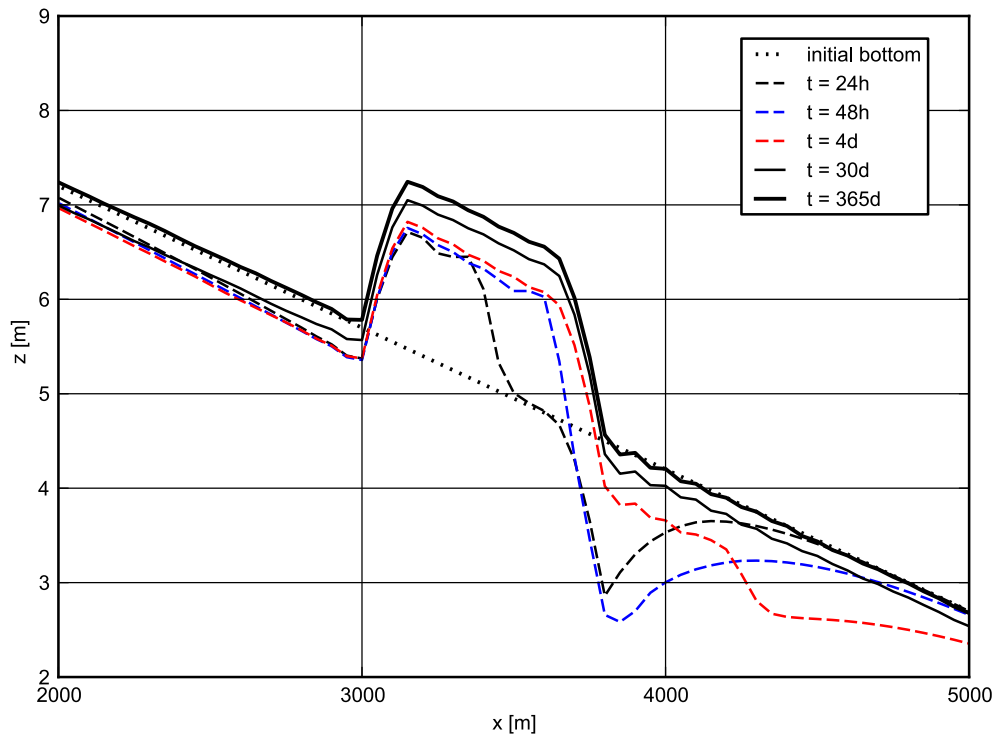
The basic morphological processes described in the introduction can be reproduced generally well with *BASEMENT*. The results with default configuration are shown in Figure 6.5. Steady state river bed conditions are reached after a total simulation runtime of about one year.

#### 6.1.3.2 Width and length of the widening

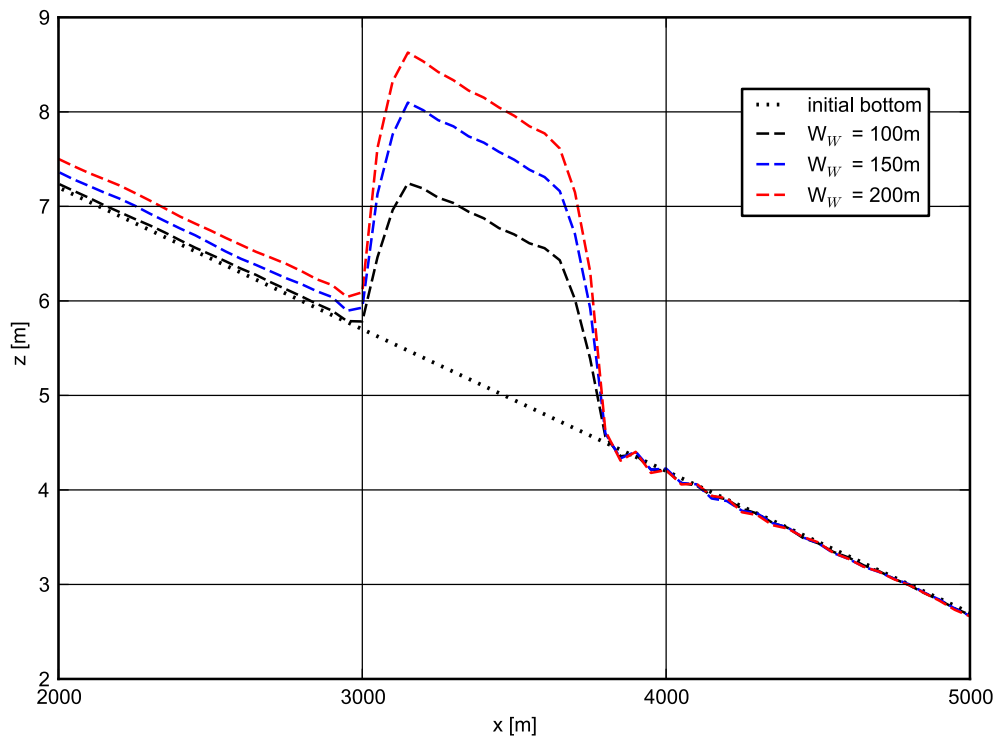
The effect of the width and the length of the widening (geometric parameters  $W_W$  and  $L_W$ ) on the longitudinal bed profile and equilibrium slope is shown in Figure 6.6 and Figure 6.7. The corresponding bottom offset  $\Delta H$  and upstream bed level change  $\Delta z_b$  is shown in Figure 6.8. For increasing width,  $\Delta H$  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 6.6). This causes a bed level change (aggradation) of 0.3 m upstream of the widening. The effect of the river widening length on  $\Delta H$  and  $\Delta z_b$  is rather small as the slope in the widening remains almost constant.  $\Delta z_b$  increases slightly upstream with increasing length of the widening.

#### 6.1.3.3 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 6.9 and Figure 6.10. The corresponding bottom offset  $\Delta H$  and upstream bed level change  $\Delta z_b$  are shown in Figure 6.11. For one geometry but varying discharge, big differences regarding the final topography are observed. For small discharges (200 and 400  $m^3/s$ ),  $\Delta H$  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  $\Delta z_b$  becomes negative. The



**Figure 6.5** Temporal evolution of the mean bottom elevation of the reference geometry with default configuration.



**Figure 6.6** Longitudinal bed profile of the initial and resulting steady state for different widening width  $W_W$ .

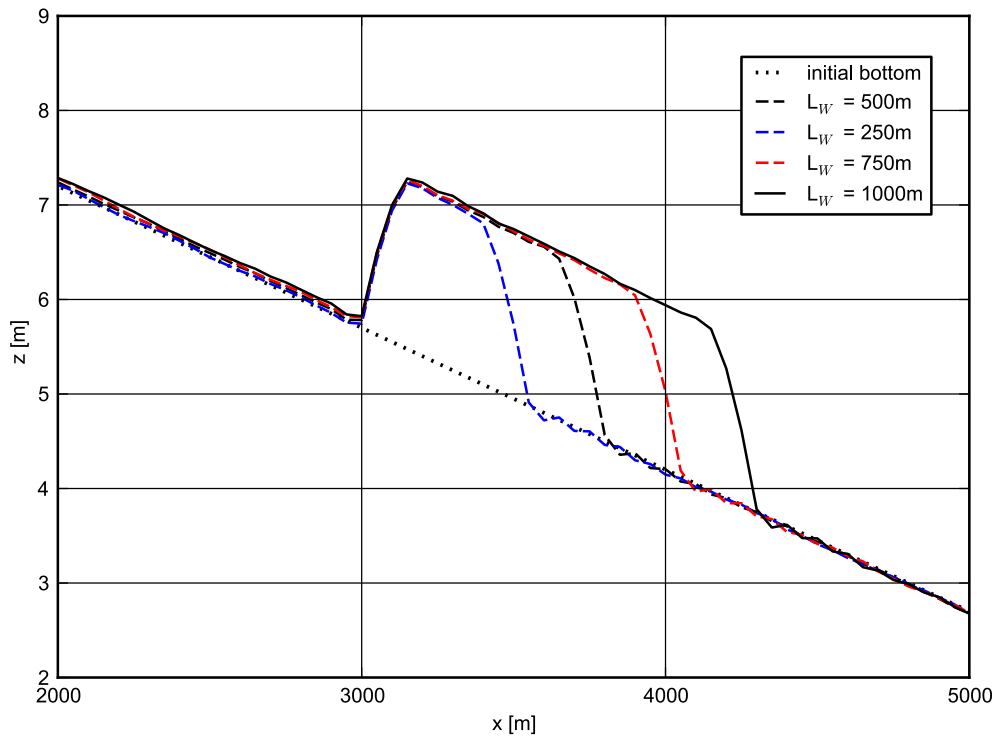


Figure 6.7 Longitudinal bed profile of the initial and resulting steady state for different widening length  $L_W$ .

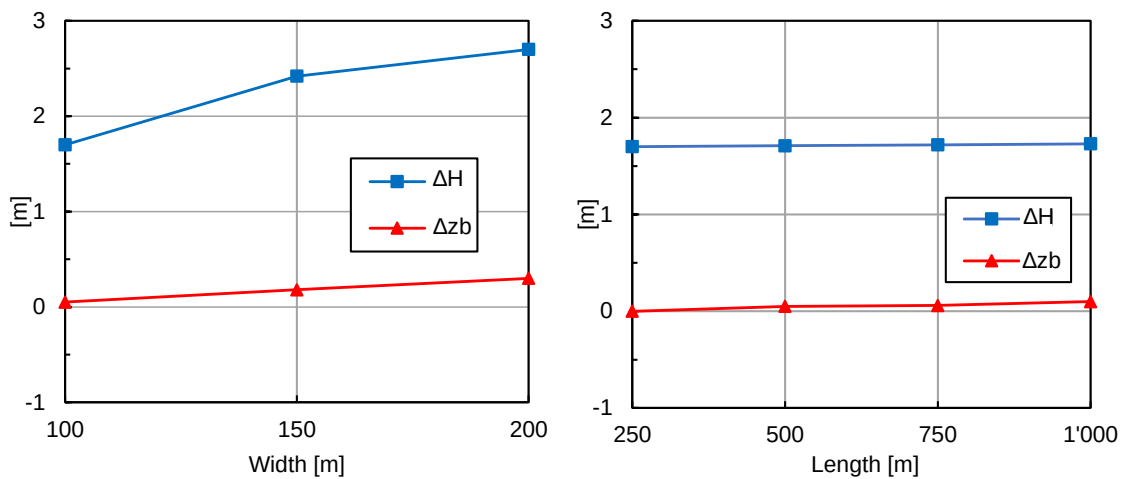
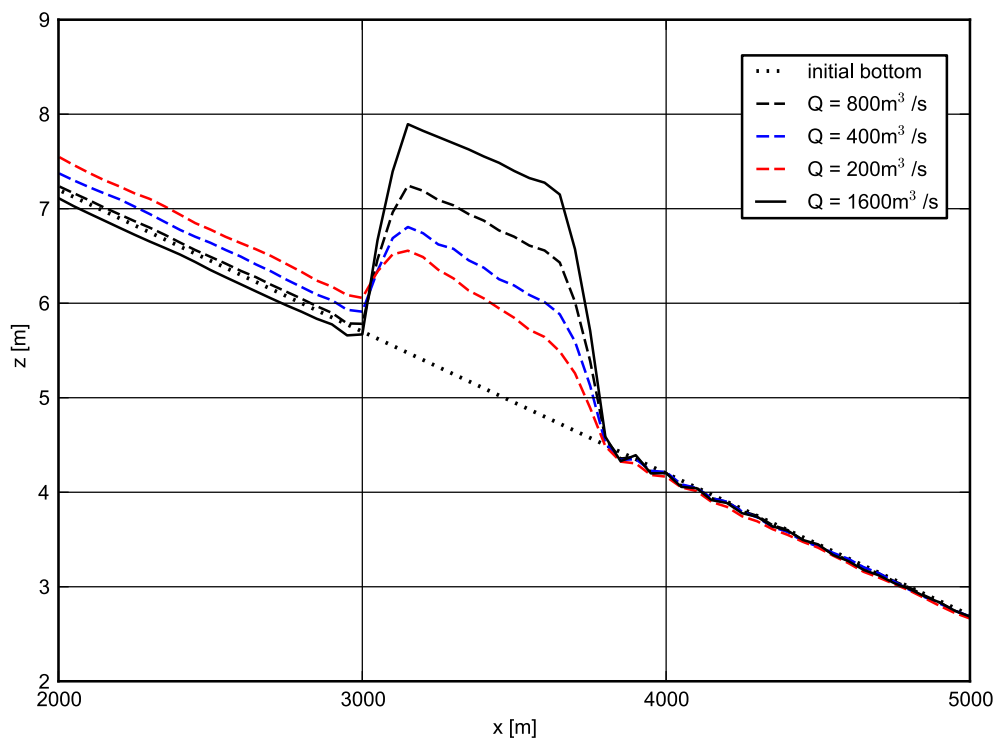


Figure 6.8 Bottom offset  $\Delta H$  and upstream displacement  $\Delta z_b$  for different  $W_W$  (left) and  $L_W$  (right), parameter variation according to table 6.2.



**Figure 6.9** Longitudinal bed profile of the initial and resulting steady state for different discharges  $Q$ .

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.

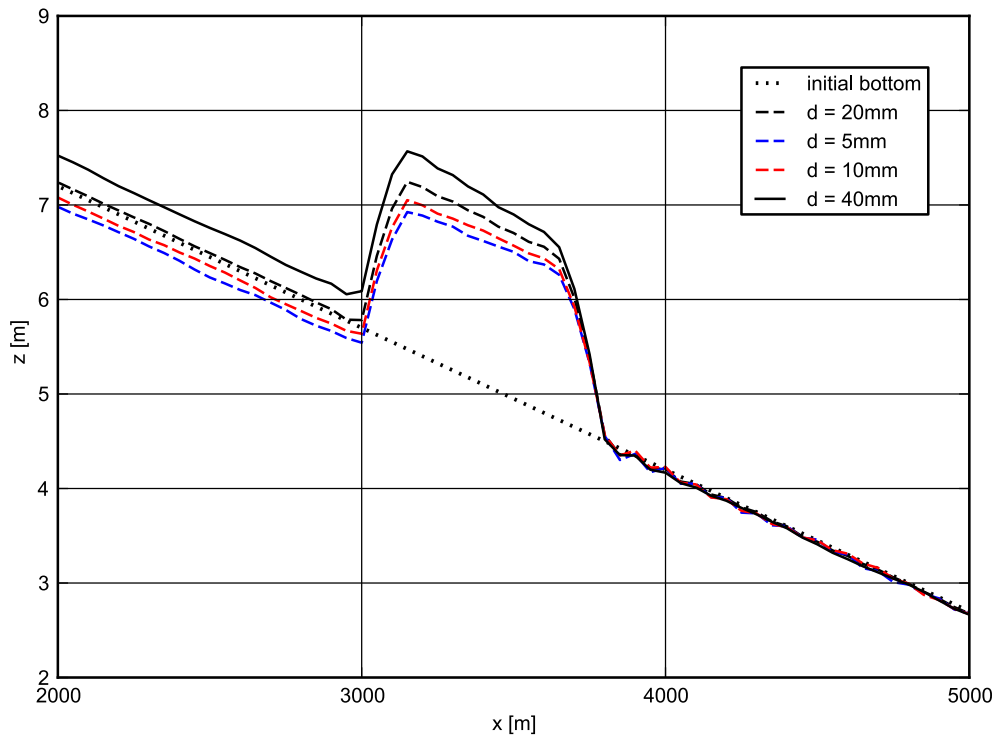


Figure 6.10 Longitudinal bed profile of the initial and resulting steady state for different mean grain size diameters  $d$ .

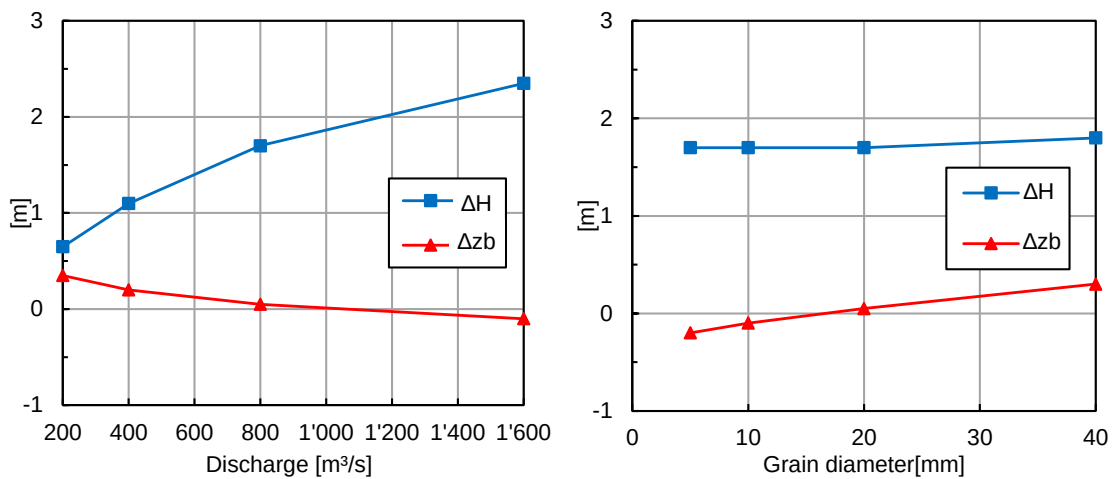


Figure 6.11 Bottom offset  $\Delta H$  and upstream displacement  $\Delta z_b$  for different discharges  $Q$  (left) and mean grain size diameter  $d$  (right) parameter variation according to table 6.2.

---

# Laterally coupled 1D-2D hydrodynamic simulation with BASEMD

## 7.1 Laterally coupled 1D-2D hydrodynamic simulation with BASEMD

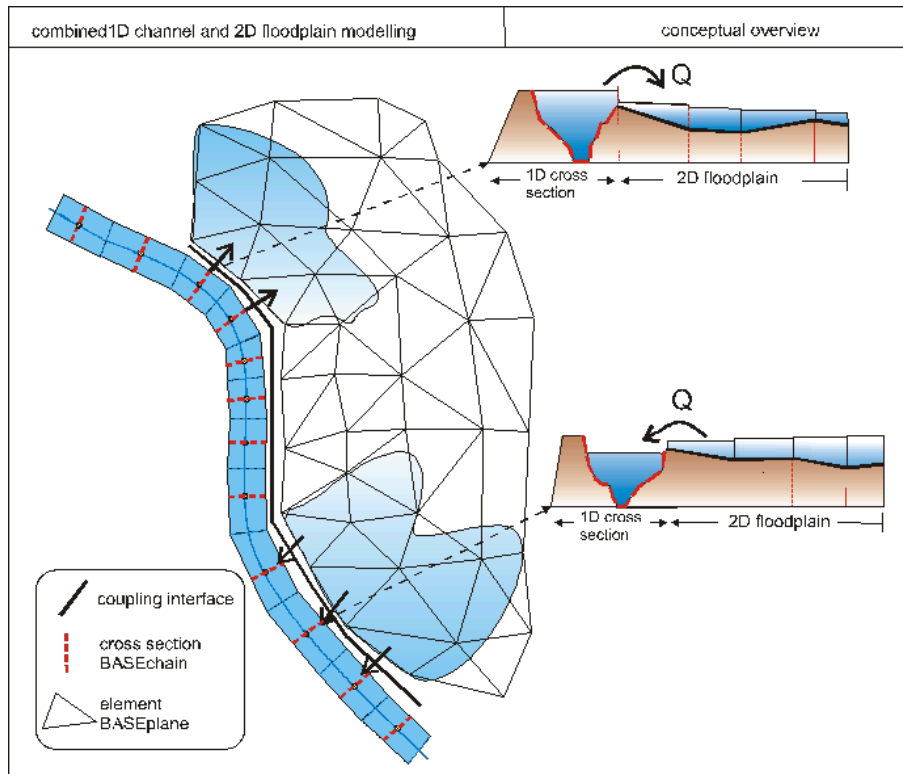
### 7.1.1 Introduction

BASEMENT offers the possibility to combine BASEchain (1D) and BASEplane (2D) domains in coupled simulations. So-called “laterally coupled” simulations simulate the channelized river flow with a 1D approach, while 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 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 as 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 domain.

Water exchange takes place at defined connections between 2D boundary edges and 1D



**Figure 7.1** 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.

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.

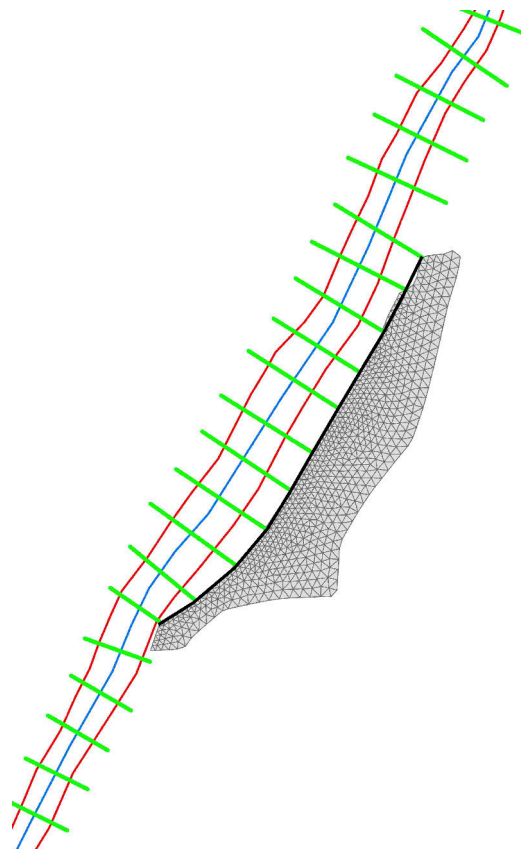
This tutorial briefly introduces the use of this lateral coupling approach based on a simple scenario. Only hydraulic computations are performed, since the lateral coupling supports no morphological computations up-to-now. The tutorial makes use of the QGIS-plugin BASEmesh (see BASEmesh tutorial Section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

## 7.1.2 Set-up of model file

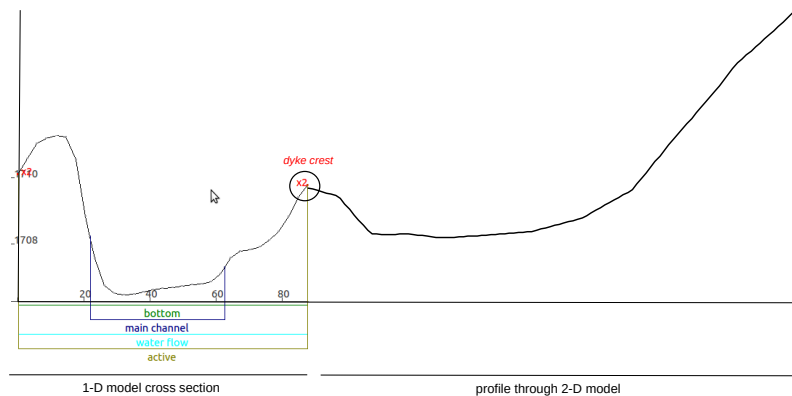
### 7.1.2.1 General remarks on mesh creation

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 7.2.





**Figure 7.2** 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 7.3** Connection of 1D cross section (left) with 2D model (transversal profile, right) at the dyke location. Water overtopping takes place via the coupling connection.

### 7.1.2.2 BASEchain (1D) river model

First, the 1D model of the river is set-up, analogous to the steps of the Thur tutorial (see Section 5.1) outlined above.

```

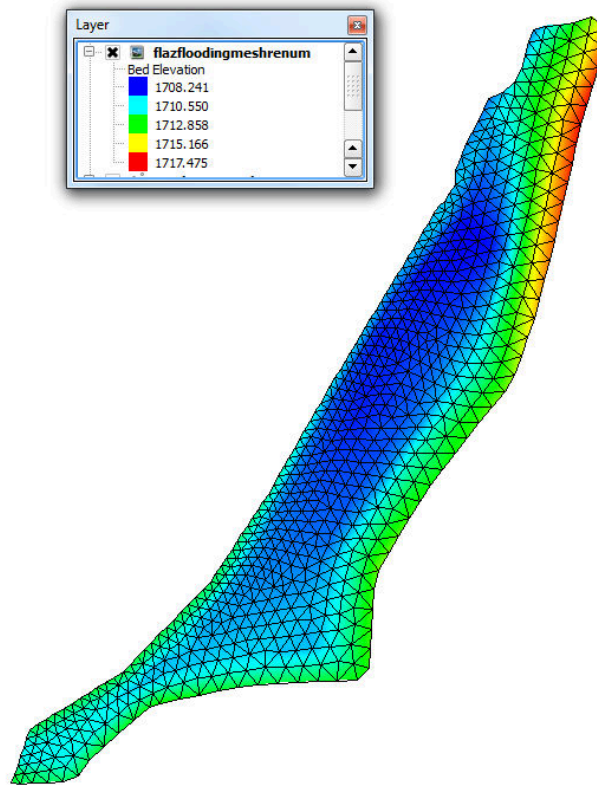
BASECHAIN_1D {
    region_name = river_flaz_1D
    GEOMETRY {
        ...
    }
    TIMESTEP {
        ...
    }
    HYDRAULICS {
        ...
    }
}

```

The topography file (\*.bmg) contains the cross section points, the main channel definitions, and the active range definitions. This data 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 7.3 shows the definition of the right dyke in the 1D cross-section and the connection to the 2D mesh (shown as transversal profile).

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

In this scenario, a steady-state inflow of  $700 \text{ m}^3/\text{s}$  is defined for the 1D model using a hydrograph inflow boundary. This discharge is large enough to overtop the dyke and to flood the 2D floodplain. The outflow boundary is modelled as a 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.



**Figure 7.4** Unstructured 2D mesh of the floodplain located to the right of the 1D river channel.

### 7.1.2.3 BASEplane (2D) floodplain model

The floodplain outside of the main river channel is modelled with a 2D BASEplane sub-domain. Figure 7.4 illustrates the corresponding 2D mesh.

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 {
    ...
  }
}

```

#### 7.1.2.4 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
    c_m = 0.6
    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 {#sec:afmodelcoupling} (Section 8.2) 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 7.1.3). 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.

### 7.1.3 Coupling connections between 1D and 2D subdomains

#### 7.1.3.1 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.

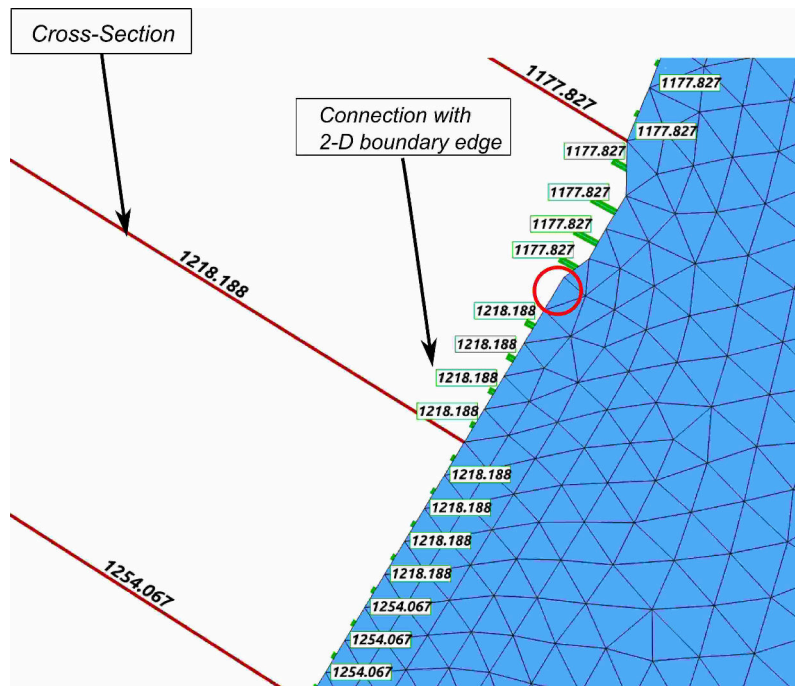
### 7.1.3.2 Automatic generation of coupling connections

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.

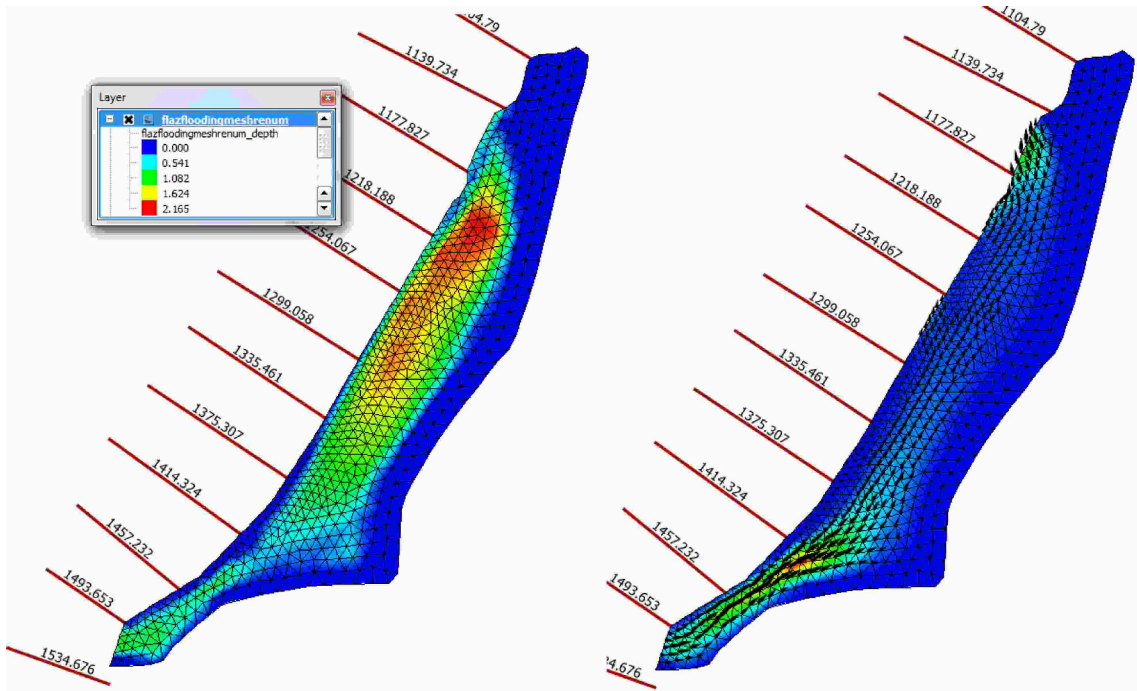
The general workflow of the automatic connection detection is as follows:



**Figure 7.5** 1D cross sections (red) connected to 2D boundary edges (blue mesh) via coupling connections (green). The labels show the cross-section names and indicate the corresponding mapping of the 2D edge  $\rightarrow$  1D cross-section. The red circle indicates a 2D boundary edge without a coupling connection, which is not taking part in the lateral coupling.

1. Open the model file in BASEMENT and set the tag: `connections_automatic = yes`.
2. Write the `setup.h5` file. On the simulation tab, Run BASEMENT. The program will automatically search for the connections and write the detected connections to file and then terminates. The simulation will then abort with the error `ERROR -> Please check the created lateral connections in the file 'lateral_connections_created.dat'` and restart with `'connections_automatic = no'` using the created file. Note: Automatic creation of lateral connections may have failed. This error is normal. Three files are written related to generation of the connections:
  - `lateral_connections_created.dat` = the coupling connection file,
  - `lateral_connections_stats.shp` = shapefile for visualizing the generated connections using GIS-software,
  - `lateral_connections_stats.txt` = tab-delimited text file for visualizing the generated connections with other software.
3. In the model file, set the tag: `connections_automatic = no`. Set the tag: `connections_filename = yes`.

The connections are shown as green lines in Figure 7.5 and are drawn from the middle of each 2D boundary edge orthogonal to the imaginary line between the dykes of two neighbored 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).



**Figure 7.6** 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.

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
```

#### 7.1.4 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 model 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 7.6. 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.





---

# Vegetation dynamics and bedload transport with BASEHPC

## 8.1 Vegetation dynamics and bedload transport with BASEHPC

### 8.1.1 Introduction

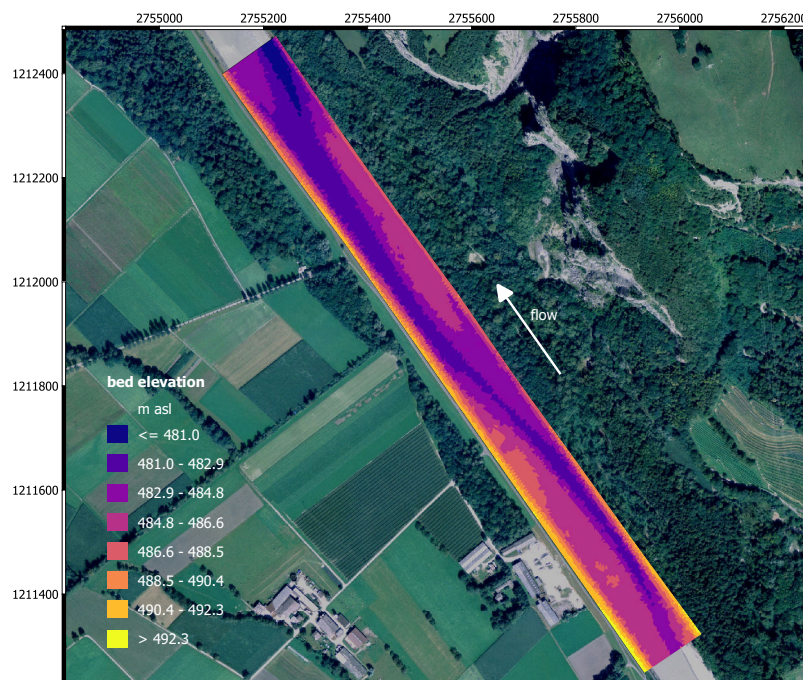
The aim of this tutorial is to show the specific setup of the configuration files for numerical simulations with BASEMENT that include vegetation and bed load transport. For this purpose, the model is applied to a 1 km-long reach of the Alpine Rhine River located at the boundary between Switzerland and Lichtenstein. This particular river reach is characterized by an alternating sequence of gravel bars, which are hotspot for riparian vegetation growth. This tutorial presents the simulation of the flood event of 2005.

### 8.1.2 Computational Mesh

The computational mesh of the Alpine Rhine river was created ad-hoc to show the main model functionalities with the vegetation module activated and for application with BASEMENT version 3. The mesh has 9799 number of cells with 5149 vertices. The MATID was defined on the gravel bars, the main channel, and the banks to differentiate the bed properties. Figure 8.1 shows the bed elevation of the river used in the simulation.

### 8.1.3 Morphology with Vegetation

The simulation in the tutorial includes a steady-state hydrodynamic simulation and an unsteady morphodynamic simulation with vegetation. The first one is used to get the initial conditions for the hydraulic variables. The unsteady simulation



*Figure 8.1 Planar view of the initial bottom elevation of the Alpine Rhine river*

The setup of the configuration files follows the same procedure seen in the Flaz tutorial for hydraulics and morphology (Section 3.1.3.1 and Section 3.1.5.1). Here only the vegetation part is shown in detail.

### 8.1.3.1 Setup the Configuration File model.json

The model is set up with the blocks HYDRAULICS, MORPHOLOGY, and VEGETATION activated. The structure of the VEGETATION block in the file model.json is:

```
{
  "SETUP":{
    "simulation_name":"tutorial_morpho_veg_flood",
    "BASEHPC": {
      "PHYSICAL_PROPERTIES": {...},
      "BASEPLANE_2D": {
        "GEOMETRY": {...},
        "HYDRAULICS": {...},
        "MORPHOLOGY": {...},
        "VEGETATION": {
          "INITIAL": {},
          "PARAMETERS": {}
        }
      }
    }
  }
}
```

The blocks INITIAL and PARAMETERS are both mandatory. The initial conditions for vegetation can be set up using a restart file from a previous simulation (as for the hydraulics) or with input values for the aboveground, belowground biomass, and the rooting depth. In the latter case, the structure is:

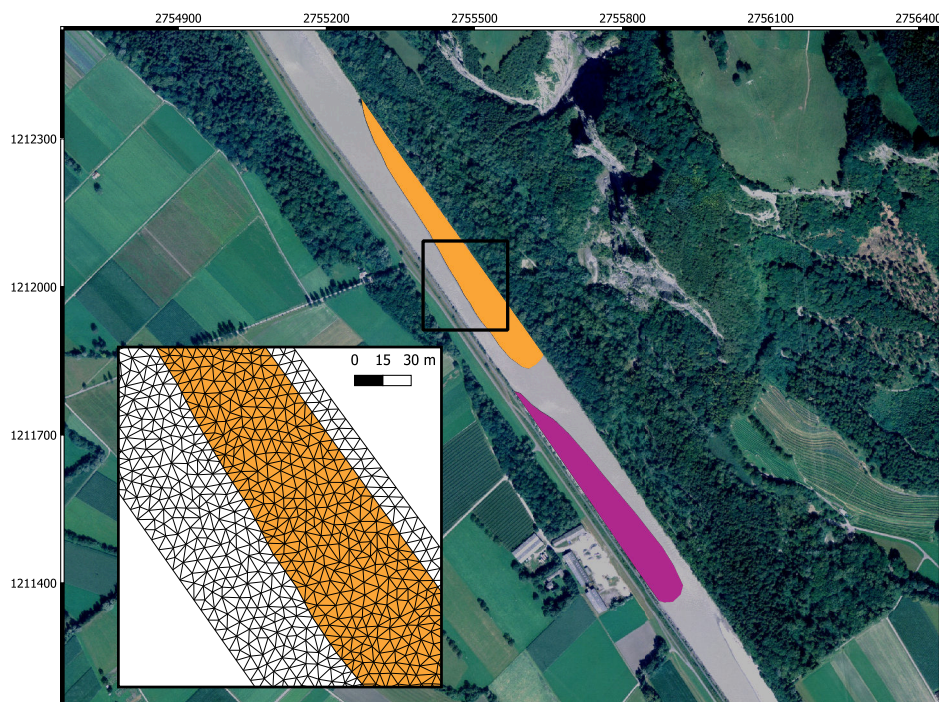
```
"INITIAL": {
  "type": "region_defined",
  "regions": [
    {
      "aboveground_biomass": 0.4,
      "belowground_biomass": 0.6,
      "rooting_depth": 1.5,
      "region_name": "bar_up"
    },
    {
      "aboveground_biomass": 0.8,
      "belowground_biomass": 0.2,
      "rooting_depth": 0.5,
      "region_name": "bar_dw"
    },
    {
      "aboveground_biomass": 0.0,
      "belowground_biomass": 0.0,
      "rooting_depth": 0.0,
      "region_name": "channel"
    },
    ...
  ]
}
```

In this tutorial the vegetation was placed only in the gravel bars and using different characteristics for each of the bar (Figure 8.2). It is important that the sum of the aboveground and belowground biomass equal 1 at maximum. The biomass values are dimensionless, while the rooting depth is expressed in meter.

