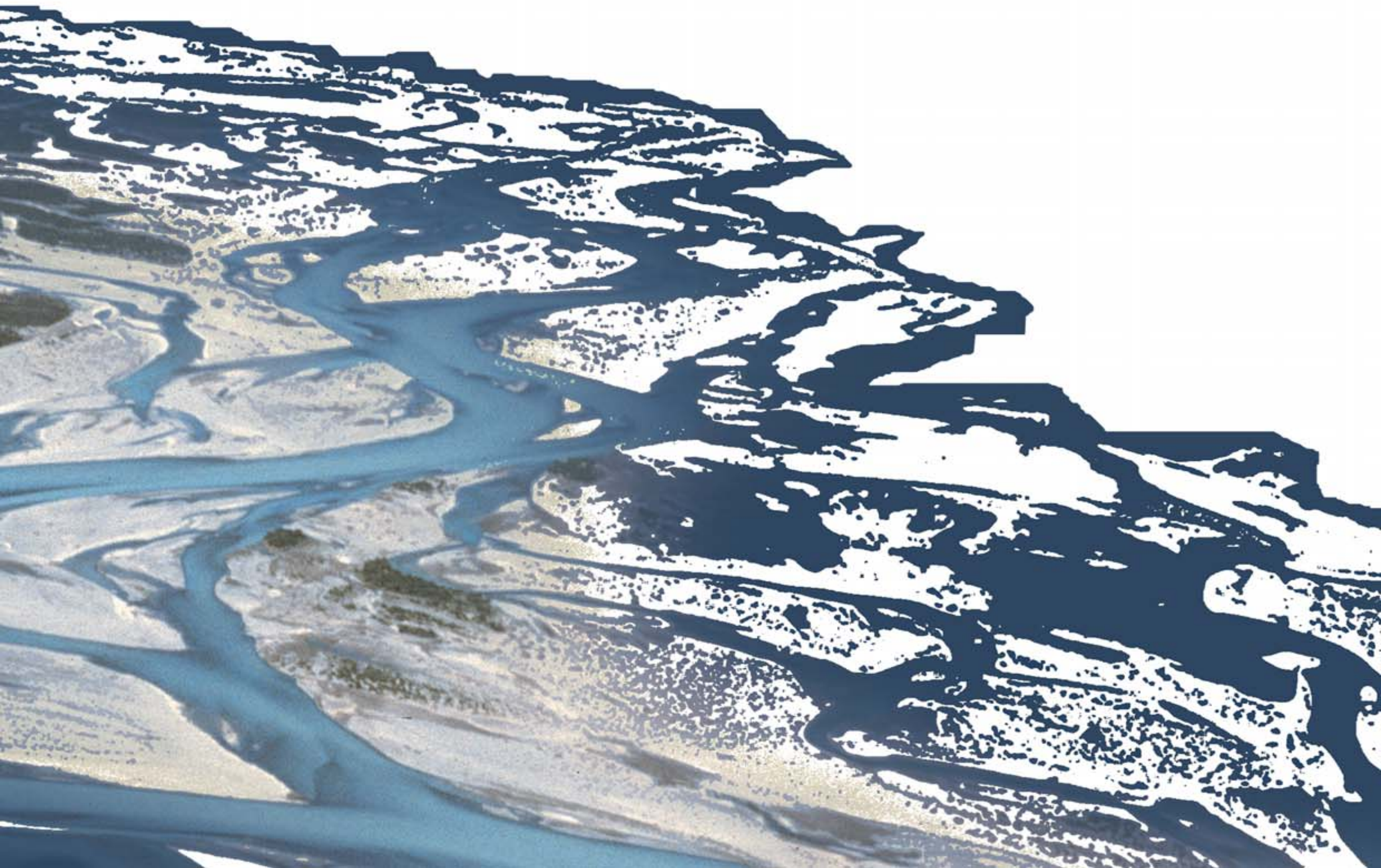
BASEMENT

BASIC SIMULATION ENVIRONMENT
FOR SIMULATION OF ENVIRONMENTAL FLOW
AND NATURAL HAZRAD SIMULATION

SYSTEM MANUALS

VERSION 3.0
September 2019



Preface

Preface to Versions 1.0 – 1.3

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

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

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

After four years of designing, implementing and testing, the software system BASEMENT has reached a state to go public. The documentation at hand confirms the invested diligence to create a transparent software system of high quality. The software, in terms of an executable computer program, and its documentation are available free of charge. It can be used by anyone who wants to run numerical simulations of rivers and sediment transport – either for training or for commercial purposes.

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

Minor

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

October, 2006

Preface to Versions 1.4

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

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

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

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

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

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

Concerning sediment transport, the one-dimensional model BASEchain now supports the modelling of fine material, either as suspended or bed load. Also the advanced models for boundary conditions are worth mentioning. On the one hand, it is now possible to model domain boundaries with momentum and on the other hand, special boundary conditions inside the computational region, such as a weir or a gate, are implemented. The fact, that the version 1.4 of BASEMENT is also available for the Linux operating system the first time, rounds off the new additions and features of the software package at hand.

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

D. Vetsch
Project Supervisor

October, 2008

Preface to Version 2.0

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

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

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

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

Moreover, the visualisation of actual results during a simulation with BASEviz has been improved and is now more interactive, i.e. the simulation can be paused, continued or the variable shown can be switched. Other improvements concern computational efficiency and sediment transport, especially gravitational bed load transport. Please refer to the release notes in the section “introduction and installation” of this manual for further details about new features and bug fixes.

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

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

May, 2010

Preface to Version 2.8

End of an era

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

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

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

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

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

May, 2018

Preface to Version 3.0

Beginning of a new software era

Computing performance has always been a challenging issue with respect to the development of the BASEMENT software. Since the first official release of the software in 2006, several approaches have been tested and implemented. For instance, parallelization features were included in version 1.4 (2008) using OpenMP, which allowed the use of nowadays common multi-core processors (CPUs). Additionally, the “cycle-step” was released with version 2.2 (2011) for accelerating morphodynamic simulations in particular. All attempts to improve the efficiency of the software were carried out with the requirement to maintain model accuracy and stability - this principle holds to this day.

However, growing complexity due to increasing number of model features led to technical constraints that posed limits to potential performance gain and to the adaption of corresponding software-technical concepts. Especially the progresses in the field of general-purpose computation on graphic processing units (GPGPU), the availability of corresponding hardware at affordable price and the availability of the Oxford Parallel Domain Specific Language OP2 for unstructured meshes, have led to the decision to rewrite the BASEMENT software from scratch. The resulting new major version 3.0 has three main pillars that contribute to improved computing performance: pure first-order finite volume discretization, strong scaling of multi-core CPU simulations and GPGPU acceleration for large computational meshes. This results in four to more than 90 times faster simulations compared to the previous version 2.8.

Version 3.0 comes with a new user interface for model setup and simulation including a workflow with separated tasks that enables customizable simulations. For now, the features of version 3.0 are still limited to 2-D hydro- and morphodynamics and bed-load transport with uniform sediment. The roadmap for further development of the BASEMENT software includes the enhancement of version 3.0 towards the simulation capabilities of version 2.8 and further maintenance of version 2.8 at least for the next five years. In doing so, mixed sediment transport, suspended load transport and the revision of the 1-D model will be implemented with high priority.

The development of the new 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 been financially supporting the BASEMENT project since 2002, first in the scope of the framework “Hydraulic Engineering and Ecology” and since 2014, as an independent project, which is greatly appreciated.

In the name of the project team
Dr. D. Vetsch, Project Director
Prof. Dr. R. Boes, Director VAW

September, 2019

Preamble

VERSION 3.0.2

March 2020

Credits

Current Team Members

Software Development, Documentation and Test (alphabetical)

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

Scientific Board

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

Former Project Members

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

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

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., Bürgler M., Caponi F., Conde D., Gerke E., Kammerer S., Koch A., Peter S., Vanzo D., Vonwiller L., Weberndorfer M. 2020. System Manuals of BASEMENT, Version 3.0. 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, 2020. https://www.basement.ethz.ch

For Software:

BASEMENT – Basic Simulation Environment for Computation of Environmental Flow and Natural Hazard Simulation. Version 3.0. © ETH Zurich, VAW, 2020.

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

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 AKNOWLEDGMENT

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 SIMULATION OF ENVIRONMENTAL FLOW
AND NATURAL HAZRAD SIMULATION

INTRODUCTION & INSTALLATION

VERSION 3.0
September 2019

Contents

1 Summary of Contents **3**

1.1 Introduction . . . 3
1.2 Content of System Manuals . . . 3
1.2.1 Introduction & Installation . . . 3
1.2.2 User Manual . . . 3
1.2.3 Reference Manual . . . 4
1.2.4 Tutorials and Test Cases . . . 4

2 Setup and First Start **5**

2.1 Setup and First Start . . . 5
2.1.1 System operator requirements . . . 5
2.1.1.1 Microsoft Windows . . . 5
2.1.1.2 Linux . . . 5
2.1.1.3 Hardware Configuration . . . 6
2.1.2 Installing under Windows . . . 6
2.1.2.1 Getting the binaries . . . 6
2.1.2.2 Installation procedure under Windows 10 . . . 6
2.1.3 Installing under Linux . . . 9
2.1.3.1 Getting the binaries . . . 9
2.1.3.2 Installation procedure . . . 9
2.2 Running BASEMENT . . . 10
2.2.1 Windows 10 . . . 10
2.2.2 Linux . . . 10

3 Migration Guide Version 2.8 to 3.0 **11**

3.1 General . . . 11
3.2 Major Changes from version 2.x to 3.x . . . 11
3.2.1 Workflow . . . 11
3.2.2 Input data . . . 12
3.2.2.1 Mesh features . . . 12
3.2.2.2 Model setup . . . 13
3.2.3 Simulation . . . 17
3.2.4 Results . . . 19
3.3 Case example . . . 19
3.3.1 Description . . . 19
3.3.2 Computational mesh . . . 19
3.3.2.1 Quality mesh . . . 20
3.3.2.2 Computational mesh . . . 20

3.3.2.3 Import of a 2.x to a compatible 3.x computational mesh . . 21
3.3.3 Setup and simulation 24
3.3.4 Results and discussion 26
3.3.4.1 Hydraulic results 26
3.3.4.2 Boundary conditions 26
3.3.4.3 Discussion 29
3.4 Performance 29
3.4.1 Introduction 29
3.4.2 Speedup 29
3.4.3 Scalability 29

4 Release Notes 33

4.1 Version 3.0 33
4.1.1 General 33
4.1.2 Hydraulics 33
4.1.3 Morphology 34
4.1.4 System manuals 34
4.2 Version 3.0.1 34
4.2.1 Bug Fixes 34
4.3 Version 3.0.2 35
4.3.1 General 35
4.3.2 Bug Fixes 35
4.3.3 Known Issues 35

5 Summary of Features Version 3.0 37

5.1 Hydrodynamic features 37
5.2 Morphodynamic features 38

6 References 41

1

Summary of Contents

1.1 Introduction

BASEMENT version 3.0 is a freeware simulation tool for hydro- and morphodynamic modelling developed at the Laboratory of Hydraulics, Hydrology and Glaziology (VAW) of the ETH Zurich. The software provides a precise and powerful tool for the simulation of river hydro- and morphodynamics. BASEMENT system manual provides information about BASEMENT version 3.0 and guides the user through the process of using BASMENT version 3.0.

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 3.0 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 version 2.x and version 3.x are described in the migration guide for the users already familiars with BASEMENT. The release notes summarize the changes introduced by BASEMENT version 3.0 and the summary of features provides an overview of the available functionalities of BASEMENT version 3.0.

1.2.2 User Manual

The user manual provides information about the simulation environment of BASEMENT version 3.0. 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 3.0.

1.2.4 Tutorials and Test Cases

This part is composed of three tutorials and two test cases. The tutorials guide the user throught the pre-processing, the numerical simulation and post-processing stages of BASEMENT version 3.0 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 acuracy of the simulations performed with BASEMENT version 3.0 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 System operator requirements

2.1.1.1 Microsoft Windows

BASEMENT version 3.0 has been tested for MS Windows 10. 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 is available for the following Linux (x86-64) systems:

- Ubuntu 16.04 (LTS), alias “Xenial Xerus”:
  - Kernel version 4.4
  - GNU C Library (glibc) version 2.23
  - VTK-version: 5.10
  - 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 18.04 (LTS), alias “Bionic Beaver”:
  - Kernel version 4.15
  - 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)

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 3.0 is **only** possible for CUDA-enabled (Compute Unified Device Architecture) GPUs produced by NVIDIA. BASEMENT version 3.0 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	Quadro P620
Memory [GB]	5	12	11	8	4	2
Architecture	Kepler	Pascal	Pascal	Pascal	Pascal	Pascal
Bandwidth[GB/s]	208	549	484	256	112	80
CUDA cores	2496	3584	3584	2432	768	512

2.1.2 Installing under Windows

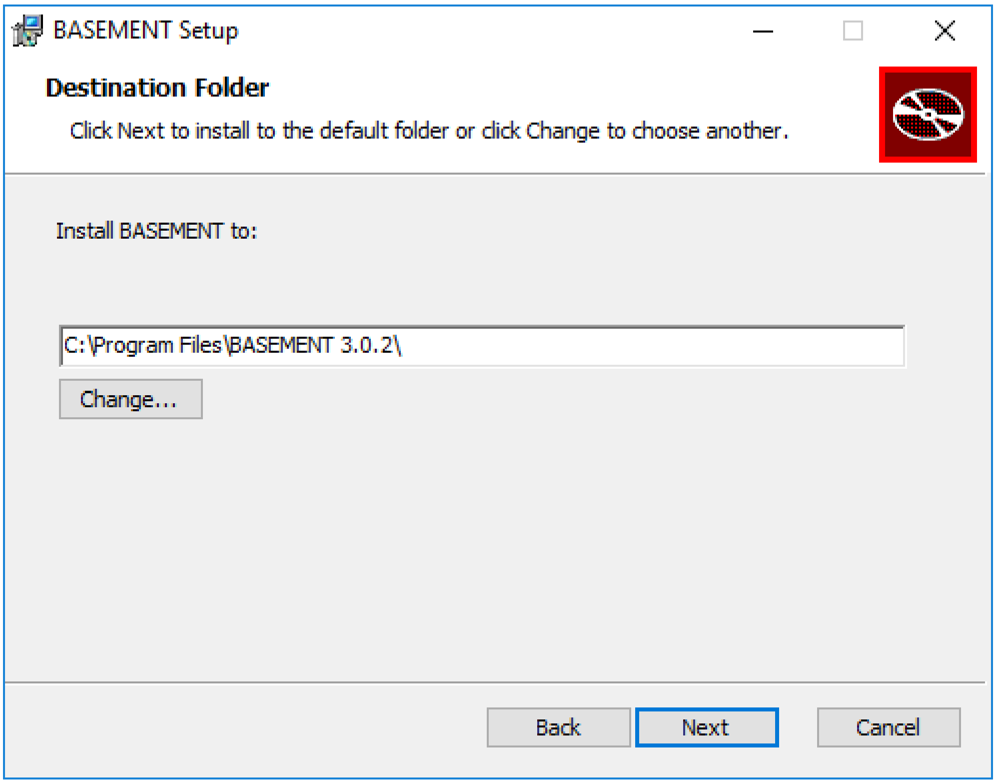
BASEMENT version 3.0 is available for Microsoft Windows Windows 10 operating system.

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 3.x) free of charge.

2.1.2.2 Installation procedure under Windows 10

Please note, that existing installations are not automatically detected by the installer. Therefore, uninstall any previous BASEMENT version before installing a more recent



Destination Folder

Click Next to install to the default folder or click Change to choose another.

Install BASEMENT to:

C:\Program Files\BASEMENT 3.0.2\

Figure 2.1 *Select the installation folder.*

version e.g. using the link in the start menu. After downloading the version 3.x 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 3.0.2" (Figure 2.1). You are free to choose any other directory.

Step 3 and 4: Confirming and finishing the installation

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 3.x (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.

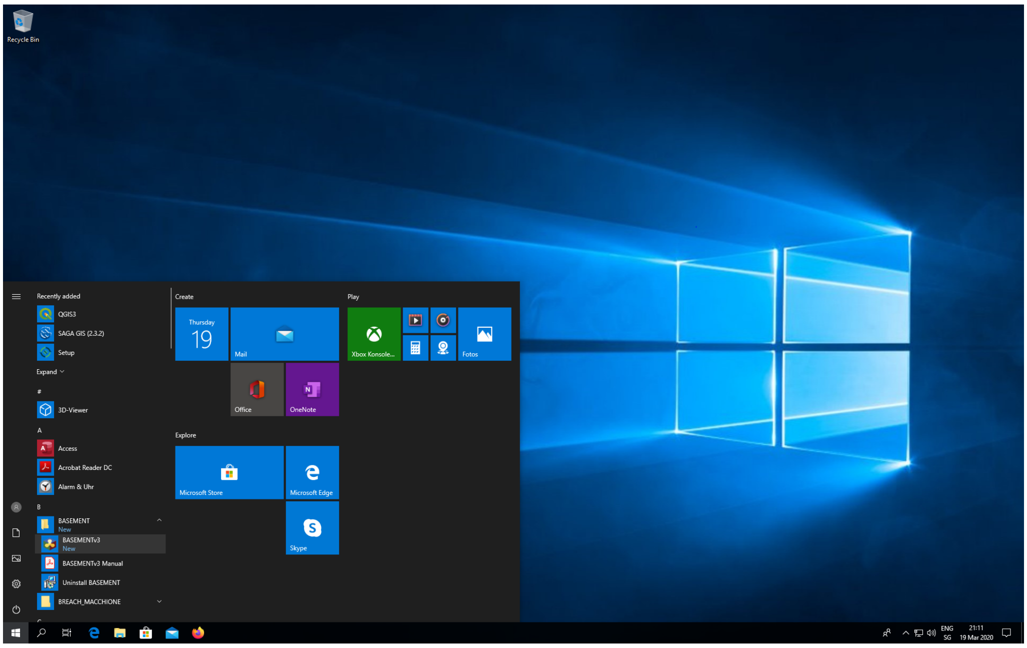


Figure 2.2 *Start BASEMENT over the icon in the Start Menu folder.*

Scenario Directory: Please choose a directory for the configuration and result files.

BASEMENT

BAsic Simulation EnvironMENT
for Computation of Environmental Flow
and Natural Hazard Simulation

Version 3.0.2 / March 2020

(C) 2020 ETH Zurich / Laboratory of hydraulics, hydrology and glaciology (VAW)

www.basement.ethz.ch

Figure 2.3 *The graphical user interface.*

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):

```
$ 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, than 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

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 Version 2.8 to 3.0

3.1 General

***Table 3.1** List of BASEMENT main features*

	Version 2.8	Version 3.0
1-D model	✓	
2-D model	✓	✓
Hydrodynamics	✓	✓
Morphodynamics:		
- Bed load	✓	✓
- Suspended load	✓	
External sub-domain	✓	
Model coupling (multi domain)	✓	
Controller	✓	
Subsurface flow	✓	
Vegetation	✓	
SMP hardware	✓	✓
GPU/HPC support		✓

3.2 Major Changes from version 2.x to 3.x

3.2.1 Workflow

Table 3.2 *Major changes in workflow*

	Version 2.8	**Version 3.0**
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)
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

3.2.2 Input data

3.2.2.1 Mesh features

Table 3.3 *Main changes regarding the computational mesh*

Version 2.8	**Version 3.0**
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
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 element_ids	Domain differentiation with regiondef

Table 3.4 *Main changes regarding the grid generation with BASEmesh*

Version 2.8	**Version 3.0**
Single procedure to generate a .2dm file with BASEmesh	Two procedures to generate a .2dm file with BASEmesh according to the mesh resolution
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

3.2.2.2 Model setup

Table 3.5 *Main changes regarding model setup*

	Version 2.8	**Version 3.0**
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 3.6 *Main changes in the* ***hydraulics*** *block of the domain BASEPLANE_2D*

	Version 2.8	**Version 3.0**
Parameters:		
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

	Version 2.8	**Version 3.0**
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
Safe mode	no	yes
Friction		
Type	Manning Strickler Chezy Yalin Darcy-Weissbach Bezzola	Manning Strickler Chezy - - Bezzola
Wall friction	yes	no
Grain size friction	yes	no
Boundary		
Type	-	**Standard**
	hydrograph	uniform_in uniform_out
	-	**Standard**
	zhydrograph	zhydrograph **Linked** zhydrograph_linked
	zero_gradient	zero_gradient_out
	weir	weir_out_constant, weir_out_dynamic **Linked** weir_linked_constant, weir_linked_dynamic
	gate	-
		Standard
	HQ_relation	hqrelation_out **Linked** 2way_hqrelation_linked, hqrelation_linked
	coupling	-
	wall	-
	-	**Internal**: wall_internal dynamic_wall_internal hqrelation_internal

	Version 2.8	**Version 3.0**
File type	hydrograph, weir, gate, hqrelation	discharge, weir elevation, hqrelation, wse
Boundary inside the computational domain	Inner boundary (weir, gate and hqrelation)	**Internal** boundary: wall, dynamic wall and h-Q relation **Linked** boundary: weir, h-Q relation
Turbulence model	yes	no
External source		
Type	source discharge	total and distributed
Sink behavior	negative source discharge values	exact, available, infinity
Initial		
Type	dry continue index_table	dry continue region_defined
Flood tracking	no	yes

Table 3.7 *Main changes in the* ***morphology*** *block of the domain BASEPLANE_2D*

	Version 2.8	**Version 3.0**
Parameter		
Active layer	yes (control_volume)	no
Porosity	porosity	sediment_porosity
Density	density	sediment_density
Starting time	- (bedload)	morphodynamic_start
morph_cycle	yes	no
morphological factor	no	yes
time scalling	no	with morphological factor

	Version 2.8	**Version 3.0**
Create new layers	yes	no
Grid perturbation (random)	distortion	-
Bedmaterial		
Grain class	Single or multi grain classes	Single grain class
Layer	Multiple layers	Single layer
Fix bed elevation	.2dm mesh or node list	over region (index)
Bedload		
Bedload transport	Simple upwind scheme	Godunov-type upwind scheme
Closure formula	mpm	MPM
	-	MPM-like (adaptable)
	engelundhansen	Engelund and Hansen
	mpmh	-
	power_law	Grass-like (adaptable)
	mpm_multi	-
	wilcockcrowe	-
	ashidamichiue	-
	parker	-
	rickenmann	-
	smartjaeggi	-
	smartjaeggi_multi	-
	wu	-
	vanrijn	-
Boundary		
- Inflow	-	**Standard**
	sediment_discharge	sedimentograph
	-	sedimentograph_warea
	-	sedimentograph_conveyance
	IOUp	equilibrium_in
	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	IKEDA
	curvature_effect_static	-

	Version 2.8	**Version 3.0**
	curvature_effect_dynamic	CURVATURE
Inner boundary	yes (weir, open)	no
Incipient motion	angle_of_repose	repose_angle
	local_slope_vanrijn	van_rijn
	local_slope_chen	chen_et_al
Gravitational transport	yes	no
Source		
Type	sediment_discharge	-
	dredge	-

3.2.3 Simulation

Table 3.8 *Main changes regarding simulation parameters*

	Version 2.8	**Version 3.0**
Command file type	run.bmc	simulation.json
Simulation time	start_time	start
	total_run_time	end
	output_time_step	out
	restart_time_step	-
	console_time_step	-
	reference_time	-
Timestep	initial_time_step	init
	minimum_time_step	minimum

	Version 2.8	**Version 3.0**
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	-
	susp_deltaz	-
	susp_total_pickup	-
	susp_total_deposition	-
	susp_grain_pickup	-
	susp_grain_deposition	-
	theta_critical	-
	grain_size	-
	grain_bedload	-
	bedload_vec	-
	saturation	-
	sediment_sum	-
	-	flow_radius
	-	flow_curvature
	-	flood_tracking
	pore_pressure	-
	-	ns_hyd_discharge
	-	ns_mor_discharge
	external_source_discharge	-
	radius_curvature	-
	radius_curvature_abs	-
	momentum	-
	water_table	-
	biomass	-
	carrying_cap	-
	source_friction	-
	source_wall_friction	-
	source_internal_friction	-
	source_bed	-
	balance_discharge_fluxes	-
	balance_momentum_fluxes	-

3.2.4 Results

Table 3.9 *Main changes regarding the results parameters*

	Version 2.8	**Version 3.0**
Command file type	run.bmc	results.json
Format	ascii, sms, tecplot, shape, vtk	xdmf
Output Type	node_centered	-
	element_centered	element_centered
	BASEviz	-
	node_history	-
	element_history	-
	stringdef_history	nodestring
	edge_history	-
	boundary_history	nodestring
	balance	-
	avs_ucd	-
	sediment_grid	-

3.3 Case example

3.3.1 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 changes between BASEMENT version 2.8 (v2.x) and 3.0 (v3.x). 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 3.10.

Table 3.10 *Geometry of trapezoidal channel*

Type	**Value**	**Unit**
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

3.3.2 Computational mesh

The topology of the computational mesh used for BASEMENT v3.x is different than for version 2.x, see Tables 3.3 and 3.4. This section describes the differences between the two

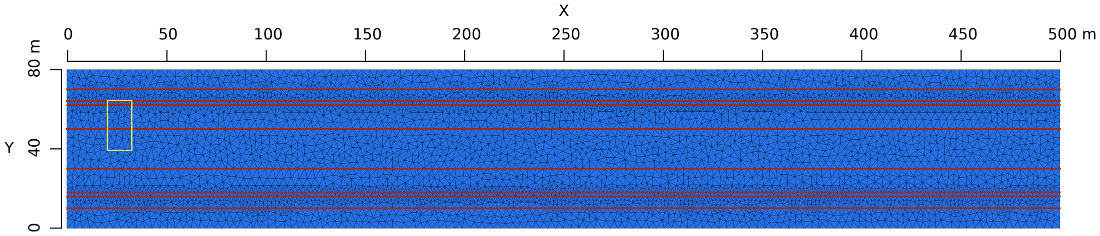


Figure 3.1 *Quality mesh used for the case example with breaklines (red)*

mesh types and provides a guideline on how to import a v2.x mesh into BASEMENT version 3.x.

3.3.2.1 Quality mesh

Table 3.11 *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 3.1) is identical for both versions, v2.x and v3.x and its attributes are listed in Table 3.11. The procedure to generate a quality mesh with QGIS using the BASEmesh plugin is explain in the Tutorial of BASEMENT v2.8 documentation.

3.3.2.2 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 BASEMENT version 3.x and version 2.x lies in the process of attributing the elevation information to the mesh cells. A small surface area (yellow rectangle, Figure 3.1) is schematically reproduced on Figure 3.2 in order to illustrate the two approaches used to create the computational mesh.

In BASEMENT version 2.x, the topographic elevation is attributed to the cell vertices (Figure 3.2 a). The quality mesh defines the location on the elevation model at which the elevation information 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 BASEMENT version 3.x, the elevation information

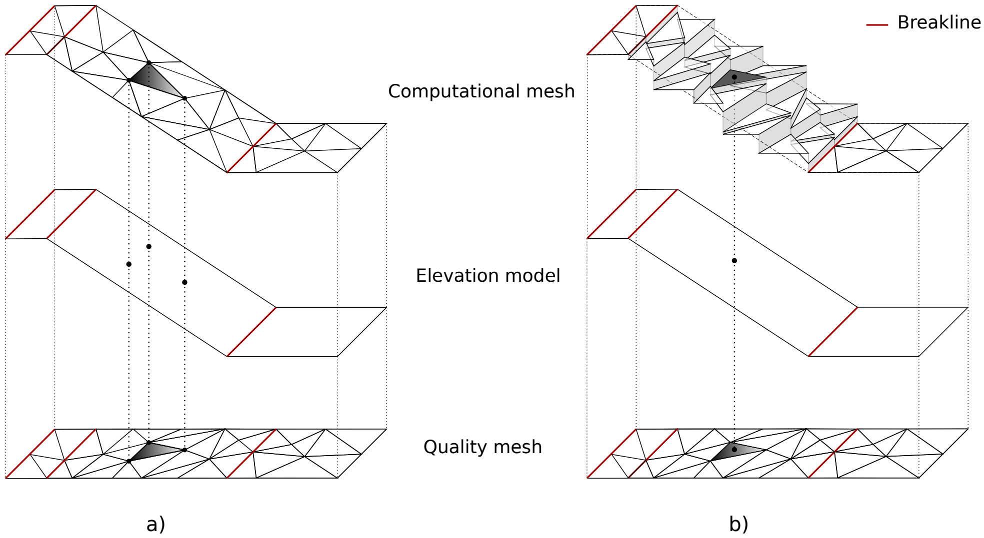


Figure 3.2 *Schematic sketch of the elevation interpolation methods with breaklines (red): a) BASEMENT version 2.x b) BASEMENT version 3.x*

is assigned to the coordinate of the cell center, resulting in a constant elevation over the cell surface (Figure 3.2 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 BASEMENT version 2.x, 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 BASEMENT version 3.x, 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 garanteed by two breaklines).

The computational mesh of the trapezoidal channel for the simulation with BASEMENT version 2.x is represented on Figure 3.3 and the computational mesh for the simulation with BASEMENT version 3.x on Figure 3.4. The flow direction is from top to bottom.

3.3.2.3 Import of a 2.x to a compatible 3.x computational mesh

The computational mesh of BASEMENT version 3.x 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 3.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 .2dm file and the stringdefs list is saved in a separate .txt file. In order to use the computational mesh version 2.x for simulations with BASEMENT version 3.x, the .2dm mesh file has to be modified:

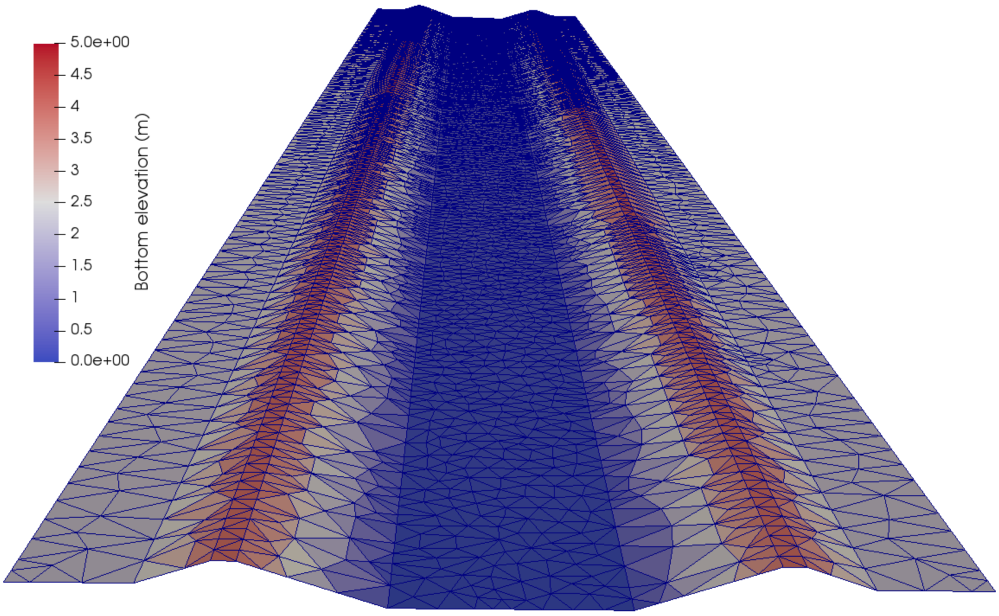


Figure 3.3 *Computational grid BASEMENT version 2.x with breaklines (view from downstream)*

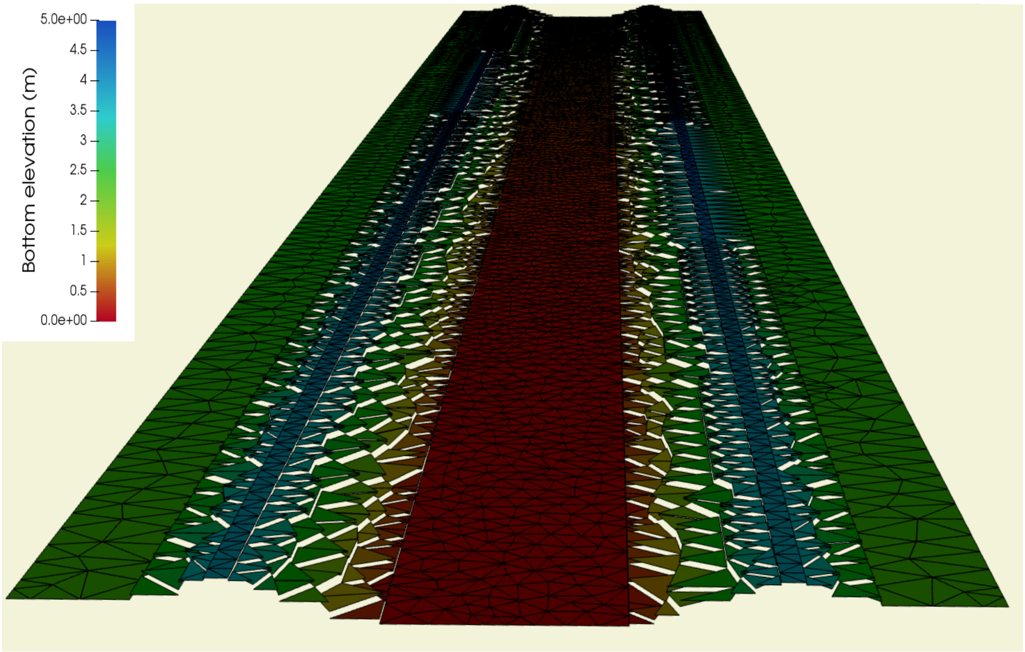
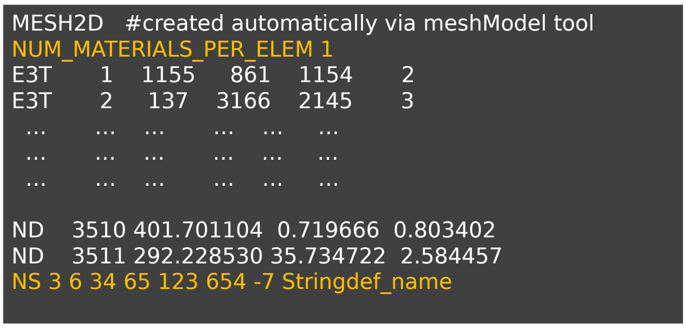


Figure 3.4 *Computational grid BASEMENT version 3.x with breaklines (view from downstream)*



