

Documentation of *FullSWOF_1D* v1.00.00 (2012-04-25)

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

2012-04-04

Contents

1	Presentation of the <i>FullSWOF_1D</i> software	1
2	Software distribution	2
2.1	How to download <i>FullSWOF_1D</i> ?	2
2.2	License	2
2.3	Installation	2
2.4	Check for proper functioning	3
3	Input and output values	3
3.1	<i>Inputs</i> directory	3
3.1.1	Space and time scales	3
3.1.2	Boundary conditions	3
3.1.3	Friction law	4
3.1.4	Numerical scheme	4
3.1.5	Topography	5
3.1.6	Initial water height and velocity	5
3.1.7	Name of output directory	6
3.1.8	Comments	6
3.1.9	Advised values	6
3.2	<i>Outputs</i> directory	6
4	For developers	7
4.1	Debugging	7
4.2	Check for performances	7
4.3	Doxygen	8
5	Validation	8
5.1	Steady state solutions	8
5.1.1	Bumps	8
5.1.2	Mac Donald's type 1D solutions	9
5.2	Transitory solutions	11
5.2.1	Dam break: case on a wet domain	12
5.2.2	Oscillations: Thacker test case	12

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 ...). For explanations concerning the numerical schemes and approximations, the reader is referred to [3], [4] and [5]. For a precise description of the structure of the software in several classes, see the Doxygen file (refman.pdf in the *doc* directory) and [2]. 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 developer (Carine LUCAS, carine.lucas@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
- http://www.cecill.info/licences/Licence_CeCILL_V2-en.html

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 fullswof.contact@listes.univ-orleans.fr.

3 Input and output values

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 3.1.9).

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>`, integer greater or equal to 2) 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 ($\frac{\text{space step}}{\text{time step}} < \text{maximum velocity}$).

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 `<Rbound>`), see [9] 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, we impose the incoming flow to be 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>` [$\text{m}^{-1/3}\text{s}$] is Manning friction coefficient.
- case 2 to choose Darcy-Weisbach law; for this choice, the **friction coefficient** `<friccoef>` [dimensionless] is 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 can 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 “ $x z$ ”. Do not forget to 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 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, see [10, 3].
- 4 to choose the bump defined by $z = \max \left(0, 0.2 - 0.05 (x - 10)^2 \right)$ (centered on $x = 10$ m), see [8, 3].

Remark 2 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.

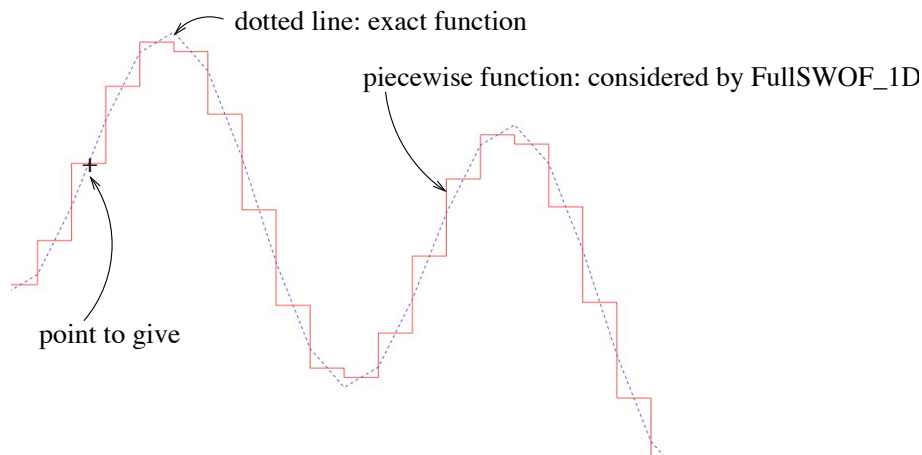


Figure 1: Piecewise approximation of 1D curves in *FullSWOF_1D*

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 initialisation 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 [1, 3].
- 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, see [1, 8].
- case 5 to have the Dressler dam break configuration used in [7, 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, see [10, 3].

3.1.7 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.

3.1.8 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.