A number of parameters are needed to run simulation with vegetation. These are all collected in the block PARAMETERS, which has the following elements:

```
"PARAMETER": {
  "burial_factor": 1,
  "plant_height_exp": 1,
  "plant_height_fact": 1,
  "uprooting_factor": 1,
  "veg_strickler_fact": 15,
  "veg_theta_critical_fact": 0.02
}
```

The parameters that are mandatory are the Strickler coefficient, the critical Shield parameter valid for completely vegetated bed, and the parameters for calculating the plant height



*Figure 8.2 Initial location of vegetation on the two bars and an extract of the computational mesh used.*

starting from the aboveground biomass (“plant\_height\_exp” and “plant\_height\_fact”). The others are by default set to 1. It is important to note that vegetation should be used in combination with the “MPM”, “MPM\_like”, or “MPM\_multi” formula for bedload transport and a “strickler” type friction closure for the hydrodynamics.

### 8.1.3.2 Setup the Configuration File simulation.json

The configuration file simulation.json defines the simulation time parameters (seconds) in the block TIME and the different output types inside the OUTPUT block.

```
{
  "SIMULATION": {
    "OUTPUT": [
      "bottom_elevation",
      "flow_velocity",
      "friction_chezy",
      "water_surface",
      "delta_z",
      "water_depth",
      "theta_critical",
      "trsp_capacity",
      "aboveground_biomass",
      "belowground_biomass",
      "rooting_depth"
    ],
  },
}
```

```
"TIME": {
  "end": 111600.0,
  "out": 11160.0,
  "start": 0.0
},
"TIMESTEP": {
  "init": 0.02,
  "minimum": 0.0001
}
}
```

There are three additional available outputs for vegetation: the aboveground biomass, the belowground biomass, and the rooting depth.

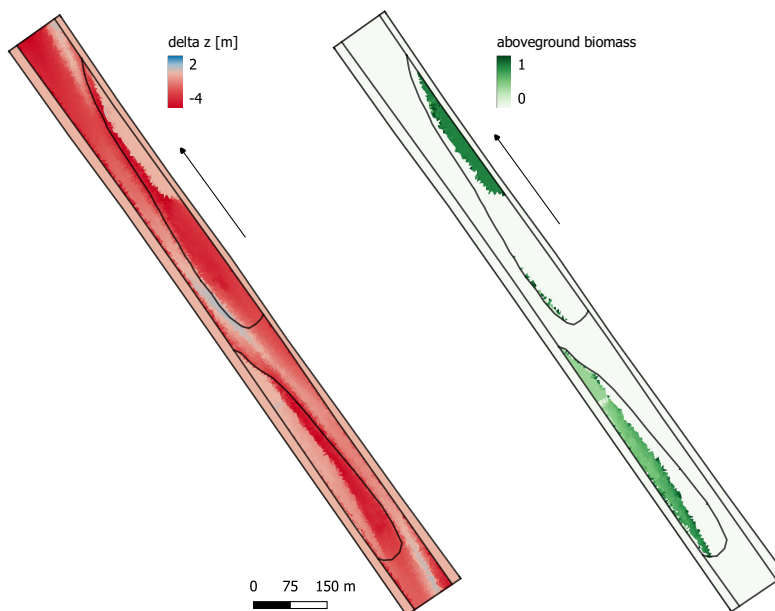
### 8.1.3.3 Set up the Configuration File `results.json`

The configuration file `results.json` defines the output format in the block `EXPORT`. Currently, `xmxf` is the only output format available.

```
{
  "RESULTS": {
    "EXPORT": [
      {"format": "xmxf"}
    ]
  }
}
```

### 8.1.3.4 Results

An example of the results of the simulation of the 2005 flood in the Alpine Rhine river considering vegetation is shown in Figure 8.3. Vegetation is removed in most parts of the bars as a result of uprooting, while the aboveground biomass of surviving vegetation changed because of erosion and deposition processes, which modify biomass redistribution.



**Figure 8.3** Results for the bed elevation and the aboveground biomass after the flood (at  $t=111600$  s). Breaklines are highlighted in black.

# 9

---

## References







**BASIC SIMULATION ENVIRONMENT  
FOR MODELLING OF ENVIRONMENTAL  
FLOWS AND NATURAL HAZARDS**

---

# **TEST CASES**

**VERSION 4.1.0  
JUNE 2024**



**BASEMENT**



# Contents

<b>1</b>	<b>Hydrodynamics</b>	<b>5</b>
1.1	Introduction . . . . .	5
1.2	Common Test Cases . . . . .	5
1.2.1	H_1: Dam break in a closed channel . . . . .	5
1.2.1.1	Intention . . . . .	5
1.2.1.2	Description . . . . .	5
1.2.1.3	Geometry and Initial Conditions . . . . .	6
1.2.1.4	Boundary Conditions . . . . .	6
1.2.2	H_2: One dimensional dam break on planar bed . . . . .	6
1.2.2.1	Intention . . . . .	6
1.2.2.2	Description . . . . .	7
1.2.2.3	Geometry and Initial Conditions . . . . .	7
1.2.2.4	Boundary Conditions . . . . .	7
1.2.3	H_3: One dimensional dam break on sloped bed . . . . .	7
1.2.3.1	Intention . . . . .	7
1.2.3.2	Description . . . . .	8
1.2.3.3	Geometry and Initial Conditions . . . . .	8
1.2.3.4	Boundary Conditions . . . . .	8
1.2.4	H_4 : Parallel execution . . . . .	8
1.2.4.1	Intention . . . . .	8
1.2.4.2	Description . . . . .	8
1.2.4.3	Geometry, initial conditions and boundary conditions . . . . .	9
1.2.5	H_5: Controlled Boundary Condition . . . . .	9
1.2.5.1	Intention . . . . .	9
1.2.5.2	Description . . . . .	9
1.3	BASEchain Specific Test Cases . . . . .	10
1.3.1	H_BC_1: Fluid at rest in a closed channel with strongly varying geometry . . . . .	10
1.3.1.1	Intention . . . . .	10
1.3.1.2	Description . . . . .	10
1.3.1.3	Geometry and Initial Conditions . . . . .	10
1.3.1.4	Boundary Conditions . . . . .	10
1.3.2	H_BC_2: bed load simulation with implicit hydraulic solution . . . . .	10
1.3.2.1	Intention . . . . .	10
1.3.2.2	Description . . . . .	11
1.3.2.3	Geometry and general data . . . . .	11
1.4	BASEplane Specific Test Cases . . . . .	11
1.4.1	H_BP_1: Rest Water in a closed area with strongly varying bottom . . . . .	11
1.4.1.1	Intention . . . . .	11

1.4.1.2	Description . . . . .	11
1.4.1.3	Geometry and Initial Conditions . . . . .	12
1.4.1.4	Boundary Conditions . . . . .	12
1.4.2	H_BP_2: Rest Water in a closed area with partially wet elements .	13
1.4.2.1	Intention . . . . .	13
1.4.2.2	Description . . . . .	13
1.4.2.3	Geometry and Initial Conditions . . . . .	13
1.4.2.4	Boundary Conditions . . . . .	13
1.4.3	H_BP_3: Dam break within strongly bended geometry . . . . .	14
1.4.3.1	Intention . . . . .	14
1.4.3.2	Description . . . . .	14
1.4.3.3	Geometry and Initial Conditions . . . . .	14
1.4.3.4	Boundary Conditions . . . . .	14
1.4.4	H_BP_4: Malpasset dam break . . . . .	15
1.4.4.1	Intention . . . . .	15
1.4.4.2	Description . . . . .	15
1.4.4.3	Geometry and Initial Conditions . . . . .	15
1.4.4.4	Boundary Conditions . . . . .	16
1.4.5	H_BP_5: Circular dam break wave . . . . .	16
1.4.5.1	Intention . . . . .	16
1.4.5.2	Description . . . . .	16
1.4.5.3	Geometry and Initial Conditions . . . . .	16
1.4.5.4	Boundary Conditions . . . . .	17
1.5	Results . . . . .	17
1.5.1	Common Test Cases . . . . .	17
1.5.1.1	H_1: Dam break in a closed channel . . . . .	17
1.5.1.2	H_2: One dimensional dambreak on planar bed . . . . .	17
1.5.1.3	H_3: One dimensional dam break on sloped bed . . . . .	19
1.5.1.4	H_4 : Parallel execution . . . . .	22
1.5.1.5	H_5 : Controlled boundary conditions . . . . .	23
1.5.2	BASEchain Specific Test Cases . . . . .	23
1.5.2.1	H_BC_1: Fluid at rest in a closed channel with strongly varying geometry . . . . .	23
1.5.2.2	H_BC_2: bed load simulation with implicit hydraulic solution . . . . .	25
1.5.3	BASEplane Specific Test Cases . . . . .	25
1.5.3.1	H_BP_1: Rest water in a closed area with strongly varying bottom . . . . .	25
1.5.3.2	H_BP_2: Rest Water in a closed area with partially wet elements . . . . .	25
1.5.3.3	H_BP_3: Dam break within strongly bended geometry . .	25
1.5.3.4	H_BP_4: Malpasset dam break . . . . .	28
1.5.3.5	H_BP_5: Circular dam break . . . . .	28
<b>2</b>	<b>Sediment Transport</b>	<b>35</b>
2.1	Introduction . . . . .	35
2.2	Common Test Cases . . . . .	35
2.2.1	ST_1: Soni et al: Aggradation due to overloading . . . . .	35
2.2.1.1	Intention . . . . .	35

2.2.1.2	Description . . . . .	35
2.2.1.3	Geometry and general data . . . . .	36
2.2.1.4	Equilibrium experiments . . . . .	36
2.2.1.5	Boundary conditions . . . . .	36
2.2.1.6	Aggradation experiments . . . . .	37
2.2.2	ST_2 : Saiedi . . . . .	37
2.2.2.1	Intention . . . . .	37
2.2.2.2	Description . . . . .	37
2.2.2.3	Geometry and general data . . . . .	37
2.2.2.4	Calibration . . . . .	37
2.2.2.5	Aggradation . . . . .	38
2.2.3	BeST_3 : Guenter . . . . .	38
2.2.3.1	Intention . . . . .	38
2.2.3.2	Description . . . . .	38
2.2.3.3	Geometry and general data . . . . .	38
2.2.3.4	Simulation procedure and boundary/initial conditions . . . . .	39
2.3	BASEchain Specific Test Cases . . . . .	39
2.3.1	ST_BC_1: Advection of suspended load . . . . .	39
2.3.1.1	Intention . . . . .	39
2.3.1.2	Description . . . . .	39
2.3.1.3	Geometry and initial conditions . . . . .	39
2.3.1.4	Boundary conditions . . . . .	40
2.3.2	ST_BC_2: Advection-Diffusion . . . . .	41
2.3.2.1	Intention . . . . .	41
2.3.2.2	Description . . . . .	41
2.3.2.3	Geometry and initial conditions . . . . .	41
2.3.2.4	Boundary conditions . . . . .	41
2.4	BASEplane Specific Test Cases . . . . .	42
2.4.1	ST_BP_1: Advection of suspended load . . . . .	42
2.4.1.1	Intention . . . . .	42
2.4.1.2	Description . . . . .	42
2.4.1.3	Geometry and initial conditions . . . . .	42
2.4.1.4	Boundary conditions . . . . .	43
2.5	Results . . . . .	44
2.5.1	Common Test Cases . . . . .	44
2.5.1.1	ST_1: Soni . . . . .	44
2.5.1.2	ST_2: Saiedi . . . . .	45
2.5.1.3	ST_3: Guenter . . . . .	47
2.5.2	BASEchain specific Test Cases . . . . .	50
2.5.2.1	ST_BC_1: Advection of suspended load . . . . .	50
2.5.2.2	ST_BC_2: Advection-Diffusion . . . . .	51
2.5.3	BASEplane specific Test Cases . . . . .	52
2.5.3.1	ST_BP_1: Advection of suspended load . . . . .	52
<b>3</b>	<b>Model Coupling</b> . . . . .	<b>55</b>
3.1	Coupling of Domains of Same Type . . . . .	55
3.1.1	COUPL_1: Sequential two-way coupling with backwater effects . . . . .	55
3.1.1.1	Intention . . . . .	55
3.1.1.2	Description . . . . .	55

3.1.1.3	Geometry and initial conditions . . . . .	55
3.1.2	COUPL_2: River network modelling . . . . .	56
3.1.2.1	Intention . . . . .	56
3.1.2.2	Description . . . . .	56
3.1.2.3	Geometry and initial conditions . . . . .	56
3.1.3	COUPL_3: Sequential coupling with sediment transport . . . . .	57
3.1.3.1	Intention . . . . .	57
3.1.3.2	Description . . . . .	57
3.1.3.3	Geometry and initial conditions . . . . .	57
3.2	Results . . . . .	58
3.2.1	Coupling Test Cases for Domains of Same Type . . . . .	58
3.2.1.1	COUPL_1: Sequential two-way coupling with backwater effects . . . . .	58
3.2.1.2	COUPL_2: River network modelling . . . . .	59
3.2.1.3	COUPL_3: Sequential coupling with sediment transport . . . . .	60
<b>4</b>	<b>Subsurface Flow</b>	<b>61</b>
4.1	BASEsub1: Saturated, confined water flow in soil . . . . .	61
4.2	BASEsub2: Water infiltration into partially-saturated soil . . . . .	61
4.3	BASEsub3: Unsaturated seepage flow . . . . .	62
<b>5</b>	<b>High-Performance Computing</b>	<b>65</b>
5.1	Introduction and Backends . . . . .	65
5.1.1	Introduction . . . . .	65
5.1.2	Backends . . . . .	65
5.2	Circular dam break . . . . .	66
5.2.1	Description . . . . .	66
5.2.2	Geometry and Initial Conditions . . . . .	66
5.2.3	Results . . . . .	67
5.2.4	Performance . . . . .	70
5.2.4.1	Ubuntu . . . . .	70
5.2.4.2	Windows . . . . .	73
5.3	Conical Dune . . . . .	75
5.3.1	Description . . . . .	75
5.3.2	Geometry and Initial Conditions . . . . .	75
5.3.3	Results . . . . .	76
5.4	Dam-break flow in a L-shaped channel with a mobile-bed . . . . .	79
5.4.1	Description . . . . .	79
5.4.2	Geometry and initial conditions . . . . .	79
5.4.3	Results . . . . .	79
5.4.4	Performance . . . . .	81
5.4.4.1	Ubuntu . . . . .	81
5.4.4.2	Windows . . . . .	83
<b>6</b>	<b>References</b>	<b>85</b>

---

## 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).



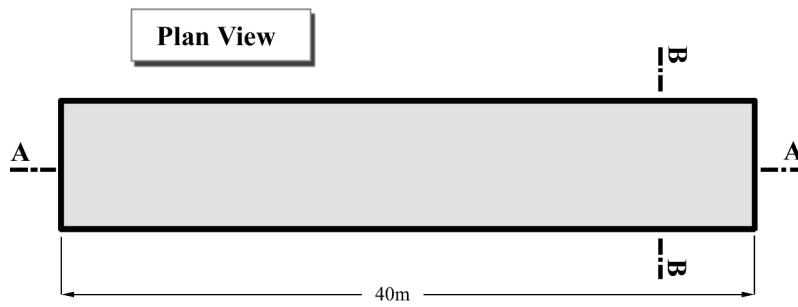


Figure 1.1 H\_1: Dam break in a closed channel, plan view

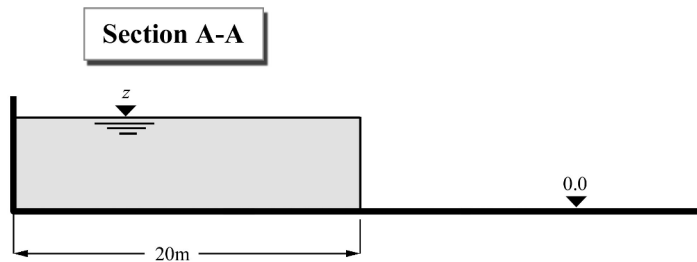


Figure 1.2 H\_1: Dam break in a closed channel, section A-A

### 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).

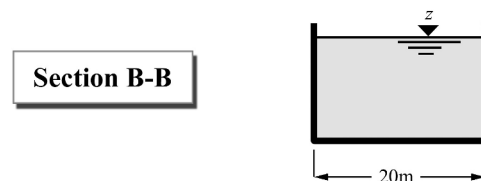
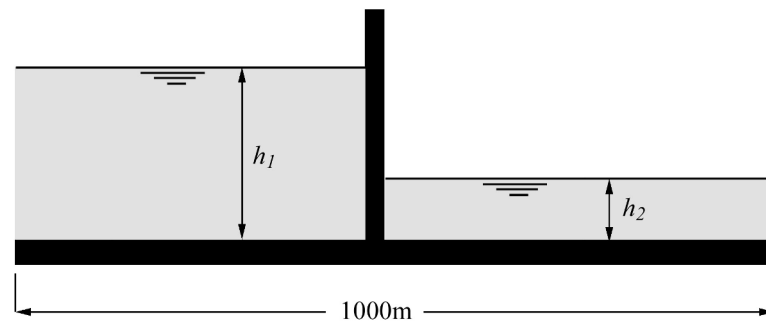


Figure 1.3 H\_1: Dam break in a closed channel, section B-B



**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.

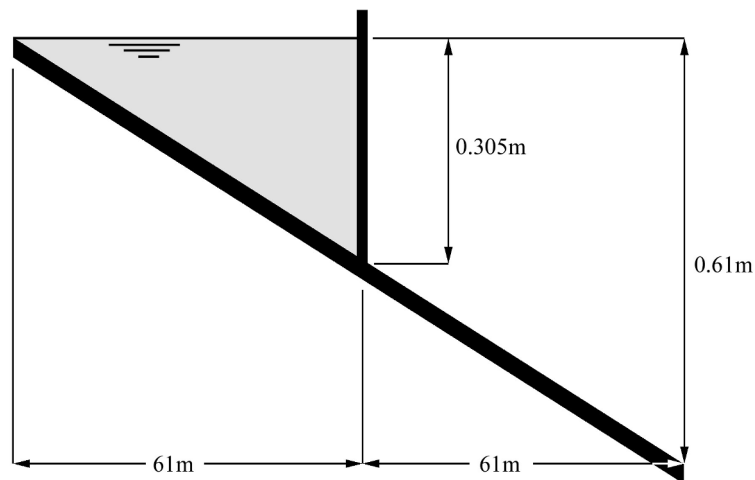
### 1.2.2.4 Boundary Conditions

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.

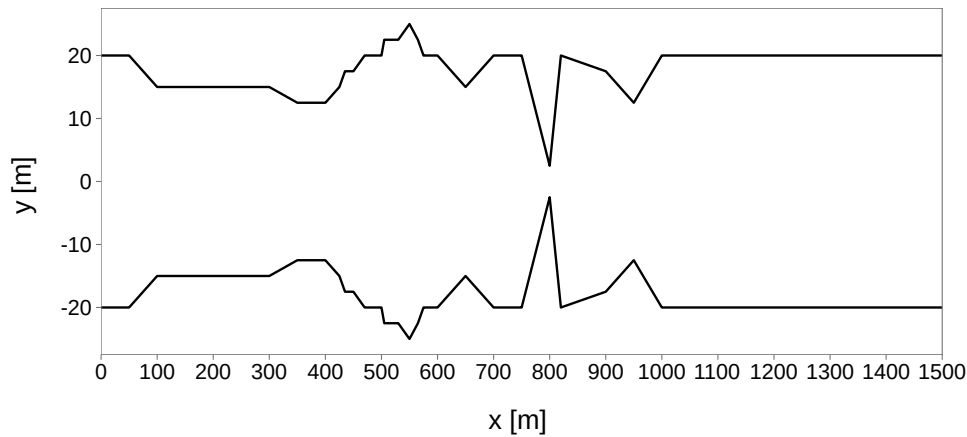
### 1.2.5.2 Description

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<sup>3</sup>/s. Over 5000 seconds it is increased up to 150 m<sup>3</sup>/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<sup>3</sup>/s to 450 m<sup>3</sup>/s and then lowered to 180 m<sup>3</sup>/s before it is again set to 300 m<sup>3</sup>/s. For the 2-D simulation, the target level is not constant, but dependent on the current inflow, which is measured upstream.

Inflow (m <sup>3</sup> /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.

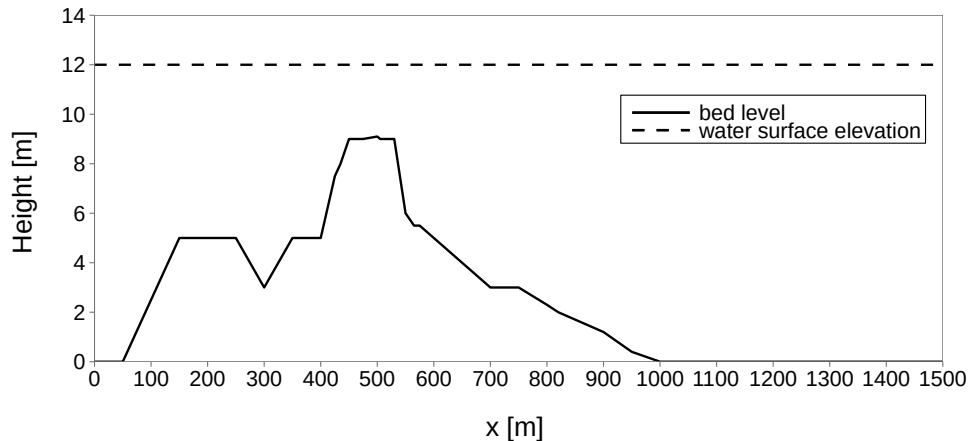
#### 1.3.1.4 Boundary Conditions

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  $0.1 \text{ m}^2$  for the wetted area and  $0.1 \text{ m}^3/\text{s}$  for the discharge.

## 1.4 BASEplane Specific Test Cases

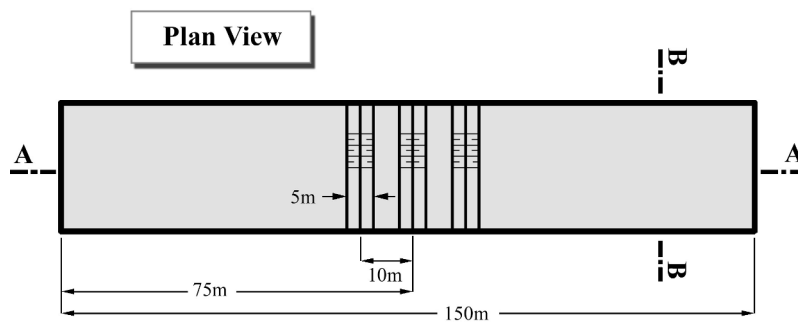
### 1.4.1 H\_BP\_1: Rest Water in a closed area with strongly varying bottom

#### 1.4.1.1 Intention

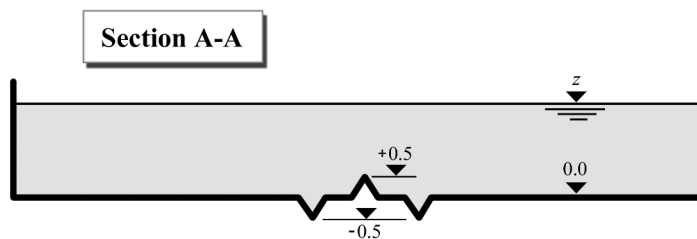
This test in two dimensions checks for the conservation of momentum. Numerical artefacts (mostly due to geometrical reasons) could generate impulse waves although there is no acting force.

#### 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*

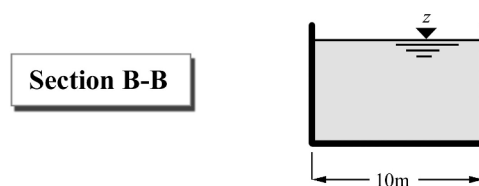
### 1.4.1.3 Geometry and Initial Conditions

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 .

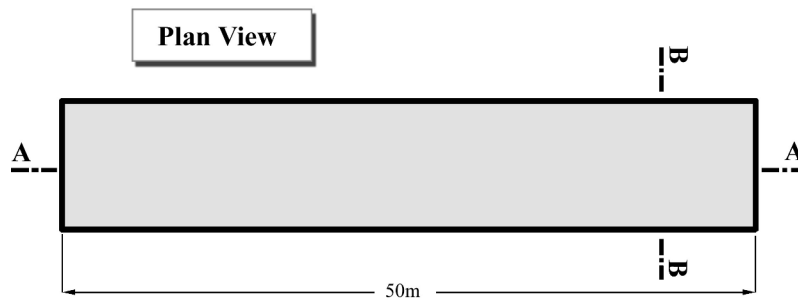
### 1.4.1.4 Boundary Conditions

No inflow and outflow. All of the boundaries are considered as a wall.

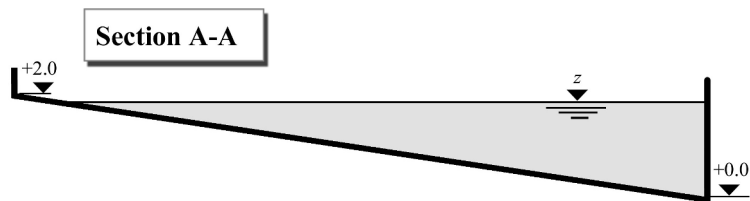
Friction: frictionless test; Manning's factor  $n = 0.0$ .



**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.

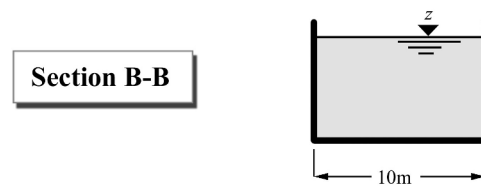
### 1.4.2.4 Boundary Conditions

No inflow and outflow. All of the boundaries are considered as a wall.

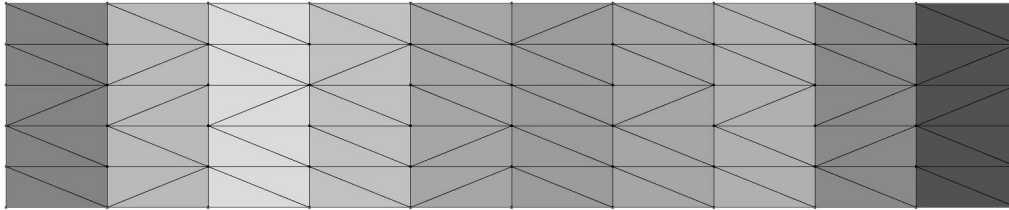
Friction: frictionless test; Manning's factor  $n = 0.0$ .

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.

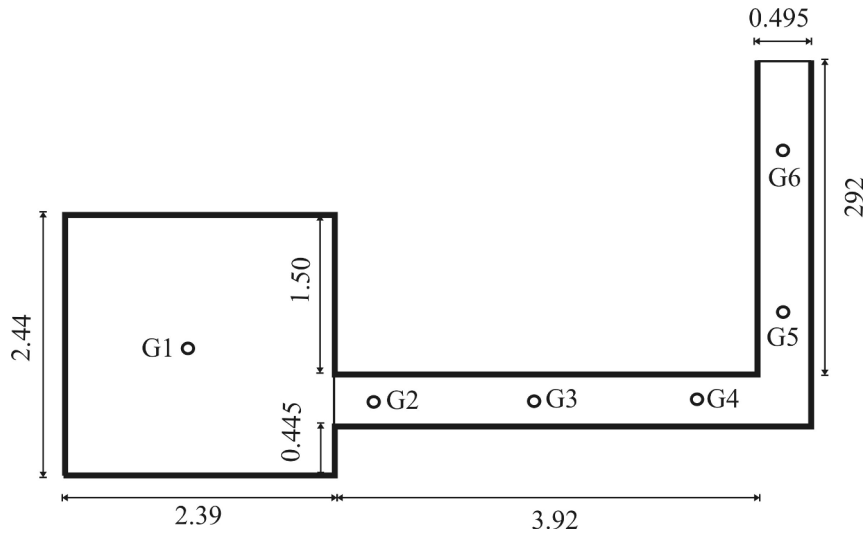
#### 1.4.3.3 Geometry and Initial Conditions

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.

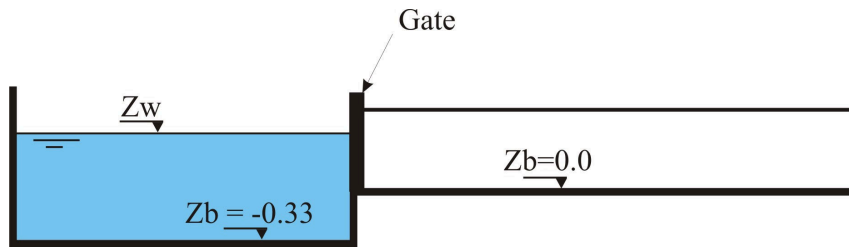
#### 1.4.3.4 Boundary Conditions

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 2<sup>nd</sup>, 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.

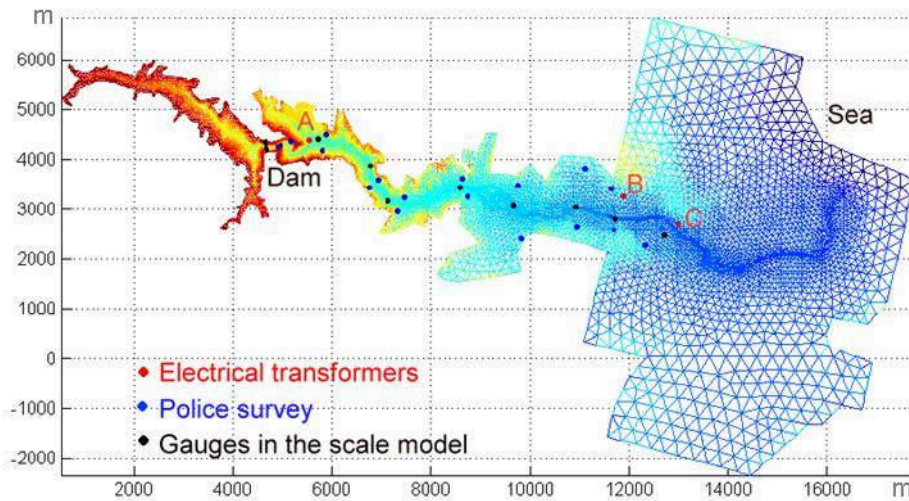


Figure 1.17 H\_BP\_4: Computational area of the malpasset dam break

#### 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

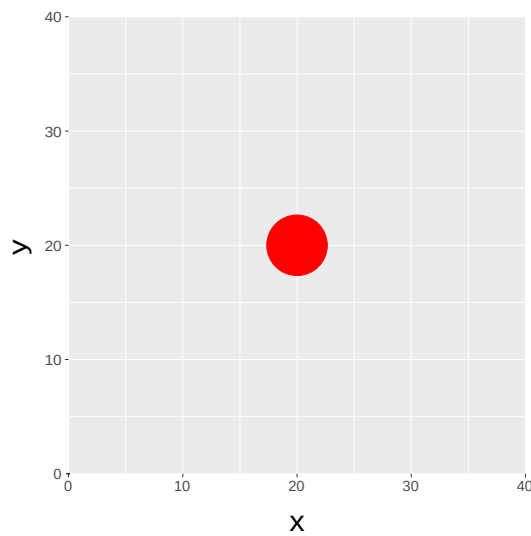
The circular dam break problem is a demanding two-dimensional test case with some distinct features. It is used to evaluate the capability to correctly model complex interactions of shock and rarefaction waves. The results are qualitatively compared to results obtained by Toro (2001), and other numerical 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<sup>3</sup>/s with decreasing tendency. Both, BASEchain and BASEplane deliver similar results.

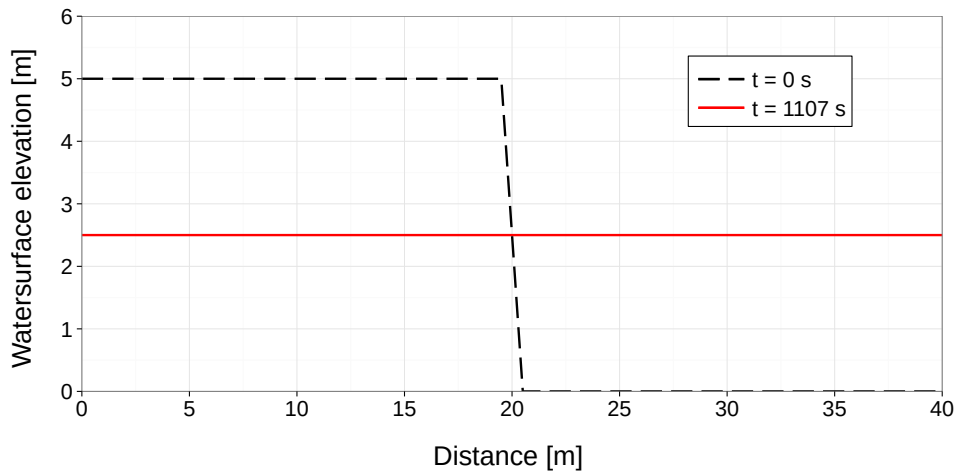
#### 1.5.1.2 H\_2: One dimensional dambreak on planar bed

##### 1.5.1.2.1 Results obtained by BASEchain

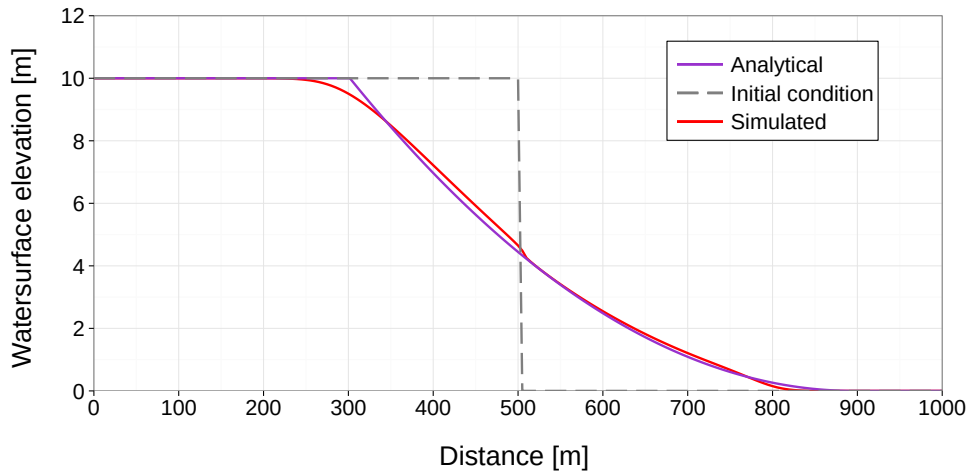
A comparison of the 1-D test case using  $h_2/h_1 = 0.000$  with the analytical solution shows an accurate behaviour of the fluid phase in time.