```
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 3.5 *Lines to add manually to the 2dm mesh file (orange)*

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 3.5). 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 3.3.3), 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 BASEMENT version 3.x. 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 3.6, where a cross section of the trapezoidal mesh illustrates the local differences between the mesh of BASEMENT version 2.x and the different interpolated meshes used in simulations with BASEMENT v3.x.

Moreover, Figure 3.7 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 3.6 and most of the interpolation methods can’t preserve the bank elevation. In the case of a numerical simulation where the exact elevation of the bank is required (e.g. to calculate the bordfull discharge), the definition of breaklines ensures the conservation of the bank elevation, independently from the chosen interpolation methods. Otherwise, the

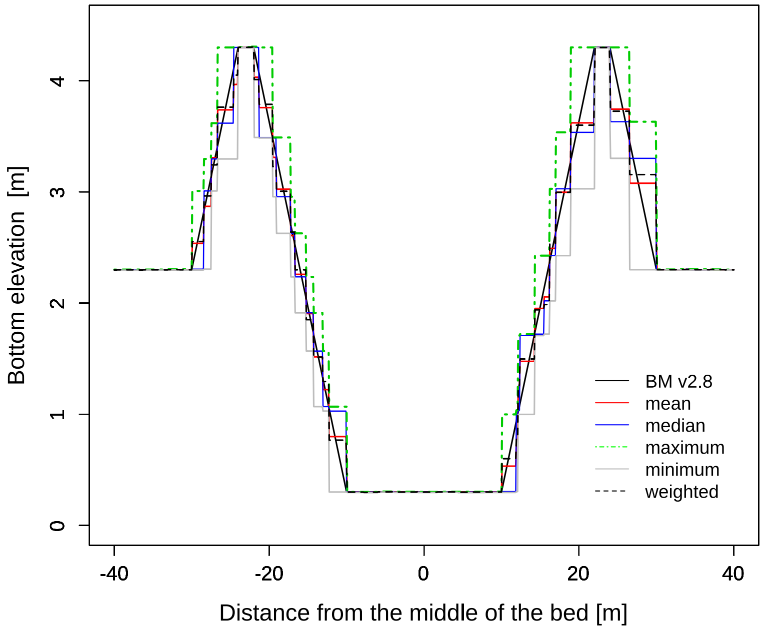


Figure 3.6 *Comparison of interpolation methods with the mesh of BASEMENT version 2.8 (BM v2.8) on a cross section at x= 150 m of the trapezoidal channel with breaklines*

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 3.8 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”.

3.3.3 Setup and simulation

The simulations were performed for all the interpolation methods using BASEMENT version 3.0. 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 3.0:

- Computational mesh (2dm), including stringdef specification
- Configuration files (model.json, simulation.json and results.json)
- Boundary condition data (.txt)

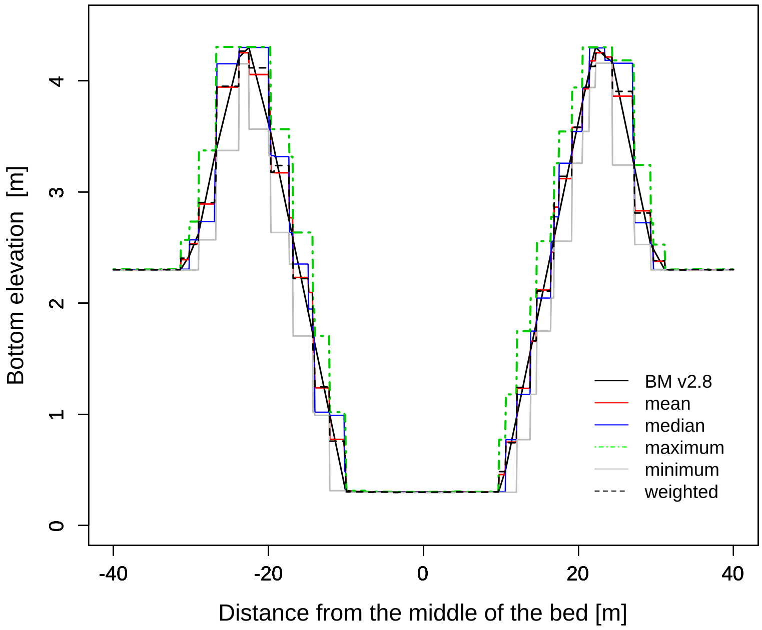


Figure 3.7 *Comparison of interpolation methods with the mesh of BASEMENT version 2.8 (BM v2.8) on a cross section at x= 150 m of the trapezoidal channel with only one breakline defined at the bank crest*

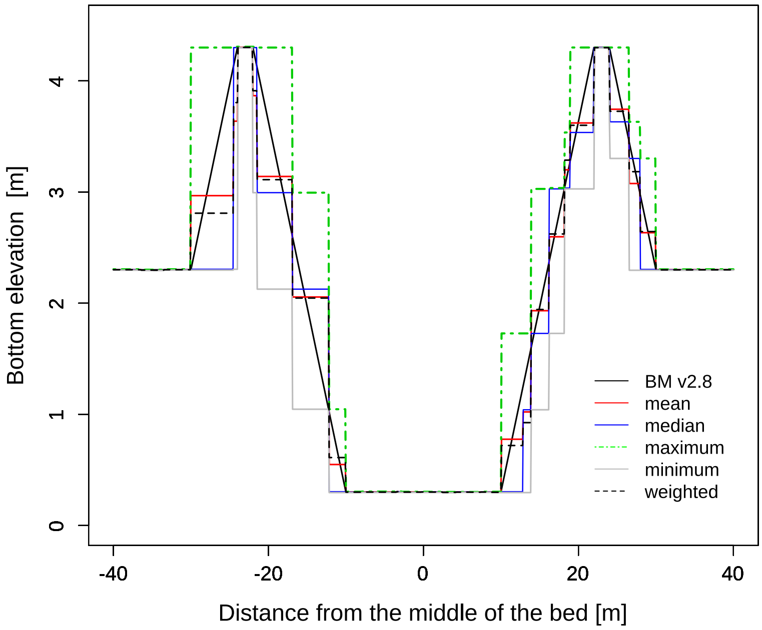


Figure 3.8 *Comparison of interpolation methods with the mesh of BASEMENT version 2.8 (BM v2.8) on a cross section at x= 150 m of the trapezoidal channel with breaklines and for a coarser mesh resolution*

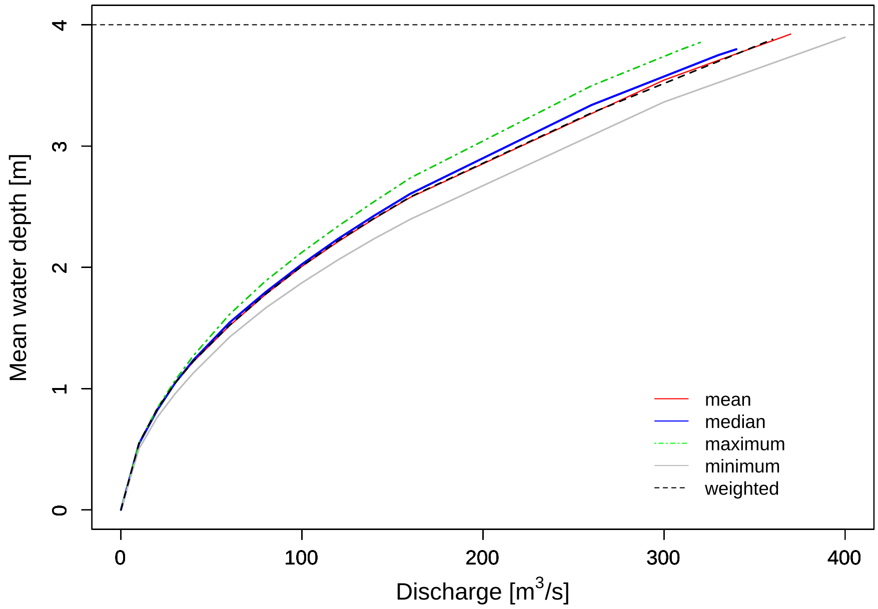


Figure 3.9 *Comparison of H-Q relations between the simulations of BASEMENT v3.0 for different interpolation types on the trapezoidal channel at x = 150 m.*

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 decribed in Section 3.3.2.3, the specification of stringsdefs, i.e. the list of nodes is included in the computational mesh (.2dm) in BASEMENT version 3.x.

3.3.4 Results and discussion

In BASEMENT version 3.x, the output data are generated either on cells (cell centered) or at the boundaries (stringdefs). Various results are available (see Table 3.9 and Table 3.8).

3.3.4.1 Hydraulic results

The result of the simulations with BASEMENT version 3.0 for different interpolation methods are compared in a stage discharge rating curve (Figure 3.9). The mesh features are summarized in Tables 3.10 and 3.11.

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”.

3.3.4.2 Boundary conditions

In BASEMENT version 3.0, the inflow data is averaged over the boundary length and the mean value is uniformly distributed over the cell edges. This assumption simplifies the

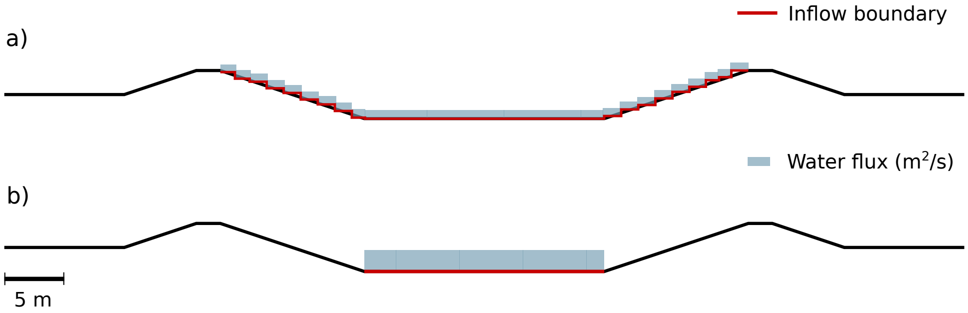


Figure 3.10 *Channel cross section and inflow boundary limit in BASEMENT version 3.0 a) Inflow boundary limit set at levee's highest point b) Reduced inflow boundary limit*

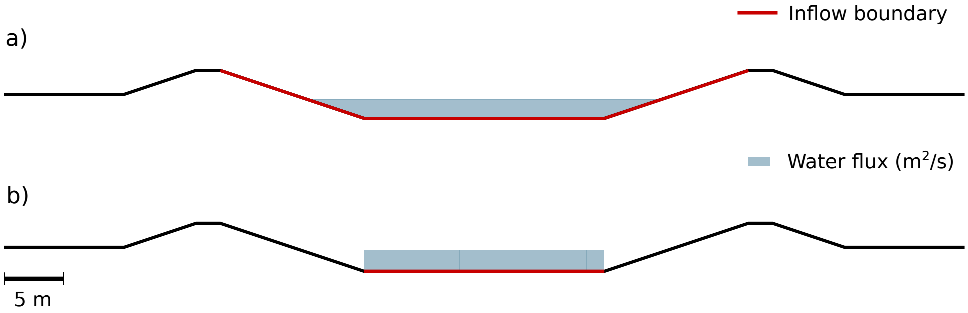


Figure 3.11 *Channel cross section and inflow boundary limit in BASEMENT version 2.8 a) Inflow boundary limit set at levee's highest point b) Reduced inflow boundary limit*

boundary conditions compared to BASEMENT version 2.8. Figure 3.10 and Figure 3.11 show two simplified representation of the averaged discharge value distribution on the element edges of the inflow boundary cross section for BASEMENT version 3.0 and BASEMENT version 2.8 respectively.

An inflow boundary defined between the top elevation of the two levees in BASEMENT version 3.0 (red line in Figure 3.10, a) generates an undesired converging flow from the bank towards the channel center (Q= 60 m^3/s) and small flux towards the floodplains as represented on Figure 3.12 for a discharge value Q= $200m^3/s$. An inflow boundary restricted to the channel bed width (Figure 3.10, 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 3.13 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 v3.0 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 50 nodes. In case of large computational mesh with fine resolution, the boundaries shall be split into several smaller stringdef of equal length and consequently, the discharge applied to the boundaries has to be adapted.

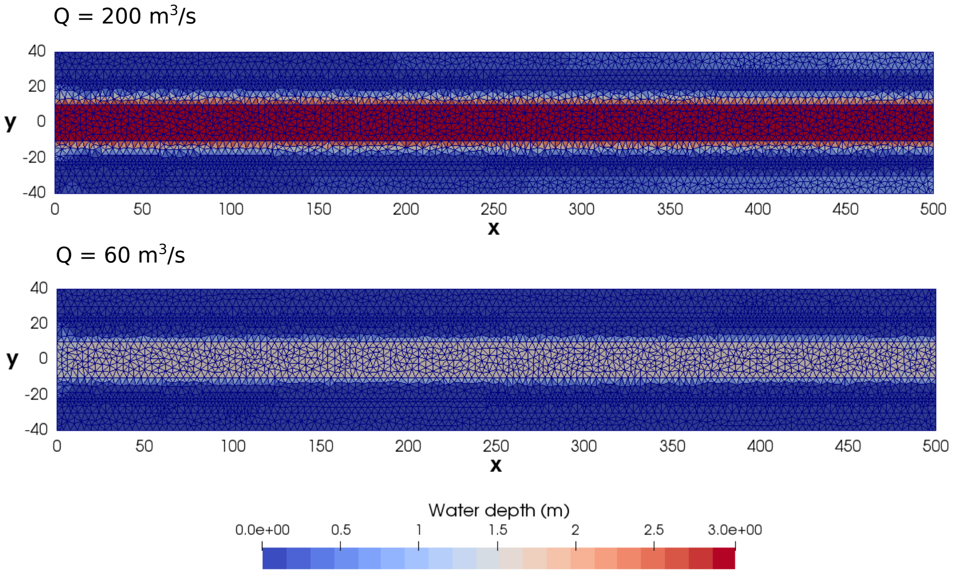


Figure 3.12 *Planar view of the simulation results using BASEMENT v3.0 of the trapezoidal channel with breaklines and for two discharge stages. Inflow boundary (x= 0.0 m) defined between the top elevation of the levees (Figure 3.10, a), inducing a converging flow from the levee towards the channel center and small fluxes towards the floodplains for higher discharge Q= 200* m^3/s

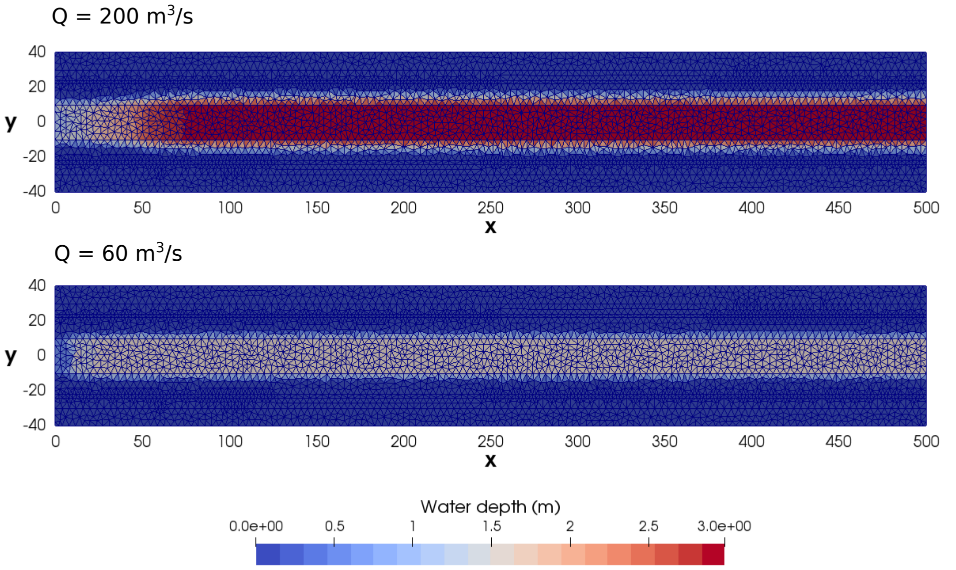


Figure 3.13 *Planar view of the simulation results using BASEMENT v3.0 of the trapezoidal channel with breaklines and for two discharge stages. Inflow boundary (x= 0.0m) restricted to the channel bed (Figure 3.10, b), inducing an increase of the flow velocity.*

3.3.4.3 Discussion

This case example of the hydraulic simulation of a trapezoidal channel pointed out the differences between BASEMENT version 2.8 and 3.0 for the topology and the boundary setup. The use of a BASEMENT v2.x mesh into BASEMENT v3.x is possible by interpolation but the simulation results may differ between the two versions due to the new topology. The simulation with BASEMENT v3.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.

3.4 Performance

3.4.1 Introduction

The performance of BASEMENT versions 2.8 and 3.0 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 more in details 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 BASEMENT version 2.8 using a CPU backend on 1, 2, 4, 8 and 12 cores and for BASEMENT version 3.0 using the CPU backend up to 32 cores and different GPU backends on Ubuntu 16.04. The backend types are listed and described in the section “Test case”. Beside of that, 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).

3.4.2 Speedup

The execution time of all the simulations is shown on Figure 3.14. The execution time increases with the computational mesh size. The CPU simulations performed with BASEMENT version 3.0 are executed faster than those performed with BASEMENT version 2.8. The execution speed increases even more if the simulations are performed on GPU processors with the best performance reached using sinle GPU processors.

The speedup $S = \frac{T_1}{T_N}$ is calculated as the ratio between serial and parallel execution time for a similar mesh size with T_1, the sequential execution time (Xeon 1) and T_N the execution time of the Nth different backends types.

3.4.3 Scalability

The speedup of the simulations performed with CPU are shown on Figure 3.15 for an increasing number of threads. The black line represents the ideal speedup according to the increasing number of threads.

