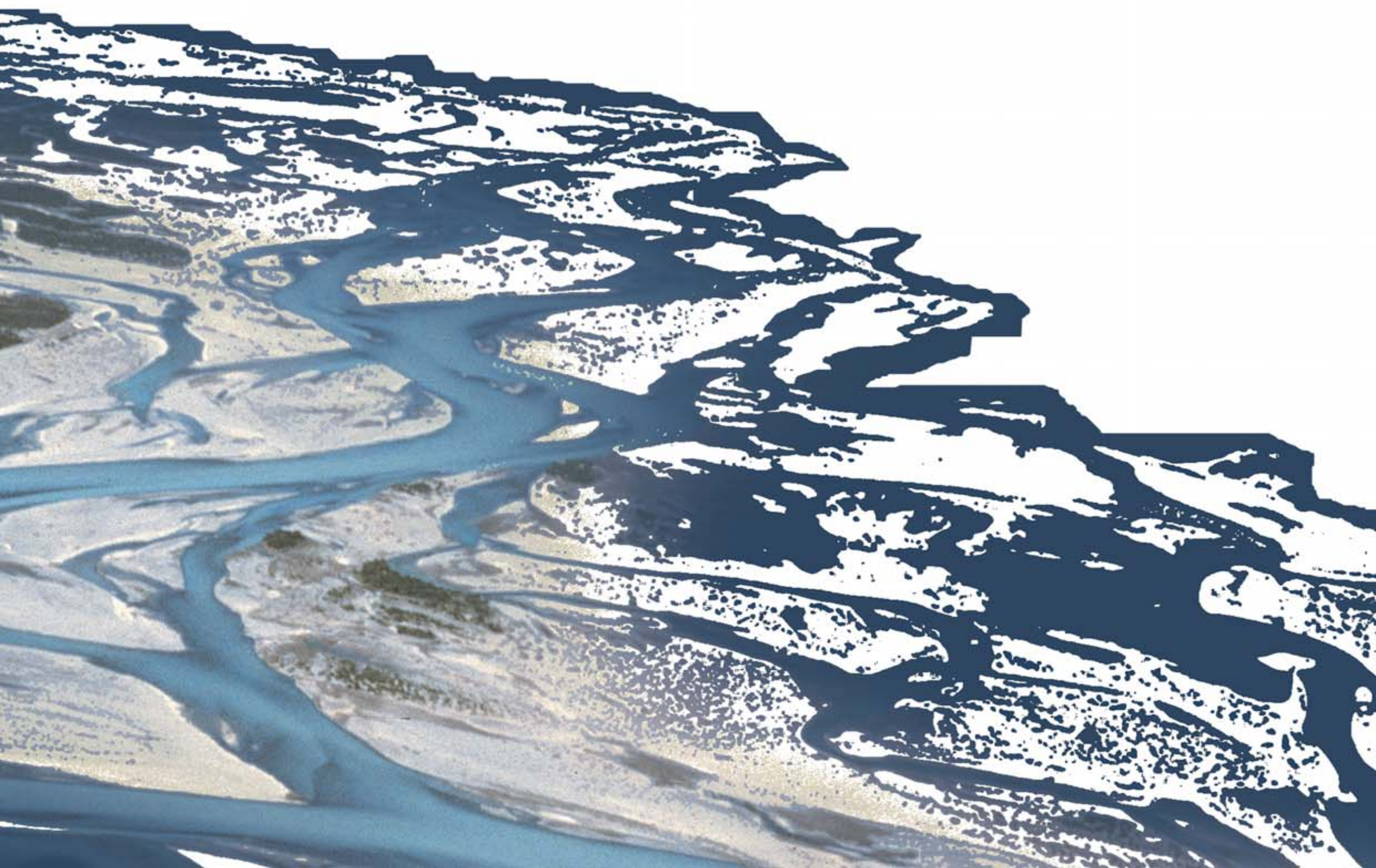
BASEMENT

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

SYSTEM MANUALS

VERSION 3.0
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

USER MANUAL

VERSION 3.0
September 2019

Contents

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.