##### 1.5.1.2.2 Results obtained by BASEplane

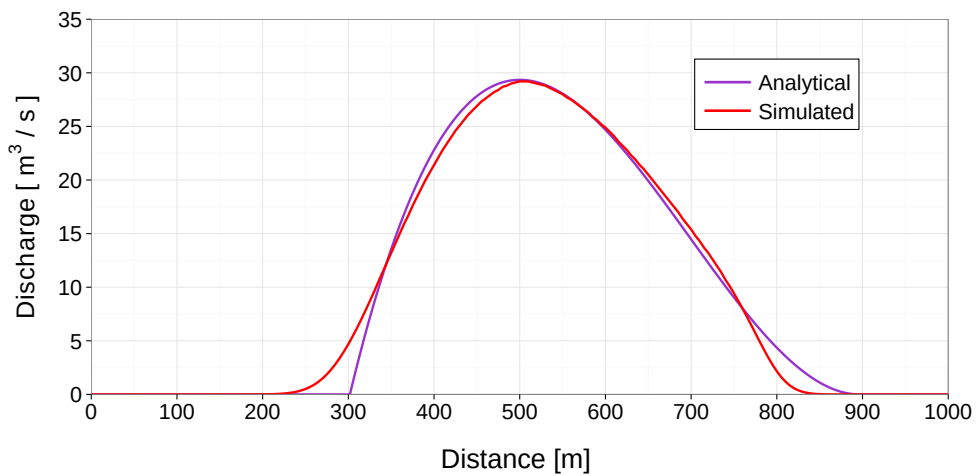
In two dimension, the dam break test case was computed using  $h_2/h_1 = 0.0$  and  $h_1 = 10$  m. The length of the channel was discretized using 1000 control volumes. The CFL number



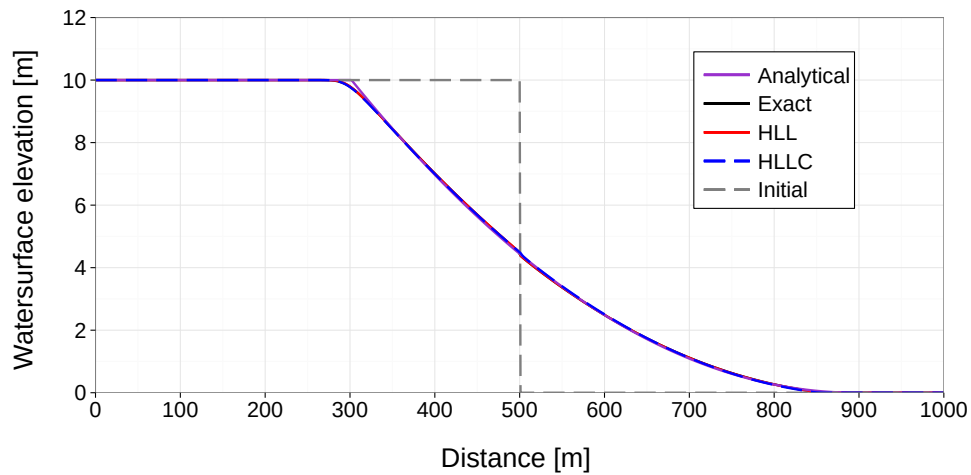
**Figure 1.19** *H\_1: Initial and final water surface elevation (BC & BP)*



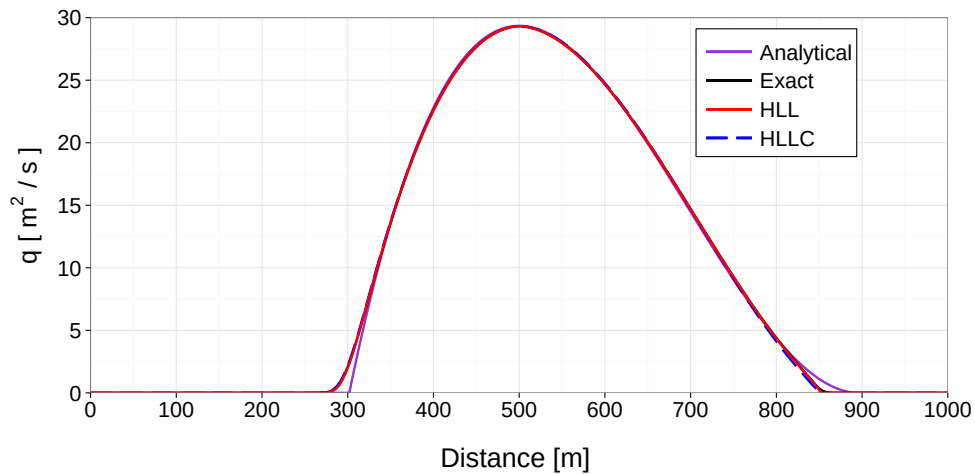
**Figure 1.20** *H\_2: Water surface elevation after 20 s (BC)*



**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.

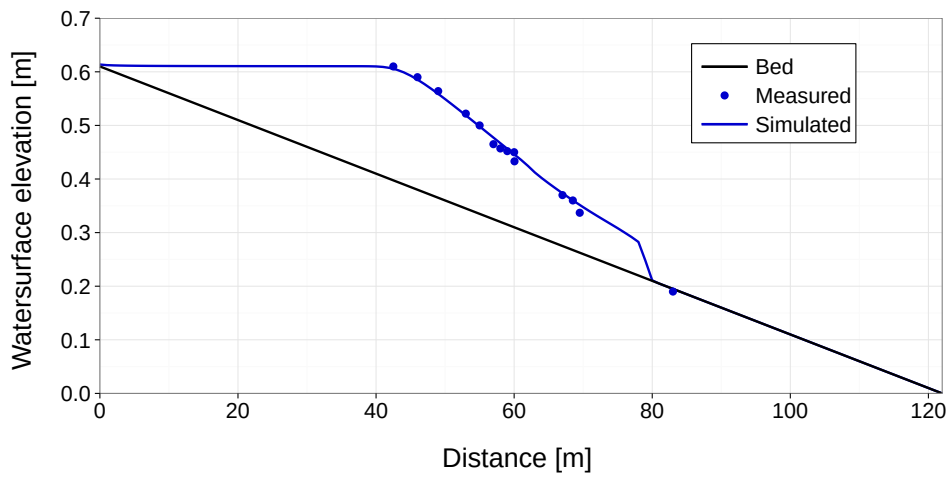


Figure 1.24 H\_3: Water surface elevation after 10 s (BC)

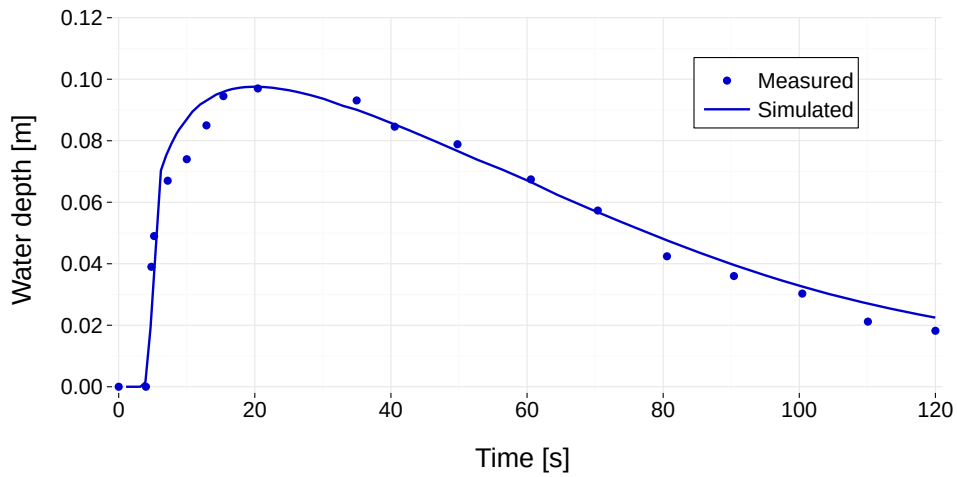


Figure 1.25 H\_3: Temporal behaviour of water depth at  $x = 70.1$  m (BC)

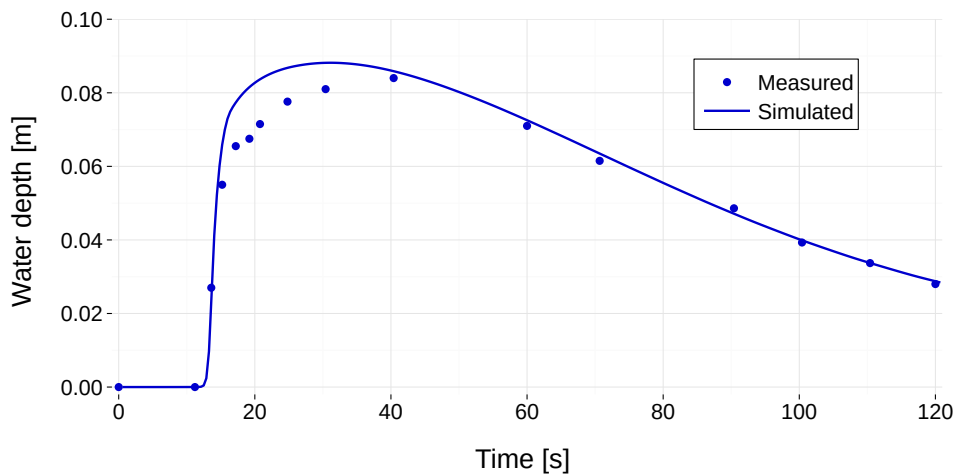


Figure 1.26 H\_3: Temporal behavior of water depth at  $x = 85.4$  m (BC)

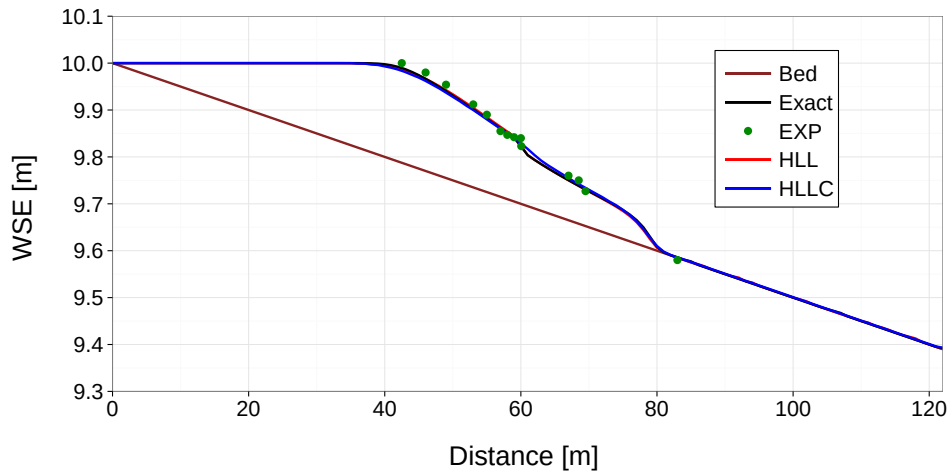


Figure 1.27  $H_3$ : Cross sectional water level after 10 s simulation time (BP)

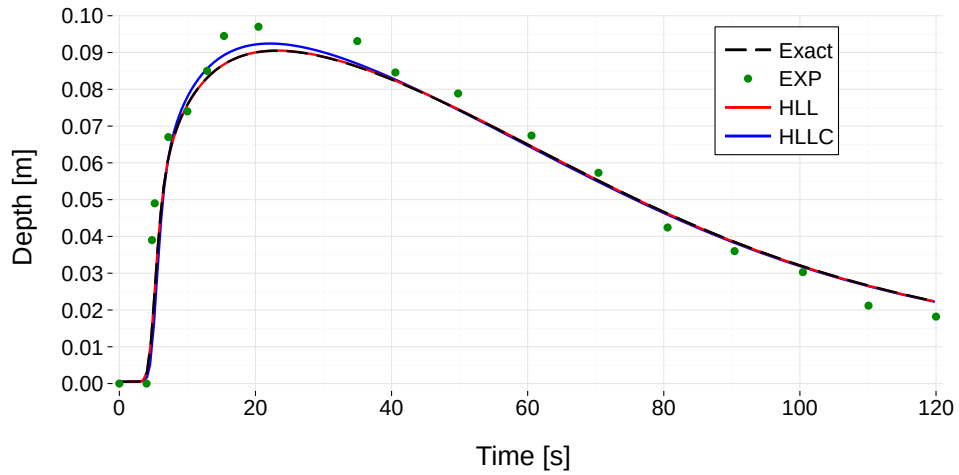


Figure 1.28  $H_3$ : Chronological sequence of water depth at point  $x = 70.1$  m (BP)

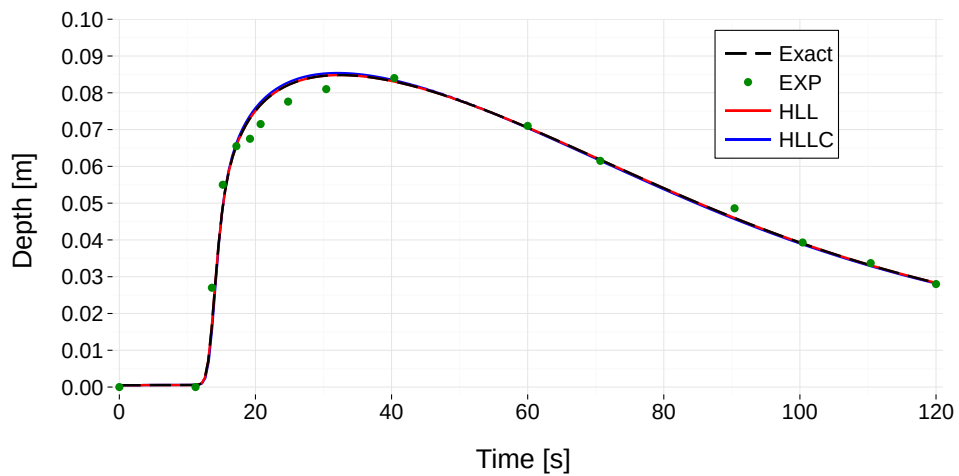


Figure 1.29  $H_3$ : Chronological sequence of water depth at point  $x = 85.4$  m (BP)



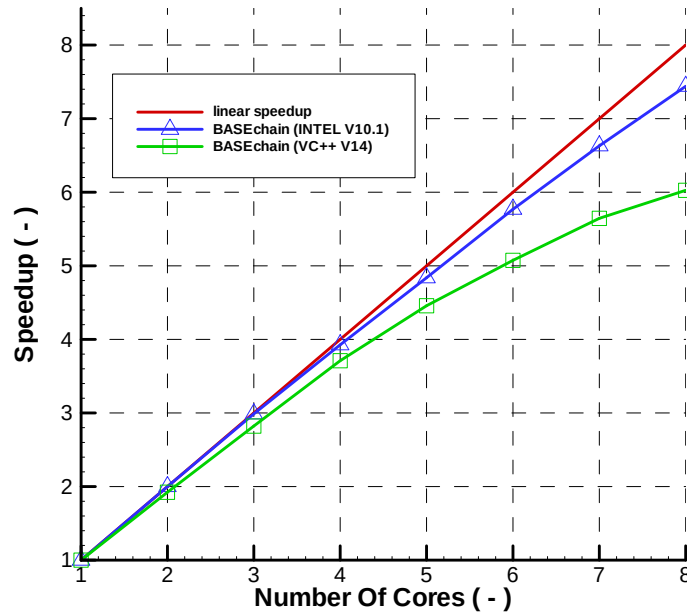


Figure 1.30  $H_4$  (BASEchain) – Parallel speedup for 1-D test case

#### 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 on 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 linearly 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.

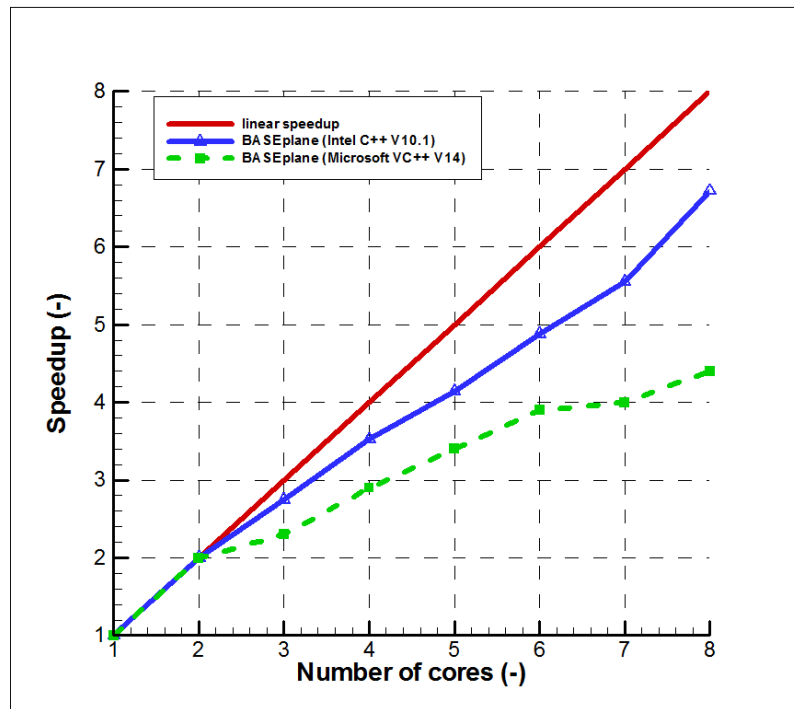


Figure 1.31  $H_4$  (BASEplane) – Parallel speedup for 2-D test case

### 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.

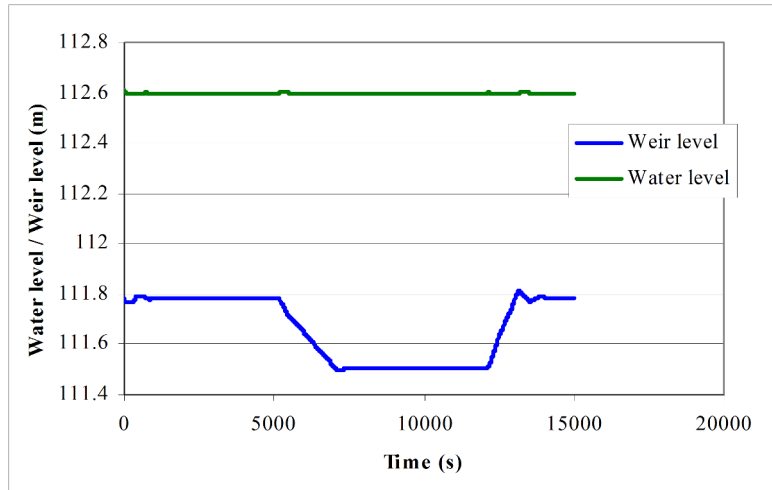
#### 1.5.1.5.2 Results for BASEplane

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  $450 \text{ m}^3/\text{s}$  down to  $180 \text{ m}^3/\text{s}$  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.

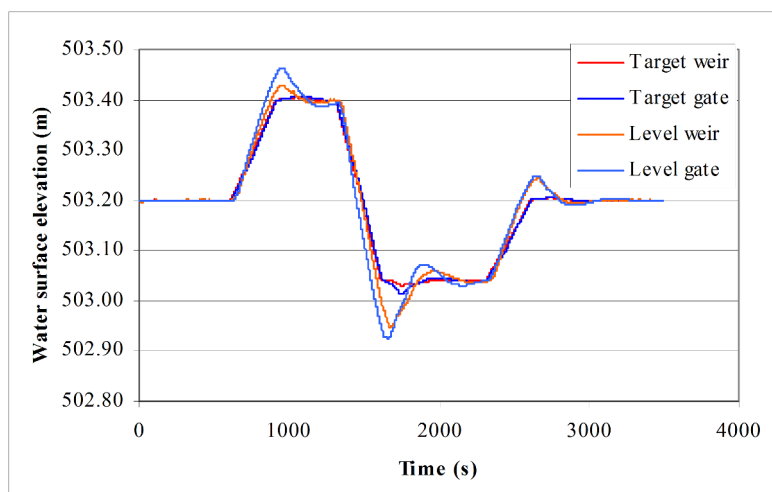
## 1.5.2 BASEchain Specific Test Cases

### 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.

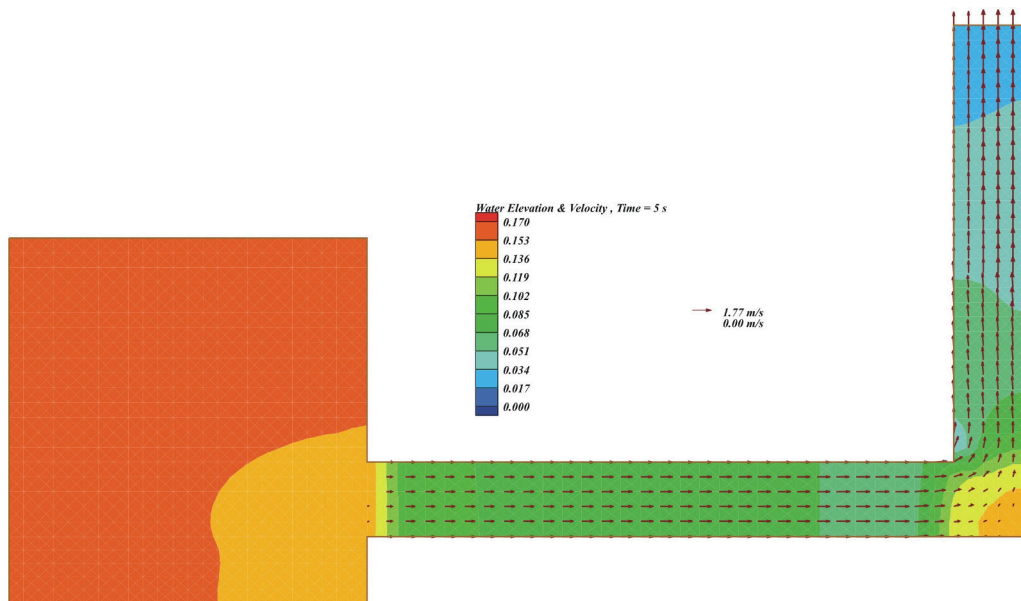


Figure 1.34 H\_BP\_3: Water level and velocity vectors 5 seconds after the dam break

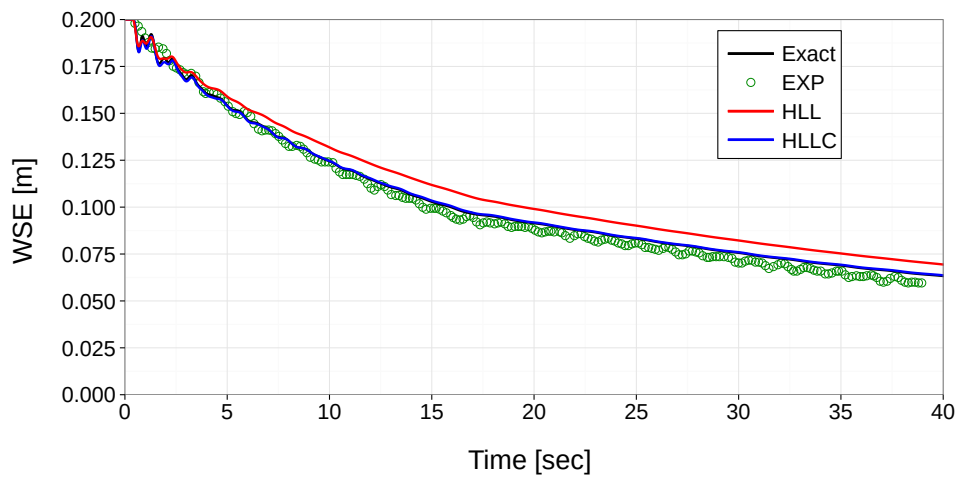


Figure 1.35 H\_BP\_3: Water surface elevation at G1

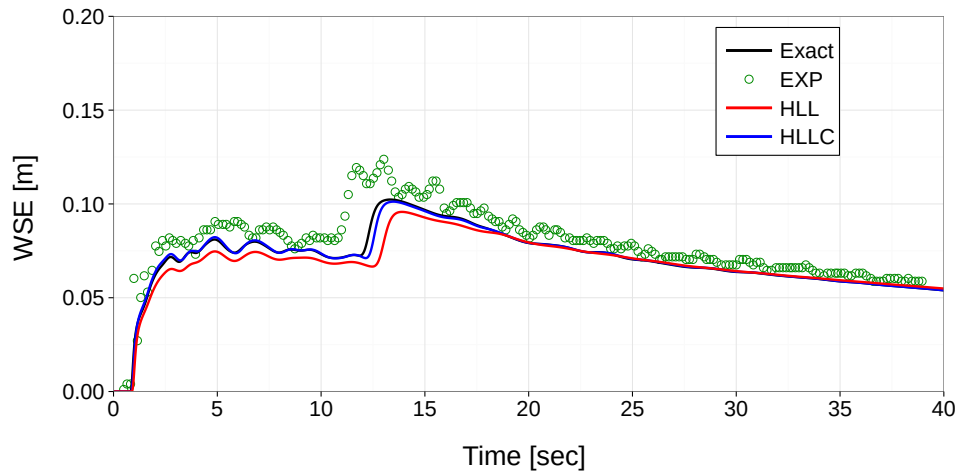


Figure 1.36 *H\_BP\_3*: Water surface elevation at G3

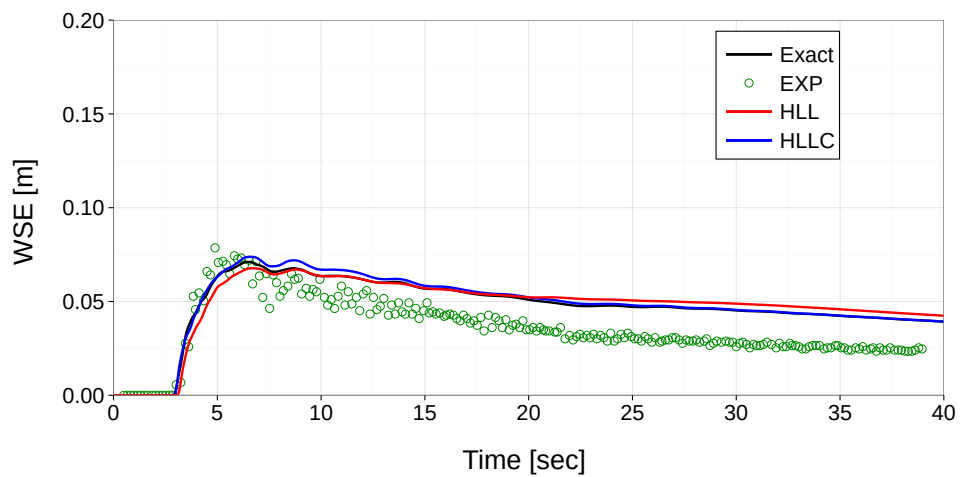
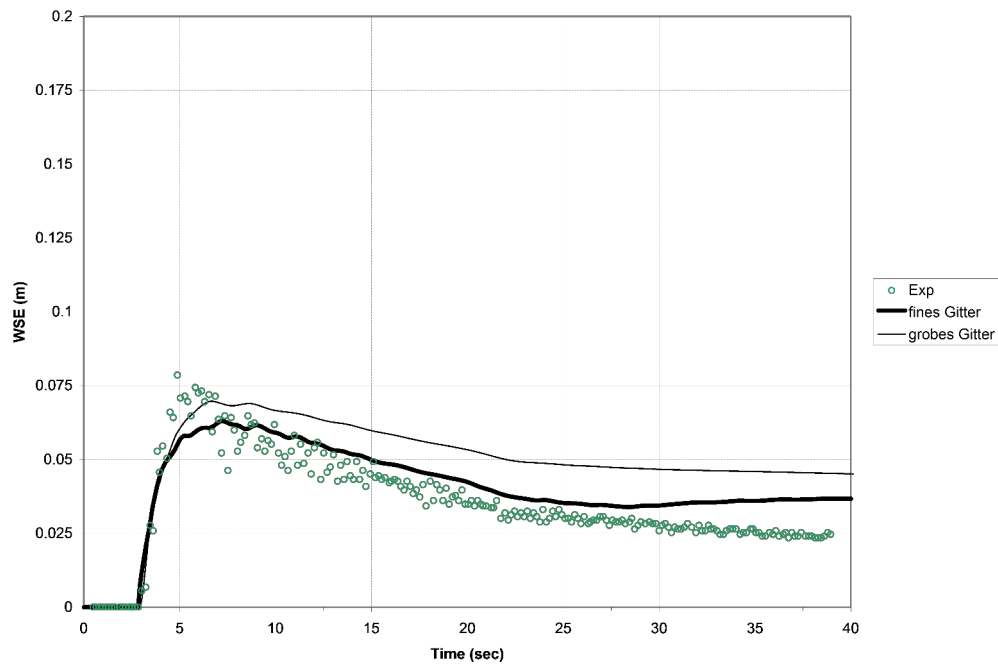


Figure 1.37 *H\_BP\_3*: Water surface elevation at G5



*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

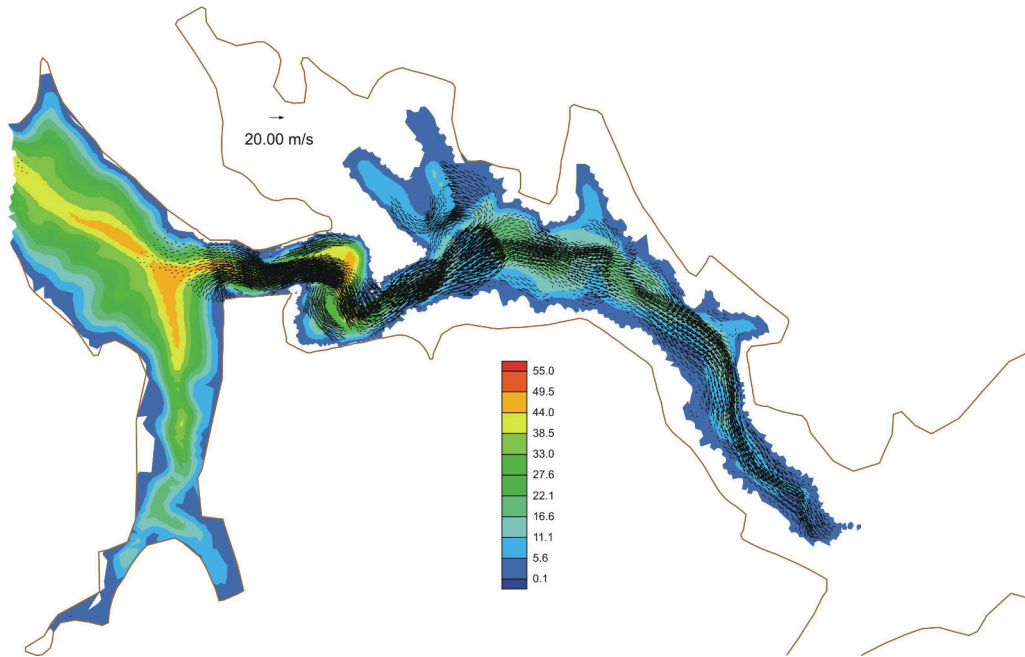


Figure 1.39 H\_BP\_4: Computed water depth and velocities 300 seconds after the dam break.

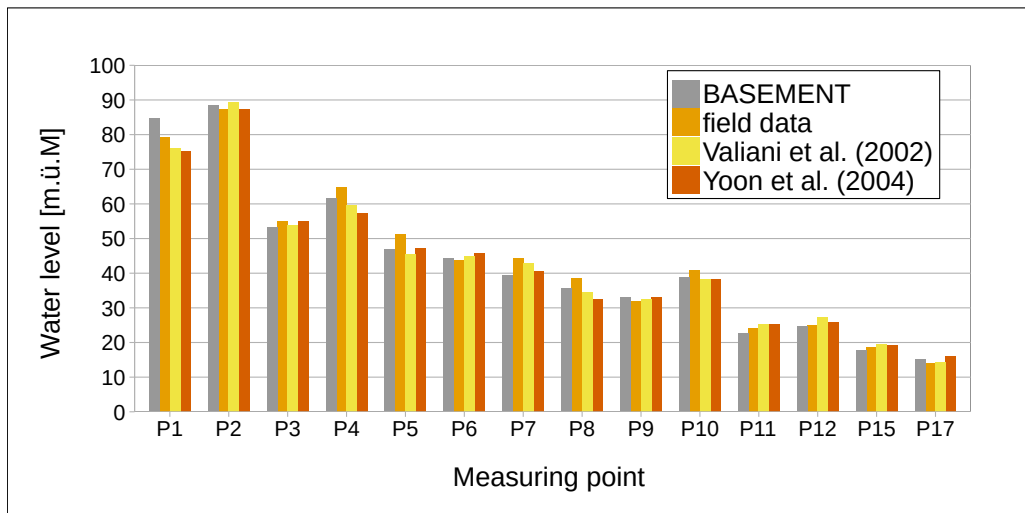
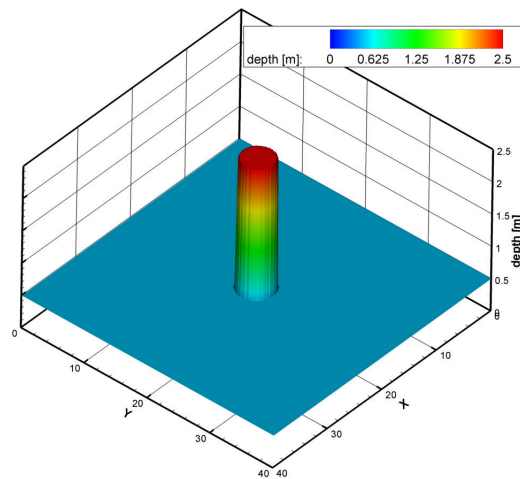


Figure 1.40 H\_BP\_4: Computed and observed water surface elevation at different control points



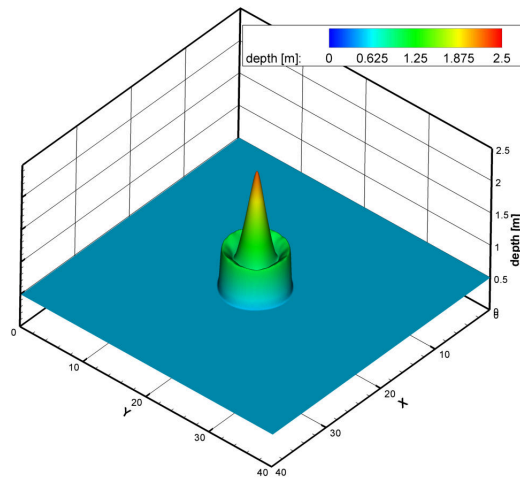


**Figure 1.41** *H\_BP\_5*: Perspective view of the circular dam break wave patterns,  $t=0.0s$

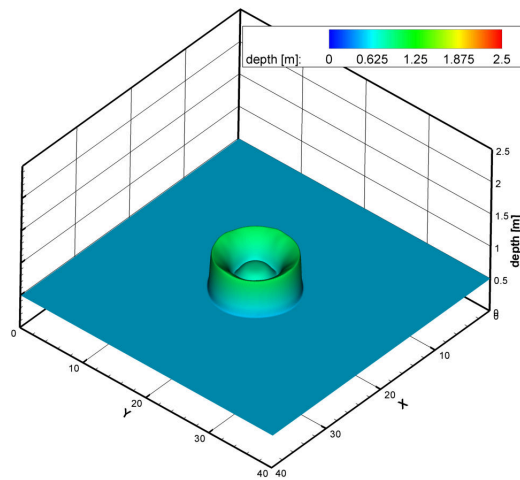
which propagates inwards while the primary shock wave continues to propagate outwards with decreasing strength. This secondary shock wave converges to the center and finally implodes. This generates a sharp jump in the surface water elevation at the center ( $t = 4.7$  s).

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.

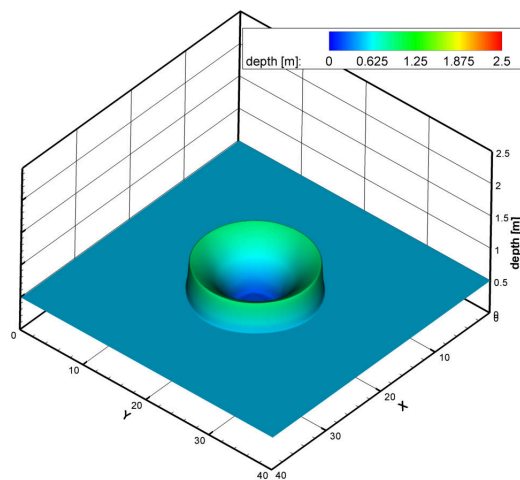
The resulting water surface and velocity profiles of the approximate Riemann solvers match well the results of the exact Riemann solver. Both approximate Riemann solvers, HLL and HLLC, reproduce the flow and wave patterns.



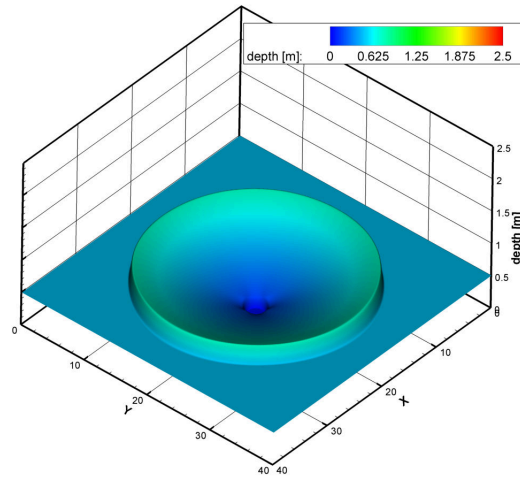
*Figure 1.42* H\_BP\_5: Perspective view of the circular dam break wave patterns,  $t=0.4s$



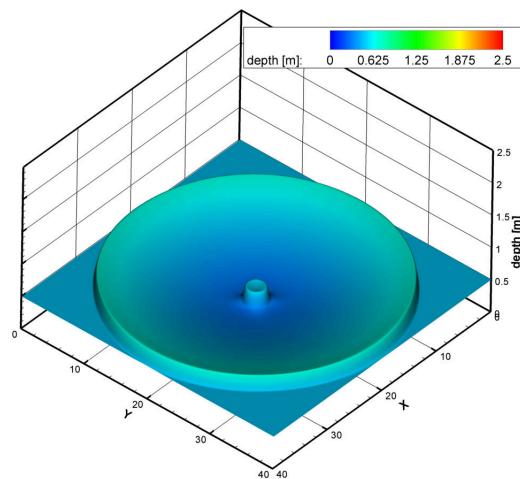
*Figure 1.43* H\_BP\_5: Perspective view of the circular dam break wave patterns,  $t=0.7s$



*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,  $t=4.7s$*

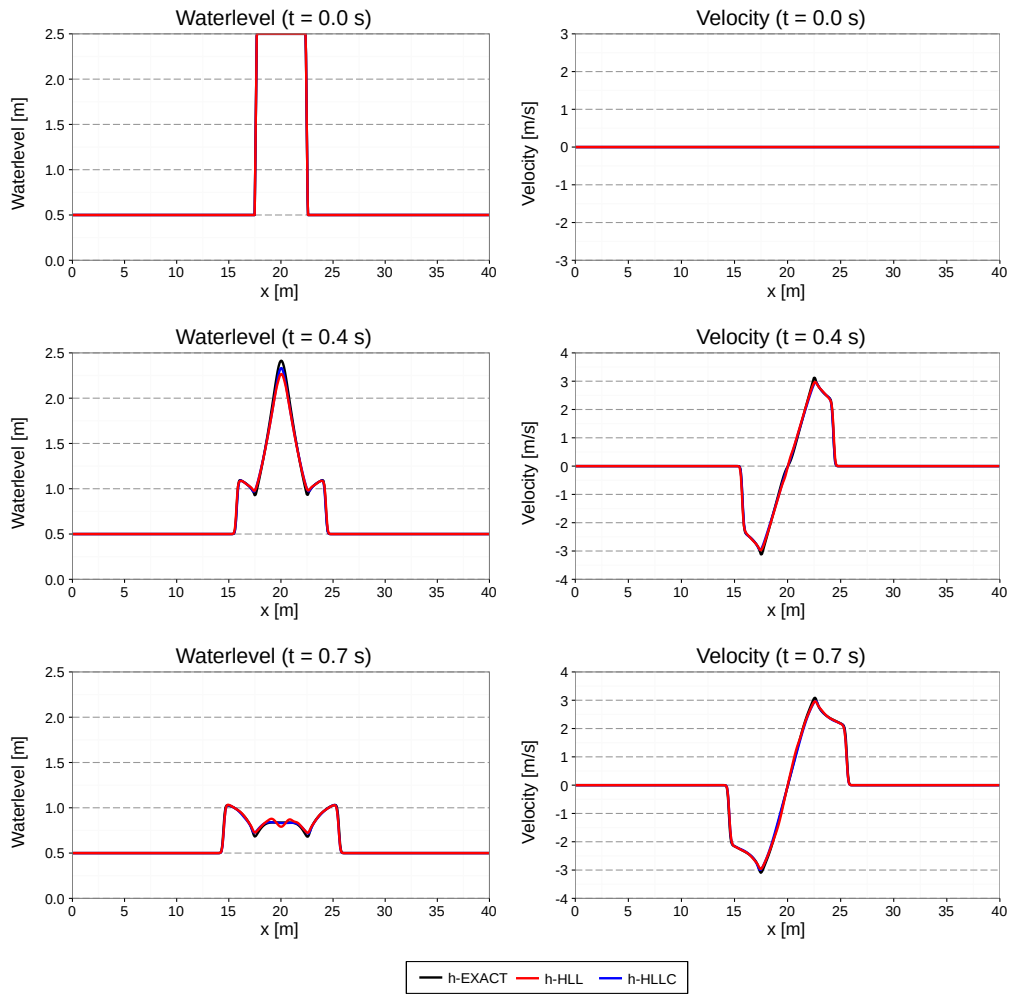


Figure 1.47 H\_BP\_5: Water surface along  $y=20m$

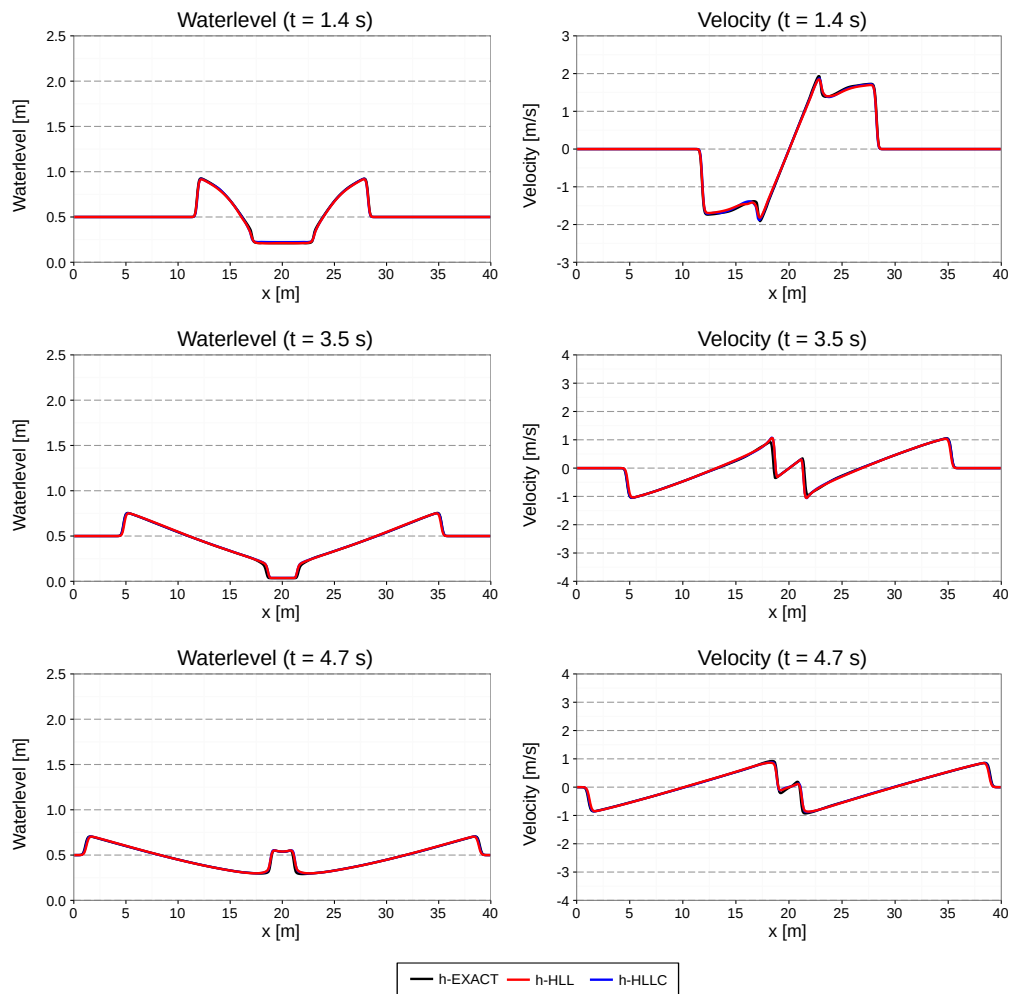


Figure 1.48 H\_BP\_5: Water surface along  $y=20m$

# 2

---

## 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<sup>3</sup>/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

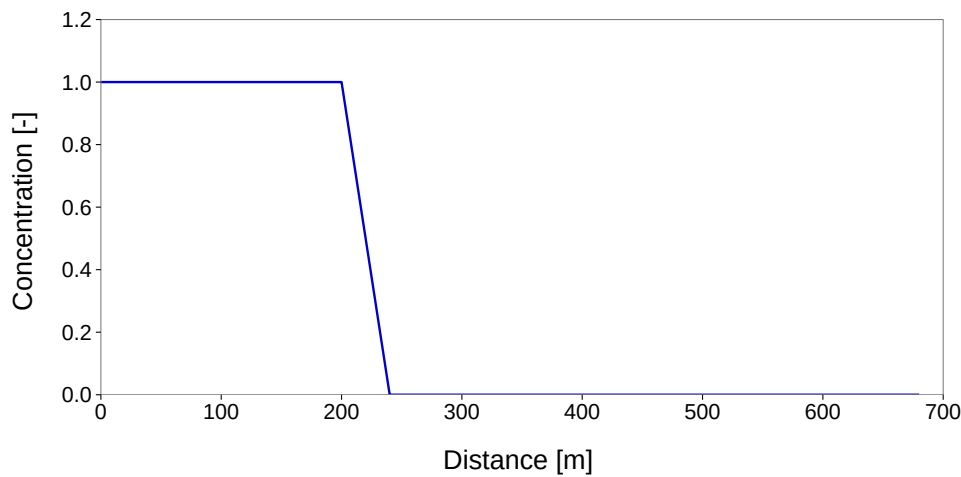
This test is intended to verify the quality of the advection schemes. The aim is to minimize the numerical diffusion even over long distances.