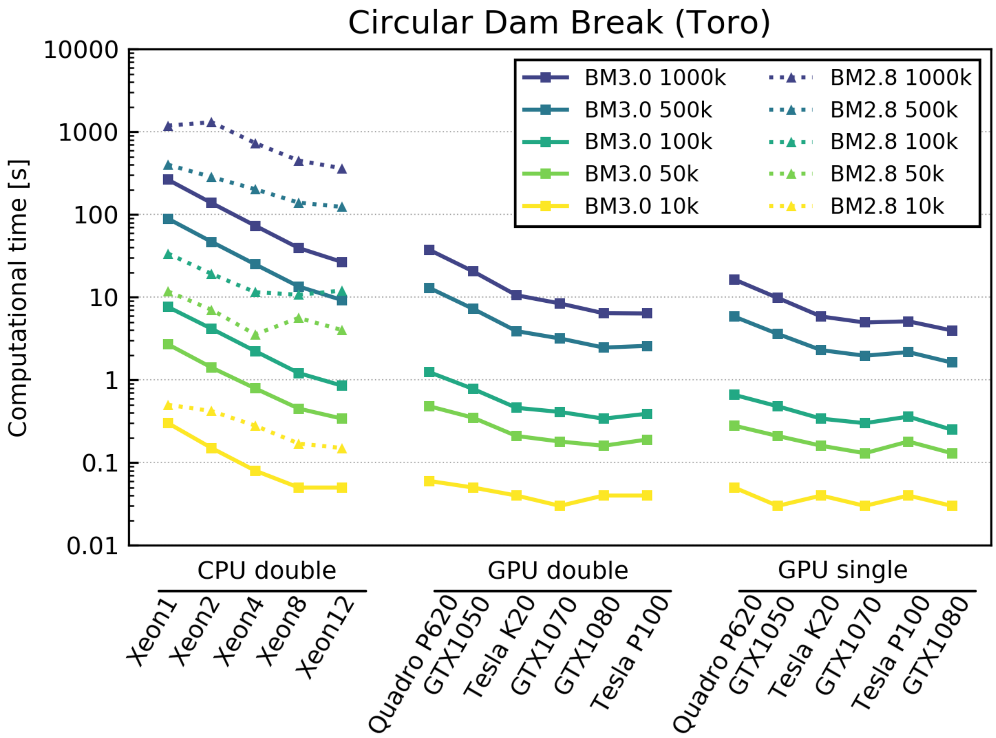


Figure 3.14 *Execution time of the circular dam break test case for different backends and mesh sizes*

The increasing number of threads for simulations performed with BASEMENT version 2.8 does not increase the speedup significally and leads to a maximal speedup (plateau) for small thread number already. The simulations performed with BASEMENT version 3.0 show a significant speedup with increasing number of threads and mesh elements. The speedup is more efficient for large meshes.

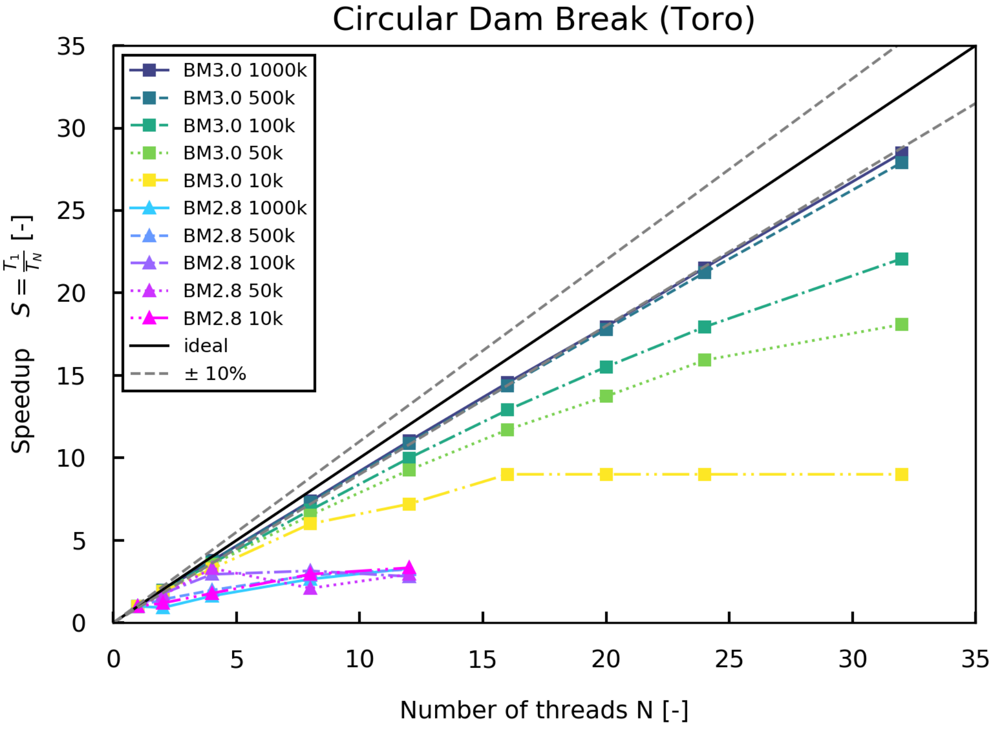


Figure 3.15 *Speedup compared to the sequential time (T_1) of the circular dam break test case performed on CPU for an increasing number of threads and different mesh sizes*

4

Release Notes

4.1 Version 3.0

4.1.1 General

- Supported operating systems: Windows 10, Linux Ubuntu 16.04 and 18.04
- Backend types: CPU, GPU (linux only), OpenMP
- New GUI (Graphic User Interface)
- New simulation workflow: numerical simulation in 3 steps (Setup, Simulation, Results) with separate executables for GUI, setup, results and for each simulation backend, e.g. for CPU, SMP and GPU
- Storage of setup and results files in HDF5 container (*.h5)
- New mesh (element centered, 1st order)
- Same mesh used for hydrodynamics and morphodynamics simulations
- Pre-processing: two procedures to generate a mesh using BASEmesh (QGIS plugin)
- Stringdef list and material indices included in the 2dm file
- json command files
- Restart and rerun
- Results in xdmf format
- The value at boundary condition is averaged over the stringdef length (hydraulics and morphology)

4.1.2 Hydraulics

- Boundary types: Wall (default), Standard, Linked (new) and Internal (new)
- Riemann Solver: HLLC, with hydrostatic reconstruction based on modified states (Duran et al., 2013)

- Sources with sink behaviours: exact, available and infinity
- Flood tracking
- Safe mode parameter

4.1.3 Morphology

- Bedload transport: HLL-type Approximate Riemann Solver (Soares-Frazão and Zech, 2011) with Godunov-type upwind scheme
- MPM-like and GRASS-like bedload formula
- Boundary conditions: Wall (default), Standard
- Geometrical (default), wetter area or conveyance weighting schemes for inflow boundary conditions 'sedimentograph' and 'transport capacity'.
- Fix bed elevation over regions (index)

4.1.4 System manuals

- Complete new manuals (Introduction and Installation, User manual, Reference Manual, Tutorials and Test cases)
- New logo
- Migration guide from version 2.x to 3.x
- Simulation workflow
- Tutorials: Flaz river for the pre-processing, setup of an hydrodynamic and morphologic simulation and post-processing
- Test cases: Circular dam break (hydraulics) and Conical dune (morphology)

4.2 Version 3.0.1

4.2.1 Bug Fixes

- The system manuals are now installed by the installers on Linux and are available in the graphical user interface (GUI).

- Correction of the structure of the file result.json in section “Tutorials and Test Cases” of the system manuals.

- Some Windows registry settings were put at incorrect locations by the Windows installer. In particular, the list of installed software provided by Windows did not include BASEMENT. This problem has been fixed.

4.3 Version 3.0.2

4.3.1 General

- BASEMENT now natively supports CUDA on Windows. This allows the use of CUDA-enabled GPUs for computation under Windows 10.
- In addition, many third party libraries and the installers have been updated. In particular, the version for Microsoft Windows uses a new installer, therefore we recommend to use the link in the start menu to uninstall previous versions of BASEMENTv3 before you update.

4.3.2 Bug Fixes

- Vector data are now loaded correctly in Paraview Versions 5.7.0 and newer.
- The setup binary does not produce an error anymore during restart simulations if the .h5 file extension is provided in the path for the restart.h5 file. The .h5 extension should explicitly be provided in the file name.
- The number of decimal places for doubles in the GUI has been increased from 3 to 6 digits.
- Minor corrections to the System Manual.

4.3.3 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.
- Currently, there is a known, but not yet resolved issue in the sediment transport solver. More specifically, the flux calculation exhibits an asymmetrical behavior depending on the orientation of the flux vector with respect to the edge normal. Up to now, we have not observed a significant impact of this on the simulation of sediment transport. We will fix this issue as soon as possible and recommend to examine the plausibility of morphodynamic simulation results critically as always.

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.

5

Summary of Features Version 3.0

5.1 Hydrodynamic features

Riemann Solver

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

Hydraulic Initial Conditions

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

Parameters

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

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)

Friction

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

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

Flood

- Flood tracking (tracking time step)

Source (water volume)

- Type: total (as discharge, m3/s), distributed (as rain, mm/h)
- Sink behaviors:
  - Exact (as prescribed)
  - Available (as prescribed or less)
  - Infinity (as much as possible)

5.2 Morphodynamic features

Bedload transport

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

Parameters

- Morphodynamic start time
- Sediment porosity

- Sediment density

Initial conditions

- Mesh file
- Continue

Incipient motion

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

Closure formula

- MPM (coefficien = 8, exponent = 15, critical threshold = 0.047)
- MPM-like (coefficient, exponent, critical threshold are adaptable)
- GRASS-like (coefficient, exponent, critical threshold are adaptable)
- Engelund and Hansen

Direction

- Lateral bed slope effect (Ikeda, 1982)
- Curvature effect

Bedload boundary conditions

- WALL: inviscid, default
- STANDARD:
  - INFLOW: equilibrium (reference_bed_elevation), sedimentograph (sediment discharge), transport capacity (boudary factor)
  - OUTFLOW: equilibrium (reference_bed_elevation)
  - Weighting scheme for transport capacity and sedimentograph: geometrical (default), wetted area, conveyance

Bed material

- Grain class
- Fixed bed

6

References

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.

Ikeda, S. (1982). Lateral Bed-Load Transport on Side Slopes. *Journal of the Hydraulics Division-Asce*, 108(11): 1369–1373.

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.

van Rijn, L.C. (1989). Handbook Sediment Transport by Current and Waves. *Delft Hydraulics Laboratory*, Delft, The Netherlands.

BASIC SIMULATION ENVIRONMENT
FOR SIMULATION OF ENVIRONMENTAL FLOW
AND NATURAL HAZRAD SIMULATION

USER MANUAL

VERSION 3.0
September 2019

BASEMENT

Contents

1 Basic Simulation Environment 3

1.1 Introduction . . . 3
1.2 General Use . . . 3
1.2.1 Problem Description . . . 3
1.2.2 Product Delineation and Employment Domains . . . 4
1.2.2.1 Product Delineation . . . 4
1.2.2.2 Employment Domains . . . 4
1.2.3 Capabilities . . . 4

2 Modelling procedure 7

2.1 General . . . 7
2.2 Preprocessing . . . 8
2.2.1 General . . . 8
2.2.2 Computational mesh . . . 8
2.2.2.1 Concept . . . 8
2.2.2.2 Breaklines . . . 8
2.2.2.3 Boundary conditions . . . 9
2.2.2.4 Mesh quality . . . 9
2.2.3 Hydrological and sediment data . . . 9
2.2.3.1 Hydrology . . . 9
2.2.3.2 Sediment transport . . . 10
2.3 Simulation Workflow . . . 10
2.4 Postprocessing . . . 10
2.4.1 Output files . . . 10
2.4.2 Result visualization . . . 11

3 Grid Generation with BASEmesh 13

3.1 General . . . 13
3.2 Installation . . . 13
3.3 Generation of small meshes . . . 14
3.4 Generation of large meshes . . . 16

4 Simulation workflow 19

4.1 General . . . 19
4.2 Pre-simulation . . . 20
4.2.1 Command files . . . 20
4.2.2 Model setup . . . 20

4.3 Simulation 21
    4.3.1 Command file 21
    4.3.2 Model backend 21
4.4 Post-simulation 21
4.5 Re-run Simulation 22
4.6 Restart Simulation 22

5 Graphical User Interface (GUI) 23

5.1 Graphical user interface 23
    5.1.1 General 23
    5.1.2 First steps 23
        5.1.2.1 Scenario directory 23
        5.1.2.2 Load and save 24
    5.1.3 Setup 24
        5.1.3.1 Adding and deleting items 24
        5.1.3.2 Help and parameter values 26
        5.1.3.3 Run BASEMENT setup 26
    5.1.4 Simulation 26
        5.1.4.1 Selecting the simulation backend 27
        5.1.4.2 Run simulation 27
    5.1.5 Results 27

6 Run the program 29

6.1 Running BASEMENT 29
    6.1.1 Graphical user interface (GUI) 29
    6.1.2 Batch mode under Linux 29
        6.1.2.1 Setup 29
        6.1.2.2 Simulation 30
        6.1.2.3 Results 31
    6.1.3 Batch mode under Windows 32

7 References 35

1

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 continual development of the software system has lead to BASEMENT version 3.0, a newly developed version motivated by an increase of efficiency, while guaranteeing the stability of the numerical models. Compared to the former versions of BASEMENT, version 3.0 has a simpler spatial discretization and improved performance. In addition, the software provides a new simulation workflow and graphic user interface (GUI).

The development process is at an early stage and focuses primarily on efficient two dimensional flows modelling with bedload transport. Further development of the software system BASEMENT is expected in the future with the implementation of a 1D model and the increase of available features and application domains.

1.2 General Use

1.2.1 Problem Description

In connection with watercourses and river areas, increasingly complex problems have to be addressed. The estimation of floods, the more frequent occurrence of restoration projects or the study of naturally shaped watercourses implicate the examination of larger regions - also outside of the actual waterway - and a more manifold shape of the channels. The simple formulas for the calculation of flow behaviour used in the past showed in several cases to be insufficient to obtain the desired information. The extent of the considered areas makes the application of hydraulic models in a laboratory - usually employed for difficult cases - impossible or too expensive. So, the numerical simulation of flow behaviour is in many cases the most obvious solution. However, existing programs have still some weak points. Some are limited in their capabilities (e.g. only steady flow 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.

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

1.2.2.2 Employment Domains

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

- Several problems in relation with the sediment transport of water courses, for instance the future development of deltas and alluvial fans, the long term evolution of the bottom of channels, or the 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 danger maps or of protection and emergency measures, considering the flow behaviour and sediment deposition both inside and outside of the main channel, as well as erosion danger, and consequences of debris flows and dam breaks.

1.2.3 Capabilities

BASEMENT has the following fundamental capabilities:

- Simulation of flow behaviour under steady and unsteady conditions in a channel as well as its transition;

- Simulation of sediment transport (bed load) under steady and unsteady conditions in a channel with arbitrary geometry;
- Simulation of erosion and deposition;

2

Modelling procedure

2.1 General

The modelling procedure involves three stages: the pre-processing, the numerical simulation and the post-processing (Figure 2.1). A numerical project is 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. The scenario directory contains all the files (input files, configuration files, output files,. . .) required to execute a numerical simulation with BASEMENT.

This section will present in details the pre- and post-processing parts, while the numerical simulation will be explained in Section 4.

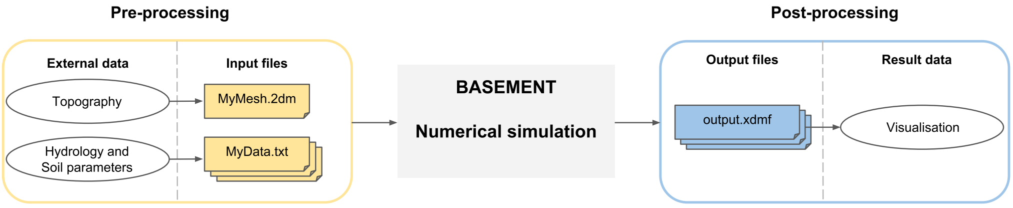


Figure 2.1 *Overview of the modelling procedure with BASEMENT*

2.2 Preprocessing

2.2.1 General

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 characterised 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.

2.2.2 Computational mesh

2.2.2.1 Concept

The numerical methods used in BASEMENT are based on a discretization of the domain topography into unstructured triangular elements. 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. Once the domain has been discretise into unstructured triangular elements, thus forming a quality mesh, the topographical elevation information has to be assigned to the quality mesh in order to generate the final computational mesh. The elevation information is attributed to the element center and is equally distributed over the element area. An appropriate definition of the element center coordinates is crucial for the generation of the computational mesh. It exists different methods to define the interpolation point coordinates of a triangular element, in BASEMENT, the average of the triangle node coordinates is used. The element edges define the boundary of the control volume and the connection between the neighboring elements.

One single computational mesh is used for hydro- and morphodynamic simulations and has to be of type “MyMesh.2dm” (Figure 2.1). 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.

2.2.2.2 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 due to the computational mesh specificity of attributing one elevation information to the cell center. The risk encountered is the loss of 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,...). 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.

2.2.2.3 Boundary conditions

Boundary conditions control the water and sediment flow on the domain. They are defined on the domain as stringdef, i.e. a selected sequence of vertices and element edges of the computational mesh located either at the border or inside the computational mesh. The sequence of vertices along the stringdef gives the stringdef direction with a left and right side. The upstream flow direction is 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.

It exists two types of boundary conditions, the external and internal boundaries. The external boundaries are defined on the domain boundary, while the internal boundary is defined inside the domain. More informations about the type of boundaries and their features can be found in the Reference Manual.

The stringdefs are listed as nodestring at the end of the computational mesh file “MyMesh.2dm”.

Please Note: In BASEMENT version 3.x, the number of nodes per nodestring is limited to 40, i.e. larger nodestrings must be split up.

2.2.2.4 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 Hydrological and sediment data

2.2.3.1 Hydrology

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 2.1). 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.

2.2.3.2 Sediment transport

The river bed is characterized by a porosity and a mean grain size diameter (m) determined from sediment or line samples. In BASEMENT version 3.0, the simulation works only for uniform sediments.

The sediment flow is defined as a specific bedload 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. The reference bed elevation has to be provided at inflow and outflow boundary conditions of type equilibrium.

2.3 Simulation Workflow

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 4.

2.4 Postprocessing

2.4.1 Output files

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 2.1.

Table 2.1 *Output types*

Cell centered	**Nodestring**
Water surface elevation, water depth, bottom elevation, Chézy friction, delta_z, specific discharge, flow velocity, flow curvature, flow radius, flood tracking	Hydraulic and morphodynamic discharge

2.4.2 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 “output.xdmf” (see Figure 2.1) for the cell centered outputs or in a text format (.csv) for the nodestring output. The software ParaView enables to visualize the results stored in “output.xdmf”.

3

Grid Generation with BASEmesh

3.1 General

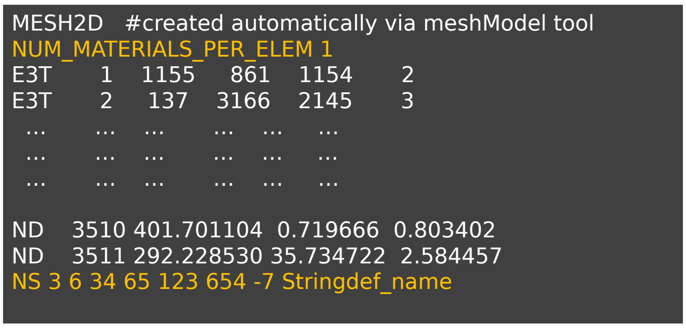
In order to provide a free and open source solution for the creation of computational meshes (Pre-Processing) and to visualize simulation results (Post-Processing) the plugin BASEmesh for the free and open source geographic information system (GIS) software QGIS was developed.

BASEmesh is a QGIS plugin developed to generate triangulated computational grids for BASEMENT based on the advanced mesh generator Triangle by Jonathan R. Shewchuk (Shewchuk, 1996) as meshing algorithm. Currently, there are two versions of BASEmesh available. BASEmesh version 1.4.4 contains all the features including a workflow to generate large meshes for BASEMENT v3.x. This version of BASEmesh is supported up to QGIS version 2.18.28. BASEmesh 1.4.5 contains the same features as version 1.4.4 but is supported and tested for QGIS version 3.10 and newer.

The generation of a computational mesh compatible with BASEMENT version 3.0 is twofold and depends on the domain size and mesh resolution. For small meshes (< 10’000 - 50’000 cells), the procedure to create a computational mesh (2dm) using BASEmesh is the same as for 2D meshes in BASEMENT version 2.8 and the tutorials can be found either in Section 3.3.5 of BASEMENT User Manual version 2.8 or in the Tutorial 1 and 2 of the Pre-Processing section in the Tutorial Manual of version 2.8 available on www.basement.ethz.ch. For larger meshes, the creation of shapefiles during the usual procedure strongly slows down the meshing process. Therefore, the pre-processing for large meshes follows the a new procedure, which avoids the shapefiles generation of of the quality mesh.

3.2 Installation

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



```
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 3.1 *Lines to add manually to the 2dm mesh file (orange)*

To install BASEmesh, follow these steps:

(1) Start QGIS

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

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

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

(5) Enter the repository address: https://people.ee.ethz.ch/~basement/qgis_plugins/qgis_plugins.xml (do not copy paste this address, because it might include line breaks)

(6) Press *OK*

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

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

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

3.3 Generation of small meshes

This section illustrates how a small computational mesh (<10’000-50’000 cells) can be generated for simulations in BASEMENT v3.0. The computational mesh is generated using the QGIS plugin BASEmesh in the exact same way as for BASEMENT version 2.8. The mesh generation process is described in detail in Section 3.3.5 of BASEMENT User Manual version 2.8 or in the Tutorial 1 and 2 of the Pre-Processing section in the Tutorial Manual of version 2.8 available on www.basement.ethz.ch. The computational mesh with the elevation information located at the mesh vertices is exported as a 2dm file and a separate stringdefs list is generated during this process. In order to use the 2dm file for simulations with BASEMENT v3.0, the stringdefs list and name have to be added manually at the end of the computational mesh (2dm) and saved under a new name (Figure 3.1). The modified 2dm file is the one used in simulation with BASEMENT v3.0.

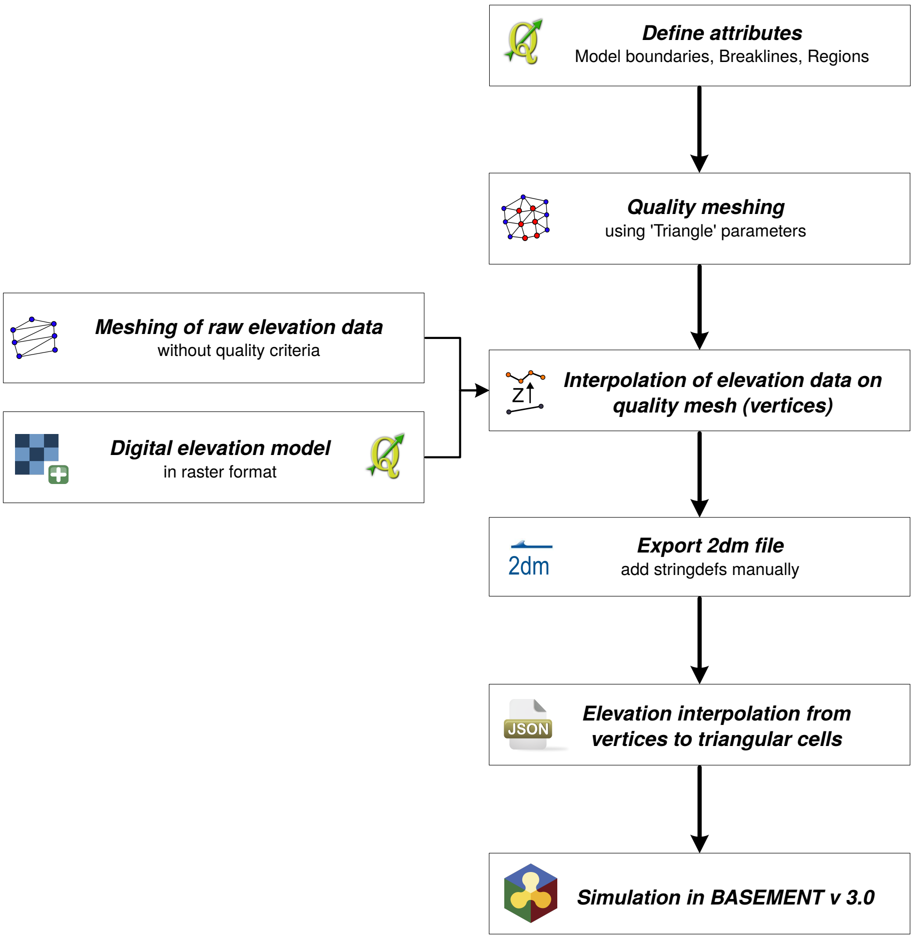
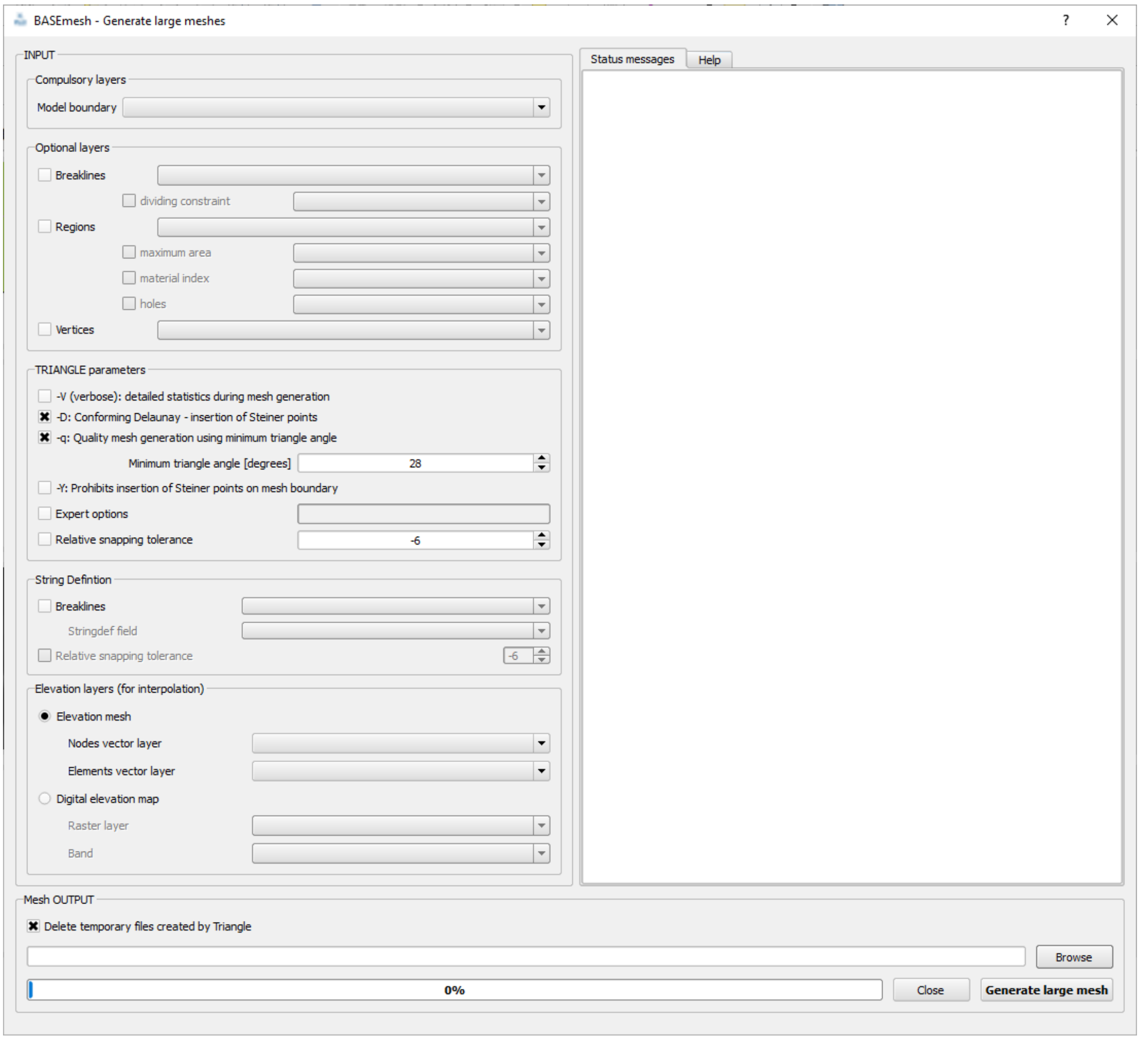


Figure 3.2 *BASEmesh workflow for small meshes and adaptations required in the 2dm and executable (.json) files for the simulation with BASEMENT v3.0*

The elevation information is saved on the mesh vertices and needs to be interpolated to obtain one uniformly distributed elevation information allocated to the corresponding cell. The interpolation method (mean, median, maximum, minimum or weighted) is selected inside the GEOMETRY block of the configuration file model.json.