3.1.9 Advised values

For overland flow, several tests have been performed, in particular on the numerical scheme [3]. 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=ENMod) <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 form as the input 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 (water height, velocity, topography, water discharge, free surface and Froude number). The number of time steps to be saved is given in the input parameter 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 (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 (in space) at the final time.

Finally, if you ran *FullSWOF_1D* in debug mode (see Section 4.1), another file is saved:

boundary_flux.dat contains the left and right fluxes at each time step, as well as the cumulative fluxes (in time).

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, an additional file 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 3 *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 L1, L2 and Linfinity 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 address 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 (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 L^AT_EX (pdf) file, 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 [6] 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 is solutions that satisfy:

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

5.1.1 Bumps

Here we present a series of steady state cases (see [8]) with a flat topography at the boundaries, no rain, no friction and no diffusion. Initial conditions satisfy the hydrostatic equilibrium

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

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:

$$z(x) = \begin{cases} 0.2 - 0.005(x - 10)^2 & \text{if } 8 \text{ m} < x < 12 \text{ m,} \\ 0 & \text{else.} \end{cases}$$

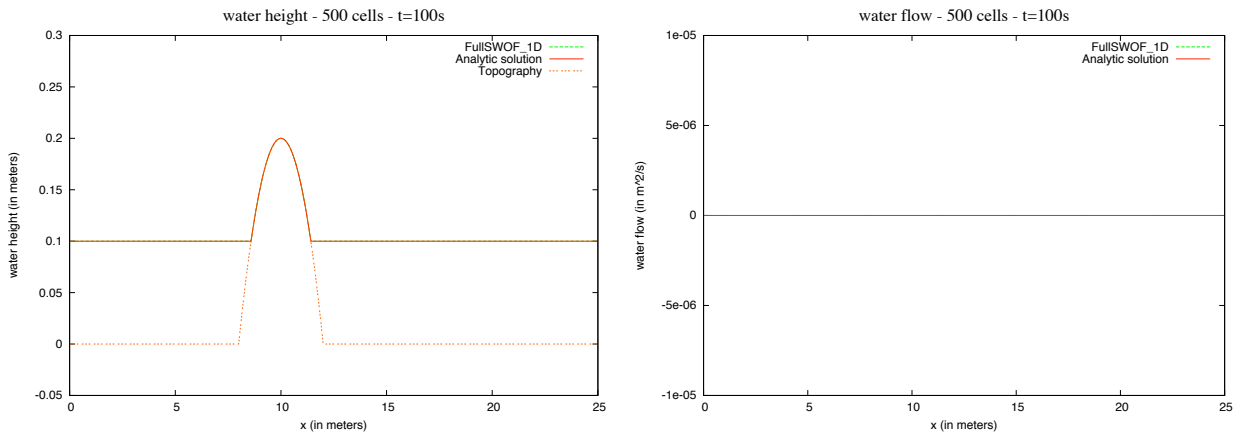


Figure 2: Emerged bump at rest

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.

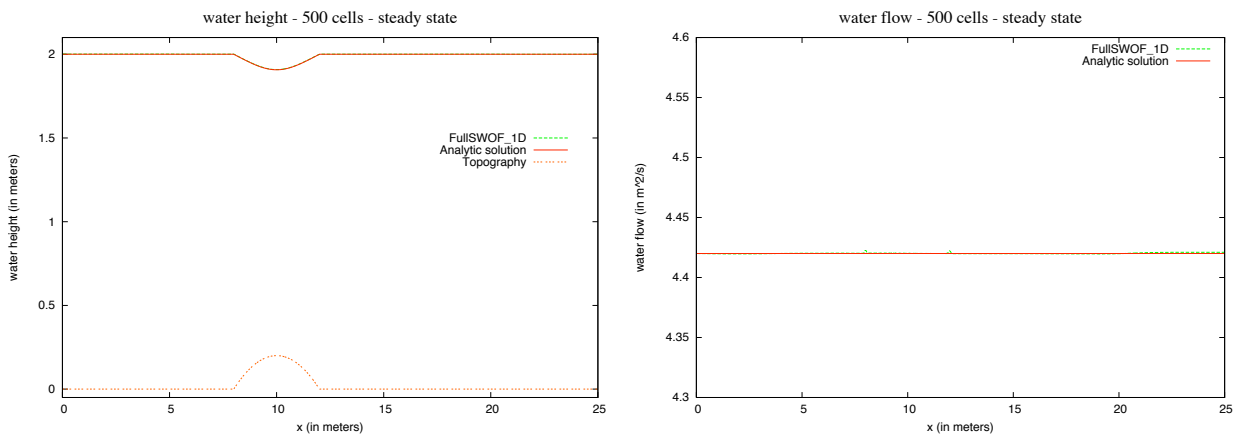


Figure 3: Fluvial bump

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.