#### 2.3.1.2 Description

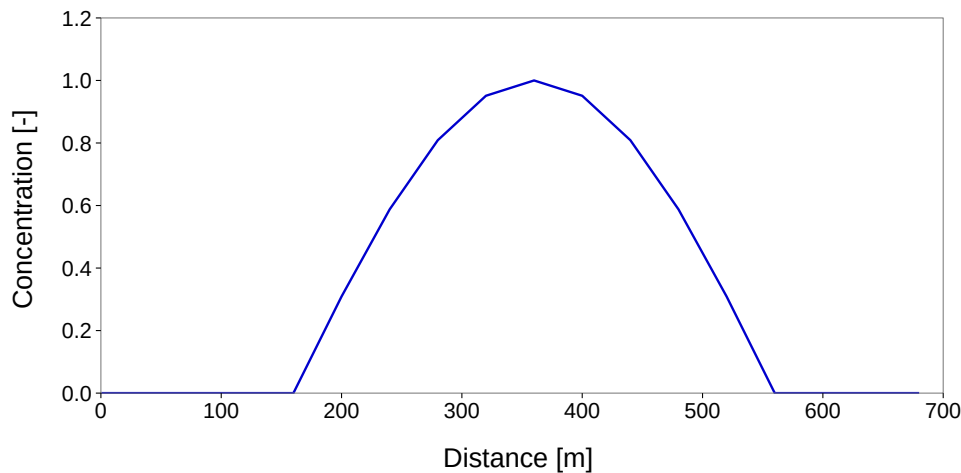
In a rectangular channel with steady flow conditions the advection of the suspended material is observed, once for a vertical front of concentration (case A) and once for a concentration with a Gaussian distribution (case B). The Quickest, Holly-Preissmann and MDPM-Scheme are tested. The diffusion is 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  $\Delta x$  is 40 m.
- The initial condition for the hydraulics is a steady discharge of 10 m<sup>3</sup>/s
- The initial conditions for the suspended load are the following:



**Figure 2.1** *ST\_BC\_1: Initial condition for case A*



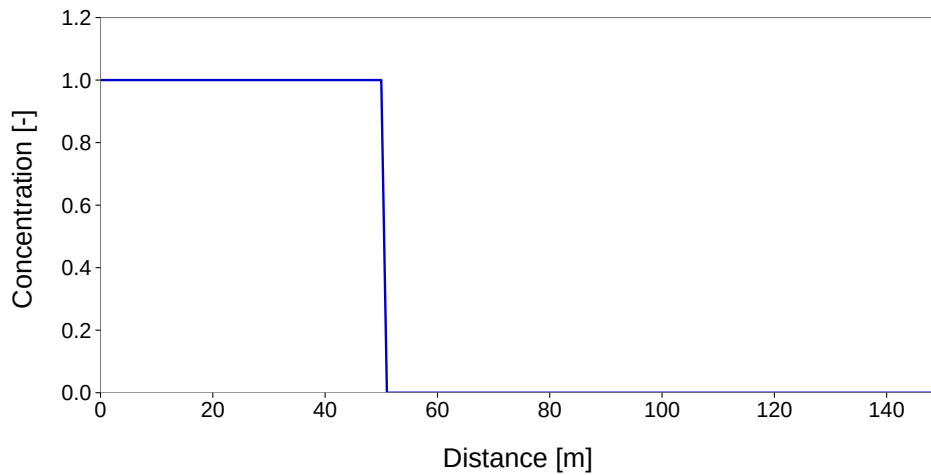
**Figure 2.2** *ST\_BC\_1: Initial condition for case B*

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  $10 \text{ m}^3/\text{s}$ .
- 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  $\Gamma$ .

### 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

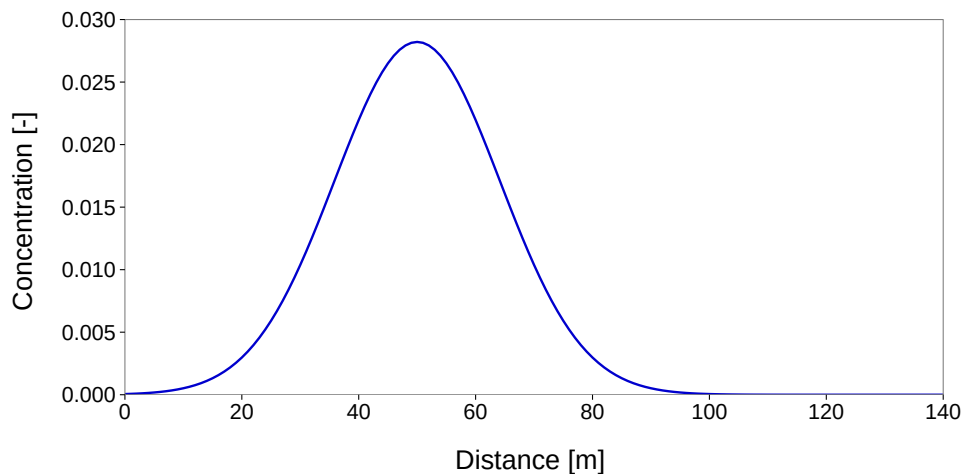
- The computational area is a rectangular channel with 1000 length, 20 m width and a slope of 1 ‰. The friction expressed as  $k_{Str}$  is 30. The cell width  $\Delta x$  is 1 m.
- The initial condition for the hydraulics is a steady discharge of 50 m<sup>3</sup>/s.
- The initial conditions for the 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<sup>3</sup>/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

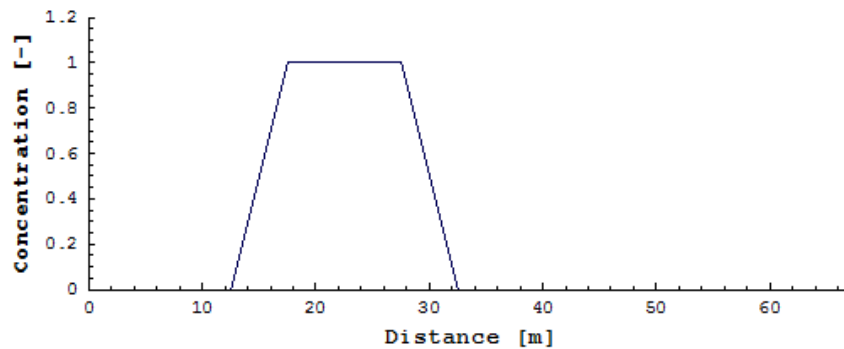
This test is intended to verify the quality of the advection scheme. The aim is to minimize the numerical diffusion even over long simulation time or distance.

#### 2.4.1.2 Description

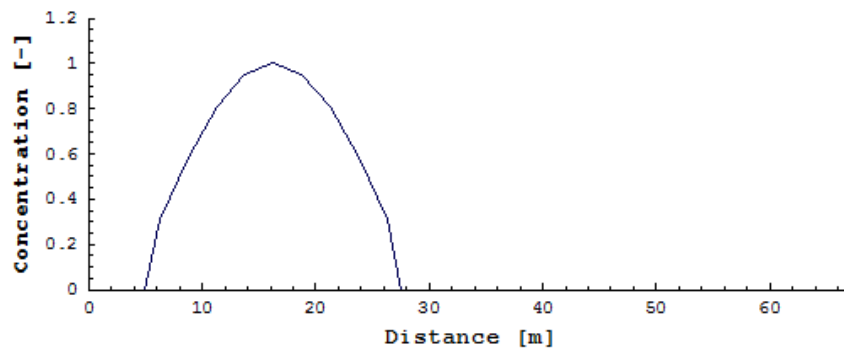
In a rectangular channel with steady flow conditions the advection of the suspended material is observed, once for a vertical front of concentration (case A) and once for a concentration with a Gaussian distribution (case B). The MDPM-Scheme is tested. The diffusion is set to 0 and there is no sediment 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<sup>3</sup>/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  $50 \text{ m}^3/\text{s}$  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.

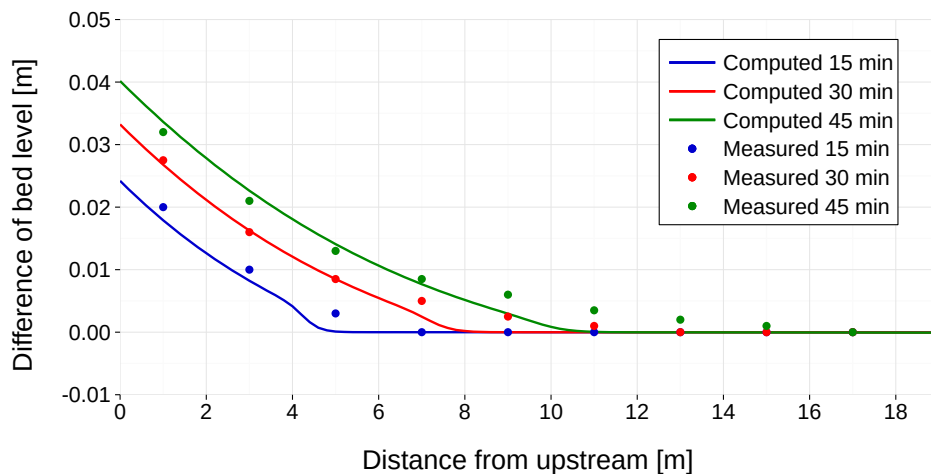


Figure 2.7 Soni Test with MPM-factor = 6.44 (BC)

## 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  $2.42 \cdot 10^{-7} \text{ m}^3/\text{s}$  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.

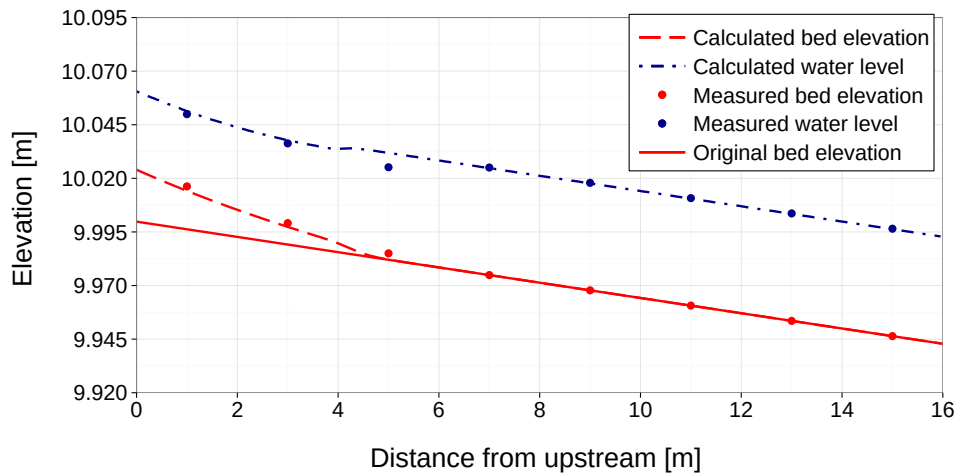
##### 2.5.1.1.2 Results obtained by BASEplane

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.

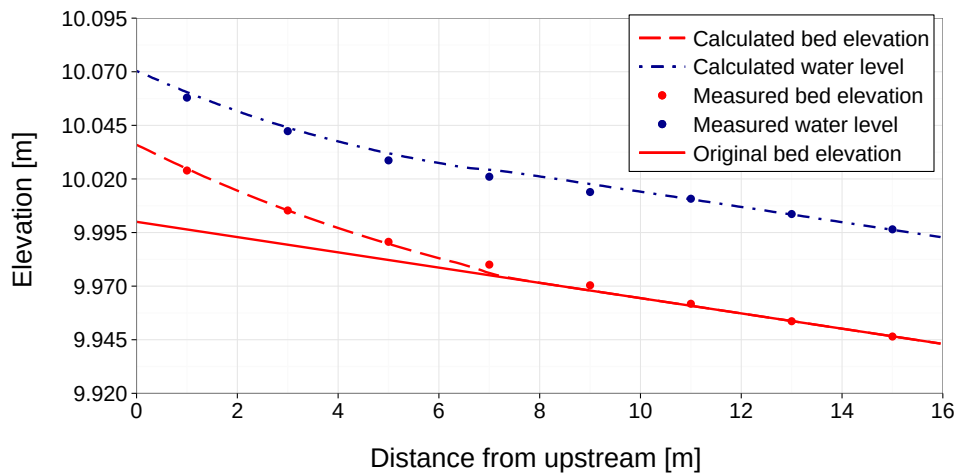
In Soni's experiments at first an equilibrium transport was established within the laboratory flume. The corresponding transport rate observed by Soni is known to be  $2.42 \cdot 10^{-7} \text{ m}^3/\text{s}$ . Then the sediment inflow was increased to 4 times the equilibrium transport.

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

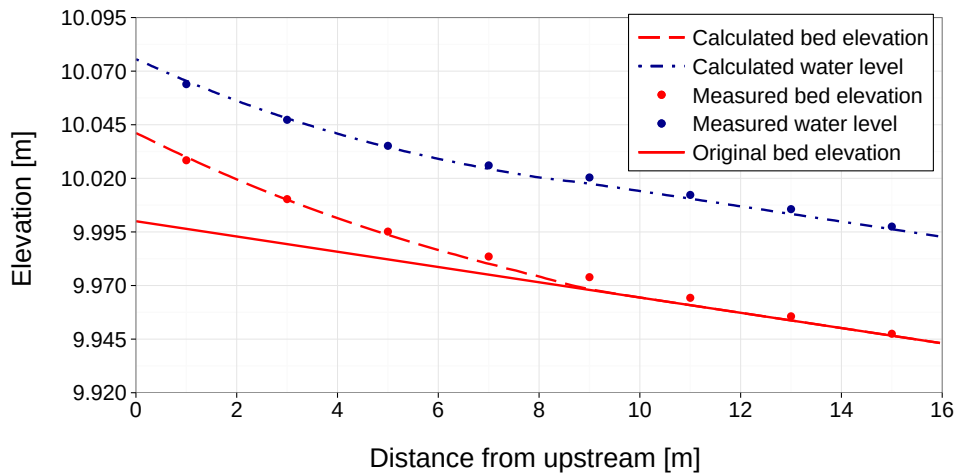
#### 2.5.1.2.1 Results obtained by BASEchain

The Saiedi test was simulated using the Meyer-Peter Müller approach for the sediment flux, which can also be formulated as:

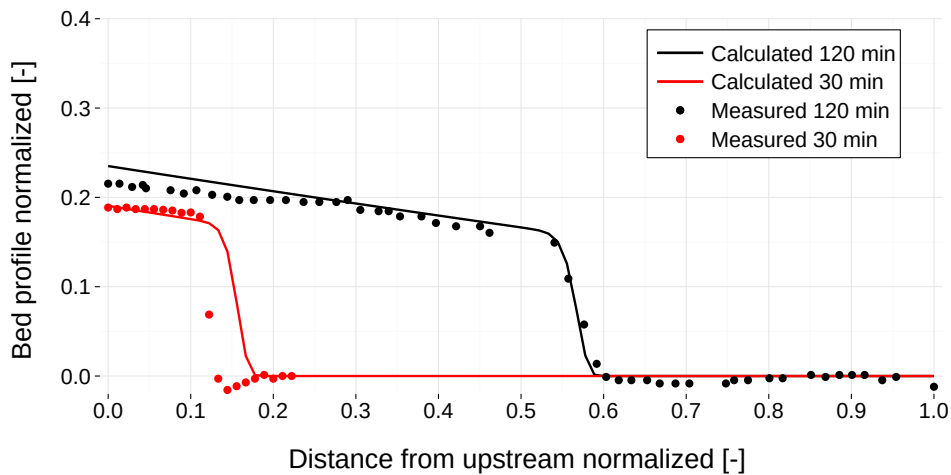
$$q_B = factor(\theta - \theta_{cr})^{3/2} \sqrt{(s-1)gd_m^{3/2}}$$

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)*

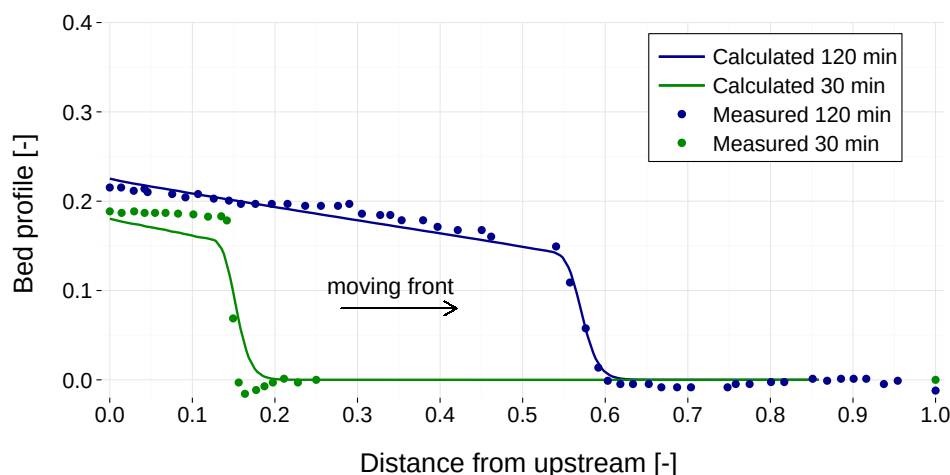
1989). In this case it was calibrated to 13.

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

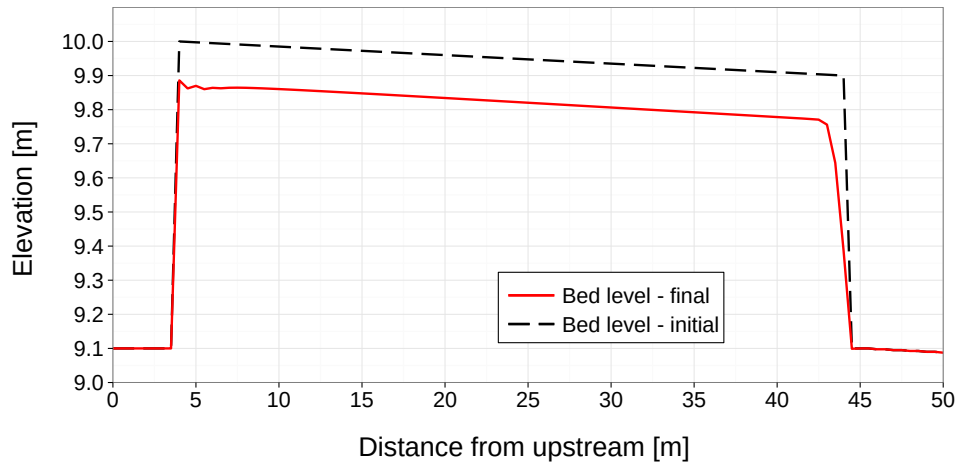


Figure 2.13 ST\_3: Guenter Test: equilibrium bed level (BC)

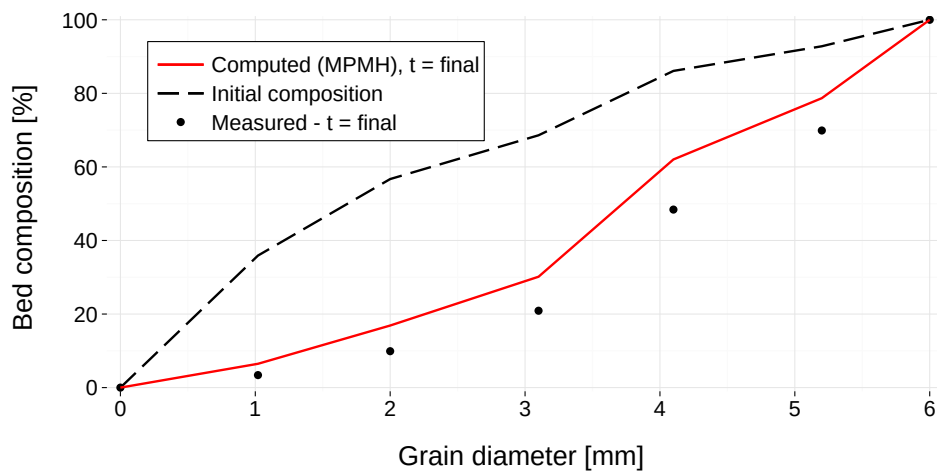
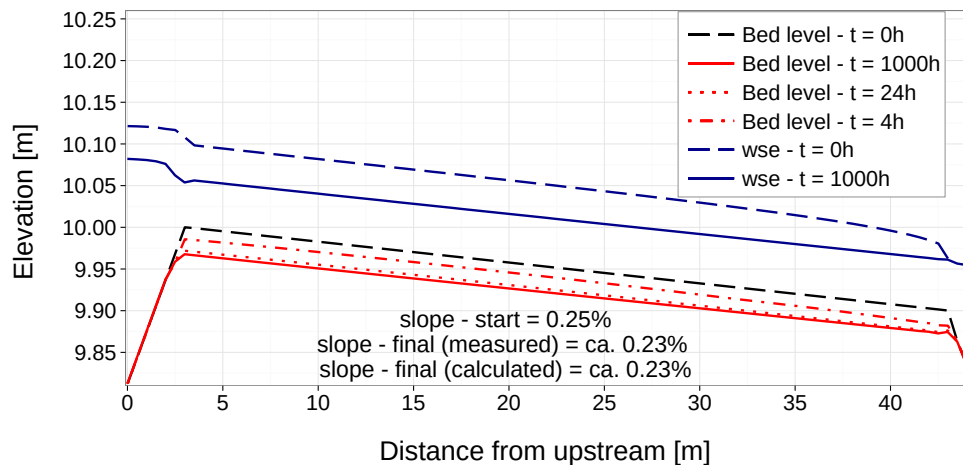


Figure 2.14 ST\_3: Guenter Test: grain size distribution (BC)



**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

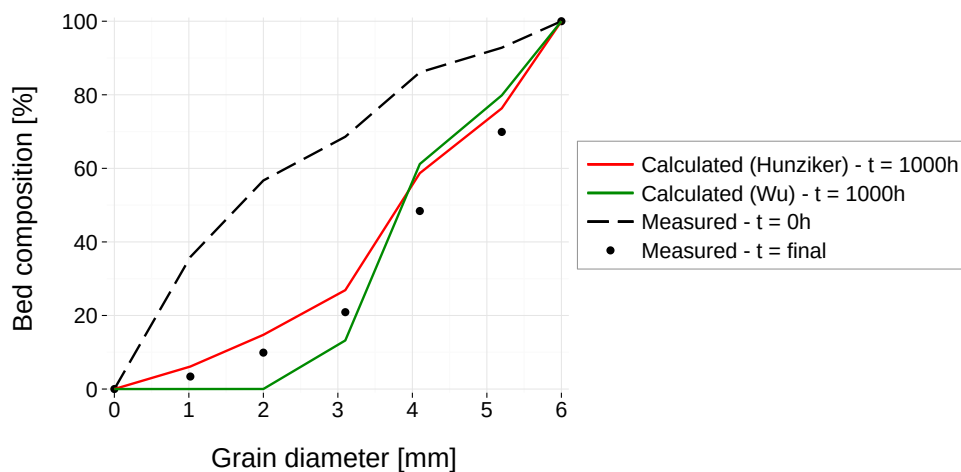


Figure 2.16 ST\_3: Guenter Test: grain size distribution (BP)

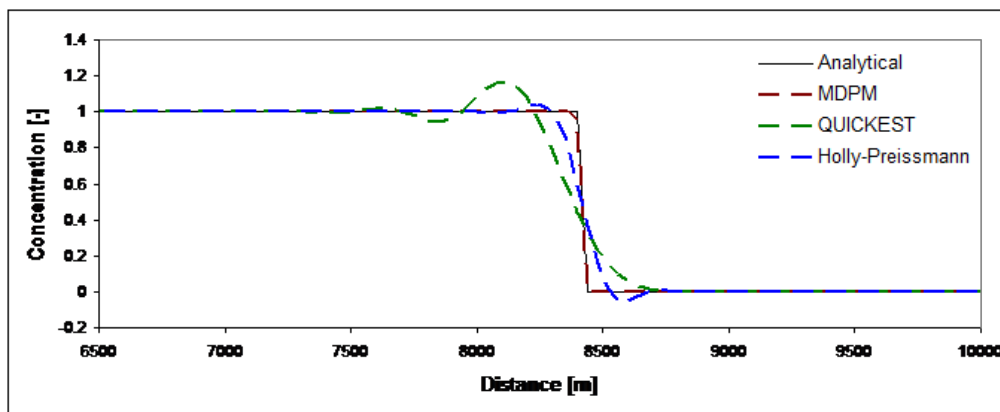


Figure 2.17 ST\_BC\_1: case A, concentration after 9060 s

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

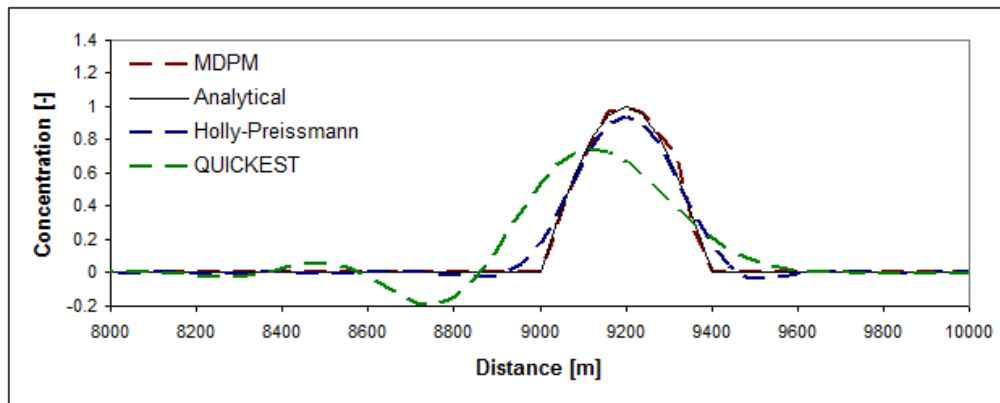


Figure 2.18 ST\_BC\_1: case B, concentration after 9772.5 s

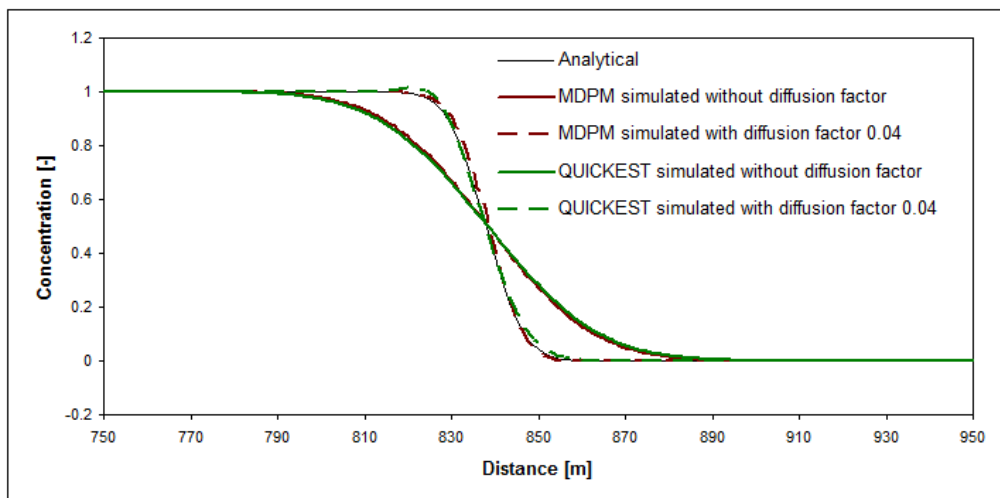


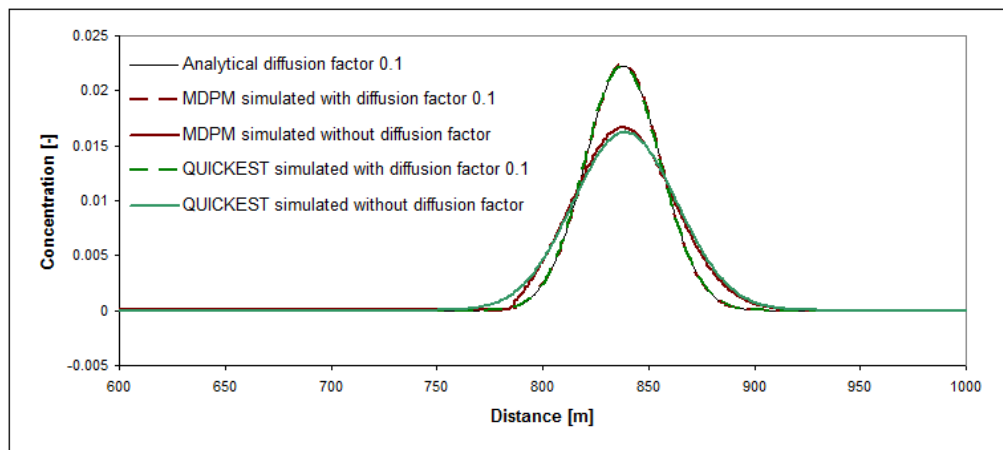
Figure 2.19 ST\_BC\_2: case A, concentration after 605 s

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 mainly 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.

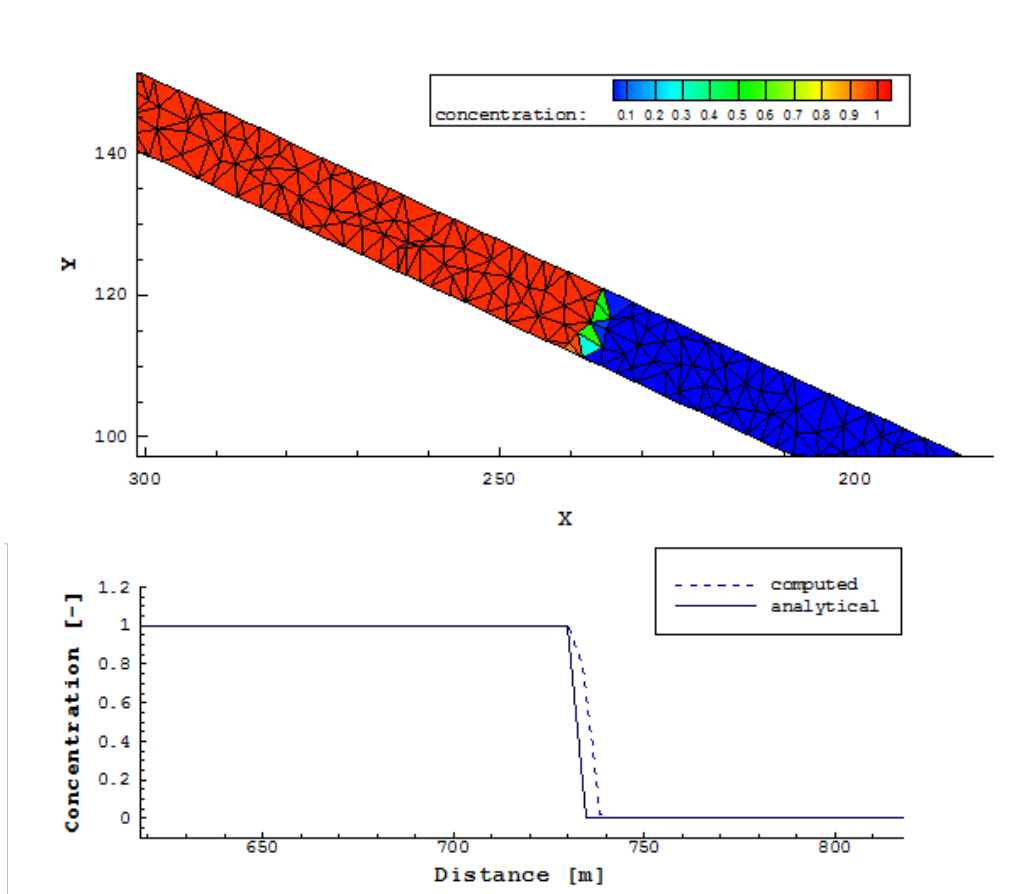


Figure 2.21 ST\_BC\_1: case A, concentration after 200 s



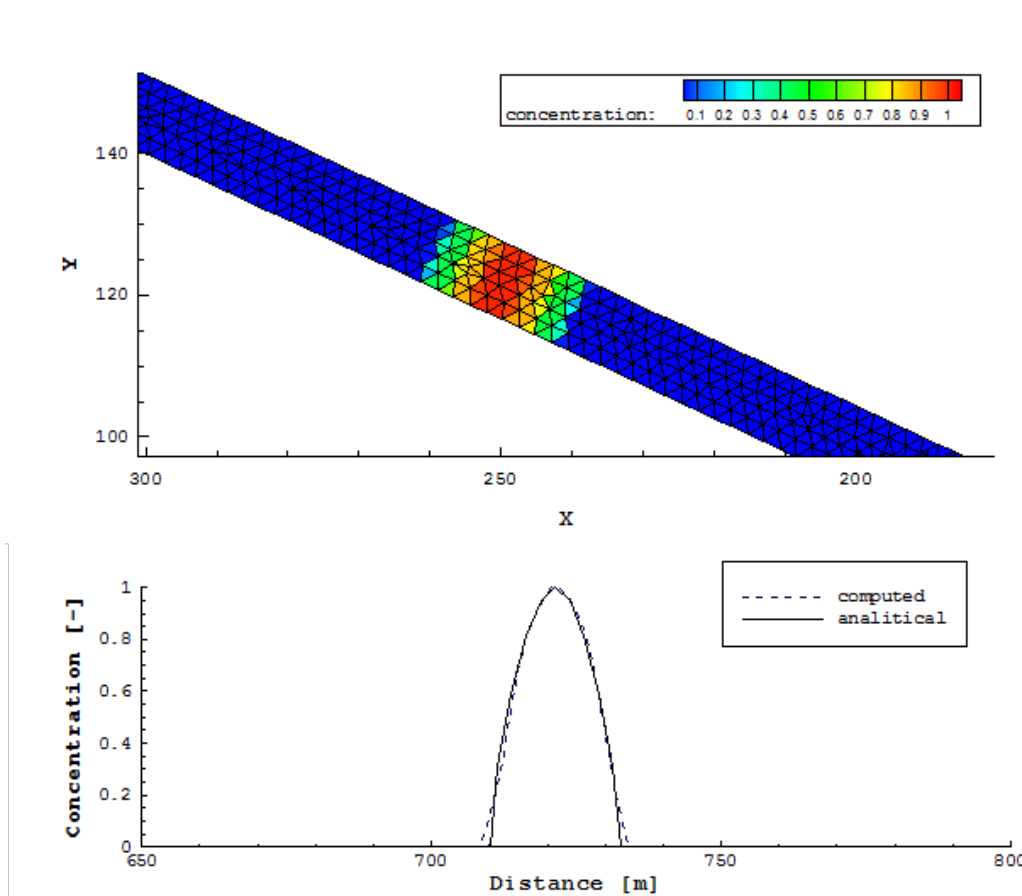


Figure 2.22 ST\_BP\_1: case B, concentration after 200 s

---

# 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

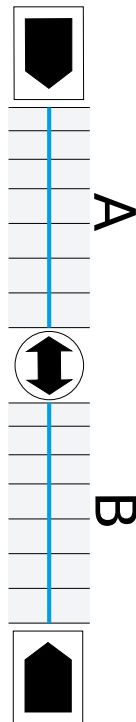
This test is intended to verify and test the coupling mechanism of a sequential coupling with mutual data exchange between the sub-domains.

