Documentation of *FullSWOF_1D* v1.01.01 (2014-02-03)

Carine Lucas, carine.lucas@univ-orleans.fr

2014-01-20

Contents

1 Presentation of the *FullSWOF_1D* software

The name *FullSWOF_1D* stands for "Full Shallow Water equations for Overland Flow in one dimension of space". In this software, the Shallow Water (or Saint-Venant) equations are solved using finite volumes and numerical methods especially chosen for hydrodynamic purposes (transitions between wet and dry areas, small water heights, steady preservation ...). A graphic user interface is available at `https://sourcesup.renater.fr/projects/fullswof-ui/`.

For explanations concerning the numerical schemes and approximations, the reader is referred to Delestre et al. [2014], Delestre [2010], Delestre et al. [2009] and Delestre and James [2009]. For a precise description of the structure of the software in several classes, see the Doxygen file (refman.pdf in the *doc* directory) and Delestre [2008]. The structure of the source code is designed to make future evolutions easy: for example, a new friction law can easily be added in the libfriction library, by creating a new friction file.

If you plan to change the code of *FullSWOF_1D*, see Section 4 for explanations on how to use the benchmarks. Doing so, you should pay attention to the license (section 2.2)

2 Software distribution

2.1 How to download *FullSWOF_1D*?

The *FullSWOF_1D* software can be downloaded on the website `https://sourcesup.renater.fr/projects/fullswof-1d/`.

2.2 License

This software is distributed under CeCILL-V2 (GPL compatible) free software license. So, you are authorized to use the Software, without any limitation as to its fields of application.

If you make changes to *FullSWOF_1D* code, you are welcome to **contribute your changes to the main repository**, directly through the website (`https://sourcesup.renater.fr/projects/fullswof-1d/`) or by contacting its main developers (fullswof.contact@listes.univ-orleans.fr). You may prefer to distribute yourself the *Modified Software*. In such a case, we ask you to **change its name** in order to avoid confusion between your software and the original one. In such a case, pay attention to the text that follows.

The license authorizes you to distribute the *Modified Software*, in source code or object code form, provided that said distribution complies with all the provisions of the *Agreement* and is accompanied by:

- a copy of the Agreement,
- a notice relating to the limitation of both the Licensor's warranty and liability,

and that, in the event that only the object code of the *Modified Software is redistributed*, you allows future users *access to the full source code of the Modified Software by indicating how to access it*, it being understood that the additional cost of acquiring the source code shall not exceed the cost of transferring the data.

For further explanation about this free software license, you should read the following links:

- `http://www.cecill.info/licences/Licence_CeCILL_V2-fr.html` (in French)
- `http://www.cecill.info/licences/Licence_CeCILL_V2-en.html` (in English)

2.3 Installation

Remark 1 *To windows' users: please, look at the application note entitled "Using Cygwin to compile and run* FullSWOF_1D*,* FullSWOF_2D *or* SWASHES *under windows".*

When you are in the *FullSWOF_1D* directory, write the following lines:

```
make cleanall
```

```
make
```

For the first tests, you can use the *Exp01* directory, where you will find examples of inputs files:

```
cd Exp01
../bin/FullSWOF_1D
```

2.4 Check for proper functioning

FullSWOF_1D comes with a set of test cases used for benchmarking (see section 5). Each test case has its own directory, which initially contains:

- the analytic solution (file analytic.dat),
- the *FullSWOF_1D* input parameters (Inputs directory)
- the benchmark outcome as computed by the developers (file comp_STANDARD.dat).

Once the software is installed on your computer, it is worth checking its proper functioning. For this, simply run the command `make benchref`. This will first compute all the test cases (the results of the computation will be stored in the Outputs_REFERENCES directories). Then, the differences with the analytic solutions will be computed and, finally, it will be checked if they are differences between your run and the one of the developers. It is expected no such difference will be found: hence, for each test case, you should get the diagnosis "Results are identical.". Otherwise, please, contact the developers at fullswof.contact@listes.univ-orleans.fr.

3 Input and output values

Remark 2 FullSWOF_UI*, a graphic user interface dedicated to* FullSWOF_1D*, is available at* `https://sourcesup.renater.fr/projects/fullswof-ui/`*.*

When launched, *FullSWOF_1D* expects two subdirectories: one for the inputs, one for the outputs. In the following sections, the notation `<x>` stands for the tag corresponding to the x variable, whereas the square brackets [·] give the unit of the variable.

3.1 *Inputs* directory

You can set the values of most of parameters in the *parameters.txt* file, located in the *Inputs* directory (the values we advise to use for overland flow can be found in Section 3.1.11). If a parameter is not set, we either affect a default value or stop the program (depending on the type of parameter), and the code will return either a warning or an error message.

3.1.1 Space and time scales

First, you have to specify the **number of grid cells** `<Nxcell>` (in space) and the **length of the domain** `<L>` [m]. For the time, you should set the values of the **duration of the simulation** `<T>` [s], the **number of times saved** (that is the number of pictures you will save to see the evolution as a movie, `<nbtimes>`) and to complete the time description you have to add the **time step** `<dt> [s]`. Note that this software is based on a fixed time-step method (not on a fixed CFL), which means that your space/time steps must be coherent with the maximum velocity of the flow ($\frac{\text{space step}}{\text{time step}}$ < maximum velocity).

3.1.2 Boundary conditions

For some boundary conditions, you may have to specify the discharge and the water height on the boundary. You must use the following principles:

Imposed discharges `<L_imp_q>` and `<R_imp_q>` [m^2/s] correspond to the discharges at the left and right boundaries, respectively. At the left boundary (*i.e.* $x = 0$), if you want an incoming flow, you have to impose a positive discharge whereas for an outgoing flow you must impose a *negative* discharge. At the opposite boundary (*i.e.* $x = x_{max}$), impose a *negative* discharge for the inflow and a positive discharge for the outflow;

Imposed water heights `<L_imp_h>` and <R_imp_h> [m] correspond to the water heights at the left and right boundaries respectively.

You must choose the **left and right boundary conditions** (`<Lbound>` and `<Lbound>`), see Lucas for more details:

- the case 1 corresponds to the imposed water height condition (based on the modified method of characteristics). For example, this condition can figure the case of a *large* overflowing reservoir at the inflow or a *large* lake at the outflow (*large* enough so that the water flow inside the domain does not affect the water height at the boundary). **Imposed discharge and water height** have to be specified. Depending on the regime (sub/super-critical) and on the boundary (inflow/outflow), the value of the imposed discharge may not be used by *FullSWOF_1D*;
- the case 2 is for the Neumann boundary condition, which means that the normal derivatives of the water height and velocity are null. Physically, this case represents an open boundary;
- in case 3, the value of the discharge is imposed. You must specify the **value of flow discharge** and the **specific water height**, but the latter will be considered only in supercritical cases;
- the case 4 models the wall condition;
- the case 5 stands for the periodic case, in which the outflow on one boundary is considered as the inflow on the other boundary. In this case, at each time step, the incoming flow is set equal to the outgoing flow.