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.

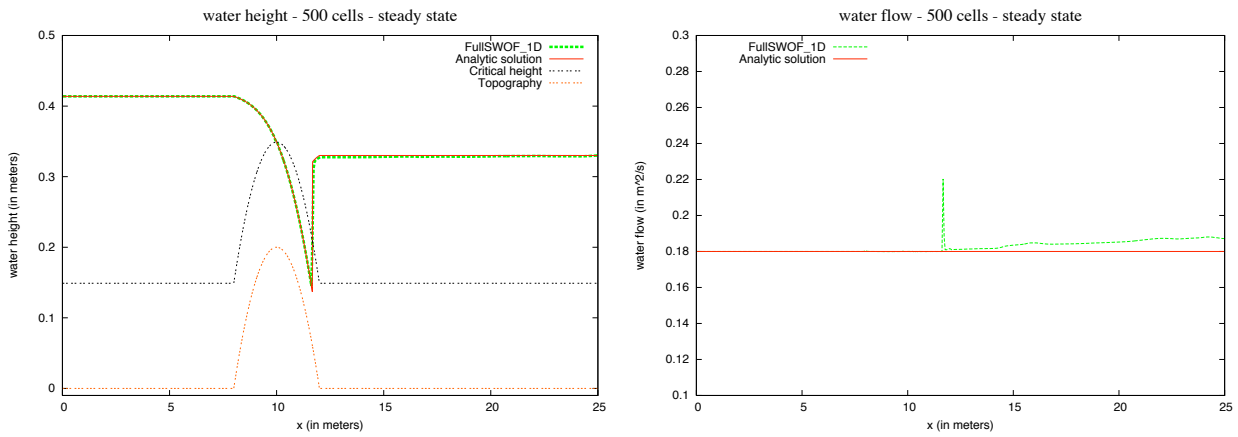


Figure 4: Transcritical bump with shock

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's friction law. In that case, the flow stays supercritical, see Figure 5.

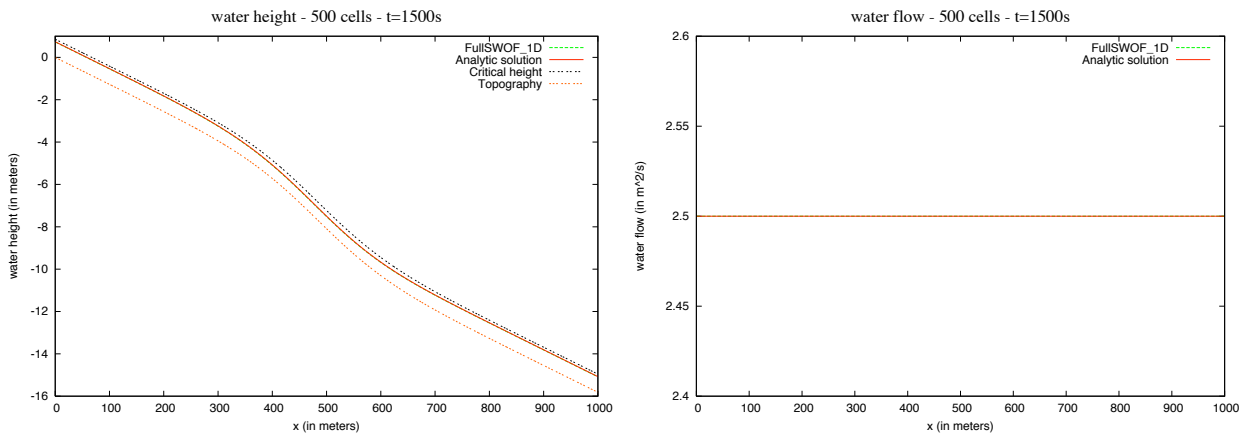


Figure 5: Torrential Mac Donald like test with Darcy-Weisbach friction coefficient

Fluvial/torrential Mac Donald like test with Manning friction coefficient The channel is 1000 m long and the discharge at equilibrium is $q = 2 \text{ m}^2/\text{s}$. 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.