#### 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

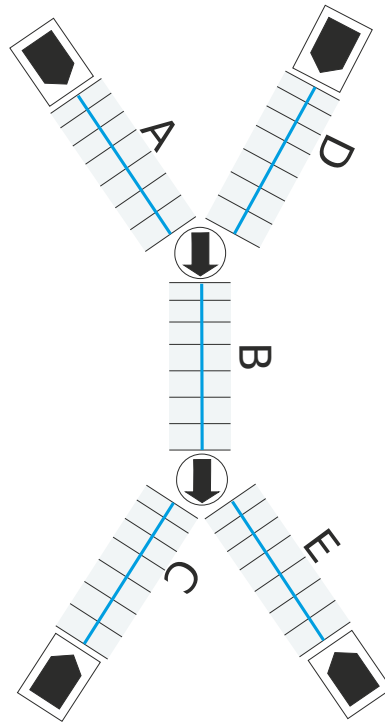
This test is intended to verify and test the coupling mechanism of a junction and a bifurcation in a simple river network configuration.

#### 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

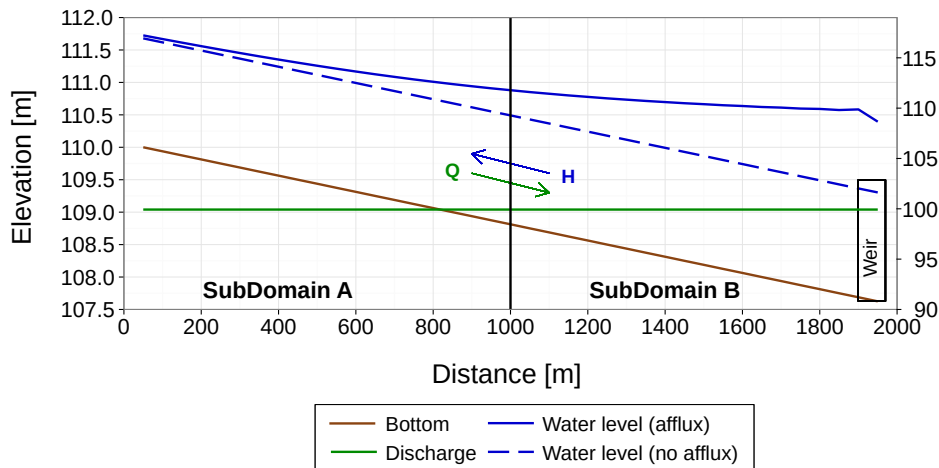
This test is intended to verify and test the coupling mechanism of a sequential coupling in a morphological simulation with bed load and suspended load transport.

#### 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  $h_q$ -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

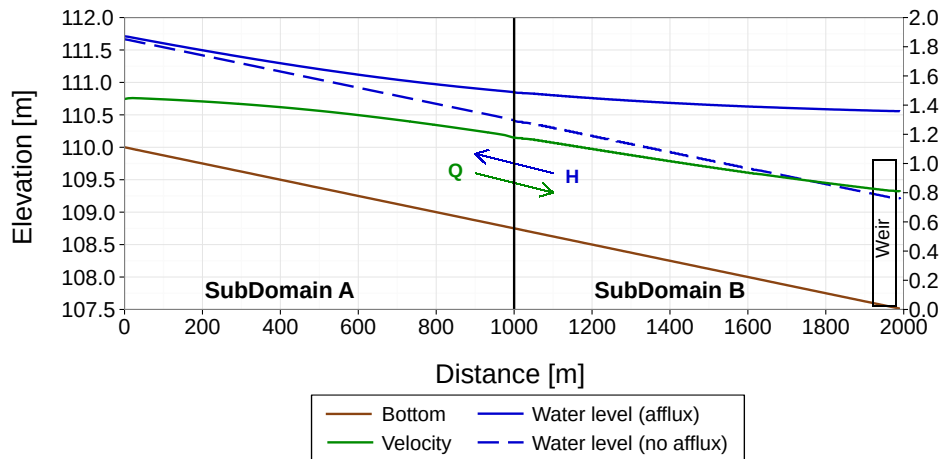
The coupling test case has been simulated starting from dry conditions in sub-domains A and B. Discharge is passed into downstream direction and water surface elevations are passed into upstream direction (two-way coupling).

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.

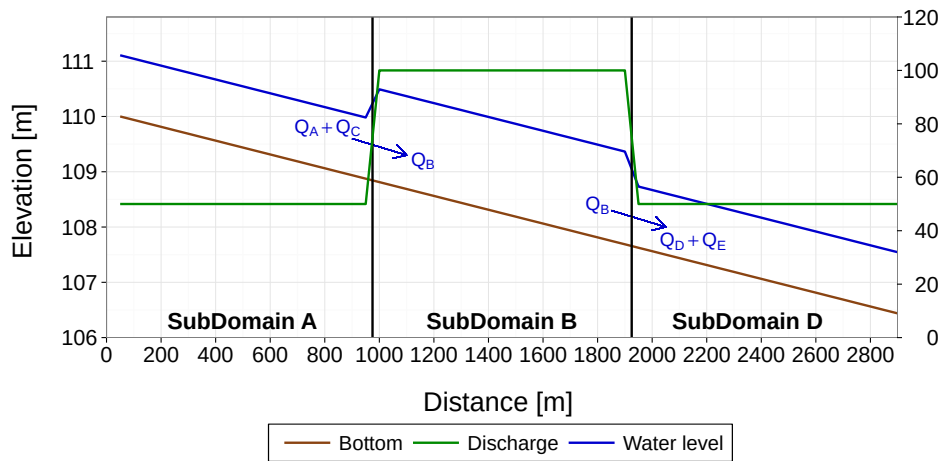
##### 3.2.1.1.2 COUPL\_1\_BP: Results obtained by BASEplane

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

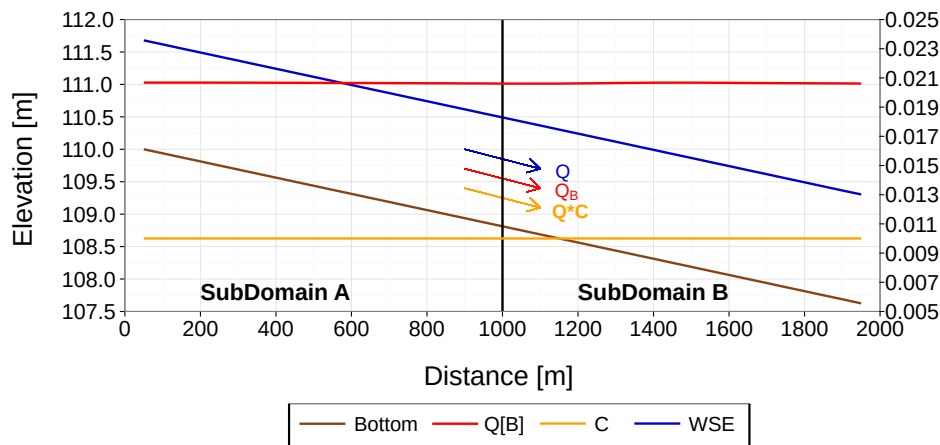


**Figure 3.5** COUPL\_2: Water surface elevation and discharge profiles in the coupled river network

### 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.

---

## 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  $1 \times 0.1 \times 1$  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  $\vartheta$  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  $\partial h / \partial x = 3$ .

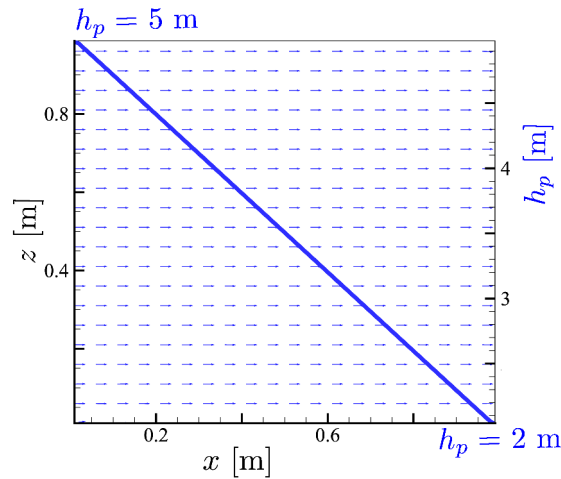
The obtained results are illustrated in Figure 4.1, for a cross sectional cut along the y-axis through the domain. The model is able to reproduce the analytical linear gradient of the pore-water pressure head within the domain.

### 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  $1 \times 0.1 \times 1$  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  $k_f = 5.55 \text{E-}7$  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  $q_{inf} = 2.78\text{E-}7$  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  $h_s = -1\text{E}7$  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  $k_f = 9.5\text{E-}4$  m/s (the value given in the reference of  $0.95\text{E-}4$  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.

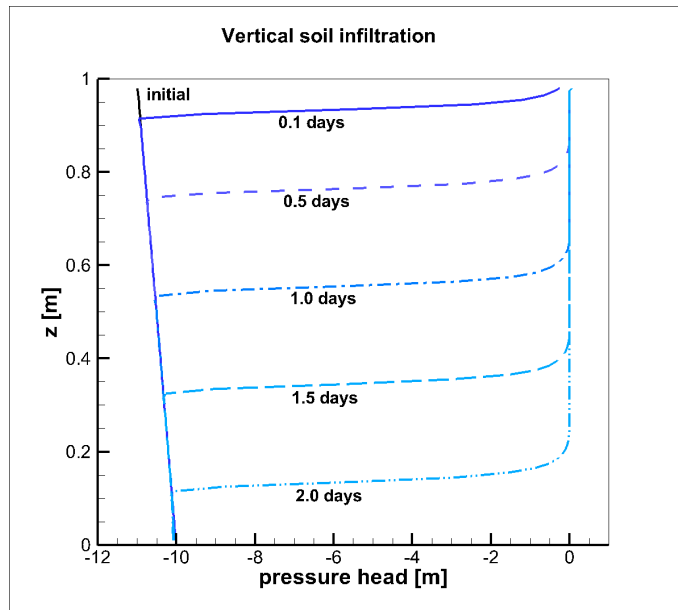


Figure 4.2 BASEsub2: Downward propagation of infiltration front into partially-saturated soil. Simulation results of BASEsub.

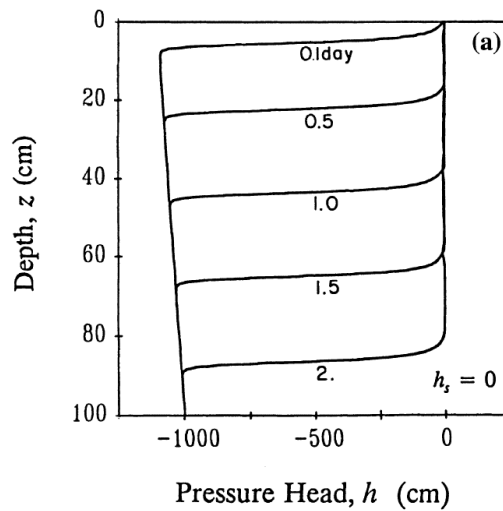


Figure 4.3 BASEsub2: Downward propagation of infiltration front into partially-saturated soil. Results of Vogel et al.

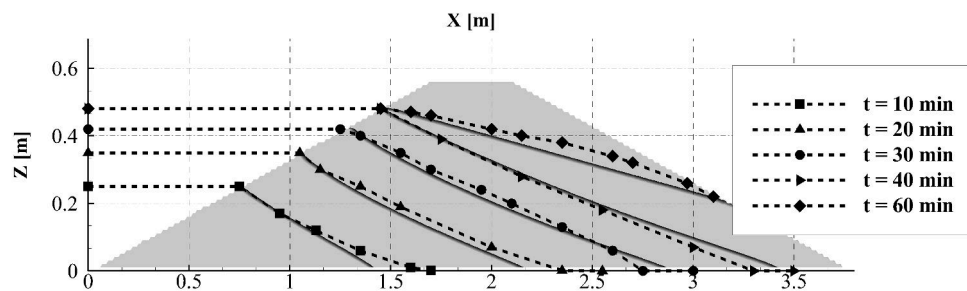


Figure 4.4 BASEsub3: Cross sectional view of measured (dashed) and simulated (bold) development of seepage line, caused by water infiltration due to a rising reservoir at the east dam side.



---

# High-Performance Computing

## 5.1 Introduction and Backends

### 5.1.1 Introduction

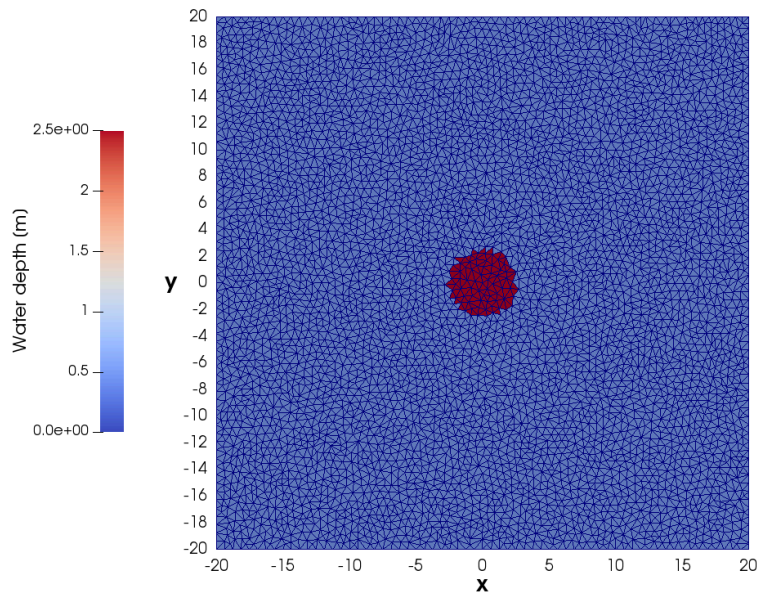
These test cases present well-defined hydro- and morphodynamic numerical simulations that are used to assess the performance and accuracy of the simulations with BASEMENT High-Performance Computing (BASEHPC). The test cases have an analytical solution to which the numerical solution is compared in order to evaluate the accuracy of the results. The numerical simulations are executed on different backends that influence the performance of the simulation.

### 5.1.2 Backends

Different backends are used to execute the simulations of the test cases. The different backends are: a central processing unit (CPU) with a sequential application on 1 core, CPU with parallel applications (OpenMP) on 2, 4, 8, 16 and 32 cores and graphic processing units (GPU) with a compute unified device architecture (CUDA) with single and double precision. The features of the GPU card are shown in Table 5.1.

**Table 5.1** Features of the GPU hardware used for the numerical simulations

Card	K20	P100	RTX 2080Ti	GTX 1080 Ti	GTX 1070 Ti	GTX 1050 Ti
Memory [GB]	5	12	11	11	8	4
Architecture	Kepler	Pascal	Turing	Pascal	Pascal	Pascal
Bandwidth [GB/s]	208	549	616	484	256	112
CUDA cores	2496	3584	4352	3584	2432	768



**Figure 5.1** Computational domain of the circular dam break (10k cells) with initial water depth.

## 5.2 Circular dam break

### 5.2.1 Description

The circular dam break problem represents a 2D Riemann problem based on the sudden collapse of an idealized circular dam over a horizontal bottom. The aim is to evaluate the ability of the model to solve the wave propagation, i.e. the complex interaction of shock and rarefaction waves induced by a dam break event. The results are quantitatively compared with a reference solution obtained as described in Toro (2001).

### 5.2.2 Geometry and Initial Conditions

The computational domain is a squared area of size 40 x 40 m and composed of unstructured triangular cells. The circular dam has a diameter of 5 m and is located at the center of the computational domain ( $x = 0$ ,  $y = 0$ ). Initially, the water is at rest and the depth inside the circular dam is 2.5 m and 0.5 m over the surrounding domain. No friction and no slope (horizontal bottom) is considered and the CFL number is set to 0.9. As the simulation starts (time 0.0 s), the dam is removed and the subsequent wave pattern is observed during 4.7 s, after which the simulation stops. Different mesh resolutions are used in the test case, the smallest mesh resolution contains exactly 9'927 cells (10k), the second 49'450 cells (50k), the third 99'416 cells (100k), the fourth 497'092 cells (500k) and the finest mesh resolution contains 994'092 cells (1000k). Figure 5.1 shows the initial conditions of the circular dam break with the computational domain of 10k cells.

### 5.2.3 Results

The evolution of the water depth is described along a reference cross-section located between  $(x = -20, y = 0)$  and  $(x = 20, y = 0)$ , see Figure 5.2. The reference solution is obtained from Toro (2001) using the exact Riemann solver on a mesh of about 1000 cells. After the collapse of the circular dam at  $t = 0.0$  s, the primary shock wave propagates outwards. At the same time, a rarefaction wave that propagates towards the center of the dam is observed. The rarefaction wave will implode at the grid center before  $t=0.4$  s and then travel in outward direction inducing a rapid drop of the water elevation which will reach a level below the initial outer water surface elevation ( $t=1.4$  s) in the center area. The primary shock wave travels outwards with decreasing strength while a secondary shock wave develops in opposite direction towards the center and finally generates a jump in the water depth at the grid center ( $t=4.7$  s).

The results on Figure 5.2 show how the solution obtained with BASEMENT version 3 (first order HLLC method) correctly converges to the reference solution as the grid is refined. The cylindrical symmetry of the wave propagation in BASEMENT is maintained, even if water surface modulations are observed along the primary shock wave.

The relative error is computed for the water depth ( $e_H$ ) and speed ( $e_u$ ) using eq. 5.1 and for various reference elevations. The result is displayed in Figure 5.3.

$$e_X = \frac{|X_{BM} - X_{ref}|}{X_{ref}} \quad (5.1)$$

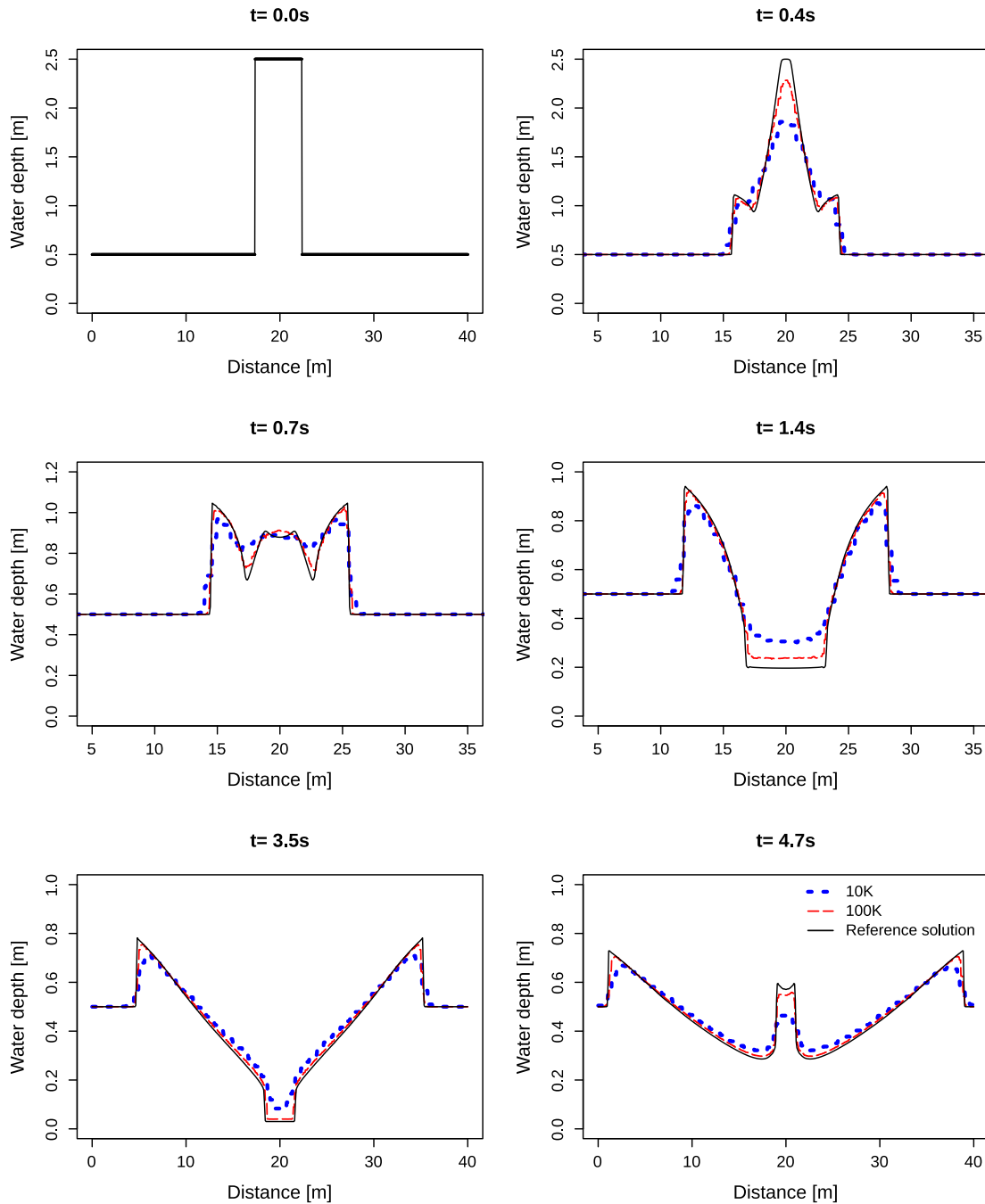
Where  $X$  stands either for the water surface elevation  $H$  or the velocity  $u$  with  $X_{BM}$  the result of the simulation with BASEMENT and  $X_{ref}$  the reference value of Toro (2001).

The relative  $L^2$  norm error (eq. 5.2, e.g Vanzo et al. (2016)) is calculated for each mesh size using the water surface elevation of the simulation result of BASEMENT ( $H_{BM}$ ) and the references solution ( $H_{ref}$ ) of Toro (2001). The result is shown on Table 5.2. The relative norm error decreases for finer meshes.

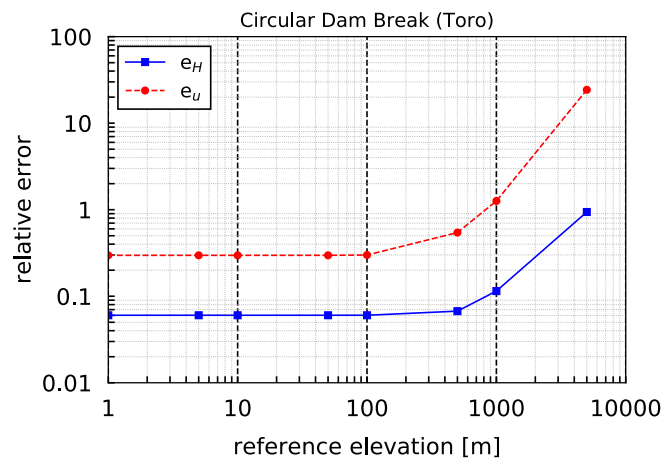
$$L^2 = \sqrt{\frac{\sum_{i=1}^N (H_{ref}(i) - H_{BM}(i))^2}{\sum_{i=1}^N H_{ref}(i)^2}} \quad (5.2)$$

**Table 5.2** Relative  $L^2$  norm error of the water surface elevation  $H$  compared to the reference solution of Toro (2001) at  $t=4.7$  s.

	10k	50k	100k	500k	1000k
$L^2$	0.1134	0.0523	0.0453	0.0184	0.0145



**Figure 5.2** Snapshots of the water depth evolution along the reference cross-section. Comparison between BASEMENT simulations with a computational grid of 10k cells, 100k cells and the reference solution of Toro (2001).



*Figure 5.3* Relative error compared to the reference solution of Toro (2001)



### 5.2.4 Performance

The performance is evaluated in terms of effective computation time and in terms of speedup  $S$ . The latter is a measure of parallelization efficiency. The speedup  $S$  is calculated as the division of the sequential runtime  $T_1$  (of each mesh size respectively) by the runtime with a certain number of cores  $T_N$  or a specific GPU card.

#### 5.2.4.1 Ubuntu

Tables 5.4, 5.5 and 5.6 contain the execution time and speedup of all the simulations performed on different backends under Linux. The backends belong to two categories, either in Central Processor Units (CPU) or Graphics Processor Units (GPU). The CPU run simulations exclusively with double precision, while GPU can perform simulations with single or double precision. More details are explained in Section 5.1.2.

**Table 5.3** Computational time (s) and speedup (in parentheses) of the circular dam break simulations for CPU simulations under Ubuntu 16.04 (Intel Xeon Gold 6154, 3.00GHz)

Mesh size	Xeon 1 Double	Xeon 2 Double	Xeon 4 Double	Xeon 8 Double	Xeon 16 Double	Xeon 32 Double
10k	0.16	0.19 (1.49x)	0.11 (2.46x)	0.07 (3.33x)	0.04 (4.18x)	0.03 (5.31x)
50k	1.88	1.08 (1.73x)	0.57 (3.29x)	0.33 (5.70x)	0.20 (9.31x)	0.15 (12.68x)
100k	5.44	3.06 (1.77x)	1.62 (3.35x)	0.89 (6.12x)	0.52 (10.4x)	0.35 (15.47x)
500k	63.63	34.49 (1.85x)	17.71 (3.59x)	9.28 (6.86x)	5.06 (12.57x)	3.07 (20.72x)
1000k	190.39	101.66 (1.87x)	51.83 (3.67x)	26.98 (7.06x)	14.36 (13.26x)	8.75 (21.77x)

**Table 5.4** Computational time (s) and speedup (in parentheses) of the circular dam break simulations for CPU simulations under Ubuntu 16.04 (Intel Xeon E5-2667 v3 @ 3.20GHz)

Mesh size	Xeon 1 Double	Xeon 2 Double	Xeon 4 Double	Xeon 8 Double	Xeon 16 Double
10k	0.16	0.19 (1.49x)	0.11 (2.46x)	0.07 (3.33x)	0.04 (4.18x) 0.03 (5.31x)
50k	1.88	1.08 (1.73x)	0.57 (3.29x)	0.33 (5.70x)	0.20 (9.31x) 0.15 (12.68x)

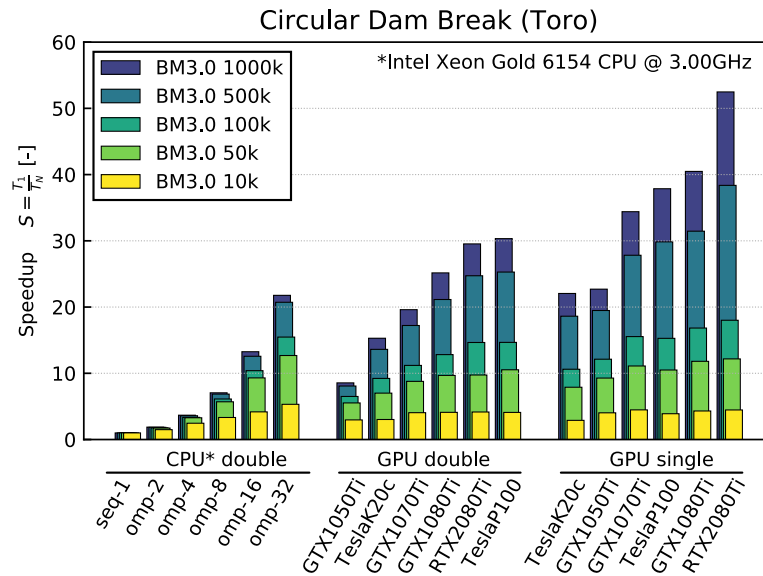
Mesh size	Xeon 1 Double	Xeon 2 Double	Xeon 4 Double	Xeon 8 Double	Xeon 16 Double
100k	5.44	3.06 (1.77x)	1.62 (3.35x)	0.89 (6.12x)	0.52 0.35 (10.4x) (15.47x)
500k	63.63	34.49 (1.85x)	17.71 (3.59x)	9.28 (6.86x)	5.06 3.07 (12.57x) (20.72x)
1000k	190.39	101.66 (1.87x)	51.83 (3.67x)	26.98 (7.06x)	14.36 8.75 (13.26x) (21.77x)

**Table 5.5** Computational time (s) and speedup (in parentheses) of the circular dam break simulations for GPU simulations

Mesh size	Xeon 1 Double	RTX 2080Ti Single	RTX 2080Ti Double	Tesla K20 Single	Tesla K20 Double	Tesla P100 Single	Tesla P100 Double
10k	0.16	0.04 (4.41x)	0.04 (4.10x)	0.06 (2.86x)	0.05 (2.99x)	0.04 (3.85x)	0.04 (4.04x)
50k	1.88	0.15 (12.18x)	0.19 (9.74x)	0.24 (7.90x)	0.27 (7.01x)	0.18 (10.5x)	0.18 (10.53x)
100k	5.44	0.30 (18.01x)	0.37 (14.65x)	0.51 (10.61x)	0.59 (9.23x)	0.36 (15.28x)	0.37 (14.67x)
500k	63.63	1.66 (38.37x)	2.57 (24.72x)	3.42 (18.61x)	4.67 (13.61x)	2.13 (29.84x)	2.52 (25.28x)
1000k	190.39	3.63 (52.48x)	6.45 (29.54x)	8.63 (22.05x)	12.45 (15.29x)	5.03 (37.87x)	6.28 (30.33x)

**Table 5.6** Computational time (s) and speedup (in parentheses) of the circular dam break simulations for GPU simulations

Mesh size	Xeon 1 Double	GTX 1050Ti Single	GTX 1050Ti Double	GTX 1070Ti Single	GTX 1070Ti Double	GTX 1080Ti Single	GTX 1080Ti Double
10k	0.16	0.04 (3.98x)	0.05 (2.93x)	0.03 (4.42x)	0.04 (4.00x)	0.04 (4.26x)	0.04 (4.05x)
50k	1.88	0.20 (9.28x)	0.34 (5.53x)	0.13 (11.1x)	0.21 (8.78x)	0.16 (11.81x)	0.19 (9.68x)

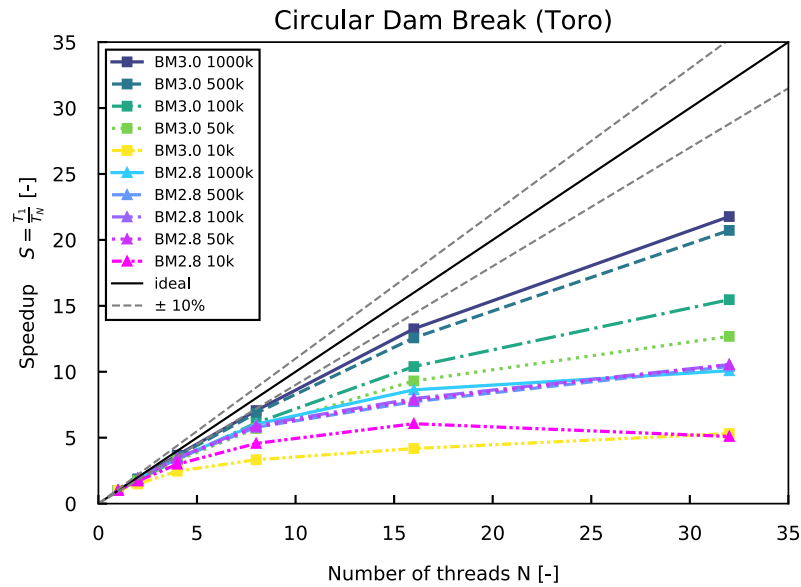


**Figure 5.4** Speedup for CPU (double precision) and GPU (single and double precision) backends for the circular dam break test case.

Mesh size	Xeon 1 Double	GTX	GTX	GTX	GTX	GTX	GTX
		1050Ti Single	1050Ti Double	1070Ti Single	1070Ti Double	1080Ti Single	1080Ti Double
100k	5.44	0.45 (12.13x)	0.84 (6.5x)	0.30 (15.54x)	0.49 (11.19x)	0.21 (16.83x)	0.42 (12.82x)
500k	63.63	3.27 (19.48x)	7.88 (8.08x)	1.96 (27.82x)	3.70 (17.21x)	2.02 (31.45x)	3.01 (21.14x)
1000k	190.39	8.55 (22.70x)	22.26 (8.55x)	5.53 (34.40x)	9.71 (19.61x)	4.70 (40.48x)	7.57 (25.16x)

Figure 5.4 shows the speedup achieved by the different backends. The sequential and OpenMP backends were run on processors of type Intel Xeon Gold 6154 (3.00GHz) with two sockets and with 18 cores per socket. The largest speedups are obtained for certain GPU cards using single precision. But also by using the GPU backend with double precision, the simulation time can be reduced significantly compared to sequential computation. It should be noted that the results of simulations with single and double precision can vary greatly depending on the problem. Further, it can be observed that the speedup greatly depends on the problem size. For smaller mesh, the speedup obtained by increasing the number of cores or by using GPU cards can be limited. The speedup increases particularly for meshes with a fine resolution (500K and 1000K) using GPU.

Figure 5.5 compares the speedup with the number of threads (CPU) between the different mesh resolutions. The simulations were performed using the CPU backend with Intel Xeon



**Figure 5.5** Speedup of the computational time  $[T_N]$  using different number of threads (CPU) compared to the time of the sequential backend Xeon1  $[T_1]$  for the circular dam break

Gold 6154 (3.00GHz) processors. The speedup is more effective for meshes with a fine resolution than for those of coarse resolution, where the increase in number of cores leads to a plateau.

#### 5.2.4.2 Windows

Tables 5.7 contains the execution time and speedup of all the simulations performed on different backends under Windows. The simulations were run on Prozessor Intel Xeon E-2174G (3.80GHz) processors with 4 cores and 2 threads per core (8 logical cores using HyperThreading). The performance scales adequately up to 4 cores. The use of 8 logical usings (HyperThreading) does not results in a significant improvement of performance compared to using 4 physical cores.

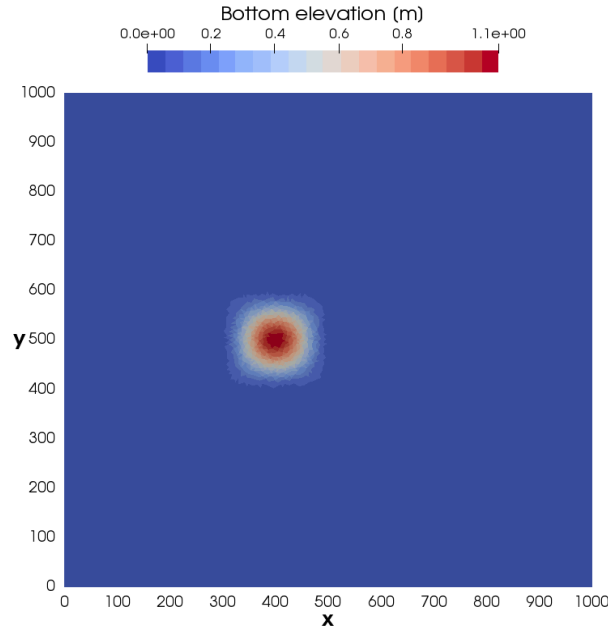
**Table 5.7** Computational time (s) and speedup (in parentheses) of the circular dam break simulations for CPU simulations under Windows 10 (Intel Xeon E-2174G, 3.80GHz)

Mesh size	Xeon 1 Double	Xeon 2 Double	Xeon 4 Double	Xeon 8 Double
10k	0.23	0.14 (1.67x)	0.08 (3.0x)	0.09 (2.49x)
50k	2.80	1.58 (1.77x)	0.98 (2.84x)	0.81 (3.44x)
100k	8.08	4.73 (1.71x)	3.16 (2.56x)	2.58 (3.13x)

---

Mesh size	Xeon 1 Double	Xeon 2 Double	Xeon 4 Double	Xeon 8 Double
500k	95.01	51.78 (1.83x)	30.55 (3.11x)	23.62 (4.02x)
1000k	275.02	149.84 (1.84x)	90.00 (3.06x)	67.25 (4.09x)

---



**Figure 5.6** Initial bottom elevation of the conical dune, maximum elevation of 1.1 m at  $[x=400, y=500]$ .

## 5.3 Conical Dune

### 5.3.1 Description

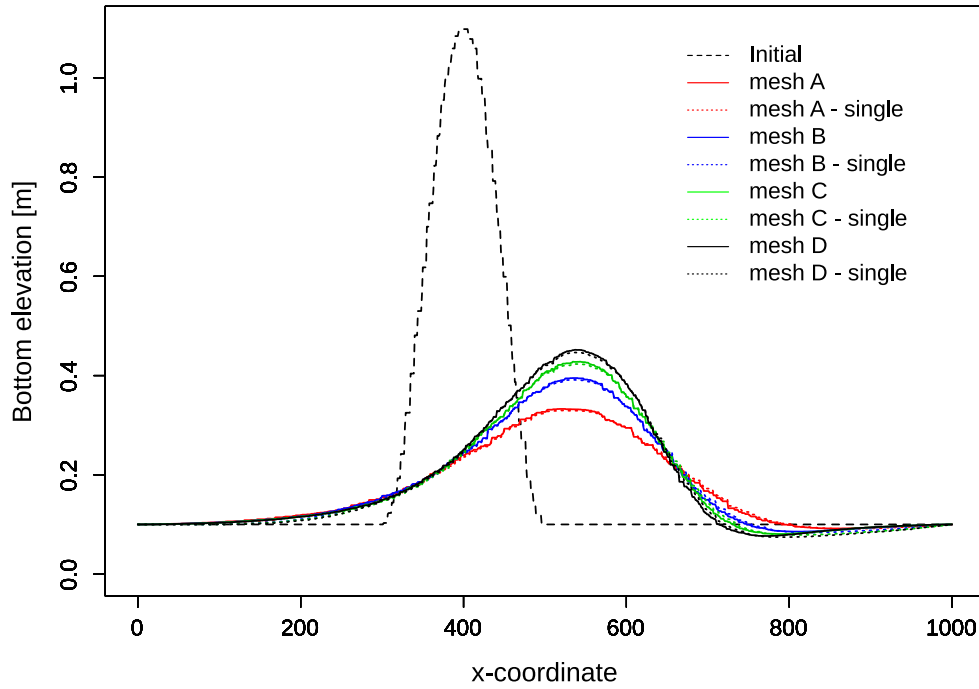
The conical dune test proposed by Hudson and Sweby (2005) is used to assess the accuracy of two dimensional morphodynamic models under sub-critical flow and without friction. The simulation describes the evolution of an initial conical dune of sediments that evolves in a star shaped pattern characterized by a spread angle  $\alpha_s$ . Numerical results are compared with the approximate analytical solution for the spread angle of the disturbance front proposed by de Vriend (1987).