```
"GEOMETRY": {
    "mesh_file": "Flaz_mesh.2dm",
    "INTERPOLATION": {
        "method": "weighted"
    }
```

More details concerning the elevation interpolation methods are given in the case example of the Migration Guide. Figure 3.2 shows the BASEmesh workflow for small meshes with the transformations required for numerical simulations with BASEMENT v3.0. The procedure for small meshes enables to use existing computational meshes of BASEMENT version 2.x (.2dm) into BASEMENT version 3.x following the two last steps of Figure 3.2.



Model boundary

☐ Breaklines

☐ dividing constraint

☐ Regions

☐ maximum area

☐ material index

☐ holes

☐ Vertices

☐ -V (verbose): detailed statistics during mesh generation

☒ -D: Conforming Delaunay - insertion of Steiner points

☒ -q: Quality mesh generation using minimum triangle angle

Minimum triangle angle [degrees] 28

☐ -Y: Prohibits insertion of Steiner points on mesh boundary

☐ Expert options

☐ Relative snapping tolerance -6

☐ Breaklines

Stringdef field

☐ Relative snapping tolerance -6

● Elevation mesh

Nodes vector layer

Elements vector layer

○ Digital elevation map

Raster layer

Band

☒ Delete temporary files created by Triangle

Figure 3.3 The generation of large meshes can be done with the new procedure in BASEmesh 1.44.

3.4 Generation of large meshes

During the quality meshing process, the BASEmesh plugin generates shape files that are significantly slowing down the procedure in case of large meshes. Therefore, a new procedure was developed to generate large computational meshes, which skips the generation of shape files. This new procedure for large mesh is available in BASEmesh versions 1.4.4 (up to QGIS version 2.18.28) and 1.4.5 (from QGIS version 3.10 and newer) under the button "XL Mesh" and combines the steps of quality meshing, elevation interpolation, nodestring definition and mesh export. The graphical user interface of the "XL Mesh" procedure is illustrated in Figure 3.3. The resulting meshes are directly compatible with BASEMENT version 3.x.

The general procedure to obtain the .2dm mesh file remains mostly the same. Be aware that all the attributes (elevation data, breaklines, points defining the regions, stringdefs,. . .) should be defined before starting the large mesh generation.

The computational mesh is generated with Jonathan Richard Shewchuk's excellent unstructured 2D-mesh generator TRIANGLE (Shewchuk, 1996) and requires the specification of all geometric information about the computational domain:

- *Model boundary*: extent of the computational domain.
- *Breaklines*: distinct interruptions of the surface slope (dyke crest, river side walls, . . .) which shall be preserved in the computational mesh.
- *Regions*: distinct regions of the mesh surrounded by breaklines, which can be characterized by a material index (matID) and maximum triangle area, or can be specified as a hole in the mesh.
- *Holes*: parts within the mesh which are excluded from modelling (e.g. buildings). These parts are defined by special points (layer region_points) surrounded by breaklines.
- *Vertices*: enforced geometric points in the mesh (e.g. measurement points).

Further, the mesh quality can be influenced by the "TRIANGLE parameters". It is important to keep in mind, that the quality of the computational mesh influences the results of your numerical analysis, e.g. stability, computation time, accuracy, etc. Parameters of major importance are:

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

In BASEMENT a list of node IDs is defined as *stringdef* or *nodestring.* They can be defined on the basis of breaklines with a *stringdef* attribute and can be used to define a boundary condition or an output along these nodes. The IDs correspond to the node IDs of the computational mesh. In comparison to the Stringdef tool, the defined stringdefs are added at the end of the .2dm mesh file with the tag "NS" for Nodestring and are **not** written in a separate file. Please Note: In BASEMENT version 3.x, the number of nodes per nodestring is limited to 40, i.e. larger nodestrings must be split up.

Finally, topographical information contained in the elevation model be be interpolated on the computational mesh, i.e. an elevation value is assigned to each node of the computational grid. As a result, the final computational mesh is obtained, which is then exported and can be used for simulations. The elevation data serving as input can be provided in two different elevation model types:

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

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

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

2. *Digital elevation map* as raster data which contains the topography as DTM. The raster elevation data is directly mapped on the computational mesh nodes without interpolation. If no corresponding raster cell is found, the elevation is set to ‘-9999’.

Be aware that the interpolation process can be time consuming.

The mesh is automatically exported in the .2dm format in the specified directory. During the meshing process temporary files are generated. These can be automatically deleted by checking the box “Delete temprary files created by Triangle”.

4

Simulation workflow

4.1 General

The simulation workflow of the software system BASEMENT (light grey rectangular background on Figure 4.1) 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) 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 4.1).

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

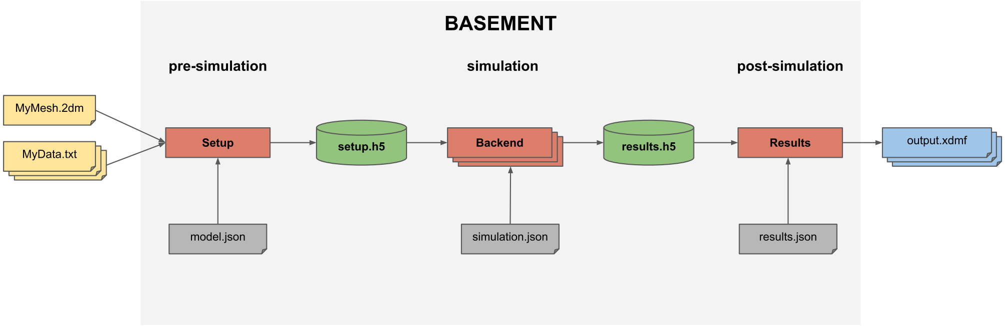


Figure 4.1 *BASEMENT simulation workflow*

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.

4.2 Pre-simulation

4.2.1 Command files

The first command file model.json defines the parameters required to run a numerical simulation on the generated computational mesh. The domain is available for 2D-simulations only and comprises two main components, the geometry and hydraulics blocks. The morphology block is also available and can be added for simulations with bed load transport.

The geometry block gives information on the computational mesh used for the 2D-simulation. The name of the computational mesh and its location in the folder have to be specified. If a computational mesh of BASEMENT version 2.x is used, the elevation interpolation method has to be defined. The stringdefs are listed by their name and the upstream flow direction should be indicated as either left or right (see the Numerical Approximation section in the Reference Manual). The regiondefs are also listed by name with the area index as parameter.

The hydraulics block contains the information about the initial conditions (dry, continue, index), the parameters (CFL, minimum water depth,. . .), the boundary conditions, friction values, external sources and flood tracking. The boundary conditions are defined by giving the corresponding stringdef name and the required type (standard, linked or internal). The friction type is assigned to the different domains (regiondef), as for external sources and flood tracking if required.

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 could be activated in order to influence the bedload transport direction.

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.

4.2.2 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.

4.3 Simulation

4.3.1 Command file

The command file simulation.json contains information about the simulation time, the type of output (see Table 2.1) 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 discharge, calculated at flow boundaries.

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.

4.3.2 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.

4.4 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 4.1). 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 is available for extracting the stringdefs results (discharge) stored in the results.h5 binary and converts them in a text format (.csv).

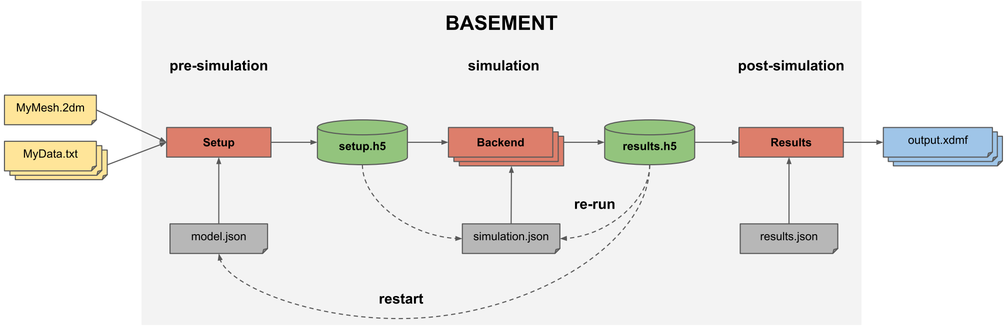


Figure 4.2 *BASEMENT simulation workflow with restart and re-run processes*

4.5 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.

4.6 Restart Simulation

Restarting a simulation (Figure 4.2) 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.

5

Graphical User Interface (GUI)

5.1 Graphical user interface

5.1.1 General

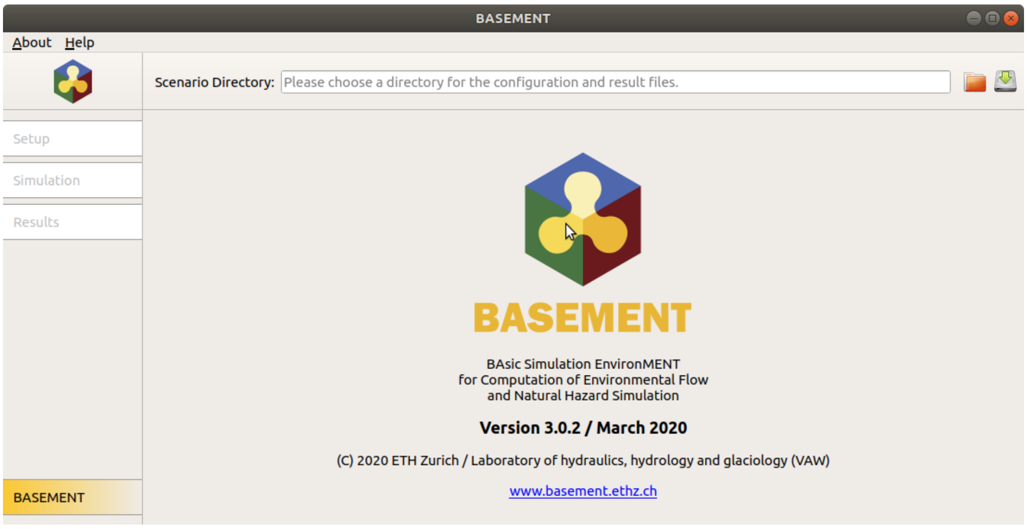
The BASEMENT graphical user interface assists the user with model configuration, numerical simulation and result export. For this purpose, the application provides a convenient way to edit the JSON configuration files and to select and run the backend executables.

5.1.2 First steps

Once started, the BASEMENT user interface application displays the welcome screen (see Figure 5.1). 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.

5.1.2.1 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.



Scenario Directory: Please choose a directory for the configuration and result files.

BASEMENT

BAsic Simulation EnvironMENT
for Computation of Environmental Flow
and Natural Hazard Simulation

Version 3.0.2 / March 2020

(C) 2020 ETH Zurich / Laboratory of hydraulics, hydrology and glaciology (VAW)

www.basement.ethz.ch

Figure 5.1 *Welcome Screen*

5.1.2.2 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 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.

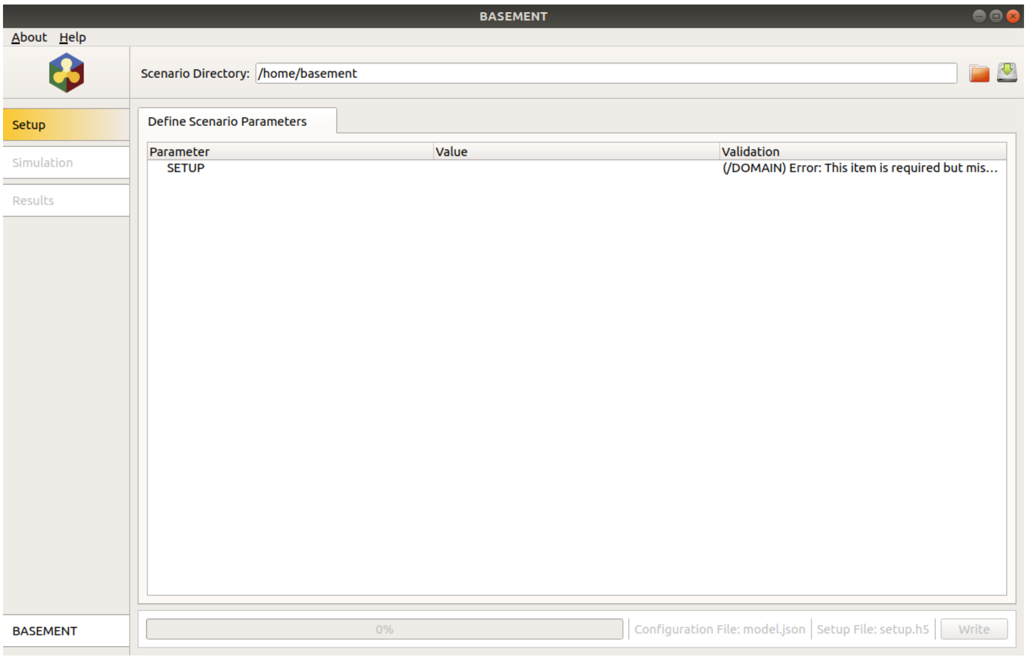
5.1.3 Setup

The setup screen (Figure 5.2) 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.

5.1.3.1 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 5.3. 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).



Scenario Directory: /home/basement

Parameter	Value	Validation
SETUP		(/DOMAIN) Error: This item is required but mis...

Figure 5.2 *Setup Screen*

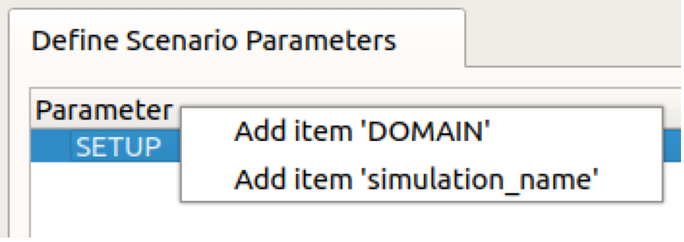
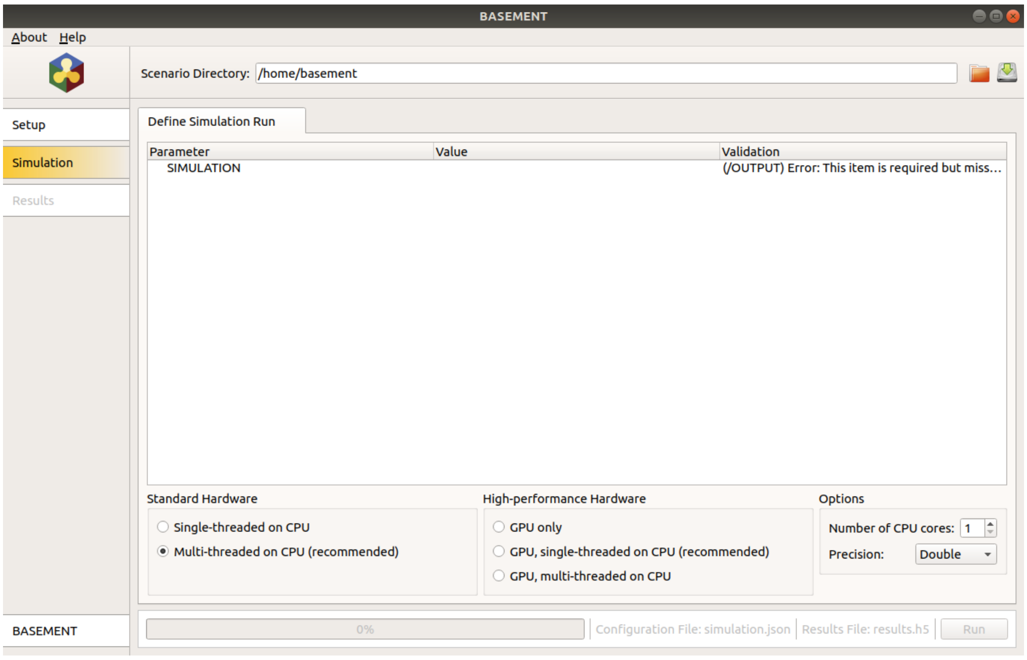


Figure 5.3 *Adding JSON Items*

Figure 5.4 *File Name Editor*



Scenario Directory: /home/basement

SIMULATION — (/OUTPUT) Error: This item is required but miss...

Standard Hardware:
- ○ Single-threaded on CPU
- ● Multi-threaded on CPU (recommended)

High-performance Hardware:
- ○ GPU only
- ○ GPU, single-threaded on CPU (recommended)
- ○ GPU, multi-threaded on CPU

Options:
Number of CPU cores: 1
Precision: Double

Configuration File: simulation.json | Results File: results.h5

Figure 5.5 *Simulation Screen*

5.1.3.2 Help and parameter values

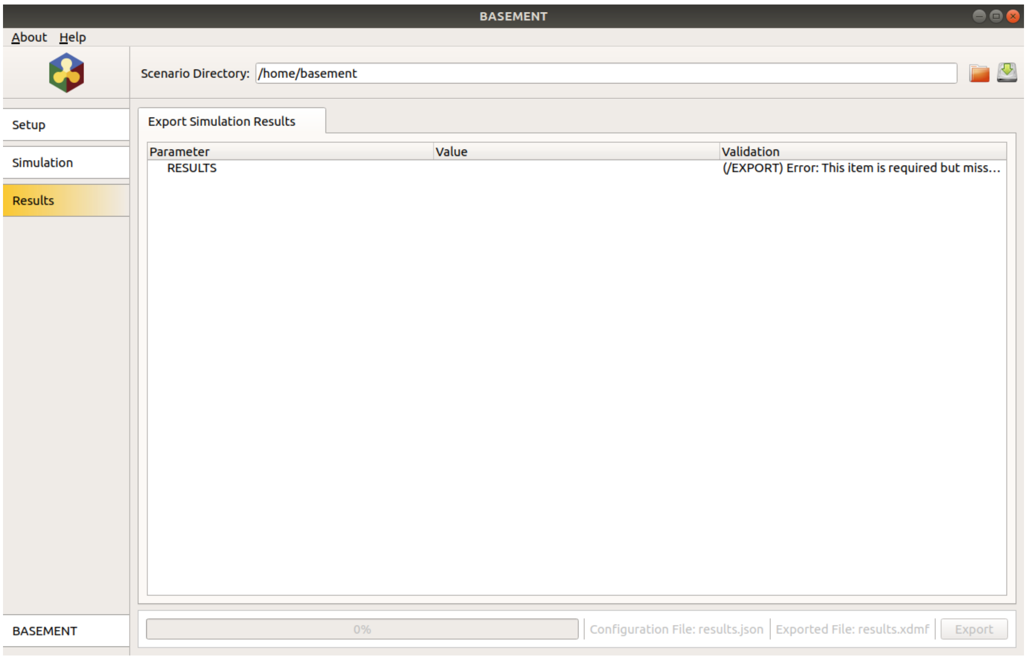
If you want to see the help for a parameter, mouse-over the parameter name and a tooltip with a parameter description appears. Double-clicking a parameter value opens a type-specific editor. In particular, you can click the “Open” icon to select a file for parameters that expect a file name (see Figure 5.4).

5.1.3.3 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.

5.1.4 Simulation

The simulation screen (Figure 5.5) 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.



Scenario Directory: /home/basement

Parameter	Value	Validation
RESULTS		(/EXPORT) Error: This item is required but miss...

Configuration File: results.json

Exported File: results.xdmf

Figure 5.6 Results Screen

5.1.4.1 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.

5.1.4.2 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.

5.1.5 Results

The results tab (Figure 5.6) 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.

6

Run the program

6.1 Running BASEMENT

6.1.1 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 5.1.

6.1.2 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.

6.1.2.1 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:

```
$ ./BMv3_BASEplane_setup -f model.json -o mySim_run.h5
```

The arguments of the setup can be obtained in the command prompt (console) with the help flag '-h'. Table 6.1 shows the setup arguments.

```
$ ./BMv3_BASEplane_setup -h
```

Table 6.1 *Command line flags and arguments for the setup*

Setup flag	Definition and arguments
-h , –help	display help information
-g , –graph	plot the tree as graph
-p , –process	level of processing (int)
-a , –archive	restore the archive (string)
-n , –nthreads	number of threads (int)
-l , –log	level of debug messages (int)
-f , –file	(required) the configuration file name
-o , –output	(required) the output name (.h5)

6.1.2.2 Simulation

The execution of the simulation depends on the backend type. There are five different backend types that can be run with single precision by adding "_single" to the backend name.

Write the following lines to execute the simulation file ("simulation.json") in batch mode using the command line on a sequential backend:

```
$ ./BMv3_BASEplane_seq -f simulation.json -r mySim_run.h5
  -o mySim_run_results.h5 -p
```

And using a single precision:

```
$ ./BMv3_BASEplane_seq_single -f simulation.json -r mySim_run.h5
  -o mySim_run_results.h5 -p
```

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

The available backends are listed below with all having the possiblity of running on single precision:

```
$ ./BMv3_BASEplane_seq

$ ./BMv3_BASEplane_omp

$ ./BMv3_BASEplane_cuda

$ ./BMv3_BASEplane_cudaC

$ ./BMv3_BASEplane_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.

The command line arguments of the simulation are shown in Table 6.2.

Table 6.2 *Command line flags and arguments for the simulation*

Setup flag	Definition and arguments
-h , –help	display help information
-p , –progress	print simulation progress
-r , –runfile	(required) h5 file name with model definition
-a , –archive	restore the archive (string)
-n , –nthreads	number of threads (int)
-l , –log	level of debug messages (int)
-f , –file	(required) the configuration file name
-o , –output	(required) the output name (.h5)

6.1.2.3 Results

The last backend converts the simulation results in output, therefore, the result file ("results.json") is executed as follow:

```
$ ./BMv3_BASEplane_results -f results.json -r mySim_run_results.h5
  -o mySim_output
```

The command line arguments for the output generation are listed in Table 6.3

Table 6.3 *Command line flags and arguments for the results*

Setup flag	Definition and arguments
-r , –results	(required) h5 file name with simulation results
-a , –archive	restore the archive (string)
-n , –nthreads	number of threads (int)
-l , –log	level of debug messages (int)
-f , –file	(required) the configuration file name
-o , –output	(required) the output name

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
./BMv3_BASEplane_setup -f /home/MyUser/Project_1/model.json
      -o /home/MyUser/Project_1/mySim_run.h5

./BMv3_BASEplane_seq -f /home/MyUser/Project_1/simulation.json
      -r /home/MyUser/Project_1/mySim_run.h5
      -o /home/MyUser/Project_1/mySim_run_results.h5

./BMv3_BASEplane_results -f /home/MyUser/Project_1/results.json
      -r /home/MyUser/Project_1/mySim_run_results.h5
      -o /home/MyUser/Project_1/mySim_output

# Project 2
./BMv3_BASEplane_setup -f /home/MyUser/Project_2/model.json
      -o /home/MyUser/Project_2/mySim_run.h5

./BMv3_BASEplane_cuda -f /home/MyUser/Project_2/simulation.json
      -r /home/MyUser/Project_2/mySim_run.h5
      -o /home/MyUser/Project_2/mySim_run_results.h5

./BMv3_BASEplane_results -f /home/MyUser/Project_2/results.json
      -r /home/MyUser/Project_2/mySim_run_results.h5
      -o /home/MyUser/Project_2/mySim_output
```

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
```

6.1.3 Batch mode under Windows

Running BASEMENT 3.x in with a graphical user interface under Microsoft Windows can be done with the same work flow as described in Section 6.1.2. 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 3.x in under the path “C:\Program Files (x86)\BASEMENTv3.0” 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 3.0.2"\bin\BMv3_BASEplane_setup.exe
      -f F:\Project_1\model.json
      -o F:\Project_1\mySim_run.h5

C:\"Program Files\BASEMENT 3.0.2"\bin\BMv3_BASEplane_seq.exe
      -f F:\Project_1\simulation.json
      -r F:\Project_1\mySim_run.h5
      -o F:\Project_1\mySim_run_results.h5 -p

C:\"Program Files\BASEMENT3.0.2"\bin\BMv3_BASEplane_results.exe
      -f F:\Project_1\results.json
      -r F:\Project_1\mySim_run_results.h5
      -o F:\Project_1\mySim_output
```

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 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\BASEMENT3.0.2"\bin\BMv3_BASEplane_setup.exe
      -f F:\Project_1\model.json
      -o F:\Project_1\mySim_run.h5

"C:\"Program Files\BASEMENT3.0.2"\bin\BMv3_BASEplane_seq.exe
      -f F:\Project_1\simulation.json
      -r F:\Project_1\mySim_run.h5
      -o F:\Project_1\mySim_run_results.h5 -p

"C:\"Program Files\BASEMENT3.0.2"\bin\BMv3_BASEplane_results.exe
      -f F:\Project_1\results.json
      -r F:\Project_1\mySim_run_results.h5
      -o F:\Project_1\mySim_output


"C:\"Program Files\BASEMENT3.0.2"\bin\BMv3_BASEplane_setup.exe
      -f F:\Project_2\model.json
      -o F:\Project_2\mySim_run.h5

"C:\"Program Files\BASEMENT3.0.2"\bin\BMv3_BASEplane_omp.exe
      -f F:\Project_2\simulation.json
      -r F:\Project_2\mySim_run.h5
      -o F:\Project_2\mySim_run_results.h5 -p -n 6
```

```
"C:\"Program Files\BASEMENT3.0.2"\bin\BMv3_BASEplane_results.exe
     -f F:\Project_2\results.json
     -r F:\Project_2\mySim_run_results.h5
     -o F:\Project_2\mySim_output
```

Then run the batch file by double clicking on it.