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 $q = 2 \text{ m}^2/\text{s}$. The boundary conditions are a torrential inflow and a fluvial outflow. At time $t = 0 \text{ s}$, the channel is initially dry and we use Darcy-Weisbach's friction law. The steady state solution is supercritical upstream and becomes subcritical through a hydraulic jump located at $x = 500 \text{ m}$, see Figure 7.

Fluvial Mac Donald like test with Manning friction coefficient The length of the channel is 100 m and the discharge at steady states is $q = 2 \text{ m}^2/\text{s}$. 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 $x = 200/3 \approx 66.67 \text{ m}$), becomes subcritical again, see Figure 8.

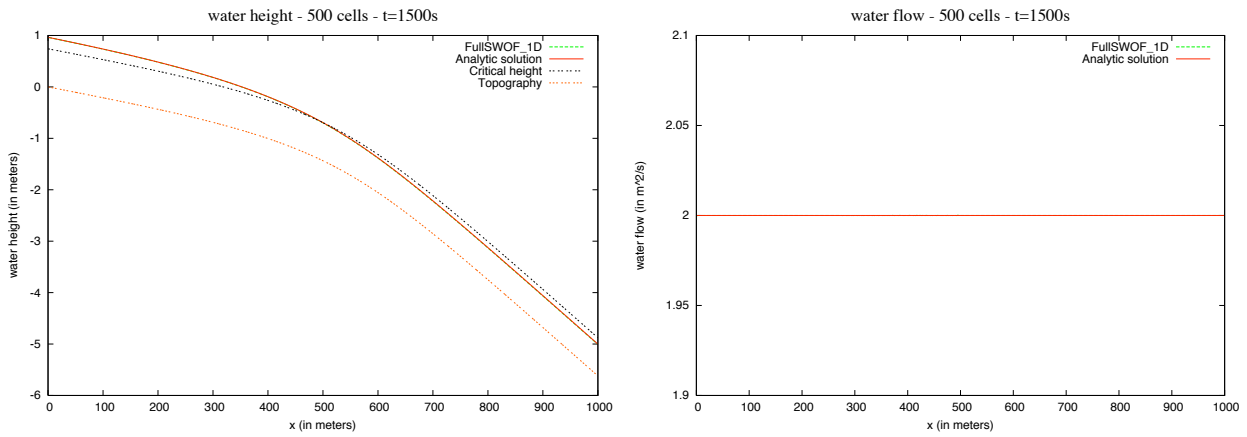


Figure 6: Fluvial/torrential Mac Donald like test with Manning friction coefficient

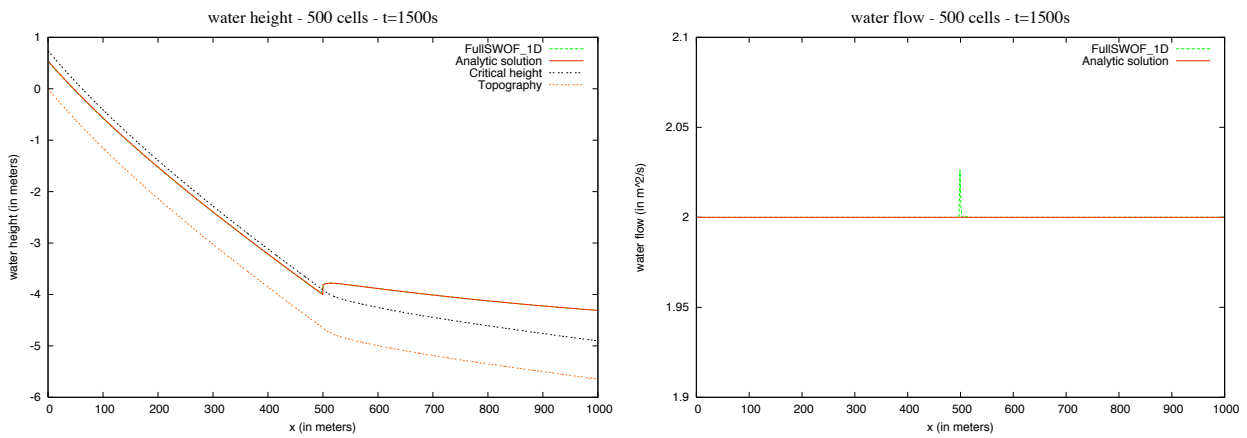


Figure 7: Torrential/fluvial Mac Donald like test with Darcy-Weisbach friction coefficient