3.1.3 Friction law

You also have to impose the **friction law** `<fric>`

- case 0 to run *FullSWOF_1D* with no friction.
- case 1 to use the Manning law; in this case, the **friction coefficient** `<friccoef>` [$\mathrm{m}^{-1/3}\mathrm{s}$] is the Manning friction coefficient.
- case 2 to choose the Darcy-Weisbach law; for this choice, the **friction coefficient** `<friccoef>` [dimensionless] is the Darcy-Weisbach friction coefficient.

3.1.4 Numerical scheme

The next seven parameters are related to the numerical scheme. You have to choose the:

Numerical flux `<flux>`.

  - case 1: Rusanov flux,
  - case 2: HLL flux,
  - case 3: HLL2 flux, that is another way of programming HLL,
  - case 4: Kinetic flux,
  - case 5: VFRoe flux.

Order of the scheme `<order>`.

  - case 1: order 1,
  - case 2: order 2.

CFL value `<cfl>`. This parameter must be smaller than 1, usually equal to 0.5 to ensure the convergence of the scheme.

Linear reconstruction `<rec>`. This parameter will play a role only for a scheme of order 2. It can be:

  - case 1: MUSCL,
  - case 2: ENO,
  - case 3: a modified ENO.

`<armortENO>`. This parameter must be specified when you are using linear reconstructions ENO or modified ENO. Its value must be between 0 and 1 (usually set to 0.25). If you take 0, it will be equivalent to choose the MUSCL reconstruction; if `<armortENO>` is set equal to 1, the reconstruction will be exactly ENO or modified ENO, depending on your previous choice.

`<modifENO>`. For the modified ENO reconstruction, you have to set the `<modifENO>` parameter between 0 and 1, usually taken equal to 0.9.

Slope limiter `<lim>`. This parameter will play a role only for a scheme of order 2. You can use either:

- case 1: the classical minmod slope limiter.
- case 2: the more complicated expression of Van Albada.
- case 3: Van Leer's reconstruction.

3.1.5 Topography

The type of input is defined by `<topo>` which can take the following values:

- 1 to load the topography from a file. You have to enter the **topography** using an ASCII file you previously generated in the format "x z". Write the **name of your topography file** in `<topo_NF>`. This file must be in the *Inputs* directory.
- 2 to use a flat topography with $z = 0$.
- 3 to have the Thacker's parabola defined by $z = \frac{1}{2}\left(\left(x - \frac{L}{2}\right)^2 - 1\right)$, where L is the length of the domain. This is used for the benchmark shown in section 5.2.2 (Figure 11) and detailed in [Delestre et al., 2013, § 4.2.1].
- 4 to choose the bump defined by $z = \max\left(0, 0.2 - 0.05\,(x-10)^2\right)$ (centered on $x = 10$ m), used for several benchmarks, see section 5.1.1 and details in [Delestre et al., 2013, § 3.1].

Remark 3 FullSWOF_1D *considers values (such as h, u, z) constant on a given cell, and the constant is given in the middle of the cell, see figure 1.*

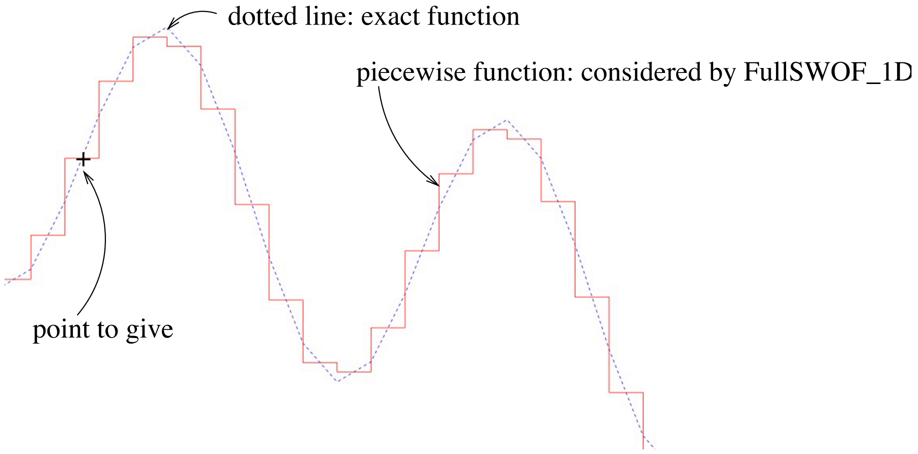

Figure 1: Piecewise approximation of 1D curves in *FullSWOF_1D*

3.1.6 Initial water height and velocity

You can impose the initial water height and velocity using `<hu_init>`:

- case 1 to load the initialization of the variables h [m] and u [m/s] from a file. This file must be in ASCII and follow the format "x h u". The **name of the file** `<huv_NF>` should be specified and this file must be in the *Inputs* directory.
- case 2 to have $h = 0$ m and $u = 0$ m/s.

- case 3 to have a wet dam ($h = 0.005$ m and $u = 0$ m/s in the left half of the domain, $h = 0.001$ m and $u = 0$ m/s in the right half of the domain). You should choose the flat topography for this case, see details in [Delestre et al., 2013, § 4.1.1].
- case 4 to have a dry dam ($h = 0.005$ m and $u = 0$ m/s in the left half of the domain, $h = 0$ m and $u = 0$ m/s in the right half of the domain). You should choose the flat topography for this case, to simulate the dry dam break shown as a benchmark in section 5.2.1 (Figure 10), and detailed in [Delestre et al., 2013, § 4.1.2].
- case 5 to have the Dressler dam break configuration, see [Delestre et al., 2013, § 4.1.3], namely ($h = 6$ m and $u = 0$ m/s in the left half of the domain, $h = 0$ m and $u = 0$ m/s in the right half of the domain). You should choose the flat topography for this case.
- case 6 to have a planar surface (with a velocity $u = 0$ m/s) in the parabola defined by $z = \max\left(0, 0.2 - 0.05\,(x - 10)^2\right)$ (centered on $x = 10$ m). You should choose the parabolic topography for this case, used as a benchmark in section 5.2.2 (Figure 11) and detailed in [Delestre et al., 2013, § 4.2.1].