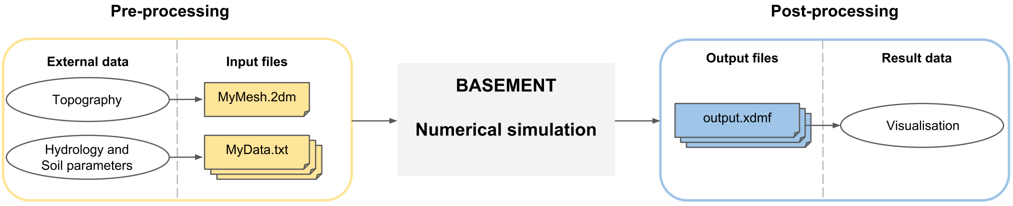


Figure 2.1 *Overview of the modelling procedure with BASEMENT*

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.

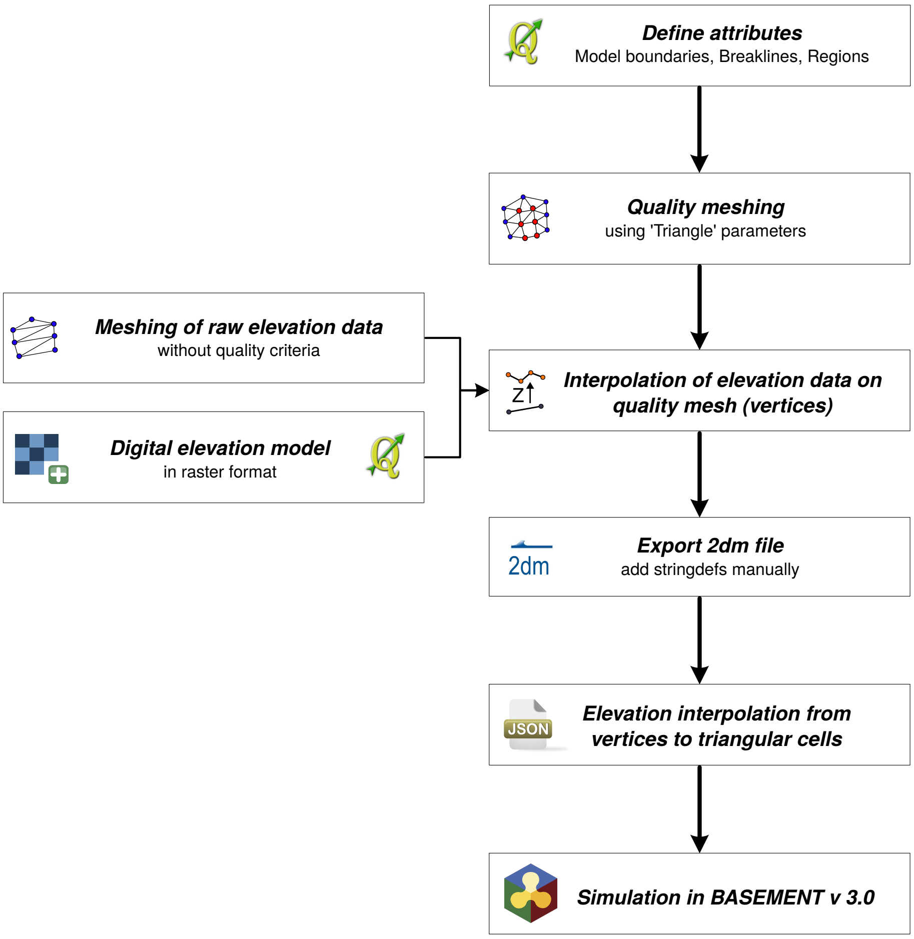


Figure 3.2 *BASEmesh workflow for small meshes and adaptations required in the 2dm and executable (.json) files for the simulation with BASEMENT v3.0*

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.