7

References

Shewchuk, J.R. (1996). Triangle: Engineering a 2D Quality Mesh Generator and Delaunay Triangulator. *Applied computational geometry: Towards geometric engineering*, *Lecture notes in computer science*, Lin, M.C. and Manocha, D. eds., No. 1148: 203–222. *Springer-Verlag*,

BASIC SIMULATION ENVIRONMENT
FOR SIMULATION OF ENVIRONMENTAL FLOW
AND NATURAL HAZRAD SIMULATION

REFERENCE MANUAL

VERSION 3.0
September 2019

Contents

1 Mathematical Models 3

1.1 Hydrodynamics . . . 3
  1.1.1 Introduction . . . 3
  1.1.2 Governing Equations . . . 4
  1.1.3 Closure relations . . . 4
    1.1.3.1 Friction terms . . . 4
    1.1.3.2 Lateral inflow/outflow . . . 5
  1.1.4 Boundary Conditions . . . 6
    1.1.4.1 External boundary conditions . . . 7
    1.1.4.2 Internal boundary conditions . . . 9
  1.1.5 Flood tracking . . . 9
1.2 Morphodynamics . . . 10
  1.2.1 Introduction . . . 10
  1.2.2 Governing Equation . . . 10
  1.2.3 Closure relations . . . 10
    1.2.3.1 Bedload sediment transport: Fundamentals . . . 11
    1.2.3.2 Closure relations for Bed Load Transport . . . 13
    1.2.3.3 Correction of Bed Load Direction . . . 14
  1.2.4 Uniform Sediment Transport . . . 16
    1.2.4.1 Governing Equation . . . 16
    1.2.4.2 Closures for Bed Load Transport . . . 16
    1.2.4.3 External Sources Terms . . . 18
  1.2.5 Boundary Conditions . . . 18
    1.2.5.1 Upstream boundary condition . . . 18
    1.2.5.2 Downstream boundary condition . . . 19

2 Numerical Models 21

2.1 General View . . . 21
2.2 Discretization . . . 23
2.3 Numerical solution of Hydrodynamics . . . 23
  2.3.1 Vectorial form of the governing equations . . . 23
  2.3.2 Spatial discretisation . . . 24
  2.3.3 Flux estimation . . . 24
    2.3.3.1 Rotational invariance of the shallow water equations . . . 24
    2.3.3.2 Computation of the flux . . . 25
    2.3.3.3 The HLLC approximated Rieman solver . . . 25
  2.3.4 Numerical stability . . . 26
  2.3.5 Discretisation of Source terms . . . 26

2.3.5.1 Bed slope source term 26
2.3.5.2 Friction source term 26
2.3.5.3 External Source Term 27
2.3.6 Solution procedure 28
2.4 Numerical solution of Morphodynamics 28
2.4.1 Numerical solution of the Exner equation 28
2.4.1.1 Fundamentals 28
2.4.1.2 Spatial discretization 29
2.4.1.3 Discretization of External Source Term 30
2.4.2 Solution procedure 30

3 References **33**

1

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 ρ 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

$$\begin{cases} \frac{\partial h}{\partial t} + \frac{\partial q_x}{\partial x} + \frac{\partial q_y}{\partial y} = S_h \\ \frac{\partial q_x}{\partial t} + \frac{\partial}{\partial x}\left(\frac{q_x^2}{h} + \frac{1}{2}gh^2\right) + \frac{\partial}{\partial y}\left(\frac{q_x q_y}{h}\right) + gh\left(S_{bx} + S_{fx}\right) = 0 \\ \frac{\partial q_y}{\partial t} + \frac{\partial}{\partial x}\left(\frac{q_y q_x}{h}\right) + \frac{\partial}{\partial y}\left(\frac{q_y^2}{h} + \frac{1}{2}gh^2\right) + gh(S_{by} + S_{fy}) = 0 \ , \end{cases} \tag{1.1}$$

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 .

The bed slope source terms

$$S_{bx}, S_{by}$$

are evaluated as follows:

$$S_{bx} = -\frac{\partial z_B}{\partial x} \quad ; \quad S_{by} = -\frac{\partial z_B}{\partial y} \tag{1.2}$$

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 of turbulent flow, hence the friction terms can be assumed proportional to the square of the depth-averaged velocity and can be written as:

Several formulae are available for S_f. All these formulae use hypothesis (H3) of a turbulent flow regime, hence the assumption that the slope of the energy line is proportional to the square of the flow velocity u. The most frequently used laws are

Adopting a quadratic friction law, the friction term is proportional to the square of the depth-averaged velocity and can be written as:

$$S_{fx} = \frac{u|\vec{u}|}{gc_f^2 h} \quad ; \quad S_{fy} = \frac{v|\vec{u}|}{gc_f^2 h} \tag{1.3}$$

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

$$c_f = \frac{k_{str} h^{1/6}}{\sqrt{g}} \tag{1.4}$$

1.1.3.1.2 Logarithmic Law

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

Chézy:

$$\begin{array}{lll} c_f = 5.75 \log\left(12\dfrac{R}{K_s}\right) & \text{for} & R > K_s \\ c_f = 5.75 \log(12) & \text{for} & R < K_s\ , \end{array} \tag{1.5}$$

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).

$$\begin{cases} c_f = 2.5\sqrt{1 - \dfrac{y_R}{h}}\ \ln\left(10.9\dfrac{R}{y_R}\right), & \text{for} \quad \dfrac{h}{y_R} > 2 \\ c_f = 1.25\sqrt{\dfrac{h}{y_R}}\ \ln\left(10.9\dfrac{R}{y_R}\right), & \text{for} \quad 0.5 \le \dfrac{h}{y_R} \le 2 \\ c_f = 1.5, & \text{for} \quad \dfrac{h}{y_R} < 0.5 \end{cases} \tag{1.6}$$

1.1.3.2 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

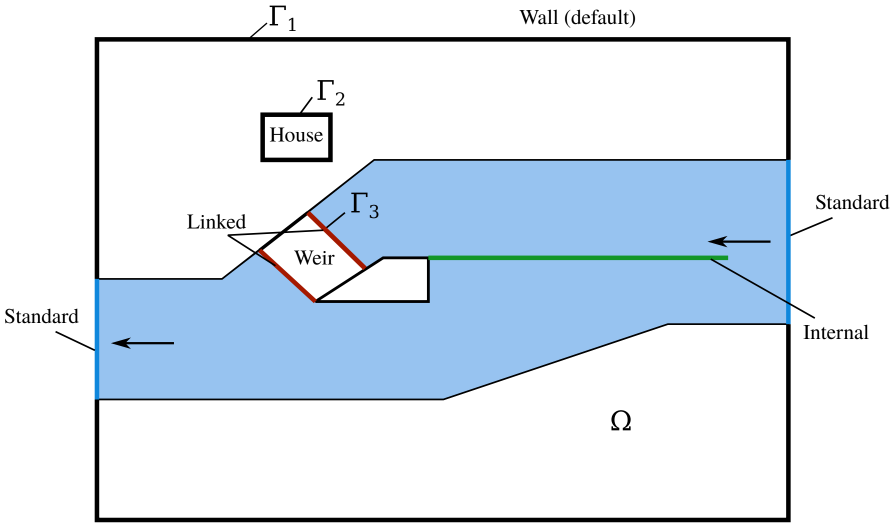


Figure 1.1 *Modeling domain and types of boundary conditions available. The flow is from right to left and a side weir (green line) divides the channel into a lower and an upper channel through the weir.* External *boundary conditions must be provided at* Γ_1, Γ_2 *and* Γ_3 *while* internal *boundary conditions can be specified in any place within* Ω

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 Γ and optionally can be specified within the interior domain Ω.

Therefore, two different types of boundary conditions can be defined:

- External boundary conditions: located at the domain boundary Γ_i
- Internal boundary conditions: located inside the domain Ω

External boundaries (at Γ) 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 External boundary conditions

At the external 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 are set as wall.

1.1.4.1.1 Wall boundary

The *Wall* or *reflective* boundary consider the boundary at Γ_i and suppose it physically consists of a fixed, reflective impermeable wall. Then the physical situation is modelled imposing that:

$$\rho \vec{u} \cdot \vec{n} = 0 \quad ; \quad \frac{\partial \vec{u}}{\partial \vec{n}} = 0 \tag{1.7}$$

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 Γ, 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 Γ 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: h is calculated assuming that local uniform flow conditions. The calculation proceeds as follows:

  $$h = \sqrt[3]{\frac{(Q/b)^2}{g c_f^2 s}} \tag{1.8}$$

  where c_f is the Chézy coefficient, b is the entire length of the boundary Γ and s is the value of the local bed slope that must be specified.

- Explicit: In this case the flow depth h is calculated as follows:

  $$h = \sqrt[3]{\frac{(Q/b)^2}{g Fr^2}} \tag{1.9}$$

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:

$$h = wse - z_B \tag{1.10}$$

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: the water depth h is calculated using equation (eq. 1.8) 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. 1.8).
- Weir: This boundary establishes a relation between the approaching discharge q constant and the water depth using the Poleni weir formula:

$$q = \frac{2}{3}\mu\sqrt{2g(h_{up} - w)^3} \tag{1.11}$$

where h_{up} is the water depth of the approaching flow and w is the weir elevation. The Poleni factor μ can be either set as constant ($\mu = 0.75$ by default) or dynamically evaluated as:

$$\mu = \frac{0.611}{a}\frac{0.75}{b}\frac{h_{up} - z_w}{w} \tag{1.12}$$

where a and b must be specified by the user (default values are $a = 0.611$ and $b = 0.075$).
- h-Q relation: The water surface elevation is determined as a function of the discharge, thus a h-Q relation has to be specified.
- zhydrograph: The water surface elevation (wse) at the boundary must be specified by the user. The depth is calculated as:

$$h = wse - z_B \tag{1.13}$$

where z_B is the bottom elevation at the boundary. The flow velocity is calculated with the Riemann solver (Hllc).
- Zero gradient (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:

$$\rho\vec{u}\cdot\vec{n} = \text{const} \quad ; \quad \frac{\partial\vec{u}}{\partial\vec{n}} = 0 \tag{1.14}$$

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.*

Linked

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 Γ_{in} and Γ_{out}. Then, one inflow boundary condition must be specified at Γ_{in} and one outflow boundary condition at Γ_{out} while in the remaining boundaries wall conditions are automatically assigned. Not necessarily, Γ_{in} and Γ_{out} must have the same number of elements.

Linked boundaries can describe a $h - Q$ relation or a weir, i.e.:

- Weir: Similar to the standard weir boundary, the weir height w has to be specified. No kinetic energy is considered.
- $h - Q$ relation: The flux is calculated given a h-Q relation (see description of the h-Q relation for standard boundaries).
- 2 ways $h-Q$ relation: 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.

1.1.4.2 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: The wall conditions (eq. 1.7) are applied on both sides of the internal boundary.
- Dynamic Wall: The wall conditions are applied on the internal boundary until reaching a threshold value (time or water depth) after which the wall is removed.
- $h - Q$ relation: A $h - Q$ relation is applied on one side of the internal boundary, while on the other side, wall conditions apply (unidirectional flow).

1.1.5 Flood tracking

The flood tracking aims at extracting the flood arrival time, the maximum water depth, flow velocity and specific discharge along the numerical simulation and over a selected domain area. The area is defined by a regiondef and is required to be flooded (wet cells). 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 Governing Equation

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 Exner equation (Exner, 1925)), i.e.:

$$(1-p)\frac{\partial z_B}{\partial t} + \frac{\partial q_{B_x}}{\partial x} + \frac{\partial q_{B_y}}{\partial y} - Sl_b = 0 \tag{1.15}$$

where p is the porosity, Sl_b is the source term specifying local input or output of sediment material (e.g. slope collapse or excavation) per unit width and q_{B_x} and q_{B_y} are the specific bed load flux in x and y direction, respectively. The Exner equation describes the bed evolution due to erosion or deposition processes, which results in changes of the bed level z_B.

1.2.3 Closure relations

In order to solve system (eq. 1.1) and equation (eq. 1.31) 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 discharges.

1.2.3.1 Bedload sediment transport: Fundamentals

The key dimensionless parameter quantifying sediment mobility is the Shields parameter defined as:

$$\theta = \frac{\tau_b}{(\rho_s - \rho) g d} \tag{1.16}$$

where τ_b is the bottom shear stress (drag force acting on the particle), d is the sediment diameter, ρ and ρ_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 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:

$$\tau_{bx} = \rho \frac{|\vec{u}|u}{c_f^2} \quad ; \quad \tau_{by} = \rho \frac{|\vec{u}|v}{c_f^2} \tag{1.17}$$

where τ_b is the bottom shear stress and ρ_s and ρ are the density of sediments and water,respectively.

1.2.3.1.1 Threshold conditions for sediment movement

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 (θ_{cr}) of the Shileds parameter. According to the Shileds' theory Shields (1936), θ_{cr} can be expressed as a function of the Reynolds number $Re^* = \frac{du_*}{\nu}$. Alternatively, the diagram of incipient motion (see Figure 1.2) can be plot 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 approximation of the original Shields diagram was proposed by van Rijn (1984):

$$\begin{array}{lll}
\theta_{cr} = 0.24(D^*)^{-1} & \text{for} & 1 \leq D^* \leq 4 \\
\theta_{cr} = 0.14(D^*)^{-0.64} & \text{for} & 4 < D^* \leq 10 \\
\theta_{cr} = 0.04(D^*)^{-0.1} & \text{for} & 10 < D^* \leq 20 \\
\theta_{cr} = 0.013(D^*)^{0.29} & \text{for} & 20 < D^* \leq 150 \\
\theta_{cr} = 0.055 & \text{for} & D^* > 150
\end{array} \tag{1.18}$$

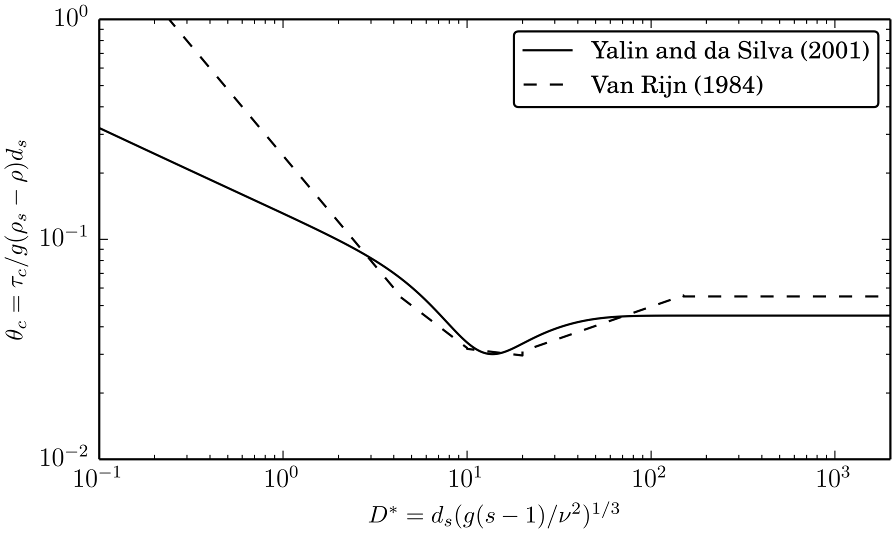


Figure 1.2 *Modified Shields diagram for initiation of sediment motion*

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

$$\theta_{cr} = 0.13 D^{*-0.392} \exp(-0.015 D^*) + 0.045\,(1 - \exp(-0.068 D^*)) \tag{1.19}$$

1.2.3.1.2 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 with correction factors for the local bed slope in the flow direction and transverse to it. The corrected critical bed shear stress becomes:

$$k_\beta k_\delta \theta_{cr} \tag{1.20}$$

The correction factors $k_\beta k_\delta$ are calculated as suggested by van Rijn (1989):

$$k_\beta = \begin{cases} \dfrac{sin(\gamma - \beta)}{sin\gamma} & \text{if slope} < 0 \\ \dfrac{sin(\gamma + \beta)}{sin\gamma} & \text{if slope} > 0 \end{cases} \qquad k_\delta = \left\{ cos\delta \sqrt{1 - \dfrac{tan^2\delta}{tan^2\gamma}} \right. \tag{1.21}$$

where β is the angle between the horizontal and the bed along flow direction, δ is the slope angle transversal to the flow direction and γ is the angle of repose of the sediment material.

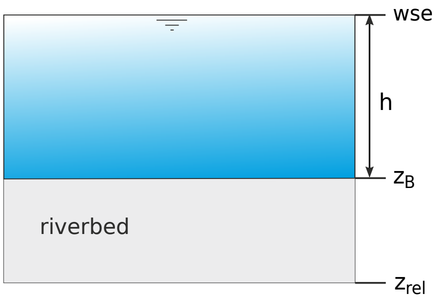


Figure 1.3 *Fixed bed concept and definition*

1.2.3.1.3 Fixed bed concept

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 1.3). The fixed bed elevation is defined relative to the initial bottom elevation z_B with $z_{rel} \leq 0$.

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.

1.2.3.2 Closure relations for Bed Load Transport

Let us now introduce the dimensionless bed load transport rate Φ also known as the Einstein bed load number, first introduced by Hans Albert Einstein in 1950, and given by

$$\Phi = \frac{q_B}{\sqrt{(s-1)gd^3}} \tag{1.22}$$

where $s = \rho_s/\rho$.

It is common practice to quantify bedload transport empirically relating Φ with either the Shields stress θ or the excess of the Shields stress θ above some appropriately defined “critical” Shields stress $(\theta - \theta_{cr}$. θ_{cr} is defined so as to fit experimenatl or field data and provide a threshold for which the bedload transport rate is too low to be of interest.

In what follows we describe the bedload transport formulas that are implemented to calculate the transport capacity $q_B = |\vec{q_B}|$ where $\vec{q_B} = (q_{B_x}, q_{B_y})$. The Shields parameter, takes the following form:

$$\theta = \frac{h\sqrt{S_{fx}^2 + S_{fy}^2}}{(s-1)} \tag{1.23}$$

and the specific bed load flux has the same direction as the water flow.

1.2.3.2.1 Meyer-Peter and Müller (MPM)

The bed load transport formula of Meyer-Peter and Müller (Meyer-Peter and Müller, 1948) reads as:

$$\Phi_B = \alpha(\theta - \theta_{cr})^m \tag{1.24}$$

Herein, α denotes the bed load coefficient, m the bed load exponent. In the original form of the formula $\alpha = 8$ and $m = 3/2$.

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 where 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 bed load coefficient α, the exponent m and the critical Shields parameter θ_{cr} can be adapted by the user in the MPM-like formula.

1.2.3.2.2 Grass formula

The Grass formula (Grass, 1981) does not require the evaluation of the Shields stress:

$$\Phi_B = \alpha(\theta - \theta_{cr})^m \tag{1.25}$$

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 and m being chosen in the range $[1 - 4]$. The two-dimensional projection of (eq. 1.33) is obtained as follows:

$$q_{Bx} = \alpha \frac{q_x |\vec{q}|^{m-1}}{h^m} \quad , \quad q_{By} = \alpha \frac{q_y |\vec{q}|^{m-1}}{h^m} \ .$$

The coefficient α characterizes the interaction between the sediment and the fluid phase. The smallest α the weaker the interaction.

1.2.3.2.3 Engelund and Hansen

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

$$\Phi_B = 0.05\sqrt{(s-1)g}\, c_f^2 \tag{1.26}$$

This formula formula does not consider the critical shear stress as threshold condition for incipient motion.

1.2.3.3 Correction of Bed Load Direction

The 2D projection of the solid discharge along x and y is obtained through standard procedures, that iare 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 riches.

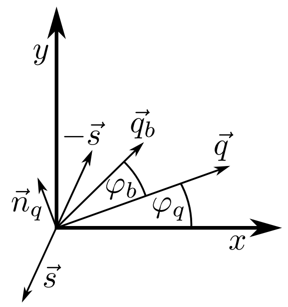


Figure 1.4 *Bed load transport deviation angle φ_b from the flow direction $\vec{q}$ due to the lateral bed slope $\vec{s}$ (Vonwiller, 2017)*

1.2.3.3.1 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, the bed load direction differs from the flow direction due to gravity acting on the bed material. Figure 1.4 illustrates the deviation of the bed load transport direction due to lateral bed slope in a Cartesian coordinate system.

The bed load direction is corrected for lateral bed slope based on the following approach (e.g. see Ikeda (1982) and Talmon et al. (1995)):

$$\tan\varphi_b = \left(\frac{-r}{\theta}\right)\vec{s}\cdot\ \vec{n}_q \quad \text{for} \quad \vec{s}\cdot\vec{n}_q < 0 \tag{1.27}$$

$$r = N_l\theta_{cr}^{1/2} \tag{1.28}$$

where φ_b = bed load direction with respect to the flow vector $\vec{q}$, N_l = lateral transport factor ($0.75 \le N_l \le 2.63$), $\vec{s} = \left(\frac{\partial z_B}{\partial x}, \frac{\partial z_B}{\partial y}\right)$ bed slope (positive uphill, negative downhill), $\vec{n}_q$ = unit vector perpendicular to $\vec{q}$ pointing in downhill direction ($\vec{s}\cdot\vec{n}_q < 0$) , θ = effective dimensionless shear stress and θ_{cr} = critical dimensionless shear stress of sediment.

The direction of the bed load transport under the influence of lateral bed slope is written as:

$$\frac{q_{B_y}}{q_{B_x}} = \tan(\varphi_b + \varphi_q) \tag{1.29}$$

1.2.3.3.2 Curvature Effect

Curvature in rivers may cause deviation of the bed load direction from the depth averaged flow direction. Due to three dimensional spiral flow motion, the bed load direction tends to 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 φ_c of the bottom shear stress $\vec{\tau}_b$ (positive counterclockwise and vice versa) from the main flow direction is determined as

$$\tan\varphi_c = \frac{|\vec{\tau}_{bn}|}{|\vec{\tau}_{bs}|} = -N_*\frac{h}{R} \tag{1.30}$$

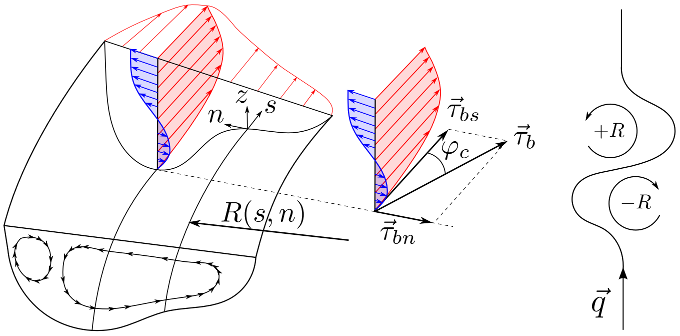


Figure 1.5 *Effect of spiral motion in river bend on bed shear stress* $\vec{\tau}_b$ *with deviation angle from main flow direction* φ_c *(Vonwiller, 2017)*

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).

1.2.4 Uniform Sediment Transport

1.2.4.1 Governing Equation

The conservation of bed material is ensured by the Exner equation (eq. 1.31), 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:

$$(1-p)\frac{\partial z_B}{\partial t} + \frac{\partial q_{B_x}}{\partial x} + \frac{\partial q_{B_y}}{\partial y} - Sl_b = 0 \tag{1.31}$$

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 bed load flux.

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 bed load flux is much slower than the water flow velocity (Soares-Frazão and Zech, 2011).

1.2.4.2 Closures for Bed Load Transport

The following section describes the bedload transport formulas that are implemented to calculate the transport capacity $q_b = |\vec{q}_b|$. The specific bed load flux has the same direction as the water flow. For practical purposes, the bed load transport formula can be calibrated by a pre-factor.

1.2.4.2.1 Meyer-Peter and Müller (MPM)

The bed load transport formula of Meyer-Peter and Müller (Meyer-Peter and Müller, 1948) is written as follows:

$$q_B = \alpha(\theta - \theta_{cr})^m \sqrt{(s-1)gd^3} \tag{1.32}$$

Herein, α denotes the bed load coefficient (originally $\alpha = 8$), m the bed load exponent (originally $m = 1.5$), q_B is the specific bed load transport rate, θ is the dimensionless bed shear stress (Shields parameter), θ_{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 where 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 bed load coefficient α, the exponent m and the critical Shields parameter θ_{cr} can be adapted by the user in the MPM-like formula.

1.2.4.2.2 Grass formula

The Grass model (Grass, 1981) proposes a simple bedload transport formula, where q_b is a function of the flow velocity and a dimensional constant α.

$$q_B = \alpha(u - u_c)^m \tag{1.33}$$

With u_c the critical velocity. The exponent m is usually set to m = 3. The threshold condition for incipient motion of grains is set to zero, meaning that the bedload transport and the fluid motion start simultaneously. The coefficient α 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.

1.2.4.2.3 Engelund and Hansen

Engelund and Hansen (1972) proposed a bedload transport formula for uniform bed material:

$$q_B = 0.05\sqrt{(s-1)g}\, c_f^2 \theta^{2.5} d^{1.5} \tag{1.34}$$

where d denotes the median sediment size of the bed material and θ the Shields parameter. The Engelund and Hansen formula for bed load transport does not consider the critical shear stress as threshold condition for incipient motion.

1.2.4.3 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.

1.2.5 Boundary Conditions

After the specification of the *closure relations* for the sediment transport, the system of governing equations (eq. 1.1) and (eq. 1.31) can be solved within the modeling domain described in Figure 1.1, provided boundary conditions (morphologic boundary conditions) are specified at the domain boundary Γ. 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 ‘sit’ on 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.

1.2.5.1 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 as function of time (constant or variable). The bed load is defined as a volumetric flow rate $Q_b = \frac{\mu_s}{\rho_s}$ $[m^3/s]$, where μ_s is the sediment mass flow rate $[kg/s]$ and ρ_s the sediment density $[kg/m^3]$. Notice that the porosity is not considered in the bed load input and is specified separately as own parameter value.
- Transport capacity: the sediment inflow is defined by calculating the equilibrium transport capacity according to the hydraulic state at the boundary. The bed load is defined as a compact volumetric flow rate (without porosity) Q_b $[m^3/s]$.
- Equilibrium: this upstream boundary condition called IOup grants a constant bed load inflow. 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.