### 5.3.2 Geometry and Initial Conditions

The computational mesh is a squared area of size 1000 x 1000 m with an initial bottom elevation  $z_{B,initial}$  (see Figure 5.6):

$$z_{B,initial} = \begin{cases} 0.1 + \sin^2\left(\frac{\pi(x-300)}{200}\right) \sin^2\left(\frac{\pi(y-400)}{200}\right) & \text{for } x \in [300, 500] \\ & \text{and } y \in [400, 600] \\ 0.1 & \text{otherwise} \end{cases} \quad (5.3)$$

An inflow boundary condition of Froude type is defined along the y-axis at  $x=0$  setting a constant unit discharge of 10 m/s. At  $x=1000$  m, a h-Q relation is defined as outflow boundary condition. The initial water depth is 10 m and the water surface elevation is kept constant at the boundaries. The Strickler coefficient is set to  $K_s = 10^7 [m^{1/3}/s]$  in order to obtain an almost frictionless simulation. The CFL number is set to 0.8, the



**Figure 5.7** Longitudinal bottom elevation profile at  $y=500$  m for initial conditions and after 100h of simulation for the meshes A, B, C and D with double and single precision

porosity to 0.4 and the morphodynamic boundary conditions are at equilibrium. Bed load transport is computed using the Grass formula with the parameters  $A_G = 0.001002$  and  $m_G = 3$  assuming a slow interaction between the sediment layer and the water flow. The analytical solution for the spread angle (eq. 5.4) proposed by de Vriend (1987) gives a value of  $\alpha_{s,ref} = 21.787^\circ$ .

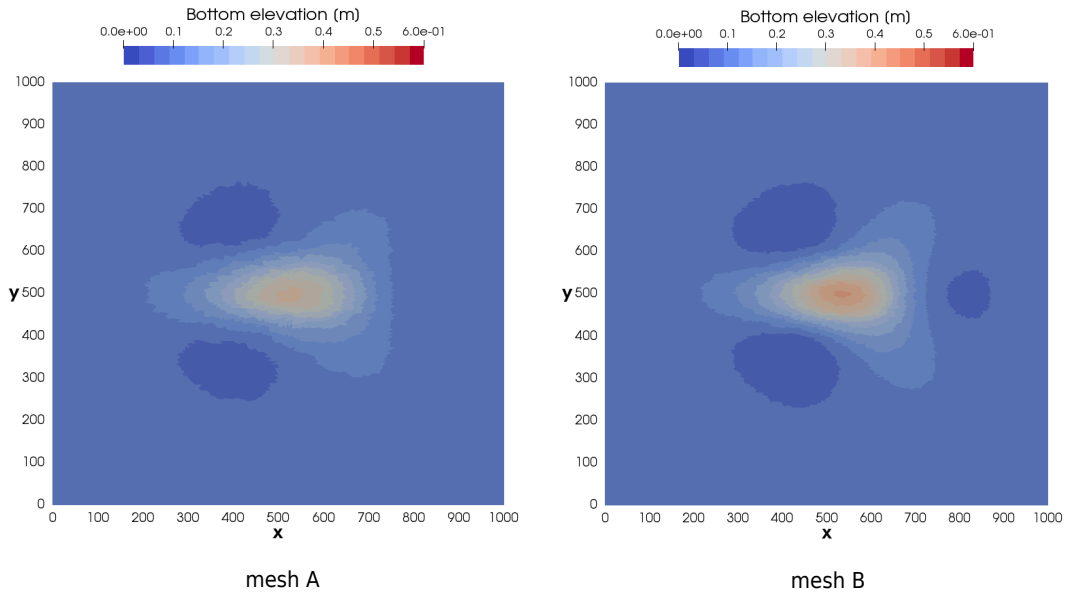
$$\alpha_s = \arctan\left(\frac{3\sqrt{3}(m_G - 1)}{9m_G - 1}\right) \quad (5.4)$$

The tests were performed with different computational mesh resolution, mesh A has 30160 cells (30k), mesh B 61201 cells (60k), mesh C 91083 cells (90k) and mesh D 126020 cells (120k). The simulation results were analyzed after 100 hours and each test was performed using different backends.

### 5.3.3 Results

Figure 5.7 shows the bed elevation profile at  $y = 500$  m along the  $x$ -coordinates at the beginning and after 100 h of the simulation for the four different meshes (A, B, C and D) performed with double and single precision. The results are exactly the same for all processor types, except for the simulation performed with CUDA-single, where the values are slightly different due to the reduced precision.

Table 5.8 summarizes the results obtained from the numerical simulations at  $t=100$ h for



**Figure 5.8** Planar view of the bottom elevation after 100 hours of simulation for the meshes A and B.

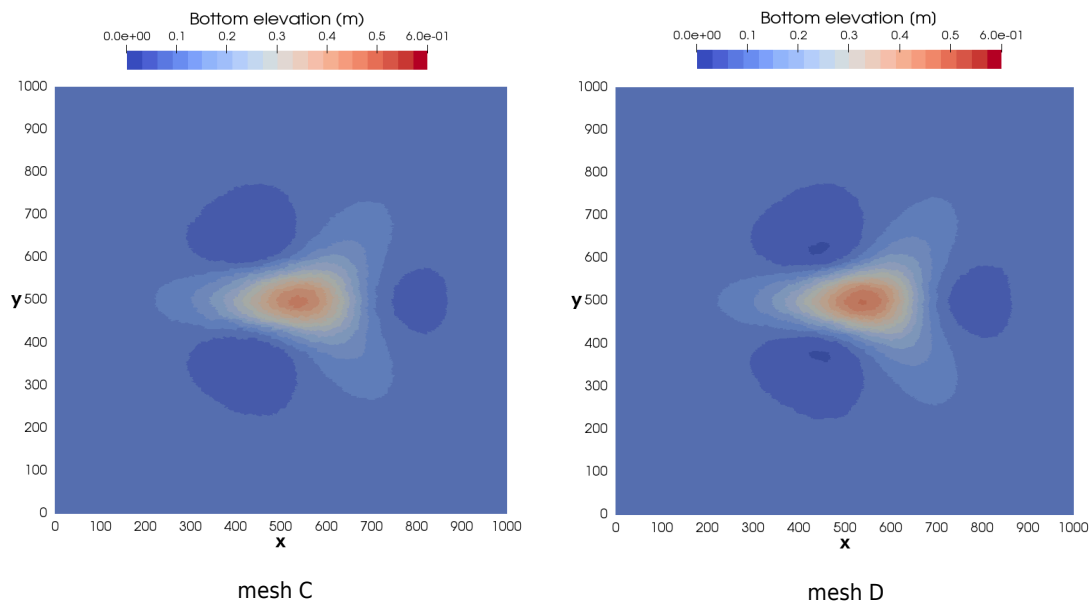
the different meshes using the backend types CPU and GPU-double. The spread angle  $\alpha_s$  is measured between  $y = 500$  m and the line passing through the initial conical dune center point and the farthest point in the  $x$ -direction where the conical dune spread for a similar bottom elevation (Figure 5.8 and Figure 5.9). The spread angle was measured at time  $t=25$ h, 50h, 75h and 100h for each mesh size and the mean value is reported in Table 5.8. The accuracy of the simulations is assessed by the relative deviation between the measure spread angle  $\alpha_s$  and the spread angle defined by de Vriend (1987)  $\alpha_{s,ref}$ .

**Table 5.8** Summary of the simulation results performed with CPU and GPU double at  $t=100$ h for different computational meshes

Mesh	Number of cells [-]	Max. cone elevation [m]	$\alpha_s$ [°]	Relative deviation [%]
A	30160	0.3327	25.57	16.0
B	61201	0.3950	25.12	14.2
C	91083	0.4279	23.94	9.4
D	126020	0.4518	23.35	6.9

The maximum bottom elevation after 100h increases with the mesh resolution while the spread angle decreases (Table 5.8). The results are more accurate for finer meshes (mesh C and mesh D, Figure 5.7), where less diffusive sediment transport is observed and thus the maximum bottom elevation after 100h is higher than for coarser meshes. This increase in accuracy for finer meshes is also observed in Figure 5.8 and Figure 5.9 by a well defined star shaped sediment transport and a spread angle that becomes closer to the value proposed by de Vriend (1987).





**Figure 5.9** Planar view of the bottom elevation after 100 hours of simulation for the meshes C and D.

## 5.4 Dam-break flow in a L-shaped channel with a mobile-bed

### 5.4.1 Description

This test case is based on the experiments described in Palumbo et al. (2008) and is a frequently used benchmark test case in literature (e.g. Siviglia et al. (2013), Soares-Frazão and Zech (2011)).

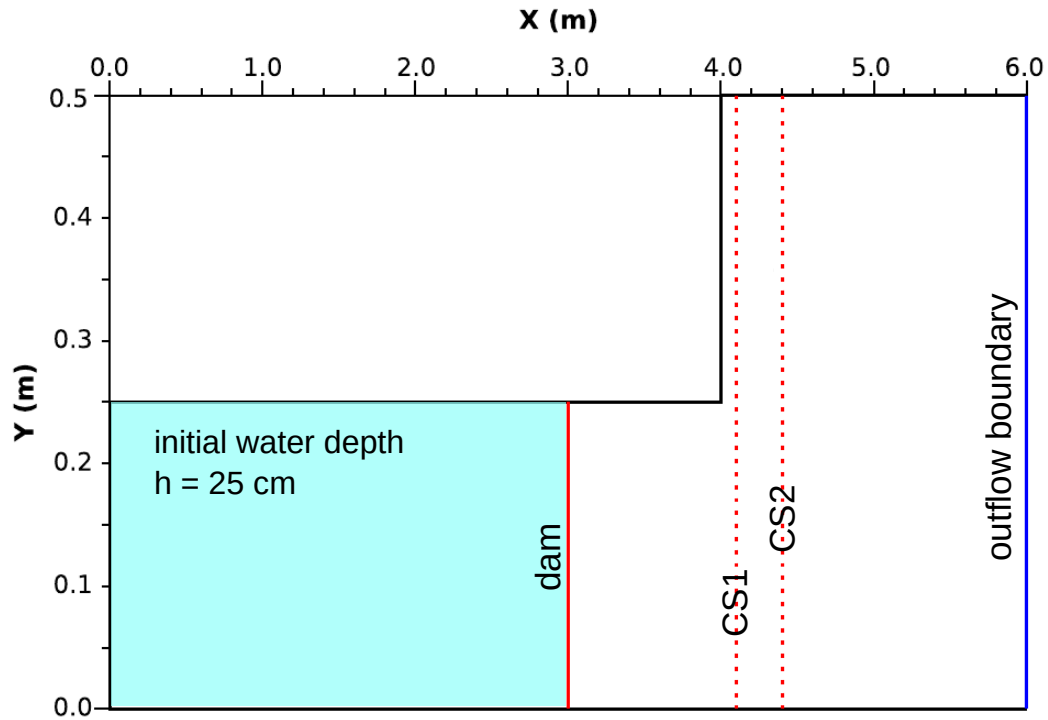
### 5.4.2 Geometry and initial conditions

This test case consist of a domain with a non-symmetrical sudden enlargement. The dimensions of the domain as illustrated in Figure 5.10. The sediment bed is composed of coarse uniform sand with grain size diameter of  $d_s = 1.82$  mm. The sediment porosity is  $p = 0.47$  and the sediment density  $\rho_s = 2680$  kg/m<sup>3</sup>. The Manning friction coefficient is set to  $n = 0.0167$  s/m<sup>1/3</sup>. The initial bed level is at an elevation of 0.1 m over the whole domain. The initial water depth upstream (left) of the dam is set to 0.25 m. The dam at is located at  $x = 3.0$  m. At time  $t = 0$  seconds, he dam is removed instantaneously, resulting in wave propagation and sediment transport.

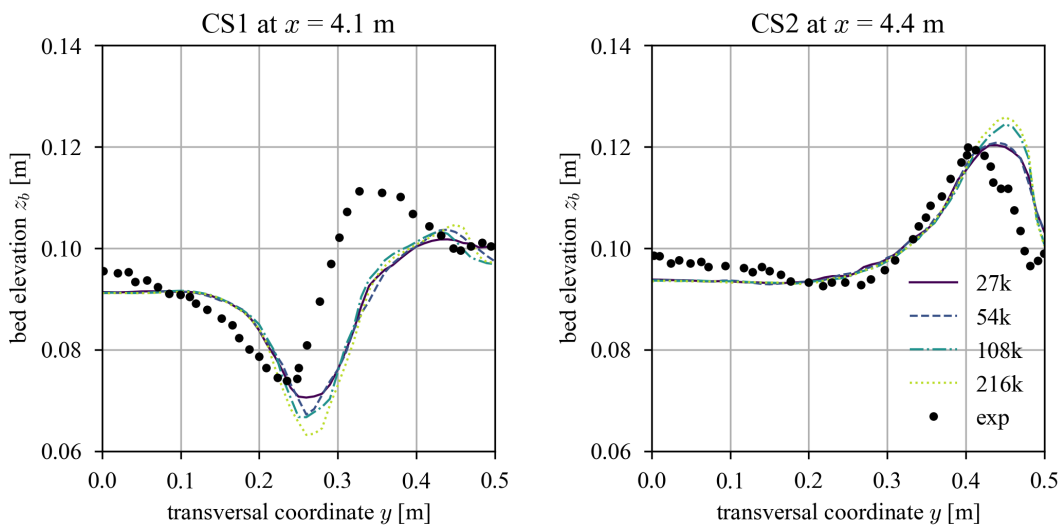
The computational domain is discretised by unstructured triangular cells. Four different sized computational grids were generated, resulting in grids with 27,000 cells (27k), 54,000 cells (54k), 108,000 cells (108k), 216,000 cells (216k). Wall boundary conditions are applied at the domain side walls. At the outflow boundary (right), a free-outflow boundary condition was applied. The sediment transport is calculated using the MPM\_ like formula with a pre-factor of 3.97 and an exponent of 1.5. The simulation lasts 12 seconds.

### 5.4.3 Results

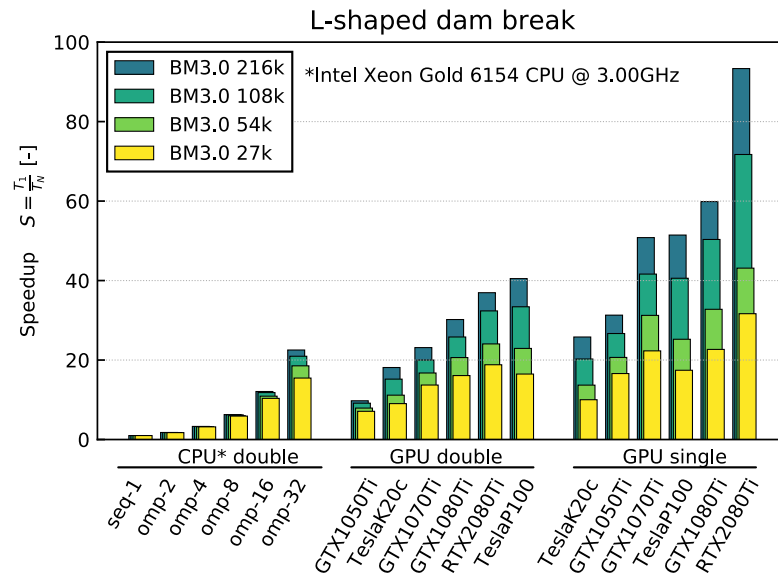
The simulated bed elevations at  $t = 12$ s are compared to the experimental results from Palumbo et al. (2008) in Figure 5.11. Overall, the simulated bed elevations are in acceptable agreement with experimental data. The scour in cross section CS1 ( $x = 4.1$  m) is matched well, while the deposition around  $y = 0.3$  m is underestimated. At cross-section CS2, the numerical results correspond well with the experimental data. In particular at location CS1, the observed deviations may be attributed to distinct 3D flow effects that are not captured by the 2D depth-averaged model (Siviglia et al. 2013, Xia et al. 2010).



**Figure 5.10** The numerical test setup for the dam-break flow in a L-shaped channel with a mobile-bed. Experimental and numerical bed elevations are compared along cross-sections CS1 and CS2. Modified from Siviglia et al. (2013).



**Figure 5.11** Experimental and numerical bed elevation are compared at cross-sections CS1 and CS2.



**Figure 5.12** Speedup of the computational time  $[T_N]$  for CPU (double) and GPU (single and double) compared to the time of the sequential backend Xeon1  $[T_1]$  for the L-shaped dam break test case.

## 5.4.4 Performance

### 5.4.4.1 Ubuntu

Simulations of this test case were performed on CPU- and GPU-backends. The largest execution times are obtained by using the sequential backend. By using the OpenMP-backend, the computation time can be reduced almost linearly with the number of cores. Also by using GPU-backends the computation time can be reduced in comparison to sequential computation, whereas the increase in performance depends on the GPU-card and the precision (single or double). The best performance using double precision is obtained with the card Tesla P100, while the RTX 2080Ti outperformed other cards when using single precision (see Tables 5.9, 5.10 and 5.11 or Figure 5.12). It should be noted that the results of simulations with single and double precision can vary greatly depending on the problem.

**Table 5.9** Computational time (s) and speedup  $S$  (inside brackets) of the L-shaped dam break simulations for CPU simulations (Intel Xeon Gold 6154, 3.00GHz)

Mesh size	Xeon 1 Double	Xeon 2 Double	Xeon 4 Double	Xeon 8 Double	Xeon 16 Double	Xeon 32 Double
30k	43.8	25.2 (1.7x)	13.7 (3.2x)	7.4 (5.9x)	4.2 (10.4x)	2.8 (15.5x)
60k	97.7	56.1 (1.7x)	30.4 (3.2x)	16.3 (6.0x)	9.0 (10.9x)	5.3 (18.5x)

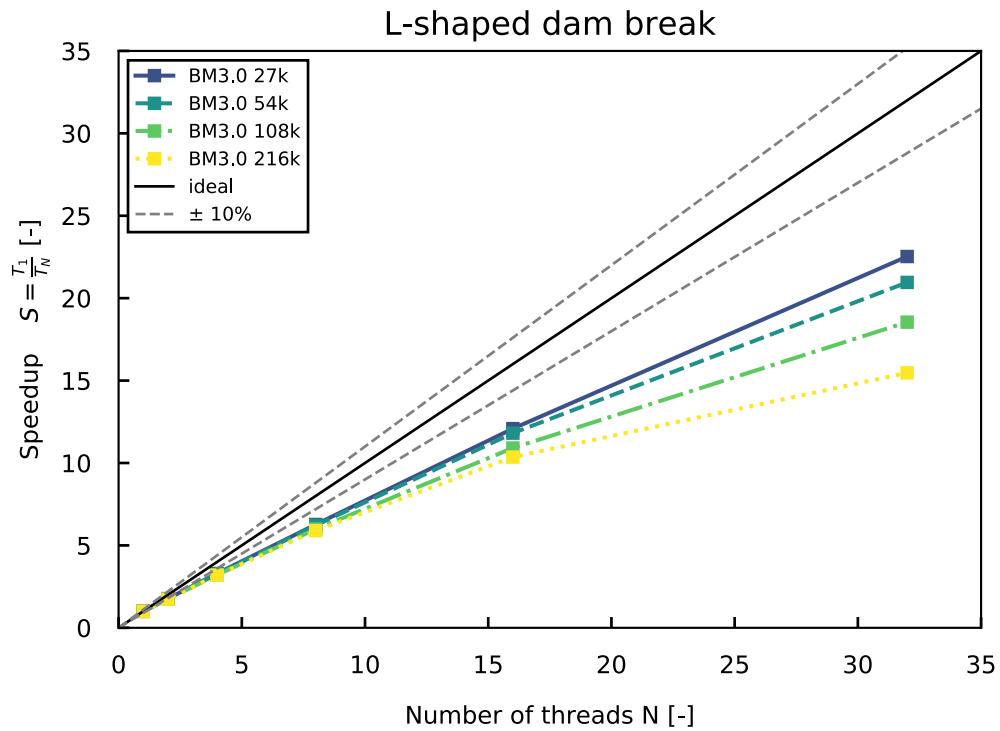
Mesh size	Xeon 1 Double	Xeon 2 Double	Xeon 4 Double	Xeon 8 Double	Xeon 16 Double	Xeon 32 Double
90k	357.3	202.7 (1.8x)	109.3 (3.3x)	57.5 (6.2x)	3.2 (11.8x)	17.0 (21.0x)
120k	1096.3	618.5 (1.8x)	332.6 (3.3x)	174.9 (6.3x)	90.7 (12.1x)	48.7 (22.5x)

**Table 5.10** Computational time (s) and speedup  $S$  (inside brackets) of the L-shaped dam break simulations for GPU simulations

Mesh size	Xeon 1 Double	RTX 2080Ti Single	RTX 2080Ti Double	Tesla K20 Single	Tesla K20 Double	Tesla P100 Single	Tesla P100 Double
30k	43.8	1.4 (31.7x)	2.3 (18.8x)	4.4 (10.0x)	4.8 (9.0x)	2.5 (17.4x)	2.7 (16.5x)
60k	97.7	2.3 (43.1x)	4.1 (24.1)	7.1 (13.7x)	8.8 (11.2x)	3.9 (25.2x)	4.3 (22.9x)
90k	357.3	5.0 (71.7x)	11.0 (32.4x)	17.6 (20.3x)	23.5 (15.2x)	8.8 (40.6x)	10.7 (33.4x)
120k	1096.3	11.7 (93.3x)	29.7 (37.0x)	42.5 (25.8x)	60.5 (18.1x)	21.3 (51.4x)	27.1 (40.5x)

**Table 5.11** Computational time (s) and speedup  $S$  (inside brackets) of the L-shaped dam break simulations for GPU simulations

Mesh size	Xeon 1 Double	GTX 1050 Single	GTX 1050 Double	GTX 1070 Single	GTX 1070 Double	GTX 1080 Single	GTX 1080 Double
30k	43.8	2.6 (16.6x)	6.2 (7.1x)	2.0 (22.3x)	3.2 (13.7x)	1.9 (22.7x)	2.7 (16.1x)
60k	97.7	4.7 (20.7x)	12.4 (7.9x)	3.1 (31.2x)	5.8 (16.7x)	3.0 (32.8x)	4.7 (20.69x)
90k	357.3	13.4 (26.7x)	39.1 (9.1x)	8.6 (41.6x)	17.9 (20.0x)	7.1 (50.3x)	13.8 (25.8x)
120k	1096.3	35.0 (31.3x)	112.4 (9.8x)	21.6 (50.8x)	47.4 (23.1x)	18.3 (59.9x)	36.3 (30.2x)



**Figure 5.13** Speedup of the L-shaped dam break test case performed on CPU for an increasing number of threads and different mesh sizes

In Figure 5.13, the speedup of the OpenMP-backend is illustrated for different mesh sizes on an Intel Xeon Gold 6154 (3.00GHz) processor on an Ubuntu system with 36 cores. The speedup of scales almost linearly up to 32 cores for this test case, indicating a good scaling behaviour.

#### 5.4.4.2 Windows

Tables 5.12 contains the execution time and speedup of the simulations performed on different backends under Windows. The simulations were run on Prozessor Intel Xeon E-2174G (3.80GHz) processors with 4 cores and 2 threads per core (8 logical cores using HyperThreading). The performance scales adequately up to 4 cores. The use of 8 logical usings (HyperThreading) does not results in a significant improvement of performance compared to using 4 physical cores.

**Table 5.12** Computational time (s) and speedup (in parentheses) for CPU simulations of L-shaped dam break with a mobile bed under Windows 10 (Intel Xeon E-2174G, 3.80GHz)

Mesh size	Xeon 1 Double	Xeon 2 Double	Xeon 4 Double	Xeon 8 Double
27k	70.0	38.3 (1.83x)	25.6 (2.73x)	16.8 (4.18x)

---

Mesh size	Xeon 1 Double	Xeon 2 Double	Xeon 4 Double	Xeon 8 Double
54k	158.8	87.3 (1.82x)	58.1 (2.73x)	37.5 (4.23x)
108k	578.5	319.6 (1.81x)	212.3 (2.72x)	137.7 (4.20x)
216k	1770.6	986.0 (1.80x)	680.9 (2.60x)	427.8 (4.14x)

---

# 6

---

## References

- de Vriend, H.J. (1987). 2DH mathematical modelling of morphological evolutions in shallow water. *Coastal engineering*, 11(1): 1–27.
- Ginzburg, I., Carlier, J.P. and Kao, C. (2004). Lattice Boltzmann approach to Richard's equation. *Proceedings of the CMWR XV, CT Miller*, 583–597. Chapel Hill, NC, USA.
- Hudson, J. and Sweby, P.K. (2005). A high-resolution scheme for the equations governing 2D bed-load sediment transport. *International Journal for Numerical Methods in Fluids*, 47: 1085–1091.
- 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.
- Palumbo, A., Soares Frazao, S., Goutière, L., Pianese, D. and Zech, Y. (2008). Dam-break flow on mobile bed in a channel with a sudden enlargement. *Proceedings of the river flow 2008 conference*,.
- 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.
- Siviglia, A., Stecca, G., Vanzo, D., Zolezzi, G. and Toro, E.F. (2013). Numerical modelling of two-dimensional morphodynamics with applications to river bars and bifurcations. *Advances in Water Resources*, 52: 243–260.
- Soares-Frazão, S. and Zech, Y. (2011). HLLC scheme with novel wave-speed estimators appropriate for two-dimensional shallow-water flow on erodible bed. *International Journal for Numerical Methods in Fluids*, 66.
- 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.
- Vanzo, D., Siviglia, A. and Toro, E.F. (2016). Pollutant transport by shallow water equations on unstructured meshes: Hyperbolization of the model and numerical solution via a novel flux splitting scheme. *Journal of Computational Physics*, 321: 1–20.
- 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 Cislserova, 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.H. 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.

**BASIC SIMULATION ENVIRONMENT  
FOR MODELLING OF ENVIRONMENTAL  
FLOWS AND NATURAL HAZARDS**

---

# **APPENDIX**

**VERSION 4.1.0  
JUNE 2024**



**BASEMENT**



---

## Third Party Software

### 1.1 Third party software licenses

Abseil

Apache License

Version 2.0, January 2004  
<https://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

#### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
  - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
  - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
  - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
  - (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained

within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. **Disclaimer of Warranty.** Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. **Limitation of Liability.** In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. **Accepting Warranty or Additional Liability.** While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<https://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

## **Brotli**

Copyright (c) 2009, 2010, 2013-2016 by the Brotli Authors.

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

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

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

## **Bzip2**

-----  
This program, "bzip2", the associated library "libbzip2", and all documentation, are copyright (C) 1996-2019 Julian R Seward. All rights reserved.

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

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.
3. Altered source versions must be plainly marked as such, and must

not be misrepresented as being the original software.

4. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

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

Julian Seward, jseward@acm.org  
bzip2/libbzip2 version 1.0.8 of 13 July 2019

---

## Curl

### COPYRIGHT AND PERMISSION NOTICE

Copyright (c) 1996 - 2022, Daniel Stenberg, <daniel@haxx.se>, and many contributors, see the THANKS file.

All rights reserved.

Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

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

Except as contained in this notice, the name of a copyright holder shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization of the copyright holder.

## Double-conversion

Copyright 2006-2011, the V8 project authors. All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

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

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT



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

### Egl-registry

Copyright (c) 2008-2018 The Khronos Group Inc.

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

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

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

---

Copyright 2013-2020 The Khronos Group Inc.  
Copyright 2007-2020 The Khronos Group Inc.

SPDX-License-Identifier: Apache-2.0

### Eigen3

Eigen is primarily MPL2 licensed. See COPYING.MPL2 and these links:  
<http://www.mozilla.org/MPL/2.0/>  
<http://www.mozilla.org/MPL/2.0/FAQ.html>

Some files contain third-party code under BSD or LGPL licenses, whence the other COPYING.\* files here.

All the LGPL code is either LGPL 2.1-only, or LGPL 2.1-or-later.  
For this reason, the COPYING.LGPL file contains the LGPL 2.1 text.

If you want to guarantee that the Eigen code that you are #including is licensed under the MPL2 and possibly more permissive licenses (like BSD), #define this preprocessor symbol:

```
EIGEN_MPL2_ONLY
```

For example, with most compilers, you could add this to your project CXXFLAGS:  
-DEIGEN\_MPL2\_ONLY

This will cause a compilation error to be generated if you #include any code that is LGPL licensed.

---

Copyright (c) 2011, Intel Corporation. All rights reserved.

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

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

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

---

Minpack Copyright Notice (1999) University of Chicago. All rights reserved

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

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The end-user documentation included with the redistribution, if any, must include the following acknowledgment:

"This product includes software developed by the University of Chicago, as Operator of Argonne National Laboratory.

Alternately, this acknowledgment may appear in the software itself, if and wherever such third-party acknowledgments normally appear.

4. WARRANTY DISCLAIMER. THE SOFTWARE IS SUPPLIED "AS IS" WITHOUT WARRANTY OF ANY KIND. THE COPYRIGHT HOLDER, THE UNITED STATES, THE UNITED STATES DEPARTMENT OF ENERGY, AND THEIR EMPLOYEES: (1) DISCLAIM ANY WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE OR NON-INFRINGEMENT, (2) DO NOT ASSUME ANY LEGAL LIABILITY OR RESPONSIBILITY FOR THE ACCURACY, COMPLETENESS, OR USEFULNESS OF THE SOFTWARE, (3) DO NOT REPRESENT THAT USE OF THE SOFTWARE WOULD NOT INFRINGE PRIVATELY OWNED RIGHTS, (4) DO NOT WARRANT THAT THE SOFTWARE WILL FUNCTION UNINTERRUPTED, THAT IT IS ERROR-FREE OR THAT ANY ERRORS WILL BE CORRECTED.

5. LIMITATION OF LIABILITY. IN NO EVENT WILL THE COPYRIGHT HOLDER, THE UNITED STATES, THE UNITED STATES DEPARTMENT OF ENERGY, OR THEIR EMPLOYEES: BE LIABLE FOR ANY INDIRECT, INCIDENTAL, CONSEQUENTIAL, SPECIAL OR PUNITIVE DAMAGES OF ANY KIND OR NATURE, INCLUDING BUT NOT LIMITED TO LOSS OF PROFITS OR LOSS OF DATA, FOR ANY REASON WHATSOEVER, WHETHER

SUCH LIABILITY IS ASSERTED ON THE BASIS OF CONTRACT, TORT (INCLUDING NEGLIGENCE OR STRICT LIABILITY), OR OTHERWISE, EVEN IF ANY OF SAID PARTIES HAS BEEN WARNED OF THE POSSIBILITY OF SUCH LOSS OR DAMAGES.

### Expat

Copyright (c) 1998-2000 Thai Open Source Software Center Ltd and Clark Cooper  
Copyright (c) 2001-2019 Expat maintainers

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

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

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

### Freetype

#### FREETYPE LICENSES

The FreeType 2 font engine is copyrighted work and cannot be used legally without a software license. In order to make this project usable to a vast majority of developers, we distribute it under two mutually exclusive open-source licenses.

This means that *you* must choose *one* of the two licenses described below, then obey all its terms and conditions when using FreeType 2 in any of your projects or products.

- The FreeType License, found in the file ``docs/FTL.TXT``, which is similar to the original BSD license *with* an advertising clause that forces you to explicitly cite the FreeType project in your product's documentation. All details are in the license file. This license is suited to products which don't use the GNU General Public License.

Note that this license is compatible to the GNU General Public License version 3, but not version 2.

- The GNU General Public License version 2, found in ``docs/GPLv2.TXT`` (any later version can be used also), for programs which already use the GPL. Note that the FTL is incompatible with GPLv2 due to its advertisement clause.

The contributed BDF and PCF drivers come with a license similar to that of the X Window System. It is compatible to the above two licenses (see files ``src/bdf/README`` and ``src/pcf/README``). The same holds for the source code files ``src/base/fthash.c`` and ``include/freetype/internal/fthash.h``; they were part of the BDF driver in earlier FreeType versions.

The gzip module uses the zlib license (see ``src/gzip/zlib.h``) which too is compatible to the above two licenses.

The MD5 checksum support (only used for debugging in development builds) is in the public domain.

--- end of LICENSE.TXT ---

## Glew

The OpenGL Extension Wrangler Library  
Copyright (C) 2002-2007, Milan Ikits <milan ikits[at]ieee org>  
Copyright (C) 2002-2007, Marcelo E. Magallon <mmagallo[at]debian org>  
Copyright (C) 2002, Lev Povalahev  
All rights reserved.

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

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- \* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- \* The name of the author may be used to endorse or promote products derived from this software without specific prior written permission.

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

Mesa 3-D graphics library  
Version: 7.0

Copyright (C) 1999-2007 Brian Paul All Rights Reserved.

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

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

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

Copyright (c) 2007 The Khronos Group Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and/or associated documentation files (the "Materials"), to deal in the Materials without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Materials, and to

permit persons to whom the Materials are furnished to do so, subject to the following conditions:

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

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

### Glibc-queue

Copyright (C) 1991-2015 Free Software Foundation, Inc.

The GNU C Library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version.

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

You should have received a copy of the GNU Lesser General Public License along with the GNU C Library; if not, write to the Free Software Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

\* All code incorporated from 4.4 BSD is distributed under the following license:

Copyright (C) 1991 Regents of the University of California.  
All rights reserved.

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

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. [This condition was removed.]
4. Neither the name of the University nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

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

### Gtest

Copyright 2008, Google Inc.

All rights reserved.

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

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

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

## Harfbuzz

HarfBuzz is licensed under the so-called "Old MIT" license. Details follow. For parts of HarfBuzz that are licensed under different licenses see individual files names COPYING in subdirectories where applicable.

Copyright © 2010,2011,2012,2013,2014,2015,2016,2017,2018,2019,2020 Google, Inc.  
Copyright © 2018,2019,2020 Ebrahim Byagowi  
Copyright © 2019,2020 Facebook, Inc.  
Copyright © 2012 Mozilla Foundation  
Copyright © 2011 Codethink Limited  
Copyright © 2008,2010 Nokia Corporation and/or its subsidiary(-ies)  
Copyright © 2009 Keith Stribley  
Copyright © 2009 Martin Hosken and SIL International  
Copyright © 2007 Chris Wilson  
Copyright © 2005,2006,2020,2021 Behdad Esfahbod  
Copyright © 2005 David Turner  
Copyright © 2004,2007,2008,2009,2010 Red Hat, Inc.  
Copyright © 1998-2004 David Turner and Werner Lemberg

For full copyright notices consult the individual files in the package.

Permission is hereby granted, without written agreement and without license or royalty fees, to use, copy, modify, and distribute this software and its documentation for any purpose, provided that the above copyright notice and the following two paragraphs appear in all copies of this software.

IN NO EVENT SHALL THE COPYRIGHT HOLDER BE LIABLE TO ANY PARTY FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OF THIS SOFTWARE AND ITS DOCUMENTATION, EVEN IF THE COPYRIGHT HOLDER HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

THE COPYRIGHT HOLDER SPECIFICALLY DISCLAIMS ANY WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE SOFTWARE PROVIDED HEREUNDER IS ON AN "AS IS" BASIS, AND THE COPYRIGHT HOLDER HAS NO OBLIGATION TO PROVIDE MAINTENANCE, SUPPORT, UPDATES, ENHANCEMENTS, OR MODIFICATIONS.

**Hdf5**

Copyright Notice and License Terms for  
HDF5 (Hierarchical Data Format 5) Software Library and Utilities

HDF5 (Hierarchical Data Format 5) Software Library and Utilities  
Copyright 2006 by The HDF Group.

NCSA HDF5 (Hierarchical Data Format 5) Software Library and Utilities  
Copyright 1998-2006 by The Board of Trustees of the University of Illinois.

All rights reserved.

Redistribution and use in source and binary forms, with or without  
modification, are permitted for any purpose (including commercial purposes)  
provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice,  
this list of conditions, and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice,  
this list of conditions, and the following disclaimer in the documentation  
and/or materials provided with the distribution.
3. Neither the name of The HDF Group, the name of the University, nor the  
name of any Contributor may be used to endorse or promote products derived  
from this software without specific prior written permission from  
The HDF Group, the University, or the Contributor, respectively.

**DISCLAIMER:**

THIS SOFTWARE IS PROVIDED BY THE HDF GROUP AND THE CONTRIBUTORS  
"AS IS" WITH NO WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED. IN NO  
EVENT SHALL THE HDF GROUP OR THE CONTRIBUTORS BE LIABLE FOR ANY DAMAGES  
SUFFERED BY THE USERS ARISING OUT OF THE USE OF THIS SOFTWARE, EVEN IF  
ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

You are under no obligation whatsoever to provide any bug fixes, patches, or  
upgrades to the features, functionality or performance of the source code  
("Enhancements") to anyone; however, if you choose to make your Enhancements  
available either publicly, or directly to The HDF Group, without imposing a  
separate written license agreement for such Enhancements, then you hereby  
grant the following license: a non-exclusive, royalty-free perpetual license  
to install, use, modify, prepare derivative works, incorporate into other  
computer software, distribute, and sublicense such enhancements or derivative  
works thereof, in binary and source code form.

Limited portions of HDF5 were developed by Lawrence Berkeley National  
Laboratory (LBNL). LBNL's Copyright Notice and Licensing Terms can be  
found here: COPYING\_LBNL\_HDF5 file in this directory or at  
[http://support.hdfgroup.org/ftp/HDF5/releases/COPYING\\_LBNL\\_HDF5](http://support.hdfgroup.org/ftp/HDF5/releases/COPYING_LBNL_HDF5).

Contributors: National Center for Supercomputing Applications (NCSA) at  
the University of Illinois, Fortner Software, Unidata Program Center  
(netCDF), The Independent JPEG Group (JPEG), Jean-loup Gailly and Mark Adler  
(gzip), and Digital Equipment Corporation (DEC).

Portions of HDF5 were developed with support from the Lawrence Berkeley  
National Laboratory (LBNL) and the United States Department of Energy  
under Prime Contract No. DE-AC02-05CH11231.

Portions of HDF5 were developed with support from Lawrence Livermore National Laboratory and the United States Department of Energy under Prime Contract No. DE-AC52-07NA27344.

-----

Portions of HDF5 were developed with support from the University of California, Lawrence Livermore National Laboratory (UC LLNL). The following statement applies to those portions of the product and must be retained in any redistribution of source code, binaries, documentation, and/or accompanying materials:

This work was partially produced at the University of California, Lawrence Livermore National Laboratory (UC LLNL) under contract no. W-7405-ENG-48 (Contract 48) between the U.S. Department of Energy (DOE) and The Regents of the University of California (University) for the operation of UC LLNL.

**DISCLAIMER:**

THIS WORK WAS PREPARED AS AN ACCOUNT OF WORK SPONSORED BY AN AGENCY OF THE UNITED STATES GOVERNMENT. NEITHER THE UNITED STATES GOVERNMENT NOR THE UNIVERSITY OF CALIFORNIA NOR ANY OF THEIR EMPLOYEES, MAKES ANY WARRANTY, EXPRESS OR IMPLIED, OR ASSUMES ANY LIABILITY OR RESPONSIBILITY FOR THE ACCURACY, COMPLETENESS, OR USEFULNESS OF ANY INFORMATION, APPARATUS, PRODUCT, OR PROCESS DISCLOSED, OR REPRESENTS THAT ITS USE WOULD NOT INFRINGE PRIVATELY- OWNED RIGHTS. REFERENCE HEREIN TO ANY SPECIFIC COMMERCIAL PRODUCTS, PROCESS, OR SERVICE BY TRADE NAME, TRADEMARK, MANUFACTURER, OR OTHERWISE, DOES NOT NECESSARILY CONSTITUTE OR IMPLY ITS ENDORSEMENT, RECOMMENDATION, OR FAVORING BY THE UNITED STATES GOVERNMENT OR THE UNIVERSITY OF CALIFORNIA. THE VIEWS AND OPINIONS OF AUTHORS EXPRESSED HEREIN DO NOT NECESSARILY STATE OR REFLECT THOSE OF THE UNITED STATES GOVERNMENT OR THE UNIVERSITY OF CALIFORNIA, AND SHALL NOT BE USED FOR ADVERTISING OR PRODUCT ENDORSEMENT PURPOSES.