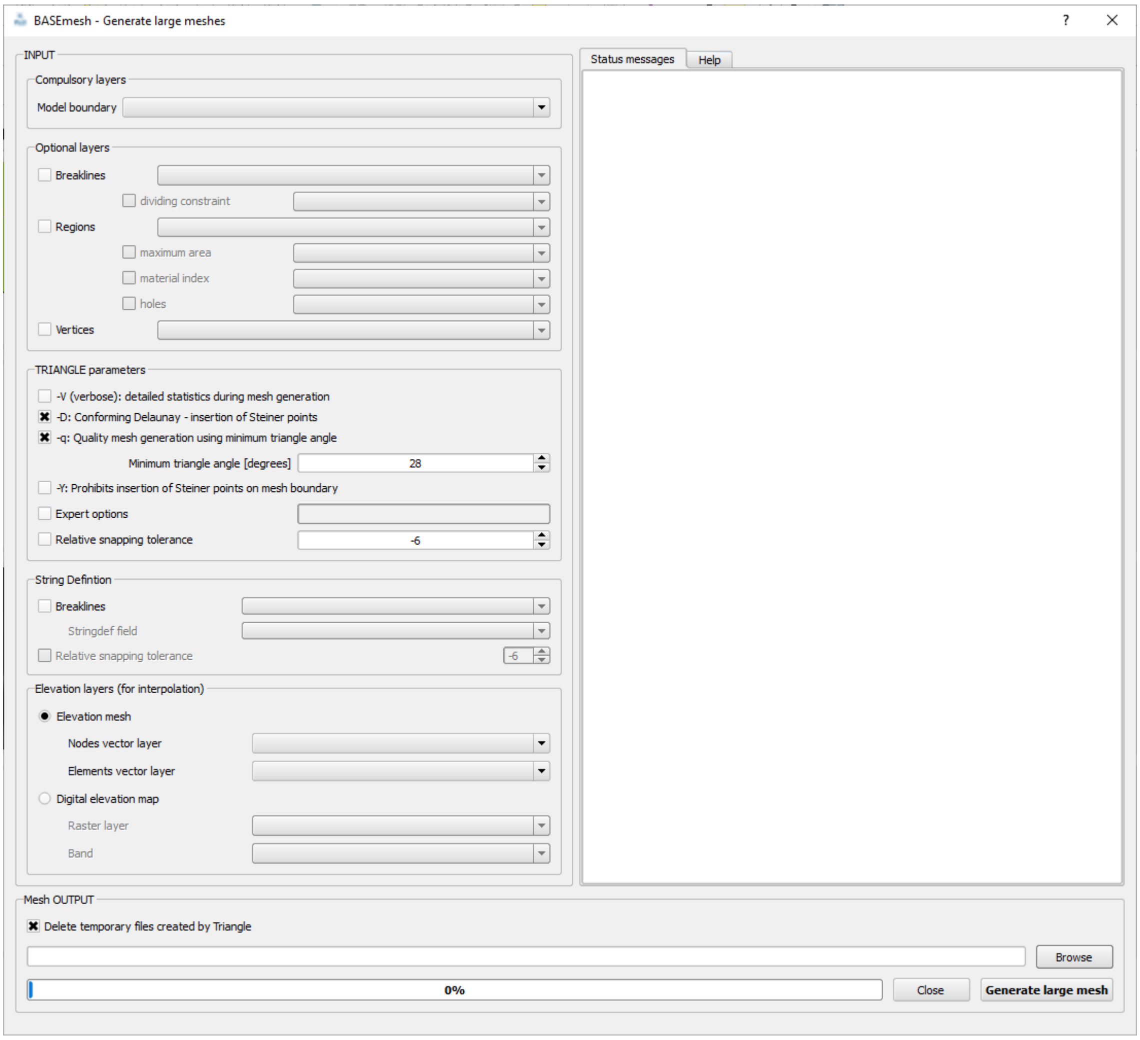


Figure 3.3 *The generation of large meshes can be done with the new procedure in BASEmesh 1.44.*