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.

$$q_b = \frac{Q_b}{L_n} \qquad \left[\frac{m^3}{s \cdot m}\right] \tag{1.35}$$

2. Wetted area weighting

$$q_b = \frac{Q_b}{A_{w,tot}} \cdot h \qquad \left[\frac{m^3}{s \cdot m^2}\right] \tag{1.36}$$

3. Conveyance weighting

$$q_b = \frac{Q_b}{K_{tot}} h\sqrt{c_f h} \qquad \left[\frac{m^3}{s \cdot m}\right] \tag{1.37}$$

with $K_{tot} = A_{w,tot}\sqrt{c_f h}$ the total conveyance and c_f the friction coefficient.

1.2.5.2 Downstream boundary condition

Two types of downstream boundary condition are available:

- Equilibrium: all sediment entering the last computational cell will leave the cell over the downstream boundary.
- Check-dam: the equilibrium downstream boundary condition is activated only if the bed level reaches a threshold value. Before reaching the threshold value, a wall type boundary is assumed.

2

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 (Figure 2.1) consisting of elements (often having triangular shape) 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.

The conservation equations can be formulated either in integral or differential form. Different numerical schemes exist to discretise the equations:

- Finite difference: The discrete values are considered as point values defined at mesh points
- Finite element: The discrete values are determined in terms of the nodal values of the mesh
- Finite volume: The discrete values are averaged over finites volumes of the mesh

In BASEMENT, the spatial discretisation of the domain is based on an unstructured mesh made of triangular elements. 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 numerical model processes the hydro- and morphodynamic equations 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.

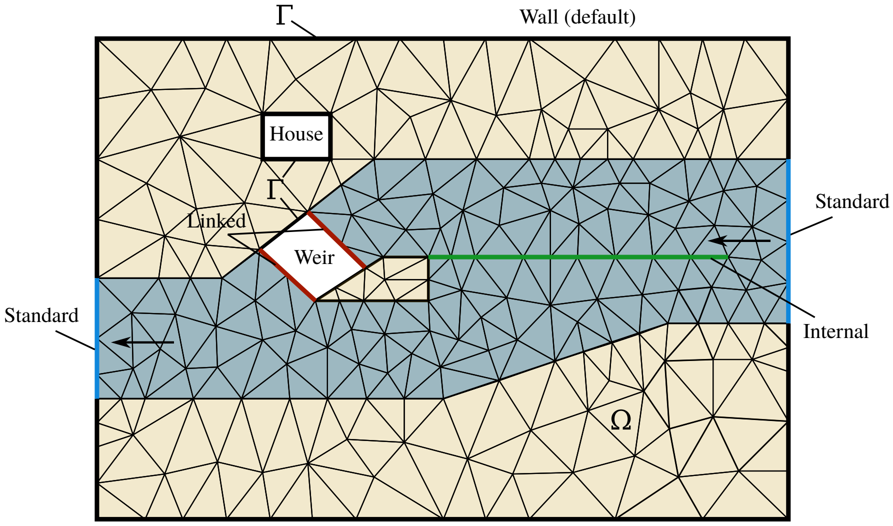


Figure 2.1 *Modeling domain, types of boundary conditions and computational mesh. The flow is from right to left and a side weir (green line) divides the channel into a lower and an upper channel through the weir.* External *boundary conditions must be provided at* Γ_1, Γ_2 *and* Γ_3 *while* internal *boundary conditions can be specified in any place within* Ω

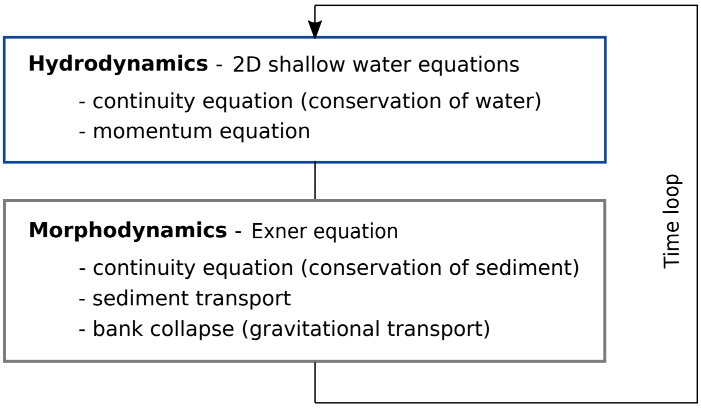


Figure 2.2 *Overview of the numerical model*

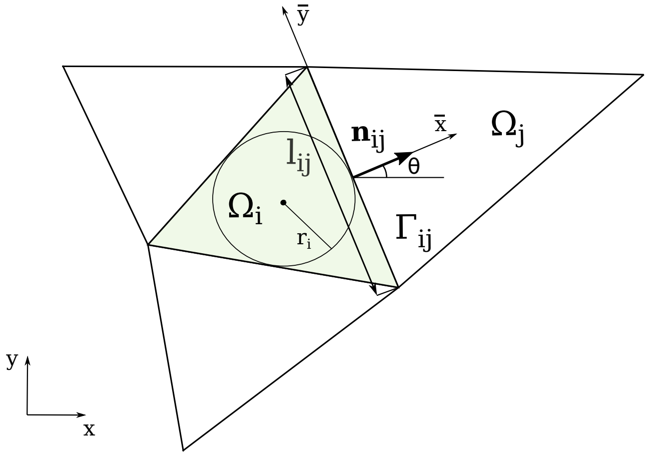


Figure 2.3 *Element (shaded triangle) of unstructured triangular mesh and used notation*

2.2 Discretization

The problem is discretised adopting a finite volume approach over unstructured triangular meshes. A conforming triangulation T_Ω of the computational domain $\Omega \subset \mathbb{R}^2$ by elements Ω_i such that $T_\Omega = \bigcup \Omega_i$, is assumed. Hereafter we will use the following notation: given a finite volume Ω_i, $j = 1, 2, 3$ is the set of indexes such that Ω_j is a neighbour of Ω_i; Γ_{ij} is the common edge of two neighbour cells Ω_i and Ω_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 Γ_{ij} and points toward the cell Ω_j (see Figure 2.3). Data are represented by cell averages $\boldsymbol{U}_i^n$ and the numerical solution sought at time $t^{n+1} = t^n + \Delta t$, is denoted by $\boldsymbol{U}_i^{n+1}$.

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:

$$\frac{\partial \boldsymbol{U}}{\partial t} + \frac{\partial \boldsymbol{F}_x}{\partial x} + \frac{\partial \boldsymbol{F}_y}{\partial y} = \boldsymbol{S} \tag{2.1}$$

where the vector of unknowns is

$$\boldsymbol{U} = \begin{pmatrix} H \\ q_x \\ q_y \end{pmatrix} \tag{2.2}$$

the vector fluxes are

$$\boldsymbol{F}_x = \begin{pmatrix} q_x \\ uq_x + \frac{1}{2}g(H^2 - 2Hz_b) \\ uq_y \end{pmatrix} \quad ; \quad \boldsymbol{F}_y = \begin{pmatrix} q_y \\ vq_x \\ vq_y + \frac{1}{2}g(H^2 - 2Hz_b) \end{pmatrix} \tag{2.3}$$

and the vector of source terms is

$$\boldsymbol{S} = \begin{pmatrix} S_h \\ gHS_x \\ gHS_y \end{pmatrix} \ . \tag{2.4}$$

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

$$\boldsymbol{U}_i^{n+1} = \boldsymbol{U}_i^n - \frac{\Delta t}{|\Omega_i|} \sum_{j=1}^{3} l_{ij} \left[\boldsymbol{F}_{ij}\right] + \Delta t \boldsymbol{S}_i \ . \tag{2.5}$$

2.3.3 Flux estimation

2.3.3.1 Rotational invariance of the shallow water equations

The flux $\boldsymbol{F}_{ij}$ 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):

$$\boldsymbol{n}_{ij} \cdot [\boldsymbol{F}_x(\boldsymbol{U}), \boldsymbol{F}_y(\boldsymbol{U})] = \boldsymbol{T}^{-1}(\theta)\boldsymbol{F}_x[\boldsymbol{T}(\theta)\boldsymbol{U}] \tag{2.6}$$

where θ is the angle between the vector $\boldsymbol{n}_{ij}$ and x-axis, measured counter clockwise from the x-axis (see Figure 2.3) and

$$\boldsymbol{T}(\theta) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & \sin\theta \\ 0 & -\sin\theta & \cos\theta \end{pmatrix} \tag{2.7}$$

being

$\boldsymbol{T}^{-1}(\theta) =$ inverse of $\boldsymbol{T}(\theta)$.

2.3.3.2 Computation of the flux

The flux $\boldsymbol{F}_{ij}$ 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 $\boldsymbol{U}$ is transformed into the local coordinate system ($\bar{x}$, $\bar{y}$) (see Figure 2.3) at the edge with the operation $\boldsymbol{T}(\theta)\boldsymbol{U}$.
- A one-dimensional, local Riemann problem is formulated and solved in the normal direction of the edge. From this calculation the new flux vector over the edge $\boldsymbol{F}[\boldsymbol{T}(\theta)\boldsymbol{U}]$ is defined.
- The flux vector, formulated in the local coordinate system is transformed back to the global coordinates (Cartesian) with $\boldsymbol{T}^{-1}\boldsymbol{F}[\boldsymbol{T}(\theta)\boldsymbol{U}]$. The sum of the fluxes of all edges of an element gives the total fluxes in 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 Rieman 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:

$$\boldsymbol{F}_{ij}^{HLLC} = \begin{cases} \boldsymbol{F}_i & if \quad 0 \leq S_i, \\ \boldsymbol{F}_{*i} = \boldsymbol{F}_i + S_i(\boldsymbol{U}_{*L} - \boldsymbol{U}_i) & if \quad S_i \leq 0 \leq S_*, \\ \boldsymbol{F}_{*j} = \boldsymbol{F}_j + S_j(\boldsymbol{U}_{*R} - \boldsymbol{U}_j) & if \quad S_* \leq 0 \leq S_j, \\ \boldsymbol{F}_j & if \quad 0 \geq S_j. \end{cases} \tag{2.8}$$

The wave speed velocities are estimated as:

$$S_i = u_i - \sqrt{gh_i}\xi_i \ ; \ S_j = u_j + \sqrt{gh_j}\xi_j \tag{2.9}$$

where $\xi_{K=(i,j)}$ is defined as:

$$\xi_K = \begin{cases} \sqrt{\frac{1}{2}\left[\frac{(h_*+h_K)h_*}{h_K^2}\right]} & if \quad h_* > h_K, \\ 1 & if \quad h_* \leq h_K. \end{cases} \tag{2.10}$$

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

$$h_* = \frac{1}{2}(h_L + h_R) - \frac{1}{4}(u_R - u_L)(h_L - h_R)/(\sqrt{gh_L} + \sqrt{gh_R}) \tag{2.11}$$

In case of dry-bed conditions, the wave speeds are estimated as the exact dry front speed, i.e.:

$$S_i = \begin{cases} u_i - 2\sqrt{gh_i} & if \quad h_i = 0, \\ \text{usual estimate} & if \quad h_i > 0, \end{cases}$$
$$S_j = \begin{cases} u_j + 2\sqrt{gh_j} & if \quad h_j = 0, \\ \text{usual estimate} & if \quad h_j > 0. \end{cases} \tag{2.12}$$

And the middle estimated wave speed S_* corresponds to the front wave speed in case of dry-bed problem.

The expression of the states $\boldsymbol{U}_{*i}, \boldsymbol{U}_{*j}$ 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 Δt 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:

$$CFL = \frac{(\sqrt{u^2+v^2}+c)\Delta t}{r_i} \tag{2.13}$$

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

$$0 < CFL \leq 1 \tag{2.14}$$

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:

$$\boldsymbol{S}_{b,i} = \sum_{j=1}^{m} l_{ij}\boldsymbol{S}_{b,ij} = \sum_{j=1}^{m} l_{ij} \begin{pmatrix} 0 \\ gH^*_{ij}(z_i - \bar{z}_{ij})\vec{\boldsymbol{n}}_{ij} \end{pmatrix} \tag{2.15}$$

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 Γ_{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)

$$\left.\begin{array}{llll} PDE: & \mathcal{A}(\boldsymbol{U}) & = & \mathcal{S}(\boldsymbol{U}) \\ IC: & \boldsymbol{U}(x,y,0) & = & \boldsymbol{U}_i^n \end{array}\right\} \text{IVP} \; .$$

where $\mathcal{A}$ represents the advective operator

$$\mathcal{A}(\boldsymbol{U}) = \frac{\partial \boldsymbol{U}}{\partial t} + \frac{\partial \boldsymbol{F}_x}{\partial x} + \frac{\partial \boldsymbol{F}_y}{\partial y} = \mathbf{0} \; ,$$

and $\mathcal{S}$ represents the frictional source term operator.

The numerical solution is then obtained by subsequently integrating *two* initial value problems (IVPs):

$$\left.\begin{array}{llll} ODEs: & \frac{\mathrm{d}\boldsymbol{U}}{\mathrm{d}t} & = & \mathcal{S}(\mathbf{U}) \\ ICs: & \boldsymbol{U}(x,y,0) & = & \boldsymbol{U}_i^n \end{array}\right\} \overset{\Delta t}{\Longrightarrow} \overline{\boldsymbol{U}}_i \quad \text{IVP1} \; ,$$

$$\left.\begin{array}{llll} PDEs: & \mathcal{A}(\mathbf{U}) & = & 0 \\ ICs: & \boldsymbol{U}(x,y,0) & = & \overline{\boldsymbol{U}}_i \end{array}\right\} \overset{\Delta t}{\Longrightarrow} \boldsymbol{U}_i^{n+1} \quad \text{IVP2} \; ,$$

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 Δt and is denoted by $\overline{\boldsymbol{U}}_i$. IVP2 is then integrated by a time step Δt, with initial condition given by the solution of IVP1 $\overline{\boldsymbol{U}}_i$. The solution of IVP2 $\boldsymbol{U}_i^{n+1}$ 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 Δt 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 δ (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 δ_i. The volume allocated is characterized by different behaviors:

$$\begin{array}{llll} \text{Exact:} & S_{h,i} = \delta_i & & \\ \text{Available:} & S_{h,i} = \delta_i & \text{if} & \delta_i \cdot \Delta t > 0 \\ & S_{h,i} = \max(\delta_i, -h_i) & \text{if} & \delta_i \cdot \Delta t < 0 \\ \text{Infinity:} & S_{h,i} = \delta_i & \text{if} & \delta_i \cdot \Delta t > 0 \\ & S_{h,i} = -h_i & \text{if} & \delta_i \cdot \Delta t < 0 \end{array} \tag{2.16}$$

Where h_i is the water depth of the element i. The external source volume is added to the initial water volume.

$$h_i^{t+1} = h_i^t + S_{h,i} \cdot \Delta t \tag{2.17}$$

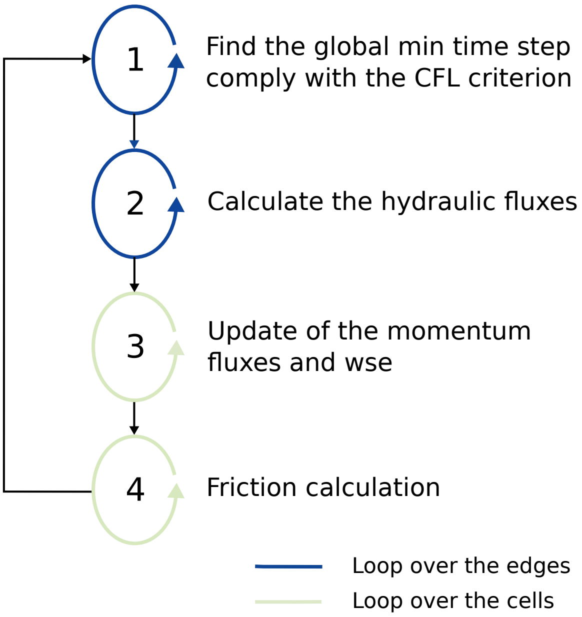


Figure 2.4 *Numerical solution procedure of hydrodynamic simulation for each time step* Δt

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 Δt through a sequence of loops over the edges or cells (Figure 2.4).

First, a global minimum time step Δt should be defined. Then, the hydraulic fluxes (liquid mass, x-momentun 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 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).

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

$$z_{Bi}^{n+1} = z_{Bi}^{n} - \frac{\Delta t}{|\Omega_i|} \sum_{j=1}^{3} \left[q_{Bij \cdot l_{ij}} \right] + \Delta t \boldsymbol{S}_i \ . \tag{2.18}$$

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 1.2.3.3.1)
    - calculation of the local curvature of the flow field, for the spiral flow correction (see section Section 1.2.3.3.2)

2. loop over the cell interfaces and:

    1. calculate the flux projection along the normal vector ($n_{ij,x}$,$n_{ij,y}$ of edge Γ_{ij}, i.e.: $q_{Bi,n} = q_{Bi,x} \cdot n_{ij,x} + q_{Bi,y} \cdot n_{ij,x}$ and $q_{Bj,n} = q_{Bj,x} \cdot n_{ij,x} + q_{Bj,y} \cdot n_{ij,x}$ 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:

$$\lambda_1 = 1/2(u_n - c - \sqrt{(u_n - c)^2 + 4a_2c^2}) \tag{2.19}$$

$$\lambda_2 = 1/2(u_n - c + \sqrt{(u_n - c)^2 + 4a_2c^2}) \tag{2.20}$$

where $u_n = u \cdot n_{ij,x} + v \cdot n_{ij,y}$, $c = \sqrt{gh}$ and $a_2 = \dfrac{\partial q_{b,n}}{\partial q_n}$ 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

$$S^- = \min(\lambda_{1,L}, \lambda_{1,R}) \tag{2.21}$$

and

$$S^+ = \max(\lambda_{2,L}, \lambda_{2,R}) \tag{2.22}$$

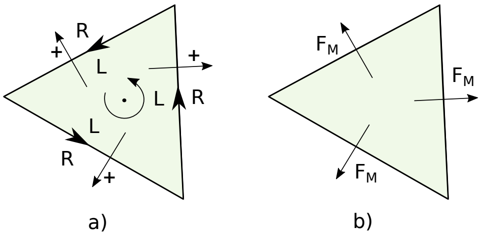


Figure 2.5 *a) Sign convention for the edge direction: counterclockwise b) Positive morphological flux direction at edges: from left (L) to right (R)*

- Flux calculation:

$$q_{Bij}^{HLL} = \begin{cases} q_{Bi,n} & if \quad S^- \geq 0, \\ \dfrac{q_{Bi,n} S^+ - q_{Bj,n} S^- + S^- S^+ \left(z_{Bj} - z_{Bi} \right)}{S^+ - S^-} & if \quad S^- < 0 < S^+, \\ q_{Bj,n} & if \quad S^+ \leq 0. \end{cases}$$

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 Sl_b describes a local input or removal of sediment mass into a river.

An external source is defined as specific mass flux δ (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 δ_i. The volume allocated is characterized by different behaviors:

$$\begin{array}{lll} \text{Exact:} & S_{b,i} = \delta_i & \\ \text{Available:} & S_{b,i} = \delta_i & \text{if} \quad \delta_i \cdot \Delta t > 0 \\ & S_{b,i} = \max(\delta_i, -(z_{Fix} - z_i) & \text{if} \quad \delta_i \cdot \Delta t < 0 \\ \text{Infinity:} & S_{b,i} = \delta_i & \text{if} \quad \delta_i \cdot \Delta t > 0 \\ & S_{b,i} = -(z_{Fix} - z_i) & \text{if} \quad \delta_i \cdot \Delta t < 0 \end{array} \tag{2.23}$$

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.

$$z_i^{t+1} = z_i^t + S_{b,i} \cdot \Delta t \tag{2.24}$$

2.4.2 Solution procedure

The numerical solution procedure of BASEMENT explains how the discretised Exner equation (eq. 1.31) 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

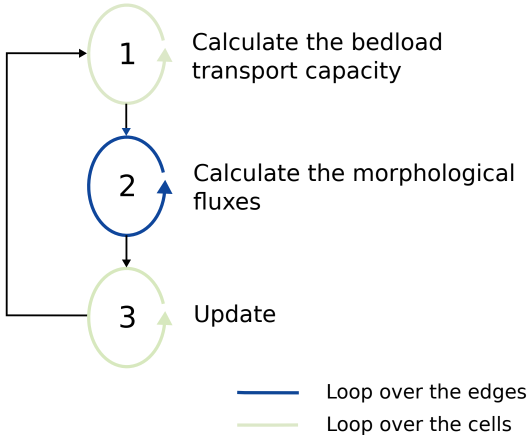


Figure 2.6 *Numerical solution procedure of morphodynamic simulation for each time step* Δt

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 Δt.

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.

3

References

Bezzola, G.R. (2002). Fliesswiederstand und Sohlenstabilität natürlicher Gerinne [PhD thesis]: Eidgenössische Technische Hochschule Zürich.

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. (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.

Ikeda, S. (1982). Lateral Bed-Load Transport on Side Slopes. *Journal of the Hydraulics Division-Asce*, 108(11): 1369–1373.

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.

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.

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., Struiksma, N. and van Mierlo, M. (1995). Laboratory measurements of the direction of sediment transport on transverse alluvial-bed slopes. *Journal of Hydraulic Research*, 33(4): 495–517.

Toro, S., E. F. (1994). Restoration of the contact surface in the HLL-Riemann solver. *Shock Waves*, 4: 25–34.

Toro, E.F. (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 I: Bed Load Transport. *Journal of Hydraulic Engineering, ASCE*, 110(10): 1431–1456.

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.

Yalin, M. and Silva, A. da (2001). Fluvial processes. *International Association of Hydraulic Engineering; Research (IAHR), Delft, The Netherlands*,

BASIC SIMULATION ENVIRONMENT
FOR SIMULATION OF ENVIRONMENTAL FLOW
AND NATURAL HAZRAD SIMULATION

TUTORIALS & TEST CASES

VERSION 3.0
September 2019

BASEMENT

Contents

1 Tutorials **3**

1.1 Hydrodynamics and sediment transport at the river Flaz 3
1.1.1 Introduction 3
1.1.2 Computational mesh 3
1.1.3 Hydraulics 3
1.1.3.1 Set up the configuration file model.json 4
1.1.3.2 Set up the configuration file simulation.json 6
1.1.3.3 Set up the configuration file results.json 7
1.1.3.4 Steady flow simulation 7
1.1.3.5 Model calibration 8
1.1.3.6 Unsteady flow simulation 8
1.1.4 Morphology 9
1.1.4.1 Set up the configuration file model.json 9
1.1.4.2 Set up the configuration file simulation.json 12
1.1.4.3 Results 12
1.2 Post-processing 12
1.2.1 ParaView 13
1.2.2 Extract data from result file 13

2 Test cases **15**

2.1 Introduction and Backends 15
2.1.1 Introduction 15
2.1.2 Backends 15
2.2 Circular dam break 16
2.2.1 Description 16
2.2.2 Geometry and initial conditions 16
2.2.3 Results 17
2.2.4 Performance 20
2.2.4.1 Ubuntu 20
2.2.4.2 Windows 21
2.3 Conical dune 24
2.3.1 Description 24
2.3.2 Geometry and initial conditions 24
2.3.3 Results 25
2.3.4 Performance 28

3 References **31**

1

Tutorials

1.1 Hydrodynamics and sediment transport at the river Flaz

1.1.1 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.

1.1.2 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 3.0 compatible computational mesh. Figure 1.1 shows the bottom elevation of the river Flaz used in this tutorial.

1.1.3 Hydraulics

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 are saved in one folder and the simulation will automatically generate a new folder called “run” which contains the output.

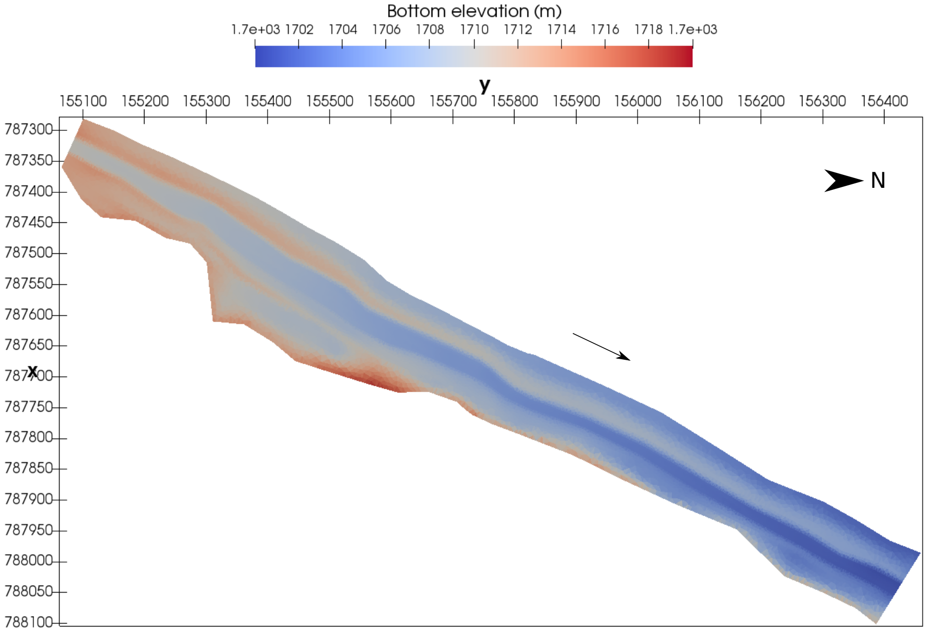


Figure 1.1 *Planar view of the initial bottom elevation of the river Flaz*

1.1.3.1 Set up the configuration file model.json

The configuration file model.json for a hydrodynamic simulation has the following structure:

```
{
   "SETUP":{
        "simulation_name":"Flaz_steady_state",
        "DOMAIN": {
            "PHYSICAL_PROPERTIES": {...},
            "BASEPLANE_2D": {
                "GEOMETRY": {...},
                "HYDRAULICS": {
                    "PARAMETER": {...},
                    "FRICTION": {...},
                    "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 link 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, 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
                    }

"FRICTION": {
                    "type": "strickler",
                    "default_friction": 30,
                    "regions": [
                            {"region_name": "one",
                             "friction": 28.0},
```

```
                            {"region_name": "two",
                             "friction": 30.0},
                            {"region_name": "three",
                             "friction": 35.0},
                            ...
                        ]
                    }

"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
```

1.1.3.2 Set up 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",
```

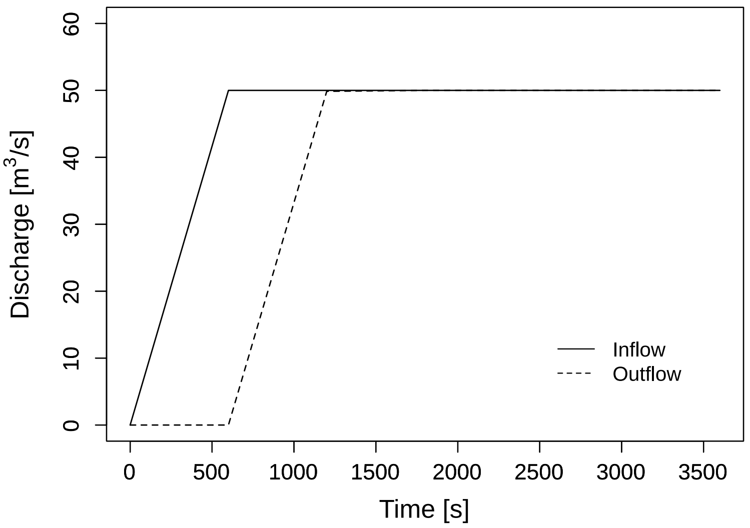


Figure 1.2 *Inflow and outflow hydrograph at boundaries*

```
            "bottom_elevation",
            "spec_discharge",
            "water_depth"
        ]
    }
}
```

1.1.3.3 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": "xdmf"}
        ]
    }
}
```

1.1.3.4 Steady flow simulation

The simulation results are stored inside the results.h5 binary. By calling the python script, 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 1.2).