-----

**HighFive**

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

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

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

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

**Icu**

UNICODE, INC. LICENSE AGREEMENT - DATA FILES AND SOFTWARE



See Terms of Use <<https://www.unicode.org/copyright.html>>  
for definitions of Unicode Inc.'s Data Files and Software.

NOTICE TO USER: Carefully read the following legal agreement.  
BY DOWNLOADING, INSTALLING, COPYING OR OTHERWISE USING UNICODE INC.'S  
DATA FILES ("DATA FILES"), AND/OR SOFTWARE ("SOFTWARE"),  
YOU UNEQUIVOCALLY ACCEPT, AND AGREE TO BE BOUND BY, ALL OF THE  
TERMS AND CONDITIONS OF THIS AGREEMENT.  
IF YOU DO NOT AGREE, DO NOT DOWNLOAD, INSTALL, COPY, DISTRIBUTE OR USE  
THE DATA FILES OR SOFTWARE.

#### COPYRIGHT AND PERMISSION NOTICE

Copyright © 1991-2022 Unicode, Inc. All rights reserved.  
Distributed under the Terms of Use in <https://www.unicode.org/copyright.html>.

Permission is hereby granted, free of charge, to any person obtaining  
a copy of the Unicode data files and any associated documentation  
(the "Data Files") or Unicode software and any associated documentation  
(the "Software") to deal in the Data Files or Software  
without restriction, including without limitation the rights to use,  
copy, modify, merge, publish, distribute, and/or sell copies of  
the Data Files or Software, and to permit persons to whom the Data Files  
or Software are furnished to do so, provided that either  
(a) this copyright and permission notice appear with all copies  
of the Data Files or Software, or  
(b) this copyright and permission notice appear in associated  
Documentation.

THE DATA FILES AND SOFTWARE ARE PROVIDED "AS IS", WITHOUT WARRANTY OF  
ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE  
WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND  
NONINFRINGEMENT OF THIRD PARTY RIGHTS.  
IN NO EVENT SHALL THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS  
NOTICE BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL INDIRECT OR CONSEQUENTIAL  
DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE,  
DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER  
TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR  
PERFORMANCE OF THE DATA FILES OR SOFTWARE.

Except as contained in this notice, the name of a copyright holder  
shall not be used in advertising or otherwise to promote the sale,  
use or other dealings in these Data Files or Software without prior  
written authorization of the copyright holder.

---

#### Third-Party Software Licenses

This section contains third-party software notices and/or additional  
terms for licensed third-party software components included within ICU  
libraries.

---

ICU License - ICU 1.8.1 to ICU 57.1

#### COPYRIGHT AND PERMISSION NOTICE

Copyright (c) 1995-2016 International Business Machines Corporation and others  
All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining  
a copy of this software and associated documentation files (the  
"Software"), to deal in the Software without restriction, including  
without limitation the rights to use, copy, modify, merge, publish,  
distribute, and/or sell copies of the Software, and to permit persons  
to whom the Software is furnished to do so, provided that the above  
copyright notice(s) and this permission notice appear in all copies of  
the Software and that both the above copyright notice(s) and this

permission notice appear in supporting documentation.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Except as contained in this notice, the name of a copyright holder shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization of the copyright holder.

All trademarks and registered trademarks mentioned herein are the property of their respective owners.

-----  
Chinese/Japanese Word Break Dictionary Data (cjdict.txt)

```
# The Google Chrome software developed by Google is licensed under
# the BSD license. Other software included in this distribution is
# provided under other licenses, as set forth below.
#
# The BSD License
# http://opensource.org/licenses/bsd-license.php
# Copyright (C) 2006-2008, Google Inc.
#
# All rights reserved.
#
# Redistribution and use in source and binary forms, with or without
# modification, are permitted provided that the following conditions are met:
#
# Redistributions of source code must retain the above copyright notice,
# this list of conditions and the following disclaimer.
# Redistributions in binary form must reproduce the above
# copyright notice, this list of conditions and the following
# disclaimer in the documentation and/or other materials provided with
# the distribution.
# Neither the name of Google Inc. nor the names of its
# contributors may be used to endorse or promote products derived from
# this software without specific prior written permission.
#
# THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND
# CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES,
# INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
# MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
# DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
# LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
# CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
# SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR
# BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF
# LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
# NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
# SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
#
#
# The word list in cjdict.txt are generated by combining three word lists
# listed below with further processing for compound word breaking. The
# frequency is generated with an iterative training against Google web
# corpora.
#
# * Libtabe (Chinese)
# - https://sourceforge.net/project/?group_id=1519
# - Its license terms and conditions are shown below.
```

```

#
# * IPADIC (Japanese)
#   - http://chasen.aist-nara.ac.jp/chasen/distribution.html
#   - Its license terms and conditions are shown below.
#
# -----COPYING.libtabe ---- BEGIN-----
#
# /*
# * Copyright (c) 1999 TaBE Project.
# * Copyright (c) 1999 Pai-Hsiang Hsiao.
# * All rights reserved.
# *
# * Redistribution and use in source and binary forms, with or without
# * modification, are permitted provided that the following conditions
# * are met:
# *
# * . Redistributions of source code must retain the above copyright
# * notice, this list of conditions and the following disclaimer.
# * . Redistributions in binary form must reproduce the above copyright
# * notice, this list of conditions and the following disclaimer in
# * the documentation and/or other materials provided with the
# * distribution.
# * . Neither the name of the TaBE Project nor the names of its
# * contributors may be used to endorse or promote products derived
# * from this software without specific prior written permission.
# *
# * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
# * "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
# * LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS
# * FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE
# * REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT,
# * INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
# * (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
# * SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
# * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
# * STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
# * ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED
# * OF THE POSSIBILITY OF SUCH DAMAGE.
# */
#
# /*
# * Copyright (c) 1999 Computer Systems and Communication Lab,
# * Institute of Information Science, Academia
# * Sinica. All rights reserved.
# *
# * Redistribution and use in source and binary forms, with or without
# * modification, are permitted provided that the following conditions
# * are met:
# *
# * . Redistributions of source code must retain the above copyright
# * notice, this list of conditions and the following disclaimer.
# * . Redistributions in binary form must reproduce the above copyright
# * notice, this list of conditions and the following disclaimer in
# * the documentation and/or other materials provided with the
# * distribution.
# * . Neither the name of the Computer Systems and Communication Lab
# * nor the names of its contributors may be used to endorse or
# * promote products derived from this software without specific
# * prior written permission.
# *
# * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
# * "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
# * LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS
# * FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE
# * REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT,
# * INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
# * (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
# * SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
# * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
# * STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)

```

```
# * ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED
# * OF THE POSSIBILITY OF SUCH DAMAGE.
# */
#
# Copyright 1996 Chih-Hao Tsai @ Beckman Institute,
# University of Illinois
# c-tsai4@uiuc.edu http://casper.beckman.uiuc.edu/~c-tsai4
# -----COPYING.libtabe-----END-----
#
# -----COPYING.ipadic-----BEGIN-----
#
# Copyright 2000, 2001, 2002, 2003 Nara Institute of Science
# and Technology. All Rights Reserved.
#
# Use, reproduction, and distribution of this software is permitted.
# Any copy of this software, whether in its original form or modified,
# must include both the above copyright notice and the following
# paragraphs.
#
# Nara Institute of Science and Technology (NAIST),
# the copyright holders, disclaims all warranties with regard to this
# software, including all implied warranties of merchantability and
# fitness, in no event shall NAIST be liable for
# any special, indirect or consequential damages or any damages
# whatsoever resulting from loss of use, data or profits, whether in an
# action of contract, negligence or other tortuous action, arising out
# of or in connection with the use or performance of this software.
#
# A large portion of the dictionary entries
# originate from ICOT Free Software. The following conditions for ICOT
# Free Software applies to the current dictionary as well.
#
# Each User may also freely distribute the Program, whether in its
# original form or modified, to any third party or parties, PROVIDED
# that the provisions of Section 3 ("NO WARRANTY") will ALWAYS appear
# on, or be attached to, the Program, which is distributed substantially
# in the same form as set out herein and that such intended
# distribution, if actually made, will neither violate or otherwise
# contravene any of the laws and regulations of the countries having
# jurisdiction over the User or the intended distribution itself.
#
# NO WARRANTY
#
# The program was produced on an experimental basis in the course of the
# research and development conducted during the project and is provided
# to users as so produced on an experimental basis. Accordingly, the
# program is provided without any warranty whatsoever, whether express,
# implied, statutory or otherwise. The term "warranty" used herein
# includes, but is not limited to, any warranty of the quality,
# performance, merchantability and fitness for a particular purpose of
# the program and the nonexistence of any infringement or violation of
# any right of any third party.
#
# Each user of the program will agree and understand, and be deemed to
# have agreed and understood, that there is no warranty whatsoever for
# the program and, accordingly, the entire risk arising from or
# otherwise connected with the program is assumed by the user.
#
# Therefore, neither ICOT, the copyright holder, or any other
# organization that participated in or was otherwise related to the
# development of the program and their respective officials, directors,
# officers and other employees shall be held liable for any and all
# damages, including, without limitation, general, special, incidental
# and consequential damages, arising out of or otherwise in connection
# with the use or inability to use the program or any product, material
# or result produced or otherwise obtained by using the program,
# regardless of whether they have been advised of, or otherwise had
# knowledge of, the possibility of such damages at any time during the
```

```
# project or thereafter. Each user will be deemed to have agreed to the
# foregoing by his or her commencement of use of the program. The term
# "use" as used herein includes, but is not limited to, the use,
# modification, copying and distribution of the program and the
# production of secondary products from the program.
#
# In the case where the program, whether in its original form or
# modified, was distributed or delivered to or received by a user from
# any person, organization or entity other than ICOT, unless it makes or
# grants independently of ICOT any specific warranty to the user in
# writing, such person, organization or entity, will also be exempted
# from and not be held liable to the user for any such damages as noted
# above as far as the program is concerned.
#
# -----COPYING.ipadic-----END-----
```

-----

Lao Word Break Dictionary Data (laodict.txt)

```
# Copyright (C) 2016 and later: Unicode, Inc. and others.
# License & terms of use: http://www.unicode.org/copyright.html
# Copyright (c) 2015 International Business Machines Corporation
# and others. All Rights Reserved.
#
# Project: https://github.com/rober42539/lao-dictionary
# Dictionary: https://github.com/rober42539/lao-dictionary/laodict.txt
# License: https://github.com/rober42539/lao-dictionary/LICENSE.txt
#         (copied below)
#
# This file is derived from the above dictionary version of Nov 22, 2020
# -----
# Copyright (C) 2013 Brian Eugene Wilson, Robert Martin Campbell.
# All rights reserved.
#
# Redistribution and use in source and binary forms, with or without
# modification, are permitted provided that the following conditions are met:
#
# Redistributions of source code must retain the above copyright notice, this
# list of conditions and the following disclaimer. Redistributions in binary
# form must reproduce the above copyright notice, this list of conditions and
# the following disclaimer in the documentation and/or other materials
# provided with the distribution.
#
# THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
# "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
# LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS
# FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE
# COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT,
# INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
# (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
# SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
# HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
# STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
# ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED
# OF THE POSSIBILITY OF SUCH DAMAGE.
# -----
```

-----

Burmese Word Break Dictionary Data (burmesedict.txt)

```
# Copyright (c) 2014 International Business Machines Corporation
# and others. All Rights Reserved.
#
# This list is part of a project hosted at:
#   github.com/kanyawtech/myanmar-karen-word-lists
#
# -----
# Copyright (c) 2013, LeRoy Benjamin Sharon
```

```

# All rights reserved.
#
# Redistribution and use in source and binary forms, with or without
# modification, are permitted provided that the following conditions
# are met: Redistributions of source code must retain the above
# copyright notice, this list of conditions and the following
# disclaimer. Redistributions in binary form must reproduce the
# above copyright notice, this list of conditions and the following
# disclaimer in the documentation and/or other materials provided
# with the distribution.
#
# Neither the name Myanmar Karen Word Lists, nor the names of its
# contributors may be used to endorse or promote products derived
# from this software without specific prior written permission.
#
# THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND
# CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES,
# INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
# MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
# DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS
# BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
# EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED
# TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
# DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON
# ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR
# TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF
# THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
# SUCH DAMAGE.
# -----

```

#### Time Zone Database

ICU uses the public domain data and code derived from Time Zone Database for its time zone support. The ownership of the TZ database is explained in BCP 175: Procedure for Maintaining the Time Zone Database section 7.

```

# 7. Database Ownership
#
# The TZ database itself is not an IETF Contribution or an IETF
# document. Rather it is a pre-existing and regularly updated work
# that is in the public domain, and is intended to remain in the
# public domain. Therefore, BCPs 78 [RFC5378] and 79 [RFC3979] do
# not apply to the TZ Database or contributions that individuals make
# to it. Should any claims be made and substantiated against the TZ
# Database, the organization that is providing the IANA
# Considerations defined in this RFC, under the memorandum of
# understanding with the IETF, currently ICANN, may act in accordance
# with all competent court orders. No ownership claims will be made
# by ICANN or the IETF Trust on the database or the code. Any person
# making a contribution to the database or code waives all rights to
# future claims in that contribution or in the TZ Database.

```

#### Google double-conversion

Copyright 2006-2011, the V8 project authors. All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- \* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

\* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

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

---

File: aclocal.m4 (only for ICU4C)  
Section: pkg.m4 - Macros to locate and utilise pkg-config.

Copyright © 2004 Scott James Remnant <scott@netsplit.com>.  
Copyright © 2012-2015 Dan Nicholson <dbn.lists@gmail.com>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

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

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA.

As a special exception to the GNU General Public License, if you distribute this file as part of a program that contains a configuration script generated by Autoconf, you may include it under the same distribution terms that you use for the rest of that program.

(The condition for the exception is fulfilled because ICU4C includes a configuration script generated by Autoconf, namely the `configure` script.)

---

File: config.guess (only for ICU4C)

This file is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 3 of the License, or (at your option) any later version.

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

You should have received a copy of the GNU General Public License along with this program; if not, see <<https://www.gnu.org/licenses/>>.

As a special exception to the GNU General Public License, if you

distribute this file as part of a program that contains a configuration script generated by Autoconf, you may include it under the same distribution terms that you use for the rest of that program. This Exception is an additional permission under section 7 of the GNU General Public License, version 3 ("GPLv3").

(The condition for the exception is fulfilled because ICU4C includes a configuration script generated by Autoconf, namely the `configure` script.)

-----  
File: install-sh (only for ICU4C)

Copyright 1991 by the Massachusetts Institute of Technology

Permission to use, copy, modify, distribute, and sell this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of M.I.T. not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. M.I.T. makes no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

### Jsoncpp

The JsonCpp library's source code, including accompanying documentation, tests and demonstration applications, are licensed under the following conditions...

Baptiste Lepilleur and The JsonCpp Authors explicitly disclaim copyright in all jurisdictions which recognize such a disclaimer. In such jurisdictions, this software is released into the Public Domain.

In jurisdictions which do not recognize Public Domain property (e.g. Germany as of 2010), this software is Copyright (c) 2007-2010 by Baptiste Lepilleur and The JsonCpp Authors, and is released under the terms of the MIT License (see below).

In jurisdictions which recognize Public Domain property, the user of this software may choose to accept it either as 1) Public Domain, 2) under the conditions of the MIT License (see below), or 3) under the terms of dual Public Domain/MIT License conditions described here, as they choose.

The MIT License is about as close to Public Domain as a license can get, and is described in clear, concise terms at:

[http://en.wikipedia.org/wiki/MIT\\_License](http://en.wikipedia.org/wiki/MIT_License)

The full text of the MIT License follows:

=====  
Copyright (c) 2007-2010 Baptiste Lepilleur and The JsonCpp Authors

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

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

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,



EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

=====  
(END LICENSE TEXT)

The MIT license is compatible with both the GPL and commercial software, affording one all of the rights of Public Domain with the minor nuisance of being required to keep the above copyright notice and license text in the source code. Note also that by accepting the Public Domain "license" you can re-license your copy using whatever license you like.

### Libharu

Copyright (C) 1999-2006 Takeshi Kanno  
Copyright (C) 2007-2009 Antony Dovgal

This software is provided 'as-is', without any express or implied warranty.

In no event will the authors be held liable for any damages arising from the use of this software.

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

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

### Libiconv

GNU LIBRARY GENERAL PUBLIC LICENSE  
Version 2, June 1991

Copyright (C) 1991 Free Software Foundation, Inc.  
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA  
Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

[This is the first released version of the library GPL. It is numbered 2 because it goes with version 2 of the ordinary GPL.]

#### Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users.

This license, the Library General Public License, applies to some specially designated Free Software Foundation software, and to any other libraries whose authors decide to use it. You can use it for your libraries, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for

this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library, or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link a program with the library, you must provide complete object files to the recipients so that they can relink them with the library, after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

Our method of protecting your rights has two steps: (1) copyright the library, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the library.

Also, for each distributor's protection, we want to make certain that everyone understands that there is no warranty for this free library. If the library is modified by someone else and passed on, we want its recipients to know that what they have is not the original version, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that companies distributing free software will individually obtain patent licenses, thus in effect transforming the program into proprietary software. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License, which was designed for utility programs. This license, the GNU Library General Public License, applies to certain designated libraries. This license is quite different from the ordinary one; be sure to read it in full, and don't assume that anything in it is the same as in the ordinary license.

The reason we have a separate public license for some libraries is that they blur the distinction we usually make between modifying or adding to a program and simply using it. Linking a program with a library, without changing the library, is in some sense simply using the library, and is analogous to running a utility program or application program. However, in a textual and legal sense, the linked executable is a combined work, a derivative of the original library, and the ordinary General Public License treats it as such.

Because of this blurred distinction, using the ordinary General Public License for libraries did not effectively promote software sharing, because most developers did not use the libraries. We concluded that weaker conditions might promote sharing better.

However, unrestricted linking of non-free programs would deprive the users of those programs of all benefit from the free status of the libraries themselves. This Library General Public License is intended to permit developers of non-free programs to use free libraries, while preserving your freedom as a user of such programs to change the free libraries that are incorporated in them. (We have not seen how to achieve this as regards changes in header files, but we have achieved it as regards changes in the actual functions of the Library.) The hope is that this will lead to faster development of free libraries.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, while the latter only

works together with the library.

Note that it is possible for a library to be covered by the ordinary General Public License rather than by this special one.

GNU LIBRARY GENERAL PUBLIC LICENSE  
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Library General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a) The modified work must itself be a software library.
- b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.
- c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.
- d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or

table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License.

Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also compile or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

- a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)
- b) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.
- c) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.
- d) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license

restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

- a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.
- b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that

system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Library General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

#### NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

#### END OF TERMS AND CONDITIONS

#### Appendix: How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
<one line to give the library's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>
```

```
This library is free software; you can redistribute it and/or
modify it under the terms of the GNU Library General Public
License as published by the Free Software Foundation; either
version 2 of the License, or (at your option) any later version.
```

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

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

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the library, if necessary. Here is a sample; alter the names:

```
Yoyodyne, Inc., hereby disclaims all copyright interest in the
library `Frob' (a library for tweaking knobs) written by James Random Hacker.
```

```
<signature of Ty Coon>, 1 April 1990
Ty Coon, President of Vice
```

That's all there is to it!

## Libjpeg-turbo

### libjpeg-turbo Licenses =====

libjpeg-turbo is covered by three compatible BSD-style open source licenses:

- The IJG (Independent JPEG Group) License, which is listed in [README.ijg] (README.ijg)

```
This license applies to the libjpeg API library and associated programs
(any code inherited from libjpeg, and any modifications to that code.)
```

- The Modified (3-clause) BSD License, which is listed below

```
This license covers the TurboJPEG API library and associated programs, as
well as the build system.
```

- The [zlib License] (<https://opensource.org/licenses/Zlib>)

```
This license is a subset of the other two, and it covers the libjpeg-turbo
SIMD extensions.
```

### Complying with the libjpeg-turbo Licenses =====

This section provides a roll-up of the libjpeg-turbo licensing terms, to the best of our understanding.

1. If you are distributing a modified version of the libjpeg-turbo source, then:



1. You cannot alter or remove any existing copyright or license notices from the source.

```

**Origin**
- Clause 1 of the IJG License
- Clause 1 of the Modified BSD License
- Clauses 1 and 3 of the zlib License

```

2. You must add your own copyright notice to the header of each source file you modified, so others can tell that you modified that file (if there is not an existing copyright header in that file, then you can simply add a notice stating that you modified the file.)

```

**Origin**
- Clause 1 of the IJG License
- Clause 2 of the zlib License

```

3. You must include the IJG README file, and you must not alter any of the copyright or license text in that file.

```

**Origin**
- Clause 1 of the IJG License

```

2. If you are distributing only libjpeg-turbo binaries without the source, or if you are distributing an application that statically links with libjpeg-turbo, then:

1. Your product documentation must include a message stating:

```

This software is based in part on the work of the Independent JPEG
Group.

```

```

**Origin**
- Clause 2 of the IJG license

```

2. If your binary distribution includes or uses the TurboJPEG API, then your product documentation must include the text of the Modified BSD License (see below.)

```

**Origin**
- Clause 2 of the Modified BSD License

```

3. You cannot use the name of the IJG or The libjpeg-turbo Project or the contributors thereof in advertising, publicity, etc.

```

**Origin**
- IJG License
- Clause 3 of the Modified BSD License

```

4. The IJG and The libjpeg-turbo Project do not warrant libjpeg-turbo to be free of defects, nor do we accept any liability for undesirable consequences resulting from your use of the software.

```

**Origin**
- IJG License
- Modified BSD License
- zlib License

```

The Modified (3-clause) BSD License  
 =====

Copyright (C)2009-2022 D. R. Commander. All Rights Reserved.  
 Copyright (C)2015 Viktor Szathmáry. All Rights Reserved.

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

- Redistributions of source code must retain the above copyright notice,

- this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
  - Neither the name of the libjpeg-turbo Project nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

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

#### Why Three Licenses?

=====

The zlib License could have been used instead of the Modified (3-clause) BSD License, and since the IJG License effectively subsumes the distribution conditions of the zlib License, this would have effectively placed libjpeg-turbo binary distributions under the IJG License. However, the IJG License specifically refers to the Independent JPEG Group and does not extend attribution and endorsement protections to other entities. Thus, it was desirable to choose a license that granted us the same protections for new code that were granted to the IJG for code derived from their software.

#### Liblzma

#### XZ Utils Licensing

=====

Different licenses apply to different files in this package. Here is a rough summary of which licenses apply to which parts of this package (but check the individual files to be sure!):

- liblzma is in the public domain.
- xz, xzdec, and lzmadec command line tools are in the public domain unless GNU getopt\_long had to be compiled and linked in from the lib directory. The getopt\_long code is under GNU LGPLv2.1+.
- The scripts to grep, diff, and view compressed files have been adapted from gzip. These scripts and their documentation are under GNU GPLv2+.
- All the documentation in the doc directory and most of the XZ Utils specific documentation files in other directories are in the public domain.
- Translated messages are in the public domain.
- The build system contains public domain files, and files that are under GNU GPLv2+ or GNU GPLv3+. None of these files end up in the binaries being built.
- Test files and test code in the tests directory, and debugging utilities in the debug directory are in the public domain.
- The extra directory may contain public domain files, and files that are under various free software licenses.

You can do whatever you want with the files that have been put into the public domain. If you find public domain legally problematic, take the previous sentence as a license grant. If you still find the lack of copyright legally problematic, you have too many lawyers.

As usual, this software is provided "as is", without any warranty.

If you copy significant amounts of public domain code from XZ Utils into your project, acknowledging this somewhere in your software is polite (especially if it is proprietary, non-free software), but naturally it is not legally required. Here is an example of a good notice to put into "about box" or into documentation:

This software includes code from XZ Utils <<https://tukaani.org/xz/>>.

The following license texts are included in the following files:

- COPYING.LGPLv2.1: GNU Lesser General Public License version 2.1
- COPYING.GPLv2: GNU General Public License version 2
- COPYING.GPLv3: GNU General Public License version 3

Note that the toolchain (compiler, linker etc.) may add some code pieces that are copyrighted. Thus, it is possible that e.g. liblzma binary wouldn't actually be in the public domain in its entirety even though it contains no copyrighted code from the XZ Utils source package.

If you have questions, don't hesitate to ask the author(s) for more information.

## Libogg

Copyright (c) 2002, Xiph.org Foundation

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

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

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

## Libpng

COPYRIGHT NOTICE, DISCLAIMER, and LICENSE  
=====

PNG Reference Library License version 2

- 
- 
- \* Copyright (c) 1995-2019 The PNG Reference Library Authors.
  - \* Copyright (c) 2018-2019 Cosmin Truta.
  - \* Copyright (c) 2000-2002, 2004, 2006-2018 Glenn Randers-Pehrson.
  - \* Copyright (c) 1996-1997 Andreas Dilger.
  - \* Copyright (c) 1995-1996 Guy Eric Schalnat, Group 42, Inc.

The software is supplied "as is", without warranty of any kind, express or implied, including, without limitation, the warranties of merchantability, fitness for a particular purpose, title, and non-infringement. In no event shall the Copyright owners, or anyone distributing the software, be liable for any damages or other liability, whether in contract, tort or otherwise, arising from, out of, or in connection with the software, or the use or other dealings in the software, even if advised of the possibility of such damage.

Permission is hereby granted to use, copy, modify, and distribute this software, or portions hereof, for any purpose, without fee, subject to the following restrictions:

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

PNG Reference Library License version 1 (for libpng 0.5 through 1.6.35)

---

libpng versions 1.0.7, July 1, 2000, through 1.6.35, July 15, 2018 are Copyright (c) 2000-2002, 2004, 2006-2018 Glenn Randers-Pehrson, are derived from libpng-1.0.6, and are distributed according to the same disclaimer and license as libpng-1.0.6 with the following individuals added to the list of Contributing Authors:

Simon-Pierre Cadieux  
Eric S. Raymond  
Mans Rullgard  
Cosmin Truta  
Gilles Vollant  
James Yu  
Mandar Sahastrabudhe  
Google Inc.  
Vadim Barkov

and with the following additions to the disclaimer:

There is no warranty against interference with your enjoyment of the library or against infringement. There is no warranty that our efforts or the library will fulfill any of your particular purposes or needs. This library is provided with all faults, and the entire risk of satisfactory quality, performance, accuracy, and effort is with the user.

Some files in the "contrib" directory and some configure-generated files that are distributed with libpng have other copyright owners, and are released under other open source licenses.

libpng versions 0.97, January 1998, through 1.0.6, March 20, 2000, are Copyright (c) 1998-2000 Glenn Randers-Pehrson, are derived from libpng-0.96, and are distributed according to the same disclaimer and license as libpng-0.96, with the following individuals added to the

list of Contributing Authors:

Tom Lane  
Glenn Randers-Pehrson  
Willem van Schaik

libpng versions 0.89, June 1996, through 0.96, May 1997, are Copyright (c) 1996-1997 Andreas Dilger, are derived from libpng-0.88, and are distributed according to the same disclaimer and license as libpng-0.88, with the following individuals added to the list of Contributing Authors:

John Bowler  
Kevin Bracey  
Sam Bushell  
Magnus Holmgren  
Greg Roelofs  
Tom Tanner

Some files in the "scripts" directory have other copyright owners, but are released under this license.

libpng versions 0.5, May 1995, through 0.88, January 1996, are Copyright (c) 1995-1996 Guy Eric Schalnat, Group 42, Inc.

For the purposes of this copyright and license, "Contributing Authors" is defined as the following set of individuals:

Andreas Dilger  
Dave Martindale  
Guy Eric Schalnat  
Paul Schmidt  
Tim Wegner

The PNG Reference Library is supplied "AS IS". The Contributing Authors and Group 42, Inc. disclaim all warranties, expressed or implied, including, without limitation, the warranties of merchantability and of fitness for any purpose. The Contributing Authors and Group 42, Inc. assume no liability for direct, indirect, incidental, special, exemplary, or consequential damages, which may result from the use of the PNG Reference Library, even if advised of the possibility of such damage.

Permission is hereby granted to use, copy, modify, and distribute this source code, or portions hereof, for any purpose, without fee, subject to the following restrictions:

1. The origin of this source code must not be misrepresented.
2. Altered versions must be plainly marked as such and must not be misrepresented as being the original source.
3. This Copyright notice may not be removed or altered from any source or altered source distribution.

The Contributing Authors and Group 42, Inc. specifically permit, without fee, and encourage the use of this source code as a component to supporting the PNG file format in commercial products. If you use this source code in a product, acknowledgment is not required but would be appreciated.

## Libtheora

Copyright (C) 2002-2009 Xiph.org Foundation

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

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

- Neither the name of the Xiph.org Foundation nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

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

#### Libxml2

Except where otherwise noted in the source code (e.g. the files hash.c, list.c and the trio files, which are covered by a similar licence but with different Copyright notices) all the files are:

Copyright (C) 1998-2012 Daniel Veillard. All Rights Reserved.

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

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

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

#### Lz4

LZ4 Library  
Copyright (c) 2011-2016, Yann Collet  
All rights reserved.

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

\* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

\* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED

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

#### Netcdf-c

Copyright 2018 Unidata

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

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

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

#### Nlohmann-json

MIT License

Copyright (c) 2013-2022 Niels Lohmann

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

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

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

#### Op2

BSD 2-Clause License - <http://www.opensource.org/licenses/bsd-license.php>

This file is part of the OP2 distribution.

Copyright (c) 2011, Gihan Mudalige, Istvan Reguly, Mike Giles, and others.  
Please see the AUTHORS file in the main source directory for details.  
All rights reserved.

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

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- \* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- \* The name of Mike Giles may not be used to endorse or promote products derived from this software without specific prior written permission.

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

## OpenMesh

OpenMesh is licensed under the 3-clause BSD License. See the LICENSE file for the complete license.

```

/* ===== *
 * * *
 * OpenMesh *
 * Copyright (c) 2001-2015, RWTH-Aachen University *
 * Department of Computer Graphics and Multimedia *
 * All rights reserved. *
 * www.openmesh.org *
 * * *
 *-----*
 * This file is part of OpenMesh. *
 *-----*
 * * *
 * Redistribution and use in source and binary forms, with or without *
 * modification, are permitted provided that the following conditions *
 * are met: *
 * * *
 * 1. Redistributions of source code must retain the above copyright notice, *
 * this list of conditions and the following disclaimer. *
 * * *
 * 2. Redistributions in binary form must reproduce the above copyright *
 * notice, this list of conditions and the following disclaimer in the *
 * documentation and/or other materials provided with the distribution. *
 * * *
 * 3. Neither the name of the copyright holder nor the names of its *
 * contributors may be used to endorse or promote products derived from *
 * this software without specific prior written permission. *
 * * *
 * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS *
 * "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED *
 * TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A *
 * PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER *
 * OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, *
 * EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, *
 * PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR *
 * PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF *
 * LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING *
 * NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS *
 *

```



```
* SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.      *
*                                                                      *
* =====*\
```

## Openssl

### Apache License

Version 2.0, January 2004  
<https://www.apache.org/licenses/>

#### TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

##### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and

subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
  - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
  - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
  - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
  - (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.
5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

## Perl

### PCRE2 LICENCE

-----

PCRE2 is a library of functions to support regular expressions whose syntax and semantics are as close as possible to those of the Perl 5 language.

Releases 10.00 and above of PCRE2 are distributed under the terms of the "BSD" licence, as specified below, with one exemption for certain binary redistributions. The documentation for PCRE2, supplied in the "doc" directory, is distributed under the same terms as the software itself. The data in the testdata directory is not copyrighted and is in the public domain.

The basic library functions are written in C and are freestanding. Also included in the distribution is a just-in-time compiler that can be used to optimize pattern matching. This is an optional feature that can be omitted when the library is built.

### THE BASIC LIBRARY FUNCTIONS

-----

Written by: Philip Hazel  
Email local part: Philip.Hazel

Email domain: gmail.com

Retired from University of Cambridge Computing Service,  
Cambridge, England.

Copyright (c) 1997-2021 University of Cambridge  
All rights reserved.

#### PCRE2 JUST-IN-TIME COMPILATION SUPPORT

-----

Written by: Zoltan Herczeg  
Email local part: hzmester  
Email domain: freemail.hu

Copyright(c) 2010-2021 Zoltan Herczeg  
All rights reserved.

#### STACK-LESS JUST-IN-TIME COMPILER

-----

Written by: Zoltan Herczeg  
Email local part: hzmester  
Email domain: freemail.hu

Copyright(c) 2009-2021 Zoltan Herczeg  
All rights reserved.

#### THE "BSD" LICENCE

-----

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

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

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

#### EXEMPTION FOR BINARY LIBRARY-LIKE PACKAGES

-----

The second condition in the BSD licence (covering binary redistributions) does  
not apply all the way down a chain of software. If binary package A includes  
PCRE2, it must respect the condition, but if package B is software that  
includes package A, the condition is not imposed on package B unless it uses  
PCRE2 independently.

End

## **Pegt1-2**

The MIT License (MIT)

Copyright (c) 2007-2020 Dr. Colin Hirsch and Daniel Frey

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

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

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

## **Pkgconf**

Copyright (c) 2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018  
pkgconf authors (see AUTHORS file in source directory).

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

This software is provided 'as is' and without any warranty, express or implied. In no event shall the authors be liable for any damages arising from the use of this software.

## **Proj**

All source, data files and other contents of the PROJ package are available under the following terms. Note that the PROJ 4.3 and earlier was "public domain" as is common with US government work, but apparently this is not a well defined legal term in many countries. Frank Warmerdam placed everything under the following MIT style license because he believed it is effectively the same as public domain, allowing anyone to use the code as they wish, including making proprietary derivatives.

Initial PROJ 4.3 public domain code was put as Frank Warmerdam as copyright holder, but he didn't mean to imply he did the work. Essentially all work was done by Gerald Evenden.

Copyright information can be found in source files.

-----

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

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

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

## Pugixml

### MIT License

Copyright (c) 2006-2022 Arseny Kapoulkine

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

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

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

## Qt5-base

### GNU LESSER GENERAL PUBLIC LICENSE

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

You may use, distribute and copy the Qt Toolkit under the terms of GNU Lesser General Public License version 3, which is displayed below. This license makes reference to the version 3 of the GNU General Public License, which you can find in the LICENSE.GPL3 file.

-----

### GNU LESSER GENERAL PUBLIC LICENSE Version 3, 29 June 2007

Copyright © 2007 Free Software Foundation, Inc. <<http://fsf.org/>>  
Everyone is permitted to copy and distribute verbatim copies of this licensedocument, but changing it is not allowed.

This version of the GNU Lesser General Public License incorporates the terms and conditions of version 3 of the GNU General Public License, supplemented by the additional permissions listed below.

#### 0. Additional Definitions.

As used herein, "this License" refers to version 3 of the GNU Lesser General Public License, and the "GNU GPL" refers to version 3 of the GNU General Public License.

"The Library" refers to a covered work governed by this License, other than an Application or a Combined Work as defined below.

An "Application" is any work that makes use of an interface provided by the Library, but which is not otherwise based on the Library. Defining a subclass of a class defined by the Library is deemed a mode of using an interface provided by the Library.

A "Combined Work" is a work produced by combining or linking an Application with the Library. The particular version of the Library with which the Combined Work was made is also called the "Linked Version".

The "Minimal Corresponding Source" for a Combined Work means the Corresponding Source for the Combined Work, excluding any source code for portions of the Combined Work that, considered in isolation, are based on the Application, and not on the Linked Version.

The "Corresponding Application Code" for a Combined Work means the object code and/or source code for the Application, including any data and utility programs needed for reproducing the Combined Work from the Application, but excluding the System Libraries of the Combined Work.

#### 1. Exception to Section 3 of the GNU GPL.

You may convey a covered work under sections 3 and 4 of this License without being bound by section 3 of the GNU GPL.

#### 2. Conveying Modified Versions.

If you modify a copy of the Library, and, in your modifications, a facility refers to a function or data to be supplied by an Application that uses the facility (other than as an argument passed when the facility is invoked), then you may convey a copy of the modified version:

- a) under this License, provided that you make a good faith effort to ensure that, in the event an Application does not supply the function or data, the facility still operates, and performs whatever part of its purpose remains meaningful, or
- b) under the GNU GPL, with none of the additional permissions of this License applicable to that copy.

#### 3. Object Code Incorporating Material from Library Header Files.

The object code form of an Application may incorporate material from a header file that is part of the Library. You may convey such object code under terms of your choice, provided that, if the incorporated material is not limited to numerical parameters, data structure layouts and accessors, or small macros, inline functions and templates (ten or fewer lines in length), you do both of the following:

- a) Give prominent notice with each copy of the object code that the Library is used in it and that the Library and its use are covered by this License.
- b) Accompany the object code with a copy of the GNU GPL and this license document.

#### 4. Combined Works.

You may convey a Combined Work under terms of your choice that, taken together, effectively do not restrict modification of the portions of the Library contained in the Combined Work and reverse engineering for debugging such modifications, if you also do each of the following:

- a) Give prominent notice with each copy of the Combined Work that the Library is used in it and that the Library and its use are covered by this License.

- b) Accompany the Combined Work with a copy of the GNU GPL and this license document.
- c) For a Combined Work that displays copyright notices during execution, include the copyright notice for the Library among these notices, as well as a reference directing the user to the copies of the GNU GPL and this license document.
- d) Do one of the following:
  - 0) Convey the Minimal Corresponding Source under the terms of this License, and the Corresponding Application Code in a form suitable for, and under terms that permit, the user to recombine or relink the Application with a modified version of the Linked Version to produce a modified Combined Work, in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.
  - 1) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (a) uses at run time a copy of the Library already present on the user's computer system, and (b) will operate properly with a modified version of the Library that is interface-compatible with the Linked Version.
- e) Provide Installation Information, but only if you would otherwise be required to provide such information under section 6 of the GNU GPL, and only to the extent that such information is necessary to install and execute a modified version of the Combined Work produced by recombining or relinking the Application with a modified version of the Linked Version. (If you use option 4d0, the Installation Information must accompany the Minimal Corresponding Source and Corresponding Application Code. If you use option 4d1, you must provide the Installation Information in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.)

#### 5. Combined Libraries.

You may place library facilities that are a work based on the Library side by side in a single library together with other library facilities that are not Applications and are not covered by this License, and convey such a combined library under terms of your choice, if you do both of the following:

- a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities, conveyed under the terms of this License.
- b) Give prominent notice with the combined library that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

#### 6. Revised Versions of the GNU Lesser General Public License.

The Free Software Foundation may publish revised and/or new versions of the GNU Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library as you received it specifies that a certain numbered version of the GNU Lesser General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that published version or of any later version published by the Free Software Foundation. If the Library as you received it does not specify a version number of the GNU Lesser General Public License, you may choose any version of the GNU Lesser General Public License ever published by the Free Software Foundation.



If the Library as you received it specifies that a proxy can decide whether future versions of the GNU Lesser General Public License shall apply, that proxy's public statement of acceptance of any version is permanent authorization for you to choose that version for the Library.

## Qwt

### Qwt License

Version 1.0, January 1, 2003

The Qwt library and included programs are provided under the terms of the GNU LESSER GENERAL PUBLIC LICENSE (LGPL) with the following exceptions:

1. Widgets that are subclassed from Qwt widgets do not constitute a derivative work.
2. Static linking of applications and widgets to the Qwt library does not constitute a derivative work and does not require the author to provide source code for the application or widget, use the shared Qwt libraries, or link their applications or widgets against a user-supplied version of Qwt.

If you link the application or widget to a modified version of Qwt, then the changes to Qwt must be provided under the terms of the LGPL in sections 1, 2, and 4.

3. You do not have to provide a copy of the Qwt license with programs that are linked to the Qwt library, nor do you have to identify the Qwt license in your program or documentation as required by section 6 of the LGPL.

However, programs must still identify their use of Qwt. The following example statement can be included in user documentation to satisfy this requirement:

```
[program/widget] is based in part on the work of
the Qwt project (http://qwt.sf.net).
```

---

### GNU LESSER GENERAL PUBLIC LICENSE Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc.  
59 Temple Place, Suite 330, Boston, MA 02111-1307 USA  
Everyone is permitted to copy and distribute verbatim copies  
of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts  
as the successor of the GNU Library Public License, version 2, hence  
the version number 2.1.]

#### Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some specially designated software packages--typically libraries--of the Free Software Foundation and other authors who decide to use it. You

can use it too, but we suggest you first think carefully about whether this license or the ordinary General Public License is the better strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish); that you receive source code or can get it if you want it; that you can change the software and use pieces of it in new free programs; and that you are informed that you can do these things.

To protect your rights, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surrender these rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free

library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

GNU LESSER GENERAL PUBLIC LICENSE  
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion

of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a) The modified work must itself be a software library.
- b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.
- c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.
- d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if

the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.

- c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.
- d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.
- e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

- a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.
- b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

#### NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR

PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
<one line to give the library's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>
```

```
This library is free software; you can redistribute it and/or
modify it under the terms of the GNU Lesser General Public
License as published by the Free Software Foundation; either
version 2.1 of the License, or (at your option) any later version.
```

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

```
You should have received a copy of the GNU Lesser General Public
License along with this library; if not, write to the Free Software
Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
```

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the library, if necessary. Here is a sample; alter the names:

```
Yoyodyne, Inc., hereby disclaims all copyright interest in the
library `Frob' (a library for tweaking knobs) written by James Random Hacker.
```

```
<signature of Ty Coon>, 1 April 1990
Ty Coon, President of Vice
```

That's all there is to it!