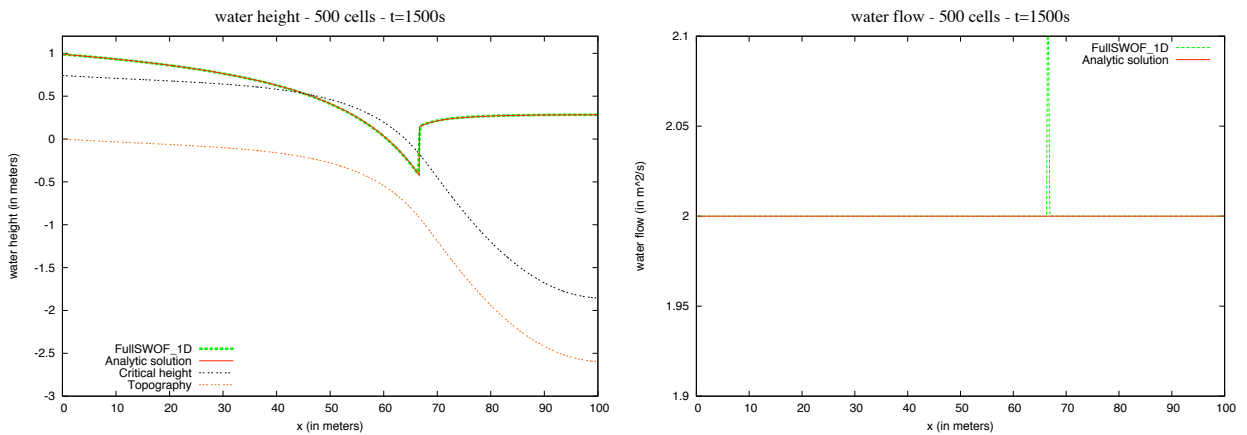


Figure 8: (Short) Fluvial Mac Donald like test with Manning friction coefficient

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 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 wet domain

In this section, we are interested in a dam break solution on a flat topography namely Stoker's solution. 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 9. This solution tests whether the code gives the location of the moving shock properly.

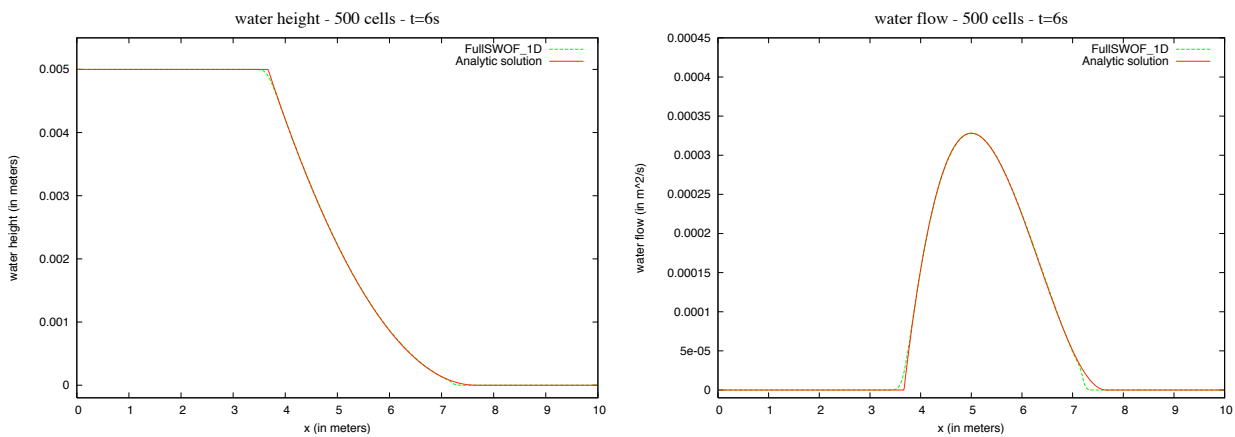


Figure 9: Dam break on a wet domain

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 10. 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.