3.1.7 Rain

For the **rain** `<rain>`, you can use:

- case 0: **No Rain**.
- case 1: An ASCII file. The **name of your rain file** `<rain_NF>` should be set and this file must be in the *Inputs* directory.
  The file must contain two columns:
  - the first one represents the time [s].
  - the second one the rain intensity [m/s].

  Each column must contain at least one value.
  In the first column, the first value must be 0 (initial time). If the file contains one value by column (namely 0 in the first column and any value in the second column), the rain intensity is constant during the run.
  If the file contains several lines with different values, that means the rain intensity will be changed during the simulation. For example, let us consider three values for the time $(0, t_1, t_2)$ and the corresponding three values for the rain intensity (a_0, a_1, a_2) (written into the second column). Let us consider $Rain(t)$ the intensity of the rain at the time t, we will get the values:

$$Rain(t) = \begin{cases} a_0 & \text{for } 0 \leq t < t_1, \\ a_1 & \text{for } t_1 \leq t < t_2, \\ a_2 & \text{for } t_2 \leq t. \end{cases}$$

  Be careful: the intensity may not change exactly at $t = t_1$ but at $t > t_1$. Indeed, t_1 might not be reached exactly. We can have $t < t_1$ for a given computing time, and the next computing $t + dt$ may be greater than t_1. So, $t = t_1$ may not exist and the change in rain intensity will take place at $t + dt > t_1$.
- case 2: the rain intensity is constant during the run. The value is equals to 10^{-5} m/s that is equivalent to 36 mm/h.

Remark 4 *You can create another case from, for example, the function in the rain_generated.cpp file (see the directory: Sources/librain_infiltration/).*

3.1.8 Infiltration

In this version of *FullSWOF_1D*, a modified bi-layer Green-Ampt is the only infiltration model (see Esteves et al. [2000]). So, you have the choice between:

- case 0 to run *FullSWOF_1D* without infiltration.

- case 1: a modified bi-layer Green-Ampt infiltration model.
  For each infiltration parameter you can either initialize it from a file or give constant value.
  If you choose to parameterize the infiltration using files, you can include a spatial variability of the infiltration parameters. For this, you have to provide ASCII files in the format "x value" and these files must be in the *Inputs* directory.
  The modified bi-layer Green-Ampt infiltration model assumes the soil to be represented by two layers:

  The first layer (top of the surface) represents a crust of **thickness** `<zcrust>` [m] and **hydraulic conductivity** `<Kc>` [m/s].
  If you want to initialize these two parameters from a file, write the **name of the thickness file** in `<zcrust_NF>` and/or the **name of your hydraulic conductivity file** in `<Kc_NF>`.

  The second layer has a **saturated hydraulic conductivity** `<Ks>` [m/s].
  If you want to initialize this parameter from a file, write the **name of your saturated hydraulic conductivity file** in `<Ks_NF>`.

  The other parameters (common values for the two layers) are:

  - the **saturated moisture content** θ_s `<dtheta>` [dimensionless] between 0 (dry) and 1 (fully saturated).
    If you want to initialize this parameter from a file, write the **name of your saturated moisture content file** in `<dtheta_NF>`.
  - the **load pressure** ψ `<Psi>` [m].
    If you want to initialize this parameter from a file, write the **name of your load pressure file** in `<Psi_NF>`.
  - the **maximum infiltration rate** i_{max} `<imax>` [m/s]. In the standard Green-Ampt model, at $t = 0$ s, if the cumulative volume of water infiltrated is zero, the infiltration capacity is infinite. To avoid this phenomena, the Green-Ampt model has been modified to be able to impose a maximum infiltration rate i_{max}.
    If you want to initialize this parameter from a file, write the **name of maximum infiltration rate file** in `<imax_NF>`.

3.1.9 Name of output directory

The default name of the output directory is *Outputs*. However, you can add a **suffix** to this name (`<suffix_o>`). This is especially useful if you are running several tests.

3.1.10 Comments

You can also add comments after the input value of each parameter. For example:
`Timestep (in seconds) <dt>:: 0.001 # try 0.01 next time`.

3.1.11 Advised values

For overland flow, several tests have been performed, in particular on the numerical scheme Delestre [2010]. Consequently, we advise the user to choose the following numerical parameters:

- `Numerical flux (1=Rus 2=HLL 3=HLL2 4=Kin 5=VFRoe) <flux>::  2`
- `Order of the scheme <order>::  2`
- `CFL value <cfl>::  0.5`
- `Reconstruction (1=MUSCL 2=ENO 3=ENOmod) <rec>::  1`
- `Limiter (1=Minmod 2=VanAlbada 3=VanLeer) <lim>::1`

If you choose the ENO or the modified ENO reconstruction, you should use:

- `AmortENO <amortENO>::  0.25`
- `ModifENO <modifENO>::  0.9`

3.2 *Outputs* directory

The results are saved in the *Outputs* directory. When starting the program, the following files are saved:

parameters.dat contains the parameters used by *FullSWOF_1D*, under the same format as the input file. The value of an unnecessary parameter is left empty, and some parameters can be set to their default value if the user did not fill them in the `parameters.txt` file.

huz_initial.dat contains the initial conditions (water height, velocity, topography, discharge and free surface).

During the computation, one file is modified in order to save several time steps:

huz_evolution.dat contains the evolution (in time) of the main variables on each cell (water height, velocity, topography, water discharge, free surface and Froude number). There are `<nbtimes>` time steps saved.
If you set `<nbtimes>` to 0, *FullSWOF_1D* will not create the **huz_evolution.dat** file.

If the software did not stop before (meaning the CFL maximum value has been reached, which implies that you must decrease the time step or increase the space step), the computation is done until the final time, and two other files are created:

huz_final.dat contains the values of the main variables on each cell (water height, velocity, topography, water discharge, free surface and Froude number) at the final time.

results.dat contains other values you may need, namely computation times and the mean Froude number [dimensionless] in space at the final time. It also contains a mass balance to check the scheme conservation at the final time. More precisely, you get:

- Infiltrated volume [m^2] is the total infiltrated water volume cumulated during the simulation.
- Stream volume [m^2] is the total water volume staying at the final time above topography.
- Complete volume (Inf+Stream) [m^2] is the sum of infiltrated volume and stream volume.
- Volume of the rain [m^2] is the total rain volume cumulated during the simulation.
- Outflow volume [m^2] is the sum of the cumulated outflow volume at all boundaries.