**Shapelib**

GNU LIBRARY GENERAL PUBLIC LICENSE  
Version 2, June 1991

```
Copyright (C) 1991 Free Software Foundation, Inc.
675 Mass Ave, Cambridge, MA 02139, USA
Everyone is permitted to copy and distribute verbatim copies
```



of this license document, but changing it is not allowed.

[This is the first released version of the library GPL. It is numbered 2 because it goes with version 2 of the ordinary GPL.]

#### Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users.

This license, the Library General Public License, applies to some specially designated Free Software Foundation software, and to any other libraries whose authors decide to use it. You can use it for your libraries, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library, or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link a program with the library, you must provide complete object files to the recipients so that they can relink them with the library, after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

Our method of protecting your rights has two steps: (1) copyright the library, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the library.

Also, for each distributor's protection, we want to make certain that everyone understands that there is no warranty for this free library. If the library is modified by someone else and passed on, we want its recipients to know that what they have is not the original version, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that companies distributing free software will individually obtain patent licenses, thus in effect transforming the program into proprietary software. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License, which was designed for utility programs. This license, the GNU Library General Public License, applies to certain designated libraries. This license is quite different from the ordinary one; be sure to read it in full, and don't assume that anything in it is the same as in the ordinary license.

The reason we have a separate public license for some libraries is that they blur the distinction we usually make between modifying or adding to a program and simply using it. Linking a program with a library, without changing the library, is in some sense simply using the library, and is analogous to running a utility program or application program. However, in a textual and legal sense, the linked executable is a combined work, a derivative of the original library, and the ordinary General Public License treats it as such.

Because of this blurred distinction, using the ordinary General Public License for libraries did not effectively promote software sharing, because most developers did not use the libraries. We concluded that weaker conditions might promote sharing better.

However, unrestricted linking of non-free programs would deprive the users of those programs of all benefit from the free status of the libraries themselves. This Library General Public License is intended to permit developers of non-free programs to use free libraries, while preserving your freedom as a user of such programs to change the free libraries that are incorporated in them. (We have not seen how to achieve this as regards changes in header files, but we have achieved it as regards changes in the actual functions of the Library.) The hope is that this will lead to faster development of free libraries.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, while the latter only works together with the library.

Note that it is possible for a library to be covered by the ordinary General Public License rather than by this special one.

GNU LIBRARY GENERAL PUBLIC LICENSE  
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Library General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a

fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a) The modified work must itself be a software library.
- b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.
- c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.
- d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany

it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also compile or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

- a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)
- b) Accompany the work with a written offer, valid for at least three years, to give the same user the materials

specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

c) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

d) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not

excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Library General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

#### NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN

WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

#### Appendix: How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
<one line to give the library's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>
```

```
This library is free software; you can redistribute it and/or
modify it under the terms of the GNU Library General Public
License as published by the Free Software Foundation; either
version 2 of the License, or (at your option) any later version.
```

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

```
You should have received a copy of the GNU Library General Public
License along with this library; if not, write to the Free
Software Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.
```

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the library, if necessary. Here is a sample; alter the names:

```
Yoyodyne, Inc., hereby disclaims all copyright interest in the
library `Frob' (a library for tweaking knobs) written by James Random Hacker.
```

```
<signature of Ty Coon>, 1 April 1990
Ty Coon, President of Vice
```

That's all there is to it!

#### SQLite3

SQLite is in the Public Domain.  
<http://www.sqlite.org/copyright.html>

#### Tecio

Tecplot, Inc. LICENSE AGREEMENT FOR Tecplot's TecIO ("TecIO")

TecIO is a software library provided by Tecplot, Inc. to enable software developed by others to write data in Tecplot's proprietary binary file

formats, .plt and .szplt, and to read Tecplot binary data in .plt and .szplt format. TecIO is included with Tecplot 360 EX and may also be downloaded from <http://www.tecplot.com/downloads/tecio-library/>.

This license applies to versions of the TecIO library distributed with Tecplot 360 EX 2016 R2 and later and covers both the serial and parallel (MPI) versions of the library.

1. This LICENSE AGREEMENT is between Tecplot, Inc. ("Tecplot"), and the Individual or Organization ("Licensee") accessing and otherwise using TecIO software in source or binary form and its associated documentation.
2. Licensee acknowledges that this is only a limited nonexclusive license. Tecplot is and remains the owner of all titles, rights, and interests in TecIO Software. Title to TecIO and all copies thereof remain with Tecplot. The Materials are copyrighted and are protected by United States copyright laws and international treaty provisions. Licensee will not remove any copyright notice from the Materials. Tecplot does not grant any express or implied right to you under Tecplot patents, copyrights, trademarks, or trade secret information.
3. Subject to the terms and conditions of this License Agreement, Tecplot hereby grants Licensee a nonexclusive, royalty-free, world-wide license to reproduce, analyze, test, perform and/or display publicly, prepare derivative works, distribute, and otherwise use TecIO alone or in any derivative version, provided, however, that Tecplot's License Agreement and Tecplot's notice of copyright, i.e., "Copyright © 1988-2016 Tecplot, Inc. All rights reserved worldwide." are retained in TecIO alone or in any derivative version prepared by Licensee.
4. In the event Licensee prepares a derivative work that is based on or incorporates TecIO or any part thereof, and wants to publish the derivative work as provided herein, Licensee hereby agrees to provide to all end users of any such work a brief summary of all changes made to TecIO, and to convey to Tecplot a copy of the modified TecIO source code within 30 days after publication of any work containing any such changes.
5. Tecplot is making TecIO available to Licensee on an "AS IS" basis. NO OTHER WARRANTIES. TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, TECPLOT, INC. AND ITS SUPPLIERS DISCLAIM ALL OTHER WARRANTIES AND CONDITIONS, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE, AND NONINFRINGEMENT, WITH REGARD TO THE SOFTWARE, AND THE PROVISION OF OR FAILURE TO PROVIDE SUPPORT SERVICES. APART FROM THE WARRANTIES STATED ABOVE, TECPLOT, INC. MAKES NO WARRANTY THAT THE SOFTWARE OR SERVICES WILL: MEET REQUIREMENTS; PROVIDE UNINTERRUPTED, TIMELY, SECURE, OR ERROR-FREE, USE OF COMPUTERS OR NETWORKS; PROVIDE RESULTS WHICH ARE ACCURATE OR RELIABLE; MEET EXPECTATIONS, OR; CORRECT ANY ERRORS IN THE SOFTWARE. TECPLOT, INC. SHALL NOT BE RESPONSIBLE FOR MISUSE OF THE SOFTWARE OR ANY LOSS OF DATA. THIS LIMITED WARRANTY GIVES YOU SPECIFIC LEGAL RIGHTS. YOU MAY HAVE OTHERS, WHICH VARY FROM STATE/JURISDICTION TO STATE/JURISDICTION.
6. TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, IN NO EVENT SHALL TECPLOT, INC. OR ITS SUPPLIERS BE LIABLE FOR ANY SPECIAL, INCIDENTAL, INDIRECT, OR CONSEQUENTIAL DAMAGES WHATSOEVER (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF BUSINESS PROFITS, BUSINESS INTERRUPTION, LOSS OF BUSINESS INFORMATION, OR ANY OTHER PECUNIARY LOSS) ARISING OUT OF THE USE OF OR INABILITY TO USE THE SOFTWARE OR THE PROVISION OF OR FAILURE TO PROVIDE SUPPORT SERVICES, EVEN IF TECPLOT, INC. HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. THIS SHALL BE TRUE EVEN IN THE EVENT OF THE FAILURE OF AN AGREED REMEDY. IN ANY CASE, TECPLOT, INC.'S ENTIRE LIABILITY FOR CLAIMS ARISING OUT OF USE OF THE SOFTWARE, SERVICE OR ARISING FROM ANY PROVISION OF THIS AGREEMENT SHALL BE LIMITED TO THE AMOUNT ACTUALLY PAID BY LICENSEE FOR THE SOFTWARE OR SERVICE COMPLAINED OF. BECAUSE SOME STATES AND JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF LIABILITY, THE ABOVE LIMITATION MAY NOT APPLY TO YOU.
7. This License Agreement will automatically terminate upon a material breach of its terms and conditions.
8. Nothing in this License Agreement shall be deemed to create any relationship of agency, partnership, or joint venture between Tecplot and Licensee. This



License Agreement does not grant permission to use Tecplot trademarks or trade name in a trademark sense to endorse or promote products or services of Licensee, or any third party.

9. By copying, installing or otherwise using TecIO, Licensee agrees to be bound by the terms and conditions of this License Agreement.

### **Tclap**

Copyright (c) 2003 Michael E. Smoot  
Copyright (c) 2004 Daniel Aarno  
Copyright (c) 2017 Google Inc.

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

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

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

### **Tiff**

Copyright (c) 1988-1997 Sam Leffler  
Copyright (c) 1991-1997 Silicon Graphics, Inc.

Permission to use, copy, modify, distribute, and sell this software and its documentation for any purpose is hereby granted without fee, provided that (i) the above copyright notices and this permission notice appear in all copies of the software and related documentation, and (ii) the names of Sam Leffler and Silicon Graphics may not be used in any advertising or publicity relating to the software without the specific, prior written permission of Sam Leffler and Silicon Graphics.

THE SOFTWARE IS PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EXPRESS, IMPLIED OR OTHERWISE, INCLUDING WITHOUT LIMITATION, ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

IN NO EVENT SHALL SAM LEFFLER OR SILICON GRAPHICS BE LIABLE FOR ANY SPECIAL, INCIDENTAL, INDIRECT OR CONSEQUENTIAL DAMAGES OF ANY KIND, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER OR NOT ADVISED OF THE POSSIBILITY OF DAMAGE, AND ON ANY THEORY OF LIABILITY, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

### **Utfcpp**

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

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

do so, all subject to the following:

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

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

### Vcpkg

Copyright (c) Microsoft Corporation

All rights reserved.

#### MIT License

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

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

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

### Vtk

/\*\*\*\*\*

Program: Visualization Toolkit  
Module: Copyright.txt

Copyright (c) 1993-2015 Ken Martin, Will Schroeder, Bill Lorensen  
All rights reserved.

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

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

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

=====\*/

## VtkexodusII

Copyright (c) 2005-2017 National Technology & Engineering Solutions of Sandia, LLC (NTESS). Under the terms of Contract DE-NA0003525 with NTESS, the U.S. Government retains certain rights in this software.

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

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

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

## Zlib

### ZLIB DATA COMPRESSION LIBRARY

zlib 1.2.12 is a general purpose data compression library. All the code is thread safe. The data format used by the zlib library is described by RFCs (Request for Comments) 1950 to 1952 in the files <http://tools.ietf.org/html/rfc1950> (zlib format), [rfc1951](http://tools.ietf.org/html/rfc1951) (deflate format) and [rfc1952](http://tools.ietf.org/html/rfc1952) (gzip format).

All functions of the compression library are documented in the file `zlib.h` (volunteer to write man pages welcome, contact [zlib@gzip.org](mailto:zlib@gzip.org)). A usage example of the library is given in the file `test/example.c` which also tests that the library is working correctly. Another example is given in the file `test/minigzip.c`. The compression library itself is composed of all source files in the root directory.

To compile all files and run the test program, follow the instructions given at the top of `Makefile.in`. In short `./configure; make test`, and if that goes well, `make install` should work for most flavors of Unix. For Windows, use

one of the special makefiles in win32/ or contrib/vstudio/ . For VMS, use make\_vms.com.

Questions about zlib should be sent to <zlib@zip.org>, or to Gilles Vollant <info@winimage.com> for the Windows DLL version. The zlib home page is <http://zlib.net/> . Before reporting a problem, please check this site to verify that you have the latest version of zlib; otherwise get the latest version and check whether the problem still exists or not.

PLEASE read the zlib FAQ [http://zlib.net/zlib\\_faq.html](http://zlib.net/zlib_faq.html) before asking for help.

Mark Nelson <markn@ieee.org> wrote an article about zlib for the Jan. 1997 issue of Dr. Dobbs's Journal; a copy of the article is available at <http://marknelson.us/1997/01/01/zlib-engine/> .

The changes made in version 1.2.12 are documented in the file ChangeLog.

Unsupported third party contributions are provided in directory contrib/ .

zlib is available in Java using the java.util.zip package, documented at <http://java.sun.com/developer/technicalArticles/Programming/compression/> .

A Perl interface to zlib written by Paul Marquess <pmqs@cpan.org> is available at CPAN (Comprehensive Perl Archive Network) sites, including <http://search.cpan.org/~pmqs/IO-Compress-Zlib/> .

A Python interface to zlib written by A.M. Kuchling <amk@amk.ca> is available in Python 1.5 and later versions, see <http://docs.python.org/library/zlib.html> .

zlib is built into tcl: <http://wiki.tcl.tk/4610> .

An experimental package to read and write files in .zip format, written on top of zlib by Gilles Vollant <info@winimage.com>, is available in the contrib/minizip directory of zlib.

Notes for some targets:

- For Windows DLL versions, please see win32/DLL\_FAQ.txt
- For 64-bit Irix, deflate.c must be compiled without any optimization. With -O, one libpng test fails. The test works in 32 bit mode (with the -n32 compiler flag). The compiler bug has been reported to SGI.
- zlib doesn't work with gcc 2.6.3 on a DEC 3000/300LX under OSF/1 2.1 it works when compiled with cc.
- On Digital Unix 4.0D (formerly OSF/1) on AlphaServer, the cc option -std1 is necessary to get gzprintf working correctly. This is done by configure.
- zlib doesn't work on HP-UX 9.05 with some versions of /bin/cc. It works with other compilers. Use "make test" to check your compiler.
- gzdopen is not supported on RISCOS or BEOS.
- For PalmOs, see <http://palmzlib.sourceforge.net/>

Acknowledgments:

The deflate format used by zlib was defined by Phil Katz. The deflate and zlib specifications were written by L. Peter Deutsch. Thanks to all the people who reported problems and suggested various improvements in zlib; they are too numerous to cite here.

Copyright notice:

(C) 1995-2022 Jean-loup Gailly and Mark Adler

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

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

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

Jean-loup Gailly            Mark Adler  
jloup@gzip.org            madler@alumni.caltech.edu

If you use the zlib library in a product, we would appreciate *not* receiving lengthy legal documents to sign. The sources are provided for free but without warranty of any kind. The library has been entirely written by Jean-loup Gailly and Mark Adler; it does not include third-party code. We make all contributions to and distributions of this project solely in our personal capacity, and are not conveying any rights to any intellectual property of any third parties.

If you redistribute modified sources, we would appreciate that you include in the file ChangeLog history information documenting your changes. Please read the FAQ for more information on the distribution of modified source versions.

## Zstd

ZSTD is dual licensed under BSD and GPLv2.

### BSD License

For Zstandard software

Copyright (c) 2016-present, Facebook, Inc. All rights reserved.

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

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

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

GNU GENERAL PUBLIC LICENSE  
Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.,  
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA  
Everyone is permitted to copy and distribute verbatim copies  
of this license document, but changing it is not allowed.

#### Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

#### GNU GENERAL PUBLIC LICENSE TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not

covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
- b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.
- c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

- a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- b) Accompany it with a written offer, valid for at least three

years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other



circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

#### NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

#### END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>
```

```
This program is free software; you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation; either version 2 of the License, or
(at your option) any later version.
```

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

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

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it
under certain conditions; type `show c' for details.
```

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

```
Yoyodyne, Inc., hereby disclaims all copyright interest in the program
`Gnomovision' (which makes passes at compilers) written by James Hacker.
```

```
<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice
```

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License.

## 1.2 Open source licenses

### Apache-2.0

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable

(except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
  - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
  - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
  - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
  - (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.
5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS,

WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

**FTL**

The FreeType Project LICENSE

-----  
2006-Jan-27

Copyright 1996-2002, 2006 by  
David Turner, Robert Wilhelm, and Werner Lemberg

## Introduction

=====

The FreeType Project is distributed in several archive packages; some of them may contain, in addition to the FreeType font engine, various tools and contributions which rely on, or relate to, the FreeType Project.

This license applies to all files found in such packages, and which do not fall under their own explicit license. The license affects thus the FreeType font engine, the test programs, documentation and makefiles, at the very least.

This license was inspired by the BSD, Artistic, and IJG (Independent JPEG Group) licenses, which all encourage inclusion and use of free software in commercial and freeware products alike. As a consequence, its main points are that:

- o We don't promise that this software works. However, we will be interested in any kind of bug reports. ('as is' distribution)
- o You can use this software for whatever you want, in parts or full form, without having to pay us. ('royalty-free' usage)
- o You may not pretend that you wrote this software. If you use it, or only parts of it, in a program, you must acknowledge somewhere in your documentation that you have used the FreeType code. ('credits')

We specifically permit and encourage the inclusion of this software, with or without modifications, in commercial products. We disclaim all warranties covering The FreeType Project and assume no liability related to The FreeType Project.

Finally, many people asked us for a preferred form for a credit/disclaimer to use in compliance with this license. We thus encourage you to use the following text:

```
""  
Portions of this software are copyright © <year> The FreeType  
Project (www.freetype.org). All rights reserved.  
""
```

Please replace <year> with the value from the FreeType version you actually use.

## Legal Terms

=====

## 0. Definitions

-----

Throughout this license, the terms 'package', 'FreeType Project', and 'FreeType archive' refer to the set of files originally distributed by the authors (David Turner, Robert Wilhelm, and Werner Lemberg) as the 'FreeType Project', be they named as alpha, beta or final release.

'You' refers to the licensee, or person using the project, where 'using' is a generic term including compiling the project's source code as well as linking it to form a 'program' or 'executable'. This program is referred to as 'a program using the FreeType engine'.

This license applies to all files distributed in the original FreeType Project, including all source code, binaries and documentation, unless otherwise stated in the file in its original, unmodified form as distributed in the original archive.

If you are unsure whether or not a particular file is covered by this license, you must contact us to verify this.

The FreeType Project is copyright (C) 1996-2000 by David Turner, Robert Wilhelm, and Werner Lemberg. All rights reserved except as specified below.

### 1. No Warranty

-----

THE FREETYPE PROJECT IS PROVIDED `AS IS' WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT WILL ANY OF THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY DAMAGES CAUSED BY THE USE OR THE INABILITY TO USE, OF THE FREETYPE PROJECT.

### 2. Redistribution

-----

This license grants a worldwide, royalty-free, perpetual and irrevocable right and license to use, execute, perform, compile, display, copy, create derivative works of, distribute and sublicense the FreeType Project (in both source and object code forms) and derivative works thereof for any purpose; and to authorize others to exercise some or all of the rights granted herein, subject to the following conditions:

- o Redistribution of source code must retain this license file ('FTL.TXT') unaltered; any additions, deletions or changes to the original files must be clearly indicated in accompanying documentation. The copyright notices of the unaltered, original files must be preserved in all copies of source files.
- o Redistribution in binary form must provide a disclaimer that states that the software is based in part of the work of the FreeType Team, in the distribution documentation. We also encourage you to put an URL to the FreeType web page in your documentation, though this isn't mandatory.

These conditions apply to any software derived from or based on the FreeType Project, not just the unmodified files. If you use our work, you must acknowledge us. However, no fee need be paid to us.

### 3. Advertising

-----

Neither the FreeType authors and contributors nor you shall use the name of the other for commercial, advertising, or promotional purposes without specific prior written permission.

We suggest, but do not require, that you use one or more of the following phrases to refer to this software in your documentation or advertising materials: 'FreeType Project', 'FreeType Engine', 'FreeType library', or 'FreeType Distribution'.

As you have not signed this license, you are not required to accept it. However, as the FreeType Project is copyrighted material, only this license, or another one contracted with the authors, grants you the right to use, distribute, and modify it. Therefore, by using, distributing, or modifying the FreeType Project, you indicate that you understand and accept all the terms of this license.

### 4. Contacts

-----

There are two mailing lists related to FreeType:

o [freetype@nongnu.org](mailto:freetype@nongnu.org)

Discusses general use and applications of FreeType, as well as future and wanted additions to the library and distribution. If you are looking for support, start in this list if you haven't found anything to help you in the documentation.

o [freetype-devel@nongnu.org](mailto:freetype-devel@nongnu.org)

Discusses bugs, as well as engine internals, design issues, specific licenses, porting, etc.

Our home page can be found at

<https://www.freetype.org>

## GPL-2.0

### GNU GENERAL PUBLIC LICENSE Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.,  
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA  
Everyone is permitted to copy and distribute verbatim copies  
of this license document, but changing it is not allowed.

#### Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software



patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE  
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
- b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.
- c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of

this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

- a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the

original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY

FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>
```

```
This program is free software; you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation; either version 2 of the License, or
(at your option) any later version.
```

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

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

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it
under certain conditions; type `show c' for details.
```

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

```
Yoyodyne, Inc., hereby disclaims all copyright interest in the program
`Gnomovision' (which makes passes at compilers) written by James Hacker.
```

<signature of Ty Coon>, 1 April 1989  
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License.

### GPL-3.0

GNU GENERAL PUBLIC LICENSE  
Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <<https://fsf.org/>>  
Everyone is permitted to copy and distribute verbatim copies  
of this license document, but changing it is not allowed.

#### Preamble

The GNU General Public License is a free, copyleft license for software and other kinds of works.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program--to make sure it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this way by its authors. You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps: (1) assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we

stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

#### TERMS AND CONDITIONS

##### 0. Definitions.

"This License" refers to version 3 of the GNU General Public License.

"Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this License. Each licensee is addressed as "you". "Licensees" and "recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based on the Program.

To "propagate" a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To "convey" a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays "Appropriate Legal Notices" to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

##### 1. Source Code.

The "source code" for a work means the preferred form of the work for making modifications to it. "Object code" means any non-source form of a work.

A "Standard Interface" means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The "System Libraries" of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an

implementation is available to the public in source code form. A "Major Component", in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The "Corresponding Source" for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

## 2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

## 3. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

## 4. Conveying Verbatim Copies.

You may convey verbatim copies of the Program's source code as you

receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

#### 5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

- a) The work must carry prominent notices stating that you modified it, and giving a relevant date.
- b) The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to "keep intact all notices".
- c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.
- d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

#### 6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

- a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.
- b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.



c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.

d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.

e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A "User Product" is either (1) a "consumer product", which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, "normally used" refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

"Installation Information" for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided,

in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

#### 7. Additional Terms.

"Additional permissions" are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

- a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or
- b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or
- c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or
- d) Limiting the use for publicity purposes of names of licensors or authors of the material; or
- e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or
- f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered "further restrictions" within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

#### 8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

#### 9. Acceptance Not Required for Having Copies.

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

#### 10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An "entity transaction" is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

#### 11. Patents.

A "contributor" is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's "contributor version".

A contributor's "essential patent claims" are all patent claims

owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. "Knowingly relying" means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is "discriminatory" if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

#### 12. No Surrender of Others' Freedom.

If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you

to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

17. Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the

Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

#### END OF TERMS AND CONDITIONS

##### How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>
```

```
This program is free software: you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation, either version 3 of the License, or
(at your option) any later version.
```

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

```
You should have received a copy of the GNU General Public License
along with this program. If not, see <https://www.gnu.org/licenses/>.
```

Also add information on how to contact you by electronic and paper mail.

If the program does terminal interaction, make it output a short notice like this when it starts in an interactive mode:

```
<program> Copyright (C) <year> <name of author>
This program comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it
under certain conditions; type `show c' for details.
```

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, your program's commands might be different; for a GUI interface, you would use an "about box".

You should also get your employer (if you work as a programmer) or school, if any, to sign a "copyright disclaimer" for the program, if necessary. For more information on this, and how to apply and follow the GNU GPL, see [<https://www.gnu.org/licenses/>](https://www.gnu.org/licenses/).

The GNU General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License. But first, please read [<https://www.gnu.org/licenses/why-not-lgpl.html>](https://www.gnu.org/licenses/why-not-lgpl.html).

#### LGPL-2.1

##### GNU LESSER GENERAL PUBLIC LICENSE Version 2.1, February 1999

```
Copyright (C) 1991, 1999 Free Software Foundation, Inc.
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.
```

[This is the first released version of the Lesser GPL. It also counts

as the successor of the GNU Library Public License, version 2, hence the version number 2.1.]

#### Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some specially designated software packages--typically libraries--of the Free Software Foundation and other authors who decide to use it. You can use it too, but we suggest you first think carefully about whether this license or the ordinary General Public License is the better strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish); that you receive source code or can get it if you want it; that you can change the software and use pieces of it in new free programs; and that you are informed that you can do these things.

To protect your rights, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surrender these rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

GNU LESSER GENERAL PUBLIC LICENSE  
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.



1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a) The modified work must itself be a software library.
- b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.
- c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.
- d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

- a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked

with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.

c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are

prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free

Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

#### NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

#### END OF TERMS AND CONDITIONS

##### How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
<one line to give the library's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>
```

```
This library is free software; you can redistribute it and/or
modify it under the terms of the GNU Lesser General Public
License as published by the Free Software Foundation; either
version 2.1 of the License, or (at your option) any later version.
```

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

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

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the library, if necessary. Here is a sample; alter the names:

```
Yoyodyne, Inc., hereby disclaims all copyright interest in the
library `Frob' (a library for tweaking knobs) written by James Random Hacker.
```

```
<signature of Ty Coon>, 1 April 1990
```

Ty Coon, President of Vice

That's all there is to it!

## MPL-2.0

Mozilla Public License Version 2.0  
=====

### 1. Definitions -----

#### 1.1. "Contributor"

means each individual or legal entity that creates, contributes to the creation of, or owns Covered Software.

#### 1.2. "Contributor Version"

means the combination of the Contributions of others (if any) used by a Contributor and that particular Contributor's Contribution.

#### 1.3. "Contribution"

means Covered Software of a particular Contributor.

#### 1.4. "Covered Software"

means Source Code Form to which the initial Contributor has attached the notice in Exhibit A, the Executable Form of such Source Code Form, and Modifications of such Source Code Form, in each case including portions thereof.

#### 1.5. "Incompatible With Secondary Licenses"

means

(a) that the initial Contributor has attached the notice described in Exhibit B to the Covered Software; or

(b) that the Covered Software was made available under the terms of version 1.1 or earlier of the License, but not also under the terms of a Secondary License.

#### 1.6. "Executable Form"

means any form of the work other than Source Code Form.

#### 1.7. "Larger Work"

means a work that combines Covered Software with other material, in a separate file or files, that is not Covered Software.

#### 1.8. "License"

means this document.

#### 1.9. "Licensable"

means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently, any and all of the rights conveyed by this License.

#### 1.10. "Modifications"

means any of the following:

(a) any file in Source Code Form that results from an addition to, deletion from, or modification of the contents of Covered Software; or

(b) any new file in Source Code Form that contains any Covered Software.

#### 1.11. "Patent Claims" of a Contributor

means any patent claim(s), including without limitation, method, process, and apparatus claims, in any patent Licensable by such Contributor that would be infringed, but for the grant of the License, by the making, using, selling, offering for sale, having

made, import, or transfer of either its Contributions or its Contributor Version.

1.12. "Secondary License"

means either the GNU General Public License, Version 2.0, the GNU Lesser General Public License, Version 2.1, the GNU Affero General Public License, Version 3.0, or any later versions of those licenses.

1.13. "Source Code Form"

means the form of the work preferred for making modifications.

1.14. "You" (or "Your")

means an individual or a legal entity exercising rights under this License. For legal entities, "You" includes any entity that controls, is controlled by, or is under common control with You. For purposes of this definition, "control" means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants and Conditions

-----

2.1. Grants

Each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

- (a) under intellectual property rights (other than patent or trademark) Licensable by such Contributor to use, reproduce, make available, modify, display, perform, distribute, and otherwise exploit its Contributions, either on an unmodified basis, with Modifications, or as part of a Larger Work; and
- (b) under Patent Claims of such Contributor to make, use, sell, offer for sale, have made, import, and otherwise transfer either its Contributions or its Contributor Version.

2.2. Effective Date

The licenses granted in Section 2.1 with respect to any Contribution become effective for each Contribution on the date the Contributor first distributes such Contribution.

2.3. Limitations on Grant Scope

The licenses granted in this Section 2 are the only rights granted under this License. No additional rights or licenses will be implied from the distribution or licensing of Covered Software under this License. Notwithstanding Section 2.1(b) above, no patent license is granted by a Contributor:

- (a) for any code that a Contributor has removed from Covered Software; or
- (b) for infringements caused by: (i) Your and any other third party's modifications of Covered Software, or (ii) the combination of its Contributions with other software (except as part of its Contributor Version); or
- (c) under Patent Claims infringed by Covered Software in the absence of its Contributions.

This License does not grant any rights in the trademarks, service marks, or logos of any Contributor (except as may be necessary to comply with the notice requirements in Section 3.4).

2.4. Subsequent Licenses

No Contributor makes additional grants as a result of Your choice to distribute the Covered Software under a subsequent version of this License (see Section 10.2) or under the terms of a Secondary License (if permitted under the terms of Section 3.3).

### 2.5. Representation

Each Contributor represents that the Contributor believes its Contributions are its original creation(s) or it has sufficient rights to grant the rights to its Contributions conveyed by this License.

### 2.6. Fair Use

This License is not intended to limit any rights You have under applicable copyright doctrines of fair use, fair dealing, or other equivalents.

### 2.7. Conditions

Sections 3.1, 3.2, 3.3, and 3.4 are conditions of the licenses granted in Section 2.1.

## 3. Responsibilities

-----

### 3.1. Distribution of Source Form

All distribution of Covered Software in Source Code Form, including any Modifications that You create or to which You contribute, must be under the terms of this License. You must inform recipients that the Source Code Form of the Covered Software is governed by the terms of this License, and how they can obtain a copy of this License. You may not attempt to alter or restrict the recipients' rights in the Source Code Form.

### 3.2. Distribution of Executable Form

If You distribute Covered Software in Executable Form then:

- (a) such Covered Software must also be made available in Source Code Form, as described in Section 3.1, and You must inform recipients of the Executable Form how they can obtain a copy of such Source Code Form by reasonable means in a timely manner, at a charge no more than the cost of distribution to the recipient; and
- (b) You may distribute such Executable Form under the terms of this License, or sublicense it under different terms, provided that the license for the Executable Form does not attempt to limit or alter the recipients' rights in the Source Code Form under this License.

### 3.3. Distribution of a Larger Work

You may create and distribute a Larger Work under terms of Your choice, provided that You also comply with the requirements of this License for the Covered Software. If the Larger Work is a combination of Covered Software with a work governed by one or more Secondary Licenses, and the Covered Software is not Incompatible With Secondary Licenses, this License permits You to additionally distribute such Covered Software under the terms of such Secondary License(s), so that the recipient of the Larger Work may, at their option, further distribute the Covered Software under the terms of either this License or such Secondary License(s).

### 3.4. Notices

You may not remove or alter the substance of any license notices (including copyright notices, patent notices, disclaimers of warranty, or limitations of liability) contained within the Source Code Form of the Covered Software, except that You may alter any license notices to



the extent required to remedy known factual inaccuracies.

### 3.5. Application of Additional Terms

You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, You may do so only on Your own behalf, and not on behalf of any Contributor. You must make it absolutely clear that any such warranty, support, indemnity, or liability obligation is offered by You alone, and You hereby agree to indemnify every Contributor for any liability incurred by such Contributor as a result of warranty, support, indemnity or liability terms You offer. You may include additional disclaimers of warranty and limitations of liability specific to any jurisdiction.

### 4. Inability to Comply Due to Statute or Regulation

-----

If it is impossible for You to comply with any of the terms of this License with respect to some or all of the Covered Software due to statute, judicial order, or regulation then You must: (a) comply with the terms of this License to the maximum extent possible; and (b) describe the limitations and the code they affect. Such description must be placed in a text file included with all distributions of the Covered Software under this License. Except to the extent prohibited by statute or regulation, such description must be sufficiently detailed for a recipient of ordinary skill to be able to understand it.

### 5. Termination

-----

5.1. The rights granted under this License will terminate automatically if You fail to comply with any of its terms. However, if You become compliant, then the rights granted under this License from a particular Contributor are reinstated (a) provisionally, unless and until such Contributor explicitly and finally terminates Your grants, and (b) on an ongoing basis, if such Contributor fails to notify You of the non-compliance by some reasonable means prior to 60 days after You have come back into compliance. Moreover, Your grants from a particular Contributor are reinstated on an ongoing basis if such Contributor notifies You of the non-compliance by some reasonable means, this is the first time You have received notice of non-compliance with this License from such Contributor, and You become compliant prior to 30 days after Your receipt of the notice.

5.2. If You initiate litigation against any entity by asserting a patent infringement claim (excluding declaratory judgment actions, counter-claims, and cross-claims) alleging that a Contributor Version directly or indirectly infringes any patent, then the rights granted to You by any and all Contributors for the Covered Software under Section 2.1 of this License shall terminate.

5.3. In the event of termination under Sections 5.1 or 5.2 above, all end user license agreements (excluding distributors and resellers) which have been validly granted by You or Your distributors under this License prior to termination shall survive termination.

```
*****
*
* 6. Disclaimer of Warranty
* -----
*
* Covered Software is provided under this License on an "as is"
* basis, without warranty of any kind, either expressed, implied, or
* statutory, including, without limitation, warranties that the
* Covered Software is free of defects, merchantable, fit for a
* particular purpose or non-infringing. The entire risk as to the
* quality and performance of the Covered Software is with You.
* Should any Covered Software prove defective in any respect, You
* (not any Contributor) assume the cost of any necessary servicing,
*
```

```
* repair, or correction. This disclaimer of warranty constitutes an
* essential part of this License. No use of any Covered Software is
* authorized under this License except under this disclaimer.
*
```

```
*****
```

```
*****
```

#### \* 7. Limitation of Liability \*

```
* ----- *
```

```
*
* Under no circumstances and under no legal theory, whether tort
* (including negligence), contract, or otherwise, shall any
* Contributor, or anyone who distributes Covered Software as
* permitted above, be liable to You for any direct, indirect,
* special, incidental, or consequential damages of any character
* including, without limitation, damages for lost profits, loss of
* goodwill, work stoppage, computer failure or malfunction, or any
* and all other commercial damages or losses, even if such party
* shall have been informed of the possibility of such damages. This
* limitation of liability shall not apply to liability for death or
* personal injury resulting from such party's negligence to the
* extent applicable law prohibits such limitation. Some
* jurisdictions do not allow the exclusion or limitation of
* incidental or consequential damages, so this exclusion and
* limitation may not apply to You.
*
```

```
*****
```

#### 8. Litigation

```
-----
```

Any litigation relating to this License may be brought only in the courts of a jurisdiction where the defendant maintains its principal place of business and such litigation shall be governed by laws of that jurisdiction, without reference to its conflict-of-law provisions. Nothing in this Section shall prevent a party's ability to bring cross-claims or counter-claims.

#### 9. Miscellaneous

```
-----
```

This License represents the complete agreement concerning the subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not be used to construe this License against a Contributor.

#### 10. Versions of the License

```
-----
```

##### 10.1. New Versions

Mozilla Foundation is the license steward. Except as provided in Section 10.3, no one other than the license steward has the right to modify or publish new versions of this License. Each version will be given a distinguishing version number.

##### 10.2. Effect of New Versions

You may distribute the Covered Software under the terms of the version of the License under which You originally received the Covered Software, or under the terms of any subsequent version published by the license steward.

##### 10.3. Modified Versions

If you create software not governed by this License, and you want to create a new license for such software, you may create and use a

modified version of this License if you rename the license and remove any references to the name of the license steward (except to note that such modified license differs from this License).

#### 10.4. Distributing Source Code Form that is Incompatible With Secondary Licenses

If You choose to distribute Source Code Form that is Incompatible With Secondary Licenses under the terms of this version of the License, the notice described in Exhibit B of this License must be attached.

##### Exhibit A - Source Code Form License Notice

-----

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

If it is not possible or desirable to put the notice in a particular file, then You may include the notice in a location (such as a LICENSE file in a relevant directory) where a recipient would be likely to look for such a notice.

You may add additional accurate notices of copyright ownership.

##### Exhibit B - "Incompatible With Secondary Licenses" Notice

-----

This Source Code Form is "Incompatible With Secondary Licenses", as defined by the Mozilla Public License, v. 2.0.

## Zlib

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

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

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