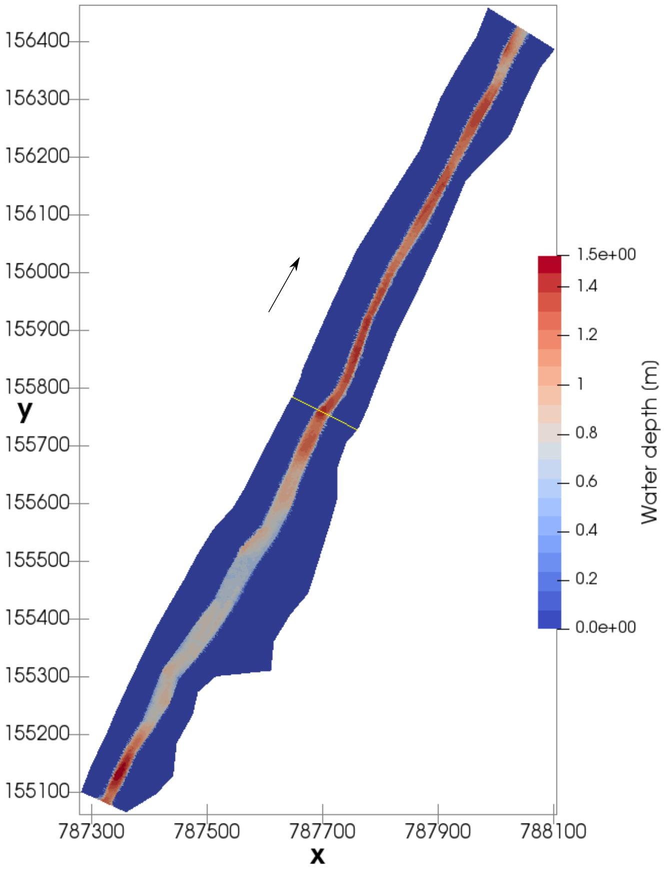


Figure 1.3 *Water depth at the end of the steady flow simulation (t=3600 s) with the cross section location (yellow line)*

1.1.3.5 Model calibration

The calibration of the friction value is done by comparing the water surface elevation between BASEMENT version 3.0 and version 2.8 at a cross section located in the middle of the channel (Figure 1.3). 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 1.4. The steady flow simulation of BASEMENT version 3.0 provides similar results to those obtained with BASEMENT version 2.8. There is no need to modify the friction value defined in Section 1.1.3.1.

1.1.3.6 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 1.1.3.1 don't change, except for the boundary block where the new discharge file (Inflow_transient.txt) replaces the stationary hydrogrpah. The initial block:

```
"INITIAL":{
```

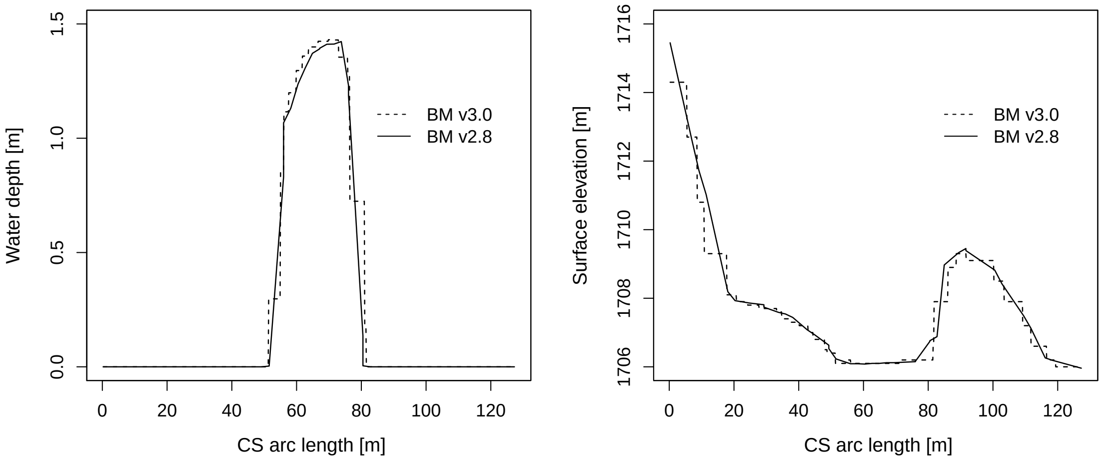


Figure 1.4 *Cross sectional view of the water depth and surface elevation facing the opposite flow direction at the end of the steady flow simulation (t=3600 s)*

```
                "type": "continue",
                "file": "../run/Flaz_steady_state_results",
                "time": 3000.0
            }
```

The running 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 the numerical simulation.

1.1.4 Morphology

1.1.4.1 Set up the configuration file model.json

The unsteady flow simulation is now converted into a 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 1.1.3.6. Additionally, the block morphology is defined inside the model.json file with the following structure

```
{
```

```
   "SETUP":{
        "simulation_name":"Flaz_unsteady_morph",
        "DOMAIN": {
            "PHYSICAL_PROPERTIES": {...},
            "BASEPLANE_2D": {
                "GEOMETRY": {...},
                "HYDRAULICS": {...},
                ""MORPHOLOGY": {
                    "INITIAL":{...},
                    "PARAMETER": {...},
                    "BEDMATERIAL": {...},
                    "BEDLOAD": {...}
                }
            }
        }
    }
}
```

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.

```
"PARAMETER": {
                        "morphodynamic_start": 0.0,
                        "sediment_porosity": 0.4,
                        "sediment_density" : 2650.0
                    }
```

The bed material is composed of uniform grains with one diameter (m). 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": {
                        "GRAIN_CLASS": {
                            "diameters": [0.050]
                        },
                        "FIXED_BED": {
                            "type": "region_defined",
                            "correction_accuracy": 0.0,
                            "max_iteration": 300,
                            "regions": [
                                {"region_name": "one",
                                 "z_rel": -0.8},
```

```
                    {"region_name": "two",
                    "z_rel": 0.0},
                    {"region_name": "three",
                    "z_rel": 0.0},
                    {"region_name": "four",
                    "z_rel": -2.0},
                    {"region_name": "five",
                    "z_rel": 0.0},
                    {"region_name": "six",
                    "z_rel": -2.0},
                    {"region_name": "seven",
                    "z_rel": -2.0},
                    {"region_name": "eight",
                    "z_rel": 0.0},
                    {"region_name": "nine",
                    "z_rel": 0.0},
                    {"region_name": "ten",
                    "z_rel": 0.0},
                    {"region_name": "eleven",
                    "z_rel": -0.2},
                    {"region_name": "twelve",
                    "z_rel": -0.4}
                ]
            }
        }
```

In the bedload block, the bedload transport formula is chosen, the lateral bed slope according to the formula of Ikeda is defined as well as the morphological boundary conditions. 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. 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.

```
"BEDLOAD": {
                    "FORMULA": {
                        "type": "MPM_like",
                        "factor":1.0,
                        "coefficient":3.2,
                        "exponent":1.6,
                        "critical_value":0.047
                    },
                    "DIRECTION": {
                        "IKEDA": {
                            "factor": 1.5
                        }
                    },
```

```
                        "BOUNDARY":{
                            "STANDARD": [
                                {
                                    "name": "inflow_MOR",
                                    "string_name": "Inflow",
                                    "type": "transport_capacity",
                                    "boundary_factor": 0.8
                                },
                                {
                                    "name": "outflow_MOR",
                                    "string_name": "Outflow",
                                    "type": "equilibrium_out",
                                    "reference_bed_elevation": 1700.68
                                }
                            ]
                        }
                    }
```

1.1.4.2 Set up the configuration file simulation.json

The simulation time defined in the simluation.json file is the same as for the unsteady flow in the hydraulics simulation and two additional outputs are defined.

```
{
        "OUTPUT": [
            "...",
            "water_depth",
            "delta_z",
            "ns_mor_discharge"
        ]
    }
}
```

1.1.4.3 Results

The morphological changes of the river bed are observed on Figure 1.5. The software ParaView was used for the post-processing of the output file (.xdmf).

1.2 Post-processing

BASEMENT version 3.x simulation results are generated inside the scenario directory and stored in binary format ".h5". These results are converted into specific output type that can be visualized and modified by external softwares. Two types of output are available, ".xdmf" for the values calculated over the cells and ".csv" for the values calculated at stringdefs. The ".xdmf" output type is defined in the configuration file results.json (Section 1.1.3.3). The delimited text file (.csv) is generated by calling the python script "BMv3NodestringResults.py".

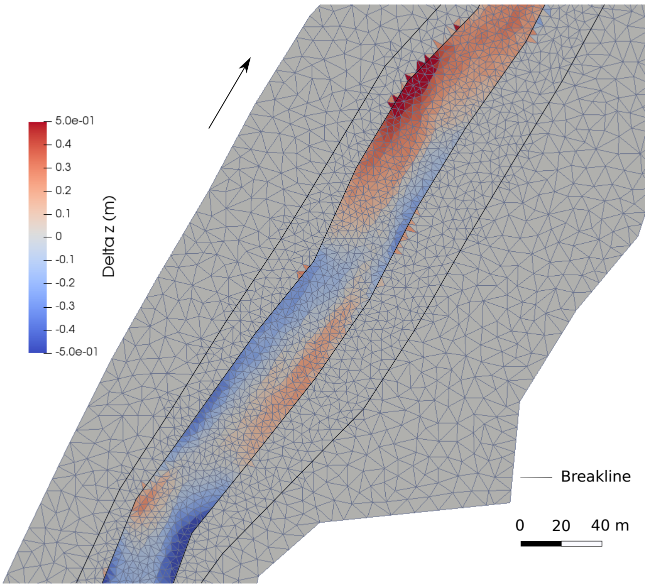


Figure 1.5 *Planar view of the widening part of the river Flaz with the change in bed elevation (Delta z) due to the flood event at the end of the morphodynamic simulation (t= 84'000 s)*

1.2.1 ParaView

The free and open source application ParaView is used to generate 2D views of BASEMENT version 3.x simulation results. The .xdmf file can be imported into ParaView to visualize the output data calculated over the cells like for example the water_depth, bottom_elevation and flow_velocity_abs.

Figure 1.5 gives an example of post-processing using ParaView for the morphological changes (delta z). A tutorial of the software ParaView is given in section 2.3 of BASEMENT 2.8 Tutorial for 3D visualization of the results. Be careful, only 2D visualization is available in ParaView for the output of BASEMENT version 3.x. Therefore, only a planar view of the results is available (see Figure 1.5).

1.2.2 Extract data from result file

This tutorial shows how to extract values from the stringdefs, i.e. boundaries to obtain the results of Figure 1.2. The python script BMv3NodestringResults.py can be downloaded from www.basement.ethz.ch and should be saved inside the scenario directory generated. The python script will read the stringdef data stored inside the _results.h5 file and convert them into a delimited text format (.csv). The data calculated at the stringdef are the discharge, the mean water surface elevation (wse), the wetted area, the mean bottom elevation, reference elevation, wetted geometric length, total water volume stored in cells of the stringdef, total conveyance of cells, the morphological flux and the bedload transport.

To generate the .csv text file:

1. Save the BMv3NodestringResults.py file inside the scenario directory
2. Open a command window

3. Change folder and go inside the scenario directory
4. Call the python script with