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

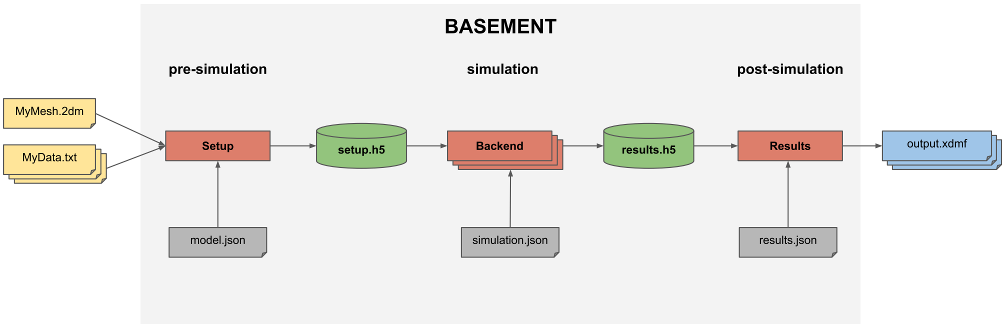


Figure 4.1 *BASEMENT simulation workflow*

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).

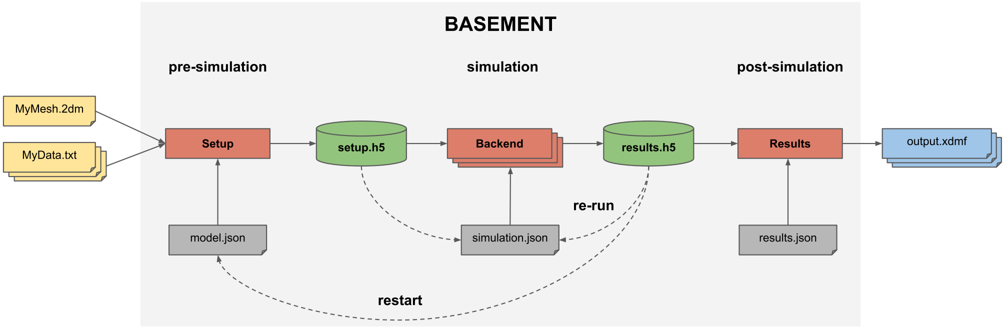


Figure 4.2 *BASEMENT simulation workflow with restart and re-run processes*

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.

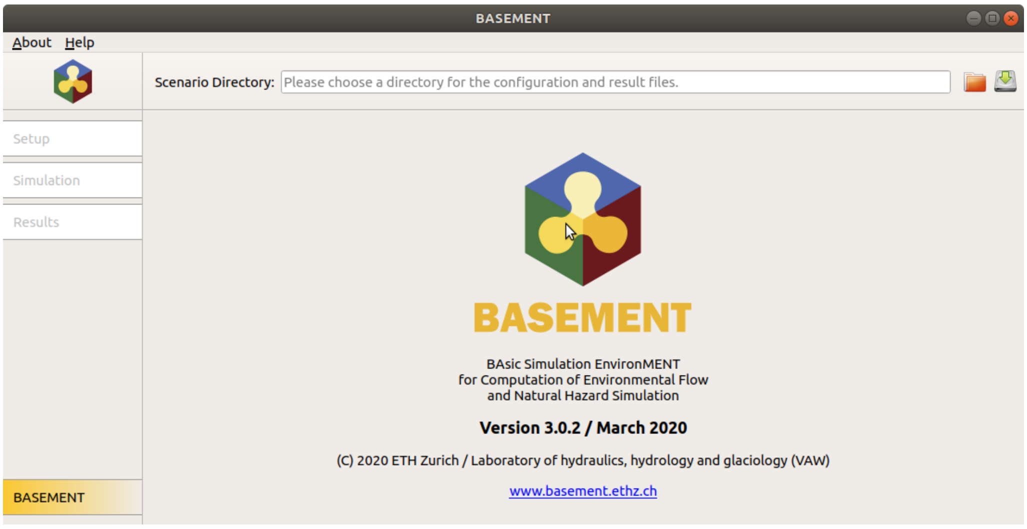


Figure 5.1 *Welcome Screen*

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).