Finally, if you ran *FullSWOF_1D* in debug mode (see Section 4.1), two other files are saved:

boundary_flux.dat contains the left and right fluxes at each time step, as well as the cumulative fluxes (in time). Recall that the second order is computed thanks to a prediction-correction method. Then, if you choose to run the code at the second order, the values of the fluxes that are printed in this file are the averages of the two fluxes used in the numerical scheme.

check_vol.dat contains the cumulated flux [m^2] at each time step. The first column is the time and the others are the cumulated fluxes for: overland flow volume (Vol_of_tot), infiltrated volume (Vol_inf_tot), rain (Vol_rain_tot). The last column contains the difference between the cumulated fluxes of the right and left boundaries (bound_flux_tot). Here again, if the scheme is at the second order, we take into account the average of the two numerical fluxes, for each boundary.

4 For developers

4.1 Debugging

If you make some changes in *FullSWOF_1D*, you will need to debug your code. The default configuration is the release mode. To change it into debug mode, you must set the `DEBUG` value to `yes` in the **make_config** file. In that case, additional output files will be created to help you in your tests (see Section 3.2).

4.2 Check for performances

FullSWOF_1D comes with a set of test cases used for benchmarking (see section 5). Each test case has its own directory, which should contains:

- the analytic solution (file analytic.dat),
- the *FullSWOF_1D* input parameters (Inputs directory)
- the benchmark outcome as computed by the developers (file comp_STANDARD.dat),
- the benchmark outcome as computed on your computer after installation (file comp_REFERENCES.dat) — see section 2.4.

After you modify *FullSWOF_1D* code, you may want to check if the performances are the same, have been degraded (a bug?), or have been improved, compared to the capabilities of the software after installation. For this, simply run the command `make bench`. This will first compute all the test cases (the results of the computation will be stored in the Outputs directories). Then, the differences with the analytic solutions will be computed and, finally it will be checked if they are differences between your current run and the run upon installation.

Remark 5 *If, at some point, you do not want to compare the performances with the run upon installation anymore, you should delete the files comp_REFERENCES.dat and the directories Outputs_REFERENCES. Then, run* `make benchref`. *The newly created comp_REFERENCES.dat file and Outputs_REFERENCES directories will be then used as references when using* `make bench`.

If there is no difference, you should get the diagnosis “Results are identical.” for each test case.

If differences are reported for one or more test cases, you may first want to look at the files diff_REF_USER.dat of the relevant test cases). After the header (lines starting with the # character), are listed a set of values:

The first column identifies the variable. DhSI stands for “Differences in water height in international system unit” (meter in the case of water height) while Dh% stands for “Differences in water height expressed as a percentage” (taking as a reference the case Outputs_REFERENCES). Differences in velocity starts with “Du” and differences in water flux starts with “Dq”.

The second column identifies the statistics for each variable. First is given the number of differences that cannot be computed (“nbdiff==NaN”), probably because it involves a division by zero. Then the number of differences equal to zero, larger than zero and smaller than zero. Following are the minimum, maximum, mean and median values, and finally the L^1, L^2 and L^∞ norms.

The third column shows the absolute differences.

The fourth column shows the relative differences (as a percentage), taking as a reference the case Outputs_REFERENCES.

So, you should be able to identify if the differences concern numerous values (or just a few), and if they are about water height, velocity or flux. At this stage in the diagnosis, it may be time to dig further by comparing the content of the Outputs and Outputs_REFERENCES directories:

- First, check for uncalled-for differences in the parameter.dat files.
- Check that the initial data are identical (files huz_initial.dat).
- Locate the differences in the final results by comparing the files huz_final.dat.
- Eventually, compare the time evolution of these differences (files huz_evolution.dat).

Based on this, you should be able to evaluate if the software results are as accurate, more accurate or less accurate than before. Since each test case addresses specific flow conditions (see section 5), you should be able to build a rationale about which part of the simulation has been impacted, and, if required, which part of the source code is involved.

4.3 Doxygen

You may wish to add some functionalities to *FullSWOF_1D* to suits your needs. Always comment the files, at the beginning of the file, using Doxygen syntax (`http://www.doxygen.org/`). Then, you will be able to create the Doxygen documentation of the whole code.

HTML documentation. In order to generate the Doxygen html file, the **Doxygen_config_file_html** file is saved in the *doc* directory. To run Doxygen, from the *FullSWOF_1D* directory, use the command:

```
doxygen doc/Doxygen_config_file_html
```

Warning: Graphviz (`http://www.graphviz.org/`) must be in your PATH to generate HTML diagrams. If not, change the HAVE_DOT parameter of the Doxygen_config_file_html file. In the *doc/html/* directory, **index.html** is created.

PDF documentation. To generate the Doxygen PDF file (using LaTeX), you must use the **Doxygen_config_file_latex** file and compile the .tex file:

```
doxygen doc/Doxygen_config_file_latex
cd doc/latex
make
```

In the *doc/latex* directory, **refman.pdf** is created.

5 Validation

This software has been validated using several analytic solutions and benchmarks from the literature, gathered in Delestre et al. [2013] and in SWASHES `https://sourcesup.renater.fr/projects/swashes/`. Some of them are already configured in the **Benchmarks** directory.

In this documentation, we recall the main characteristics of these tests, and give the results of *FullSWOF_1D*.

5.1 Steady-state solutions

In this section, we focus on a family of steady-state solutions, that are solutions satisfying:

$$\partial_t h = \partial_t u = 0.$$

5.1.1 Bumps

Here we present a series of steady-state cases (see [Delestre et al., 2013, § 3.1]) with a flat topography at the boundaries, no rain, no friction and no diffusion. Initial conditions satisfy the hydrostatic equilibrium

$$h + z = Cst \text{ and } q = 0 \text{ m}^2/\text{s}. \tag{1}$$

These solutions test the preservation of steady-states and the boundary conditions treatment.

In the following cases, we choose a domain of length $L = 25$ m with a topography given by:

$$z(x) = \begin{cases} 0.2 - 0.05(x-10)^2 & \text{if } 8 \text{ m} < x < 12 \text{ m}, \\ 0 & \text{else.} \end{cases}$$

Emerged bump at rest In the case of a lake at rest with an emerged bump, the water height is smaller than the topography in order to have emergence of some parts of the bump. In such a configuration, starting from the steady-state, the velocity must be null and the water surface should stay flat, see Figure 2.