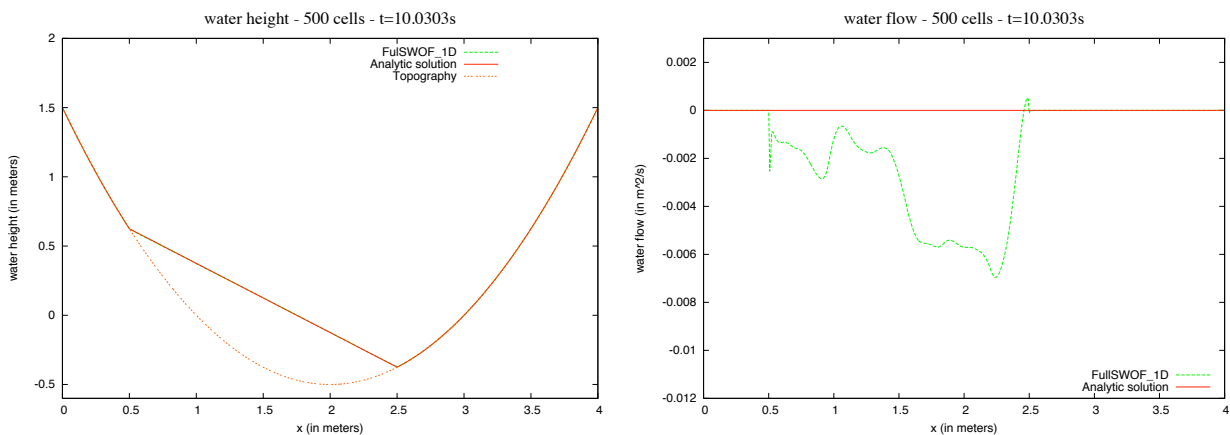


Figure 10: Thacker test case

References

- [1] E. Audusse, M.-O. Bristeau, and B. Perthame. Kinetic schemes for Saint-Venant equations with source terms on unstructured grids. Technical Report 3989, INRIA, 2000.
- [2] Olivier Delestre. Ecriture d'un code C++ pour la simulation en hydrologie. Master's thesis, Université d'Orléans, Orléans, France, September 2008. Available from: <http://dumas.ccsd.cnrs.fr/dumas-00446163/fr/>.
- [3] Olivier Delestre. *Simulation du ruissellement d'eau de pluie sur des surfaces agricoles*. PhD thesis, Université d'Orléans, Orléans, France, July 2010. Available from: <http://tel.archives-ouvertes.fr/tel-00531377>.
- [4] 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.
- [5] 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. Available from: <http://hal.archives-ouvertes.fr/hal-00426694/fr>.
- [6] Olivier Delestre, Carine Lucas, Pierre-Antoine Ksinant, Frédéric Darboux, Christian Laguerre, Thi Ngoc Tuoi Vo, François James, and Stéphane Cordier. SWASHES: a library of Shallow-Water Analytic Solutions for Hydraulic and Environmental Studies. Submitted.
- [7] Robert F. Dressler. Hydraulic resistance effect upon the dam-break functions. *Journal of Research of the National Bureau of Standards*, 49(3):217–225, September 1952.
- [8] Nicole Goutal and François Maurel. Proceedings of the 2nd workshop on dam-break wave simulation. Technical Report HE-43/97/016/B, Electricité de France, Direction des études et recherches, 1997.
- [9] Carine Lucas. How to choose boundary conditions? Study of the various cases. Application note. Available from: <https://sourcesup.renater.fr/docman/view.php/877/3962/BoundaryConditions.pdf>.
- [10] William Carlisle Thacker. Some exact solutions to the nonlinear shallow-water wave equations. *Journal of Fluid Mechanics*, 107:499–508, 1981. Available from: <http://journals.cambridge.org/action/displayAbstract?fromPage=online&aid=389055&fulltextType=RA&fileId=S0022112081001882>, doi:10.1017/S0022112081001882.