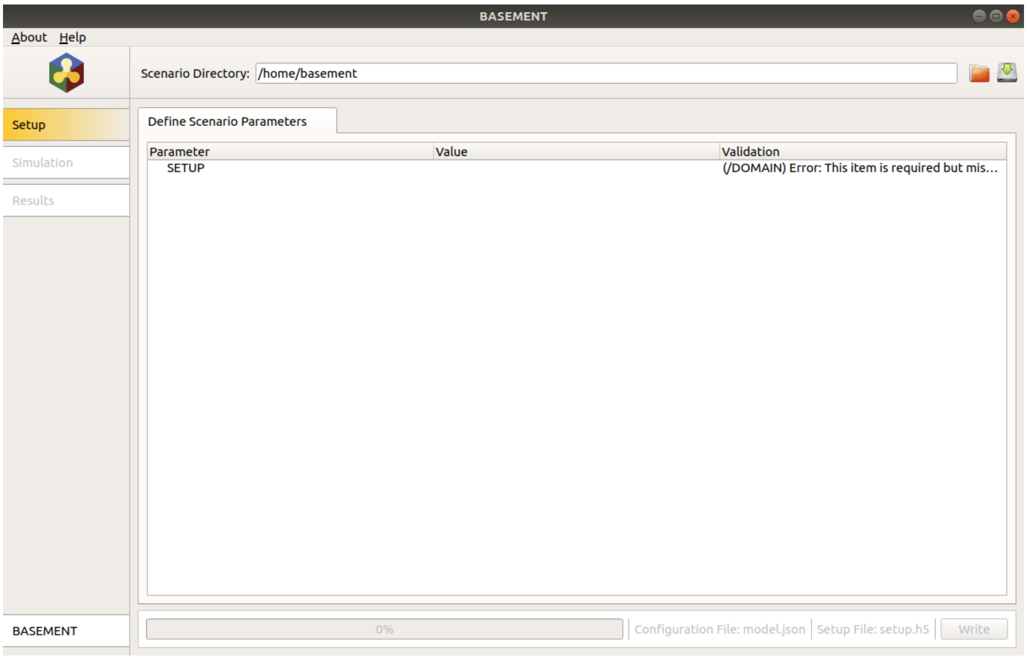


Figure 5.2 *Setup Screen*

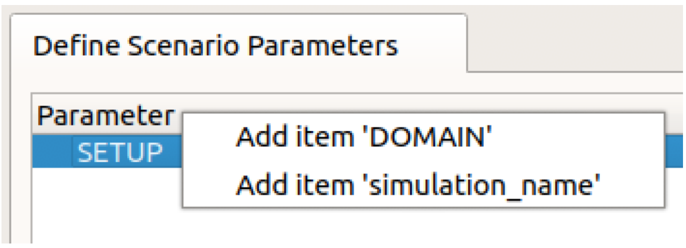


Figure 5.3 *Adding JSON Items*

Figure 5.4 *File Name Editor*

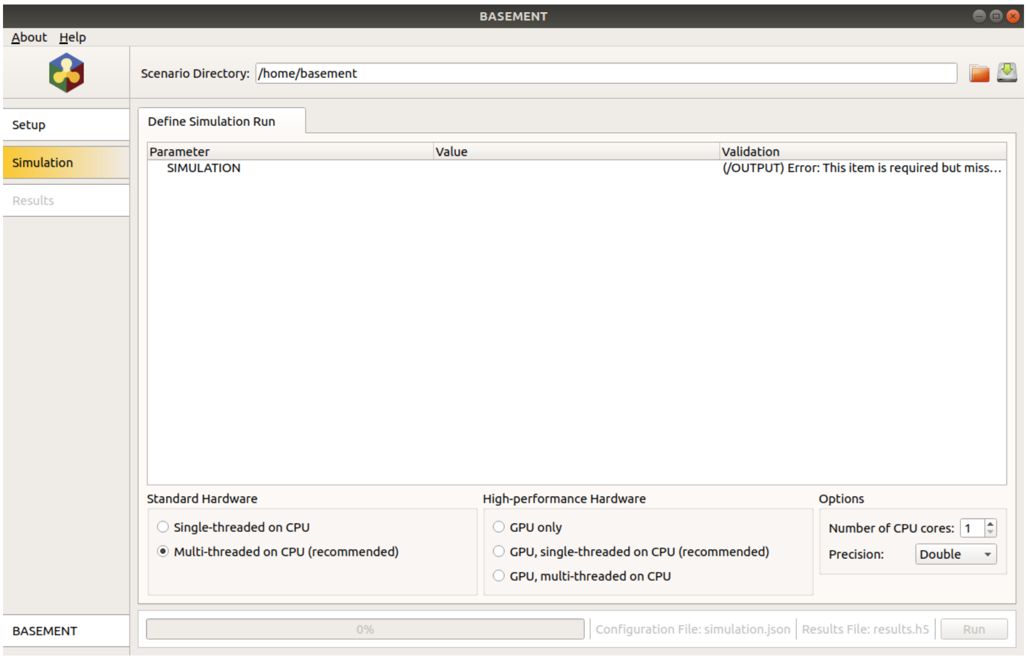


Figure 5.5 *Simulation Screen*

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.

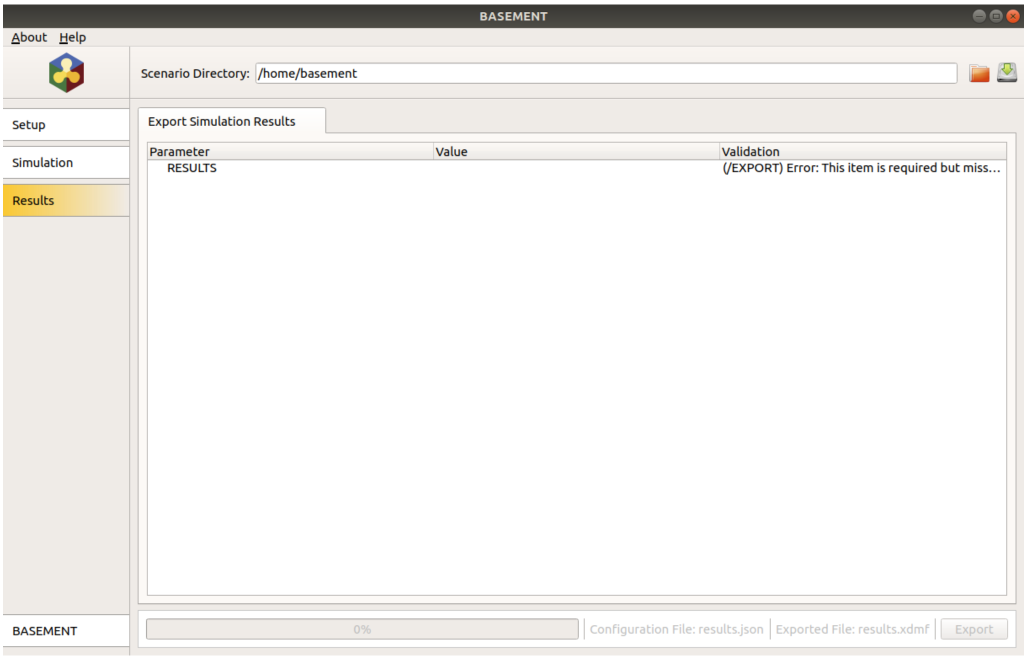


Figure 5.6 *Results Screen*

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

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
```

Double-conversion

Egl-registry

Freetype

```
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

Google Test

Harfbuzz

HDF5

HighFive

JSON for Modern C++

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
```

OP2

OpenMesh

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.
```