Fluvial bump After testing a steady-state at rest, the user can increase the difficulty with dynamical steady-states. In the case of a subcritical flow, the water height is constant when the topography is constant, decreases (respectively increases) when the bed slope increases (resp. decreases). The water height reaches its minimum at the top of the bump, see Figure 3.

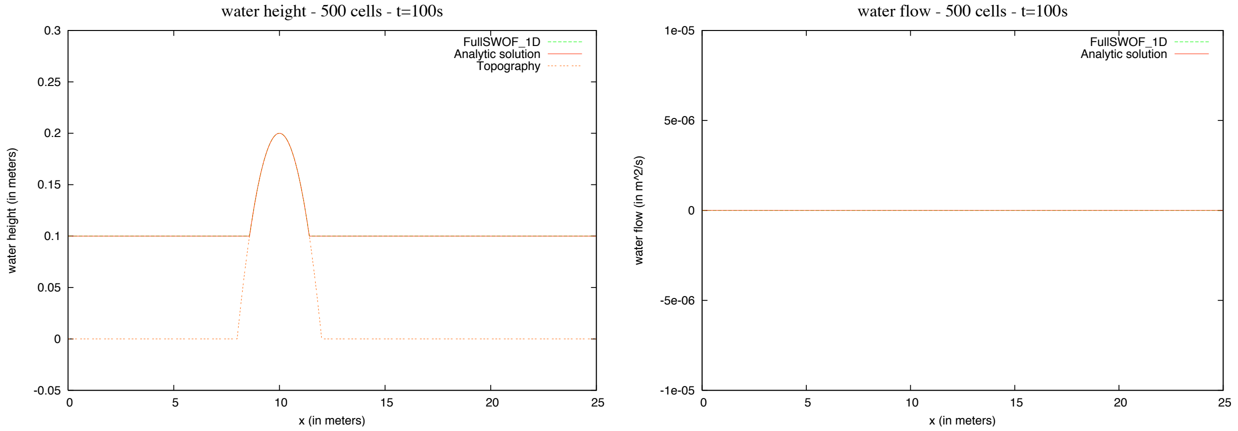

Figure 2: Emerged bump at rest

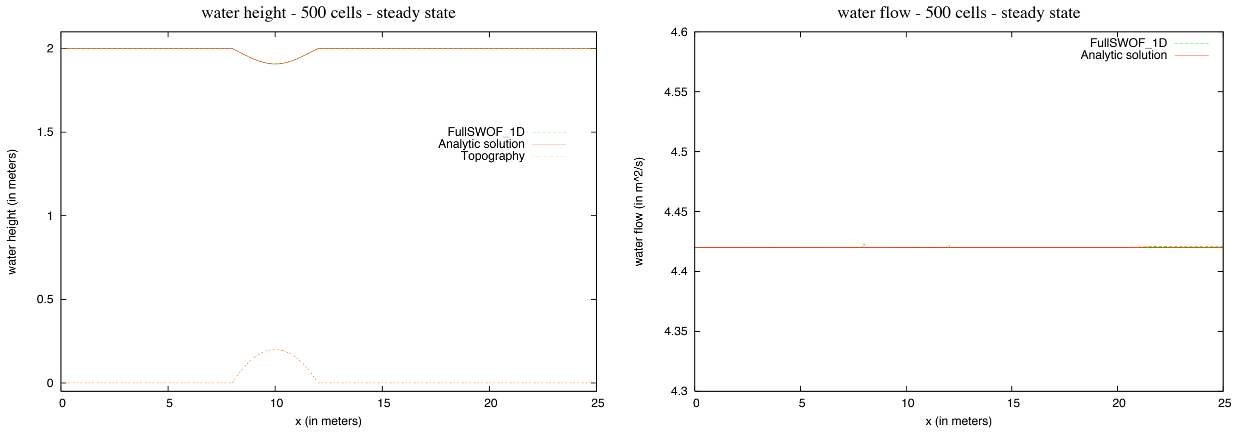

Figure 3: Fluvial bump

Transcritical bump with shock When we consider a transcritical flow with shock, the flow is fluvial upstream, becomes supercritical at the top of the bump but it becomes again fluvial after a hydraulic jump, see Figure 4.

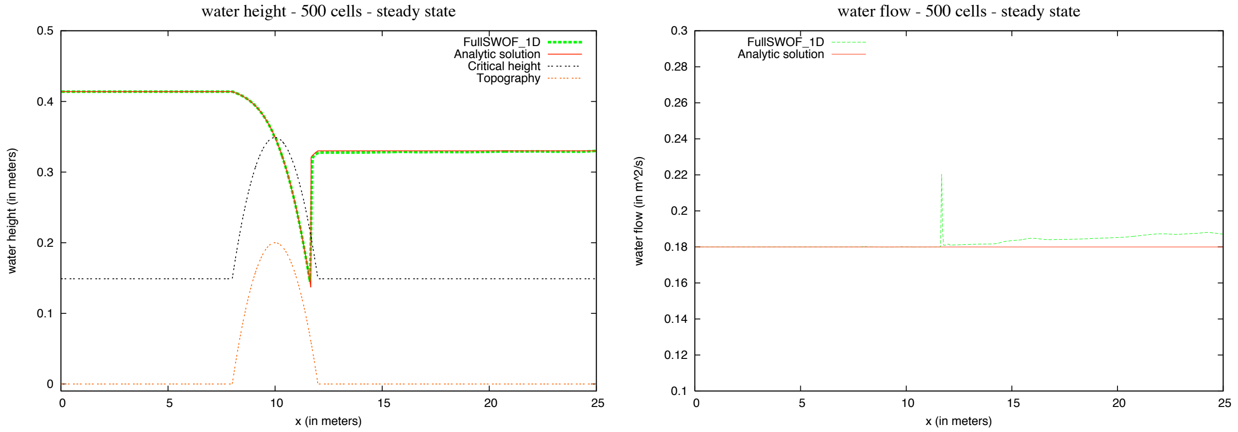

Figure 4: Transcritical bump with shock

5.1.2 Mac Donald's type 1D solutions

We give here some steady-state solutions of the Shallow-Water system with varying topography and friction term. The water height profile and the discharge are given and we compute the corresponding topographies which are used as input parameters for *FullSWOF_1D*.

The solutions explained in this section are more intricate than the ones of the previous section, as the topography can vary near the boundary. Consequently they give a better validation of the boundary conditions. If we have friction at the bottom, the following solutions can prove if the friction terms are coded in order to satisfy the steady-states.