```
$ python BMv3NodestringResults.py
```

As output, a result.csv, discharge.csv and time step.csv are generated. The discharge.csv provides the discharge value for each stringdef (columns) listed in the same order of appearance as in the .2dm file. The rows stand for the simulation time step at which the output are generated (first row is the initial time t= 0.0 s). The number of time step is given as a list in the time step.csv file. Every available result of the stringdefs is defined in the header of the results.csv file, where the rows correspond to the list of stringdefs in the same order of appearance as in the .2dm file and for every time step.

2

Test cases

2.1 Introduction and Backends

2.1.1 Introduction

The test cases present well-defined hydro- and morphodynamic numerical simulations that are used to asses the performance and accuracy of the simulations with BASEMENT version 3.0. 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.

2.1.2 Backends

Different backends are used to execute the simulations of the test cases. 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, double graphic processing units (GPU) with a compute unified device architecture (CUDA) running on 1 core for low application programming interface (API) and finally single GPUs for high API. The features of the GPU backends are shown in Table 2.1.

Table 2.1 *Features of the GPU hardware used for the numerical simulations*

Card	K20	P100	GTX 1080 Ti	GTX 1070 Ti	GTX 1050 Ti	Quadro P620
Memory [GB]	5	12	11	8	4	2
Architecture	Kepler	Pascal	Pascal	Pascal	Pascal	Pascal
Bandwidth [GB/s]	208	549	484	256	112	80
CUDA cores	2496	3584	3584	2432	768	512

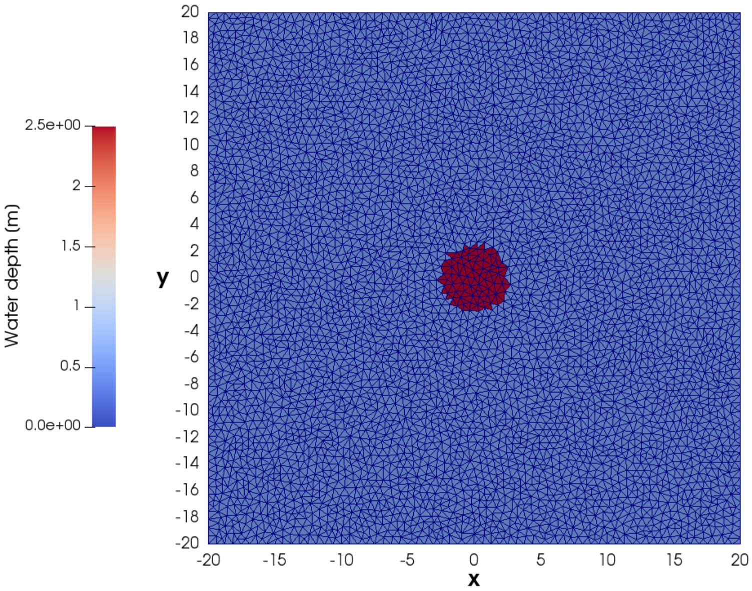


Figure 2.1 *Computational domain of the circular dam break (10k cells) with initial water depth.*

2.2 Circular dam break

2.2.1 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).

2.2.2 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 2.1 shows the initial conditions of the circular dam break with the computational domain of 10k cells.

2.2.3 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 2.2. 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 2.2 show how the solution obtained with BASEMENT version 3.0 (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. 2.1 and for various reference elevations. The result is displayed in Figure 2.3.

$$e_X = \frac{|X_{BM} - X_{ref}|}{X_{ref}} \tag{2.1}$$

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. 2.2, e.g Vanzo et al. (2016)) is calcutated 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 2.2. The relative norm error decreases for finer meshes.

$$L^2 = \sqrt{\frac{\sum_{i=1}^{N}(H_{ref}(i) - H_{BM}(i))^2}{\sum_{i=1}^{N} H_{ref}(i)^2}} \tag{2.2}$$

Table 2.2 *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

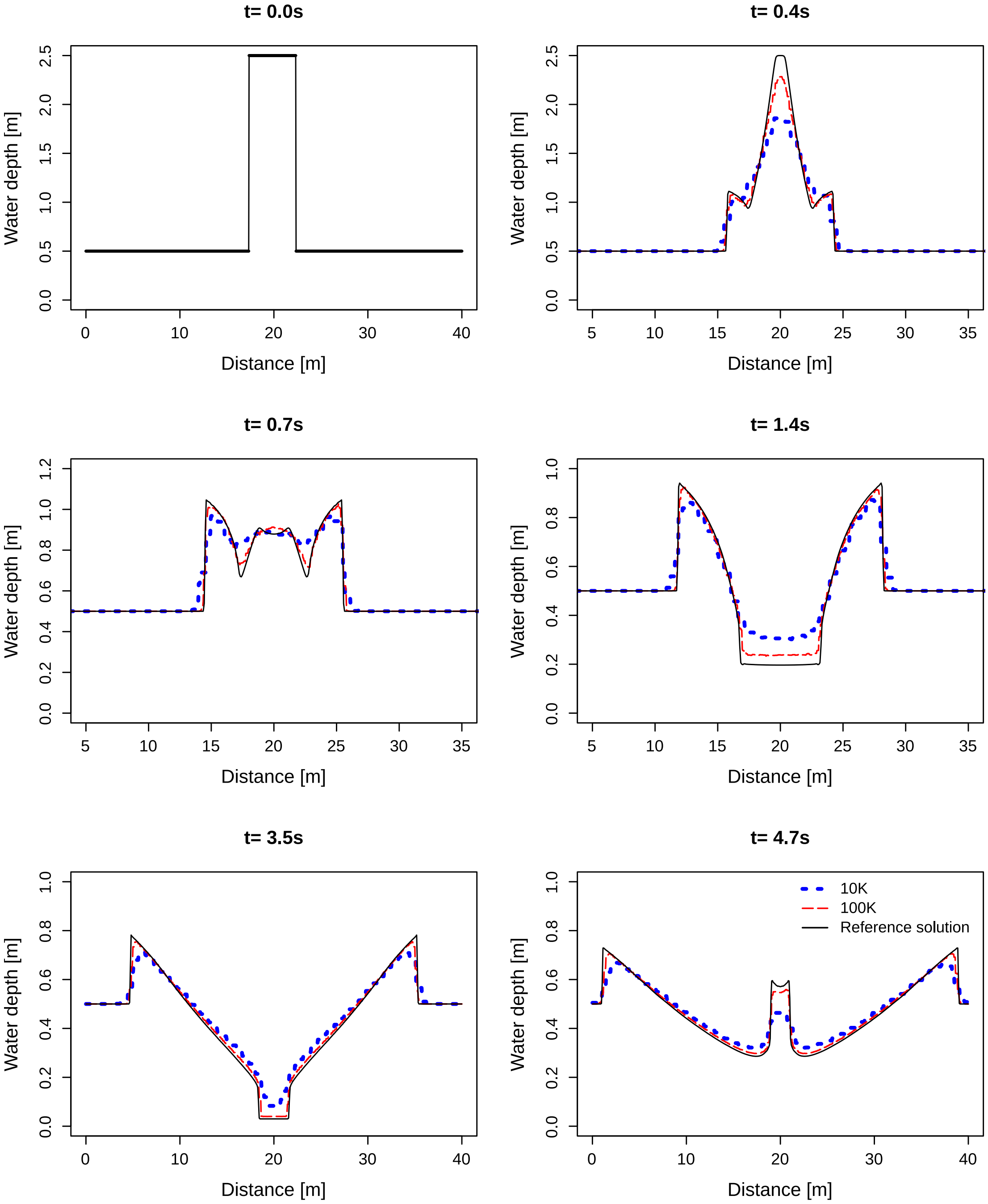


Figure 2.2 *Snapshots of the water depth evolution along the reference cross-section. Comparison between BASEMENT simulations with a computational grid of 10k cells, 100k cells and the reference solution of Toro (2001).*

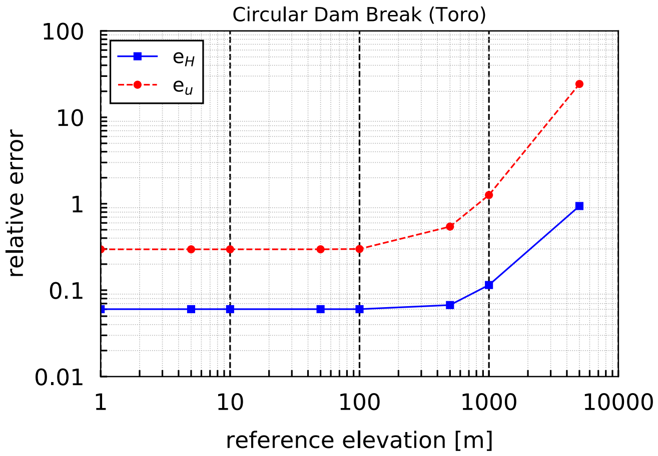


Figure 2.3 *Relative error compared to the reference solution of Toro (2001)*

2.2.4 Performance

The performance is measured as speedup S, which is defined as the ratio between the reference execution time T_1 (Xeon 1) and execution time of parallel or GPU backends T_N. The speedup states how much faster the parallel and GPU applications run compared to the reference execution time. The actual speedup is determined by several factors whose influences largely depend on size and type of the simulation.

2.2.4.1 Ubuntu

Tables 2.3, 2.4 and 2.5 contain the execution time and speedup of all the simulations performed on different backends under Linux. The backends belong to two cathegories, either in Central Processor Units (CPU) or Graphics Processor Units (GPU). The CPU run simulations exclusively with a double precision, while GPU can perform simulations with a single or double precision. More details are explained in Section 2.1.2.

Table 2.3 *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	0.36	0.19 (1.89x)	0.11 (3.27x)	0.06 (6.00x)	0.04 (9.00x)	0.04 (9.00x)
50k	3.98	2.08 (1.91x)	1.13 (3.52x)	0.61 (6.52x)	0.34 (11.71x)	0.22 (18.09x)
100k	11.48	5.98 (1.92x)	3.19 (3.60x)	1.68 (6.83x)	0.89 (12.90x)	0.52 (22.08x)
500k	137.57	70.18 (1.96x)	36.76 (3.74x)	18.82 (7.31x)	9.57 (14.38x)	4.93 (27.90x)
1000k	409.59	200.36 (1.96x)	108.98 (3.76x)	55.55 (7.37x)	28.15 (14.55x)	14.38 (28.48x)

Table 2.4 *Computational time (s) and speedup (in parentheses) of the circular dam break simulations for GPU simulations*

Mesh size	Xeon 1 Double	Quadro P620 Single	Quadro P620 Double	Tesla K20 Single	Tesla K20 Double	Tesla P100 Single	Tesla P100 Double
10k	0.36	0.05 (7.20x)	0.06 (6.00x)	0.04 (9.00x)	0.04 (9.00x)	0.04 (9.00x)	0.04 (9.00x)
50k	3.98	0.28 (14.21x)	0.48 (8.29x)	0.16 (24.88x)	0.21 (18.95x)	0.18 (22.11x)	0.19 (20.95x)

Mesh size	Xeon 1 Double	Quadro P620 Single	Quadro P620 Double	Tesla K20 Single	Tesla K20 Double	Tesla P100 Single	Tesla P100 Double
100k	11.48	0.66 (17.39x)	1.24 (9.26x)	0.34 (33.76x)	0.46 (24.96x)	0.36 (31.89x)	0.39 (29.44x)
500k	137.57	5.83 (23.60x)	12.94 (10.63x)	2.29 (60.07x)	3.89 (35.37x)	2.17 (63.40x)	2.58 (53.32x)
1000k	409.59	16.39 (24.99x)	37.32 (10.98x)	5.84 (70.14x)	10.56 (38.79x)	5.11 (80.15x)	6.35 (64.50x)

Table 2.5 *Computational time (s) and speedup (in parentheses) of the circular 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
10k	0.36	0.03 (12.00x)	0.05 (7.20x)	0.03 (12.00x)	0.03 (12.00x)	0.03 (12.00x)	0.04 (9.00x)
50k	3.98	0.21 (18.95x)	0.35 (11.37x)	0.13 (30.62x)	0.18 (22.11x)	0.13 (30.62x)	0.16 (24.88x)
100k	11.48	0.48 (23.92x)	0.78 (14.72x)	0.30 (38.27x)	0.41 (28.00x)	0.25 (45.92x)	0.34 (33.76x)
500k	137.57	3.60 (38.21x)	7.25 (18.98x)	1.96 (70.19x)	3.17 (43.40x)	1.62 (84.92x)	2.46 (55.92x)
1000k	409.59	9.80 (41.79x)	20.57 (19.91x)	4.94 (82.91x)	8.40 (48.76x)	3.96 (103.43x)	6.40 (64.00x)

Figure 2.4 shows the speedup achieved by the different backends, where the biggest time improvement is observed for the GPU single precision. The speedup increases particularly for meshes with a fine resolution (500K and 1000K) using GPU.

Figure 2.5 compares the speedup with the number of threads (CPU) between the different mesh resolutions. The simulations were performed using the CPU backend type Intel Xeon Gold 6154 (3.00GHz). 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.

2.2.4.2 Windows

Tables 2.6 contains the execution time and speedup of all the simulations performed on different backends under Windows. The simulations were run on Intel Xeon E5-2667 v3, 3.20GHz processors with a double precision.

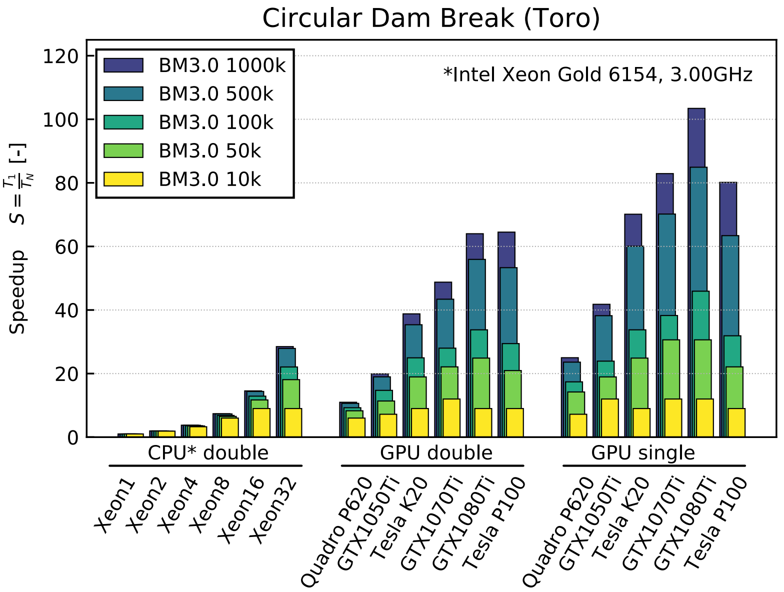


Figure 2.4 *Speedup of the computational time [T_N] for CPU (double) and GPU (single and double) compared to the time of the sequential backend Xeon1 [T_1] for the circular dam break*

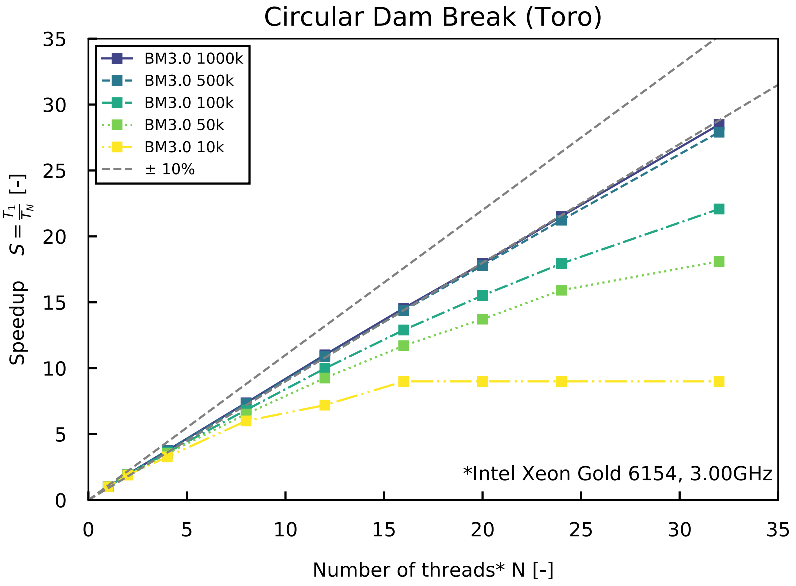


Figure 2.5 *Speedup of the computational time [T_N] using different number of threads (CPU) compared to the time of the sequential backend Xeon1 [T_1] for the circular dam break*

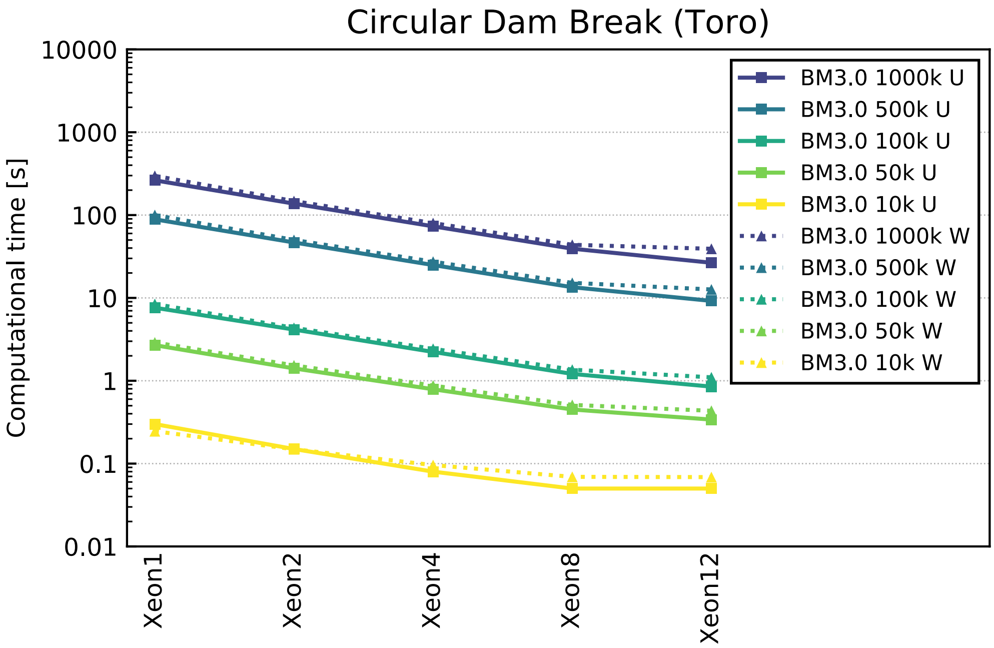


Figure 2.6 *Comparison of computational time using different number of threads (CPU) under Windows and Ubuntu on Intel Xeon E5-2667 v3, 3.20GHz processors.*

Table 2.6 *Computational time (s) and speedup (in parentheses) of the circular dam break simulations for CPU simulations under Windows 10 (Intel Xeon E5-2667 v3, 3.20GHz)*

Mesh size	Xeon 1 Double	Xeon 2 Double	Xeon 4 Double	Xeon 8 Double	Xeon 12 Double
10k	0.25	0.17 (1.53x)	0.10 (2.50x)	0.07 (3.57x)	0.07 (3.57x)
50k	2.90	1.53 (1.90x)	0.88 (3.30x)	0.51 (5.68x)	0.43 (6.74x)
100k	8.44	4.35 (1.94x)	2.44 (3.45x)	1.36 (6.2x)	1.10 (7.67x)
500k	100.12	50.09 (1.99x)	27.40 (3.69x)	15.20 (6.58x)	12.65 (7.91x)
1000k	297.61	147.53 (2.00x)	80.06 (3.71x)	43.81 (6.9x)	39.18 (7.58x)

The comparison of the executation time between Windows and Ubuntu in Figure 2.6 shows a very similar performance. The scalability under Ubuntu seems slightly better than under Windows 10.

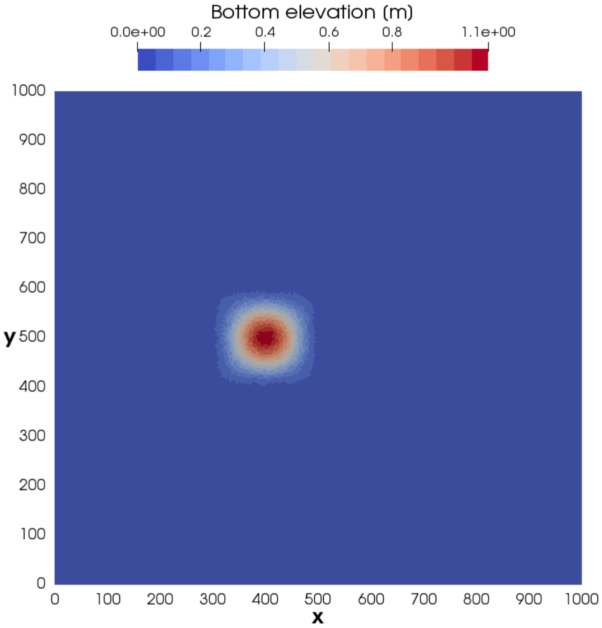


Figure 2.7 *Initial bottom elevation of the conical dune, maximum elevation of 1.1 m at [x=400, y=500].*

2.3 Conical dune

2.3.1 Description

The conical dune test proposed by Hudson and Sweby (2005) is used to assess the performance 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 α_s. Numerical results are compared with the approximate analytical solution for the spread angle of the disturbance front proposed by de Vriend (1987).

2.3.2 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 2.7):

$$z_{B,initial} = \begin{cases} 0.1 + sin^2\left(\frac{\pi(x-300)}{200}\right) sin^2\left(\frac{\pi(y-400)}{200}\right) & \text{for } x \in [300, 500] \\ & \text{and } y \in [400, 600] \\ 0.1 & \text{otherwise} \end{cases} \tag{2.3}$$

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

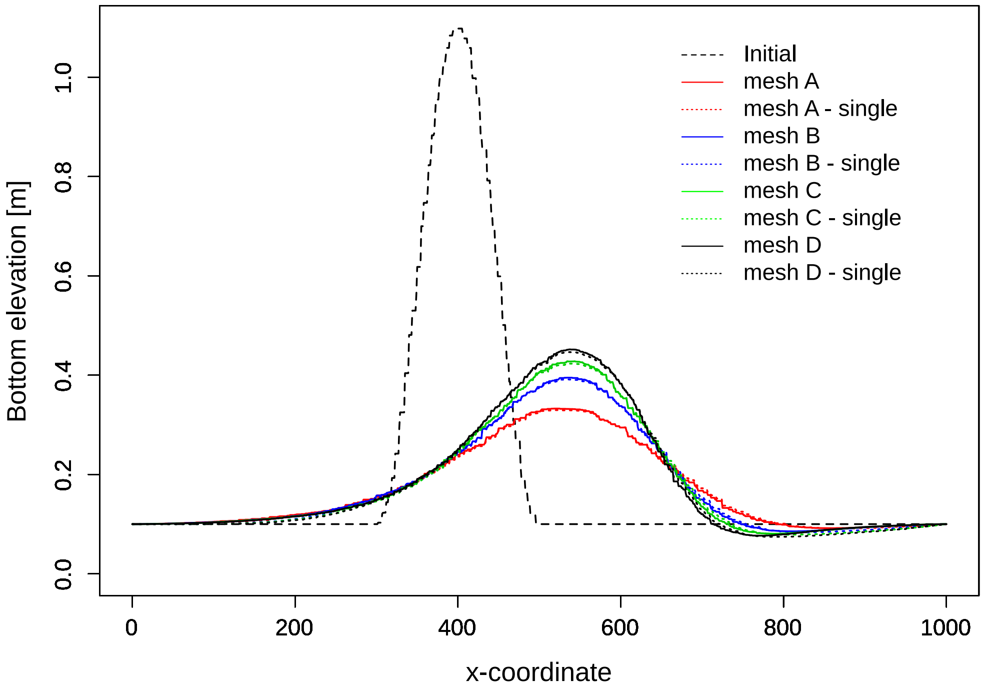


Figure 2.8 *Longitudinal bottom elevation profile at y= 500 m for initial conditions and after 100h of simulation for the meshes A, B, C and D with double and single precision*

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. 2.4) proposed by de Vriend (1987) gives a value of $\alpha_{s,ref} = 21.787°$.

$$\alpha_s = arctan\left(\frac{3\sqrt{3}(m_G - 1)}{9m_G - 1}\right) \tag{2.4}$$

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.

2.3.3 Results

Figure 2.8 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 2.7 summarizes the results obtained from the numerical simulations at t=100h for

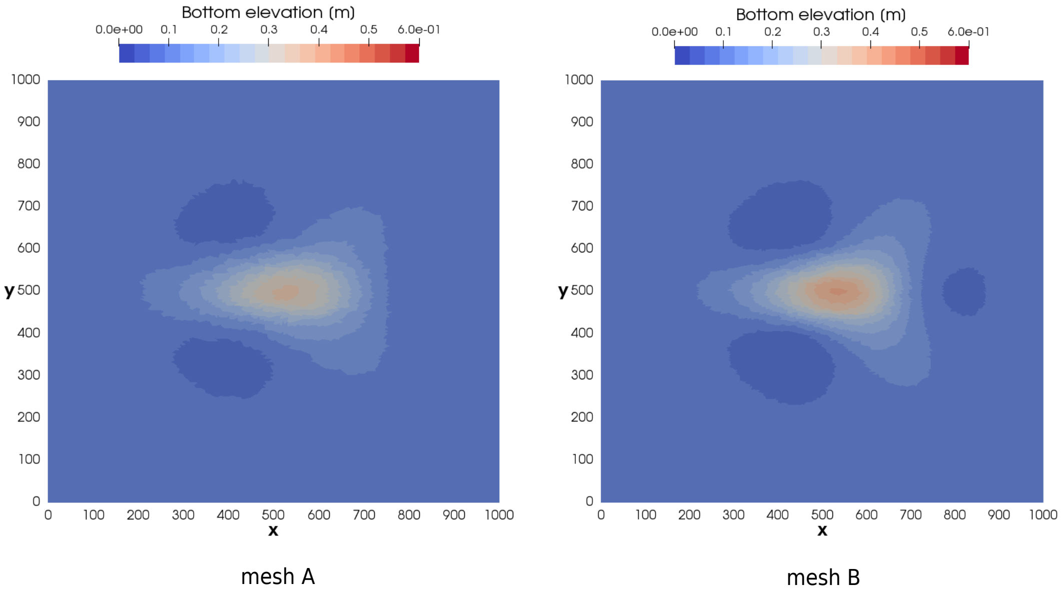


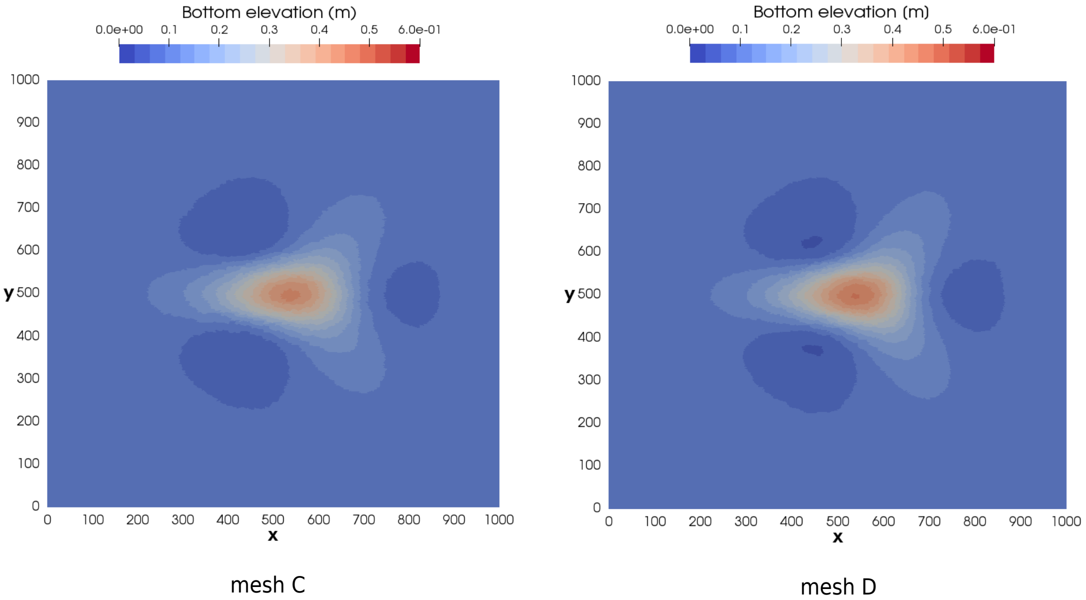
Figure 2.9 *Planar view of the bottom elevation after 100 hours of simulation for the meshes A and B.*

the different meshes using the backend types CPU and GPU-double. The spread angle α_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 2.9 and Figure 2.10). The spread angle was measured at time t=25h, 50h, 75h and 100h for each mesh size and the mean value is reported in Table 2.7. The accuracy of the simulations is assessed by the relative deviation between the measure spread angle α_s and the spread angle defined by de Vriend (1987) $\alpha_{s,ref}$.

Table 2.7 *Summary of the simulation results performed with CPU and GPU double at t=100h for different computational meshes*

Mesh	Number of cells [-]	Max. cone elevation [m]	α_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 2.7). The results are more accurate for finer meshes (mesh C and mesh D, Figure 2.8), 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 2.9 and Figure 2.10 by a well defined star shaped sediment transport and a spread angle that becomes closer to the value proposed by de Vriend (1987).



mesh C

mesh D

Figure 2.10 *Planar view of the bottom elevation after 100 hours of simulation for the meshes C and D.*

2.3.4 Performance

In total, 18 simulations were performed on different backends. The execution time depends on the mesh size, the computing processor and the number of cores used. The execution time is at maximum using the sequential platform and decreases using CPU accordingly to the number of cores used. The best performance reaches a speedup of two orders of magnitude using the GPU backend (Tables 2.8, 2.9 and 2.10).

The performance of the different running platforms is compared with respect to the sequential running time (1 core). CUDA GPU-based processor proves its efficiency by showing a speedup increase for finer meshes, while the speedups of the CPU-based OpenMP platform increase with the increasing number of core, independently from the mesh type.

Table 2.8 *Computational time (h) and speedup S (inside brackets) of the conical dune simulations for CPU simulations (Intel Xeon E5-2667 v3, 3.20GHz)*

Mesh size	Xeon 1 Double	Xeon 2 Double	Xeon 4 Double	Xeon 8 Double	Xeon 12 Double	Xeon 14 Double
30k	7.26	3.88 (1.87x)	2.01 (3.61x)	1.04 (6.95x)	0.75 (9.71x)	0.65 (11.09x)
60k	23.08	11.95 (1.93x)	6.10 (3.78x)	3.15 (7.33x)	2.19 (10.52x)	1.94 (11.93x)
90k	38.47	20.45 (1.88x)	10.47 (3.67x)	5.40 (7.13x)	3.58 (10.74x)	3.27 (11.75x)
120k	61.30	32.48 (1.89x)	16.52 (3.71x)	8.43 (7.27x)	5.87 (10.44x)	5.19 (11.81x)

Table 2.9 *Computational time (s) and speedup S (inside brackets) of the conical dune simulations for GPU simulations*

Mesh size	Xeon 1 Double	Quadro P620 Single	Quadro P620 Double	Tesla K20 Single	Tesla K20 Double	Tesla P100 Single	Tesla P100 Double
30k	7.26	0.59 (12.34x)	1.12 (6.49x)	0.53 (13.69x)	0.38 (18.98x)	0.28 (25.85x)	0.32 (22.44x)
60k	23.08	1.40 (16.46x)	2.98 (7.73x)	1.15 (20.08x)	0.78 (29.61x)	0.51 (45.66x)	0.63 (36.45x)
90k	38.47	2.06 (18.64x)	4.71 (8.17x)	1.68 (22.87x)	1.10 (35.07x)	0.67 (57.51x)	0.84 (45.70x)
120k	61.30	3.13 (19.64x)	7.40 (8.29x)	2.41 (25.47x)	1.45 (42.15x)	0.87 (70.25x)	1.22 (50.19x)

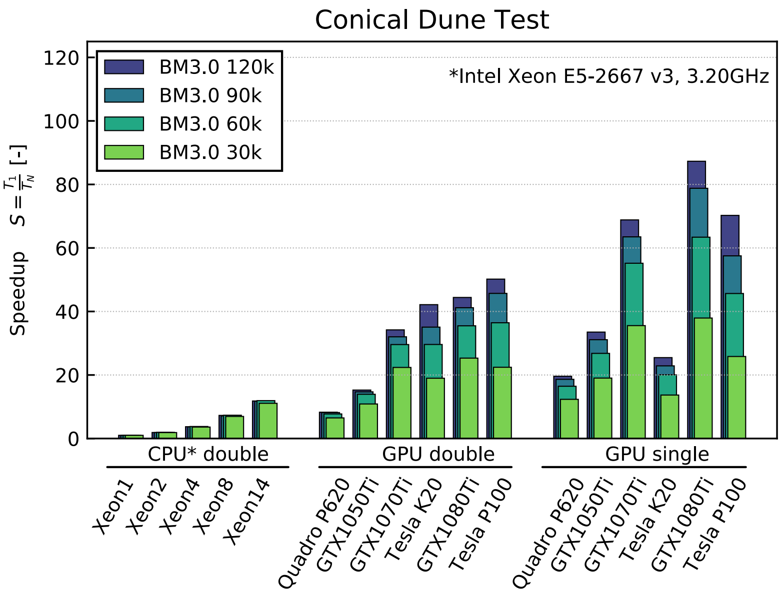


Figure 2.11 *Speedup of the computational time [T_N] for CPU (double) and GPU (single and double) compared to the time of the sequential backend Xeon1 [T_1] for the conical dune test case*

Table 2.10 *Computational time (s) and speedup S (inside brackets) of the conical dune 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	7.26	0.38 (19.03x)	0.67 (10.87x)	0.20 (35.55x)	0.32 (22.38x)	0.19 (37.94x)	0.29 (25.31x)
60k	23.08	0.86 (26.81x)	1.66 (13.93x)	0.42 (55.19x)	0.78 (29.58x)	0.36 (63.39x)	0.65 (35.49x)
90k	38.47	1.24 (31.11x)	2.62 (14.69x)	0.61 (63.50x)	1.20 (32.02x)	0.49 (78.79x)	0.93 (41.18x)
120k	61.30	1.83 (33.51x)	4.02 (15.26x)	0.89 (68.83x)	1.79 (34.21x)	0.70 (87.30x)	1.38 (44.41x)

Figure 2.11 shows the speedup achieved by the different backends, where the biggest time improvement is observed for the GPU single precision.

Figure 2.12 compares the speedup with the number of threads (CPU) between the different mesh resolutions. The simulations were performed using the CPU backend on an Intel Xeon E5-2667 v3 (3.20GHz) processor.

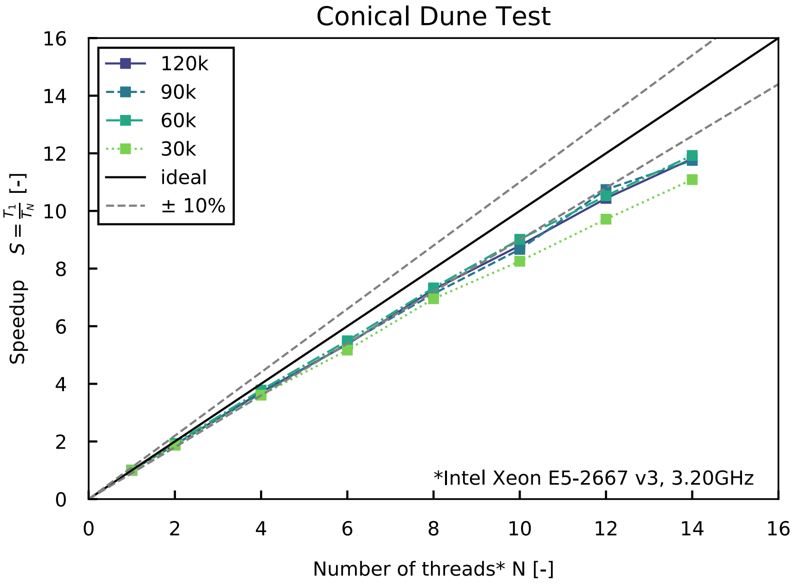


Figure 2.12 *Speedup of the computational time [T_N] using different number of threads (CPU) compared to the time of the sequential backend Xeon1 [T_1] for the conical dune test case*

3

References

de Vriend, H. (1987). 2DH mathematical modelling of morphological evolutions in shallow water. *Coastal engineering*, 11(1): 1–27.

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.

Toro, E.F. (2001). Shock-Capturing Methods for Free-Surface Shallow Flows. *John Wiley*, Chichester, New York.

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.

BASIC SIMULATION ENVIRONMENT
FOR SIMULATION OF ENVIRONMENTAL FLOW
AND NATURAL HAZRAD SIMULATION

APPENDIX

VERSION 3.0
September 2019

1

Third Party Software

1.1 Third party software copyright notices

Abseil

Copyright Google Inc.

License: Apache 2.0

Bzip2

This program, "bzip2", the associated library "libbzip2", and all documentation, are copyright (C) 1996-2010 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@bzip.org
bzip2/libbzip2 version 1.0.6 of 6 September 2010

Double-conversion

Copyright 2006-2011, the V8 project authors. All rights reserved.

License: BSD 2.0

Egl-registry

Copyright (c) 2008-2018 The Khronos Group Inc.
Copyright (c) 2013-2017 The Khronos Group Inc.
Copyright (c) 2013-2017 The Khronos Group Inc.
Copyright (c) 2007-2016 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.

Freetype

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 `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 `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 file src/bdf/README and src/pcf/README). The same holds for the files `fthash.c' and `fthash.h'; their code was 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.

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.

Google Test

Copyright 2008, Google Inc.

All rights reserved.

License: BSD 2.0

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 Google, Inc.
Copyright © 2019 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 © 2006 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

HDF5 (Hierarchical Data Format 5) Software Library and Utilities
Copyright (c) 2006-2018, The HDF Group.

NCSA HDF5 (Hierarchical Data Format 5) Software Library and Utilities
Copyright (c) 1998-2006, 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.

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 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.

HDF5 is available with the SZIP compression library but SZIP is not part of HDF5 and has separate copyright and license terms. See SZIP Compression in HDF Products (www.hdfgroup.org/doc_resource/SZIP/) for further details.

HighFive

License: BSL 1.0

JSON for Modern C++

Copyright (c) 2013-2019 Niels Lohmann

License: MIT

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)2009-2019 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.
```

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 Sahastrabuddhe
    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.

OP2

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.

License: BSD 2.0

OpenMesh

Copyright (c) 2001-2015, RWTH-Aachen University

Department of Computer Graphics and Multimedia

All rights reserved.

www.openmesh.org

License: BSD 2.0

PCRE2

```
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.

Release 10 of PCRE2 is 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: ph10
Email domain:     cam.ac.uk

University of Cambridge Computing Service,
Cambridge, England.

Copyright (c) 1997-2017 University of Cambridge
All rights reserved.


PCRE2 JUST-IN-TIME COMPILATION SUPPORT
--------------------------------------

Written by:       Zoltan Herczeg
Email local part: hzmester
Emain domain:     freemail.hu

Copyright(c) 2010-2017 Zoltan Herczeg
All rights reserved.


STACK-LESS JUST-IN-TIME COMPILER
--------------------------------

Written by:       Zoltan Herczeg
Email local part: hzmester
Emain domain:     freemail.hu

Copyright(c) 2009-2017 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.
```

Qt

```
Copyright (C) 2018 The Qt Company Ltd and other contributors

License: LGPL 3.0
```

SimpleXdmf

```
Copyright (c) 2017 hsimyu

License: MIT
```

TCLAP

```
Copyright (c) 2003 Michael E. Smoot
Copyright (c) 2004 Daniel Aarno
Copyright (c) 2017 Google Inc.

License: MIT
```

Vcpkg

```
Copyright (c) Microsoft Corporation

All rights reserved.

License: MIT
```

Zlib

Copyright (C) 1995-2017 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

1.2 Third party software licenses

LGPL 3.0

GNU LESSER 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.

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.

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/>.

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>.

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.

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.

BSL 1.0

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.

BSD 2.0

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 <organization> 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 <COPYRIGHT HOLDER> 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.

MIT

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.

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:

- We don't promise that this software works. However, we will be interested in any kind of bug reports. (`as is' distribution)

- You can use this software for whatever you want, in parts or full form, without having to pay us. (`royalty-free' usage)

- 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:

- 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.

- 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:

- 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.

- 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