Qt

SimpleXdmf

TCLAP

Vcpkg

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

MIT

FTL

```
  We  specifically  permit  and  encourage  the  inclusion  of  this
  software, with  or without modifications,  in commercial products.
  We  disclaim  all warranties  covering  The  FreeType Project  and
  assume no liability related to The FreeType Project.


  Finally,  many  people  asked  us  for  a  preferred  form  for  a
  credit/disclaimer to use in compliance with this license.  We thus
  encourage you to use the following text:

   """
    Portions of this software are copyright © <year> The FreeType
    Project (www.freetype.org).  All rights reserved.
   """

  Please replace <year> with the value from the FreeType version you
  actually use.


Legal Terms
===========

0. Definitions
--------------

  Throughout this license,  the terms `package', `FreeType Project',
  and  `FreeType  archive' refer  to  the  set  of files  originally
  distributed  by the  authors  (David Turner,  Robert Wilhelm,  and
  Werner Lemberg) as the `FreeType Project', be they named as alpha,
  beta or final release.

  `You' refers to  the licensee, or person using  the project, where
  `using' is a generic term including compiling the project's source
  code as  well as linking it  to form a  `program' or `executable'.
  This  program is  referred to  as  `a program  using the  FreeType
  engine'.

  This  license applies  to all  files distributed  in  the original
  FreeType  Project,   including  all  source   code,  binaries  and
  documentation,  unless  otherwise  stated   in  the  file  in  its
  original, unmodified form as  distributed in the original archive.
  If you are  unsure whether or not a particular  file is covered by
  this license, you must contact us to verify this.

  The FreeType  Project is copyright (C) 1996-2000  by David Turner,
  Robert Wilhelm, and Werner Lemberg.  All rights reserved except as
  specified below.

1. No Warranty
--------------

  THE FREETYPE PROJECT  IS PROVIDED `AS IS' WITHOUT  WARRANTY OF ANY
  KIND, EITHER  EXPRESS OR IMPLIED,  INCLUDING, BUT NOT  LIMITED TO,
  WARRANTIES  OF  MERCHANTABILITY   AND  FITNESS  FOR  A  PARTICULAR
  PURPOSE.  IN NO EVENT WILL ANY OF THE AUTHORS OR COPYRIGHT HOLDERS
  BE LIABLE  FOR ANY DAMAGES CAUSED  BY THE USE OR  THE INABILITY TO
  USE, OF THE FREETYPE PROJECT.

2. Redistribution
-----------------

  This  license  grants  a  worldwide, royalty-free,  perpetual  and
  irrevocable right  and license to use,  execute, perform, compile,
  display,  copy,   create  derivative  works   of,  distribute  and
  sublicense the  FreeType Project (in  both source and  object code
  forms)  and  derivative works  thereof  for  any  purpose; and  to
  authorize others  to exercise  some or all  of the  rights granted
  herein, subject to the following conditions:

    o Redistribution of  source code  must retain this  license file
```

```
    (`FTL.TXT') unaltered; any  additions, deletions or changes to
    the original  files must be clearly  indicated in accompanying
    documentation.   The  copyright   notices  of  the  unaltered,
    original  files must  be  preserved in  all  copies of  source
    files.

  o Redistribution in binary form must provide a  disclaimer  that
    states  that  the software is based in part of the work of the
    FreeType Team,  in  the  distribution  documentation.  We also
    encourage you to put an URL to the FreeType web page  in  your
    documentation, though this isn't mandatory.

These conditions  apply to any  software derived from or  based on
the FreeType Project,  not just the unmodified files.   If you use
our work, you  must acknowledge us.  However, no  fee need be paid
to us.

3. Advertising
--------------

Neither the  FreeType authors and  contributors nor you  shall use
the name of the  other for commercial, advertising, or promotional
purposes without specific prior written permission.

We suggest,  but do not require, that  you use one or  more of the
following phrases to refer  to this software in your documentation
or advertising  materials: `FreeType Project',  `FreeType Engine',
`FreeType library', or `FreeType Distribution'.

As  you have  not signed  this license,  you are  not  required to
accept  it.   However,  as  the FreeType  Project  is  copyrighted
material, only  this license, or  another one contracted  with the
authors, grants you  the right to use, distribute,  and modify it.
Therefore,  by  using,  distributing,  or modifying  the  FreeType
Project, you indicate that you understand and accept all the terms
of this license.

4. Contacts
-----------

There are two mailing lists related to FreeType:

  o freetype@nongnu.org

    Discusses general use and applications of FreeType, as well as
    future and  wanted additions to the  library and distribution.
    If  you are looking  for support,  start in  this list  if you
    haven't found anything to help you in the documentation.

  o freetype-devel@nongnu.org

    Discusses bugs,  as well  as engine internals,  design issues,
    specific licenses, porting, etc.

Our home page can be found at

  https://www.freetype.org
```