Torrential Mac Donald like test with Darcy-Weisbach friction coefficient We consider a 1000 m long channel with a constant discharge on the whole domain. The flow is supercritical both at inflow and at outflow. Initially, the channel is dry and we choose Darcy-Weisbach friction law. In that case, the flow stays supercritical, see Figure 5 and details in [Delestre et al., 2013, § 3.2.1].

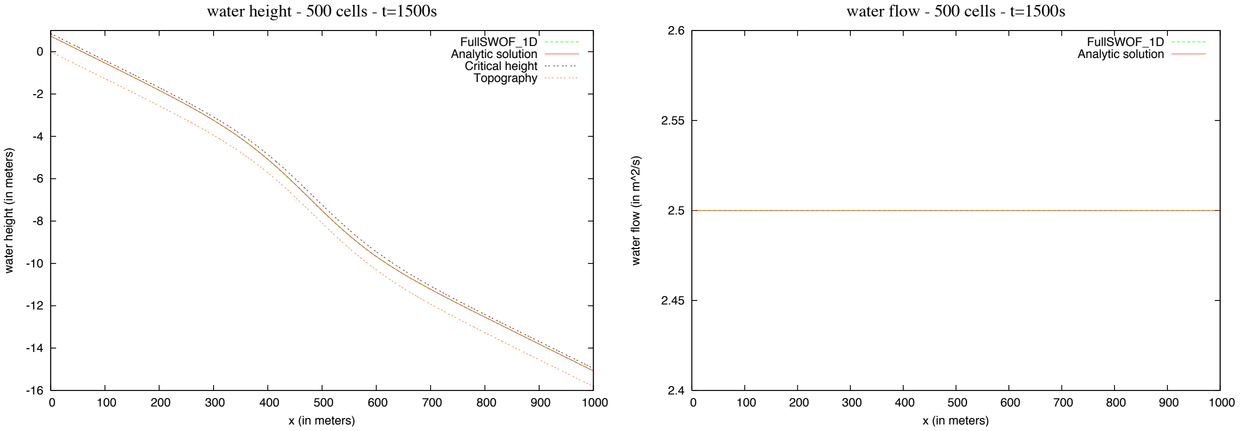

Figure 5: Torrential Mac Donald like test with Darcy-Weisbach friction coefficient

Fluvial/torrential Mac Donald like test with Manning friction coefficient The channel is 1000 m long and the discharge at equilibrium is $q = 2$ m^2/s. The flow is subcritical upstream and supercritical downstream. As initial conditions, we consider a dry channel with Manning's law. Thus we get a transcritical flow (from fluvial to torrential via a transonic point), see Figure 6 and details in [Delestre et al., 2013, § 3.2.1].

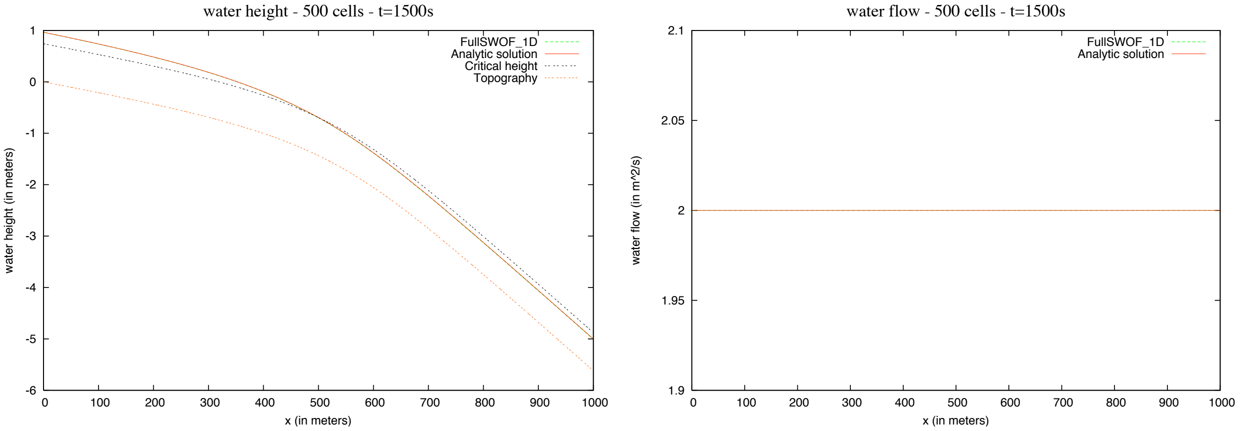

Figure 6: Fluvial/torrential Mac Donald like test with Manning friction coefficient

Torrential/fluvial Mac Donald like test with Darcy-Weisbach friction coefficient As in the previous cases, the domain is 1000 m long and the discharge is $q = 2$ m^2/s. The boundary conditions are a torrential inflow and a fluvial outflow. At time $t = 0$ s, the channel is initially dry and we use Darcy-Weisbach friction law. The steady-state solution is supercritical upstream and becomes subcritical through a hydraulic jump located at $x = 500$ m, see Figure 7 and details in [Delestre et al., 2013, § 3.2.1].

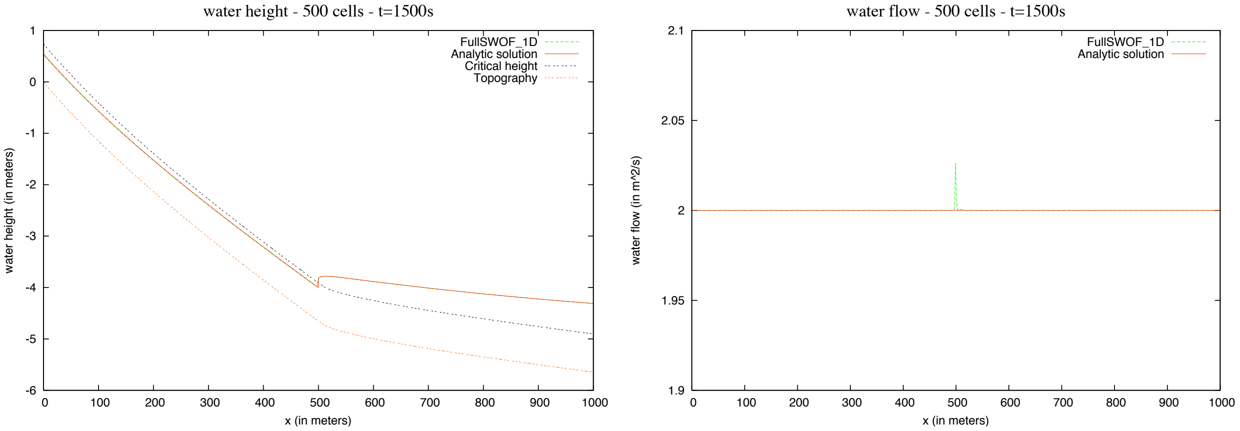

Figure 7: Torrential/fluvial Mac Donald like test with Darcy-Weisbach friction coefficient

Fluvial Mac Donald like test with Manning friction coefficient The length of the channel is 100 m and the discharge at steady-states is $q = 2$ m^2/s. The flow is fluvial both upstream and downstream. To have a case including two kinds of flow (subcritical and supercritical) and two kinds of transition (transonic and shock), we consider a channel filled with water and with Manning's friction law. The inflow is subcritical, becomes supercritical via a sonic point, and, through a shock (located at $x = 200/3 \approx 66.67$ m), becomes subcritical again, see Figure 8 and details in [Delestre et al., 2013, § 3.2.2].

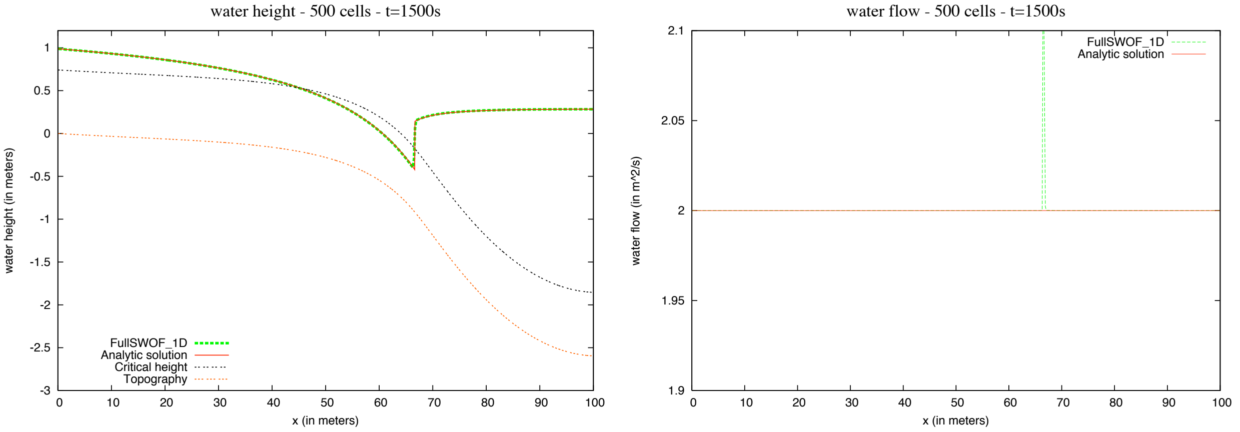

Figure 8: (Short) Fluvial Mac Donald like test with Manning friction coefficient

Torrential Mac Donald test with rain and Darcy-Weisbach friction coefficient The channel length remains unchanged (1000 m), but, as the flow is supercritical, we consider constant discharge ($q = 2.5$ m^2/s) and water height at the inflow and a free outflow. At initial time, the channel is dry and we choose Darcy-Weisbach friction law with $f = 0.065$.

In this benchmark, there is no rain until 1500 s and after the rain intensity is 0.001 m/s until the end (see Figure 9 and [Delestre et al., 2013, § 3.3.1]).

5.2 Transitory solutions

In the previous section, we gave steady-state solutions of increasing difficulties. These solutions can be used to check if the numerical methods are able to keep/catch steady-state flows. But even if the initial condition differs from the expected steady-state, we do not have information about the transitory behavior. Thus, in this section, we give some transitory solutions that may improve the validation of the numerical methods. Moreover, as these cases have wet/dry transitions, one can check the ability of the schemes to capture the evolution of these fronts (*e.g.* some methods may fail and

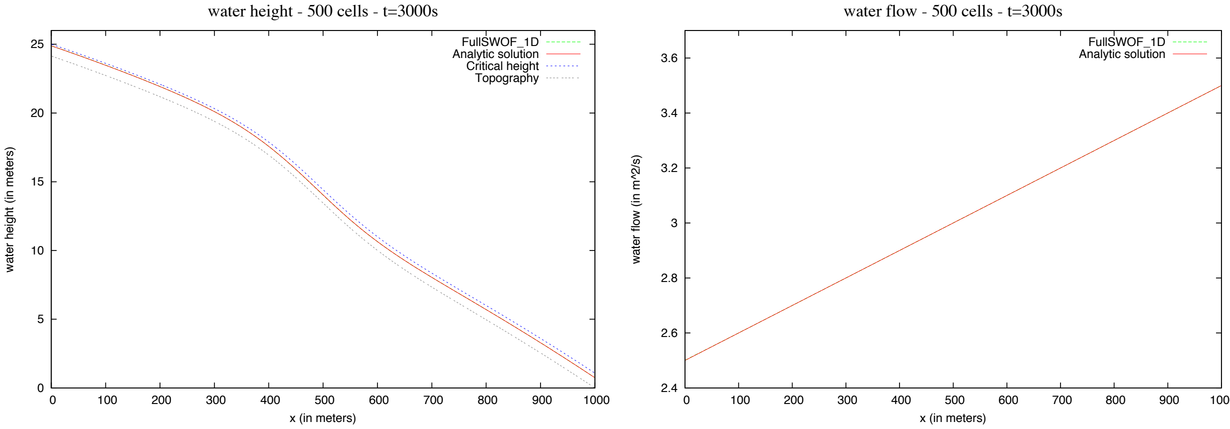

Figure 9: Torrential MacDonald: Rain test with Darcy-Weisbach friction coefficient

give negative water height). At last, we give a periodic transitory solution in order to check whether the schemes are numerically diffusive or not.

5.2.1 Dam break: case on a dry domain

In this section, we are interested in a dam break solution on a flat topography, see [Delestre et al., 2013, § 4.1.2]. The initial condition for this configuration is the following Riemann problem

$$h(x) = \begin{cases} h_l & \text{for } 0 \text{ m} \leq x \leq x_0, \\ h_r = 0 & \text{for } x_0 < x \leq L, \end{cases}$$

with $h_l \geq h_r$ and $u(x) = 0$ m/s.

At time $t \geq 0$, we have a left-going rarefaction wave (or a part of parabola) that reduces the initial depth h_l into h_m, and a right-going shock that increases the initial height h_r into h_m, see Figure 10. This solution tests whether the code gives the location of the moving shock properly.

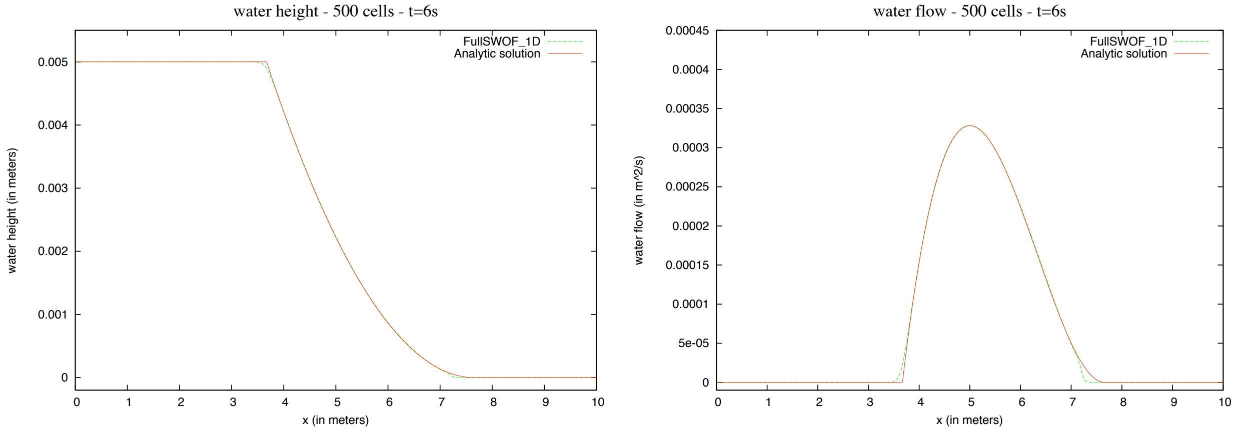

Figure 10: Dam break on a dry domain

5.2.2 Oscillations: Thacker test case

In this section, we are interested in Thacker's solution. It is a periodic solution (without friction) where the topography is a parabolic bowl and the free surface remains planar in time, see Figure 11 and [Delestre et al., 2013, § 4.2.1]. This is an analytic solution with a variable slope (in space) for which the wet/dry transitions are moving. It also tests the ability of schemes to simulate flows with comings and goings and, as the water height is periodic in time, the numerical diffusion of the scheme.

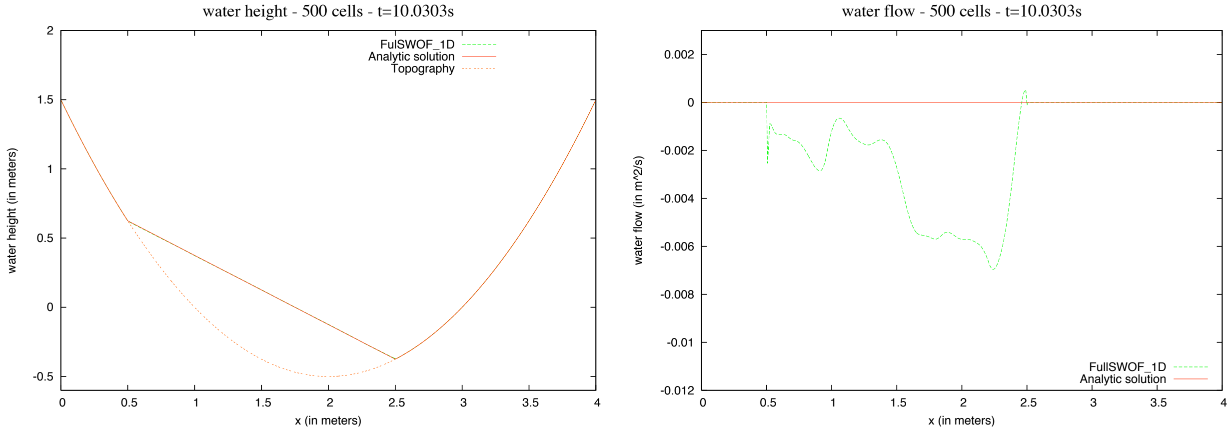

Figure 11: Thacker test case

References

Olivier Delestre. Ecriture d'un code C++ pour la simulation en hydrologie. Master's thesis, Université d'Orléans, Orléans, France, September 2008. URL `http://dumas.ccsd.cnrs.fr/dumas-00446163/fr/`.

Olivier Delestre. *Simulation du ruissellement d'eau de pluie sur des surfaces agricoles*. PhD thesis, Université d'Orléans, Orléans, France, July 2010. URL `http://tel.archives-ouvertes.fr/tel-00531377`.

Olivier Delestre and François James. Simulation of rainfall events and overland flow. In *Proceedings of X International Conference Zaragoza-Pau on Applied Mathematics and Statistics, Jaca, Spain*, Monografías Matemáticas García de Galdeano, 2009. URL `http://hal.archives-ouvertes.fr/hal-00426694/fr`.

Olivier Delestre, Stéphane Cordier, François James, and Frédéric Darboux. Simulation of rain-water overland-flow. In E. Tadmor, J.-G. Liu, and A. Tzavaras, editors, *Proceedings of the* 12th *International Conference on Hyperbolic Problems*, volume 67 of *Proceedings of Symposia in Applied Mathematics*, pages 537–546, University of Maryland, College Park (USA), 2009. Amer. Math. Soc.

Olivier Delestre, Carine Lucas, Pierre-Antoine Ksinant, Frédéric Darboux, Christian Laguerre, Thi Ngoc Tuoi Vo, François James, and Stéphane Cordier. SWASHES: a compilation of shallow water analytic solutions for hydraulic and environmental studies. *International Journal for Numerical Methods in Fluids*, 72(3):269–300, 2013. ISSN 1097-0363. doi: 10.1002/fld.3741. URL `http://hal.archives-ouvertes.fr/hal-00628246`.

Olivier Delestre, Frédéric Darboux, Francois James, Carine Lucas, Christian Laguerre, and Stephane Cordier. FullSWOF: A free software package for the simulation of shallow water flows. 30 pages, January 2014. URL `http://hal.archives-ouvertes.fr/hal-00932234`.

Michel Esteves, Xavier Faucher, Sylvie Galle, and Michel Vauclin. Overland flow and infiltration modelling for small plots during unsteady rain: numerical results versus observed values. *Journal of Hydrology*, 228:265–282, 2000.

Carine Lucas. How to choose boundary conditions? Study of the various cases. Application note. URL `https://sourcesup.renater.fr/docman/view.php/877/3962/BoundaryConditions.pdf`.