A shallow water code

Simon Chabot

February 27, 2015

Abstract

This short report introduces the SW code I have been working on for the first four weeks of my Master work placement. The code uses a fast and stable well-balanced finite volumes second order scheme with hydrostatic reconstruction. The flux terms are approximated by a Roe solver with an entropy fix to handle dry areas. Then, as one will see, the code has tested with a large amount of test cases.

The shallow water model, initial derived by Saint-Venant, describes a flow of water over a slowly varying topography $z(x,t)$, where x is the horizontal component. The flow is described by the height of water $h(x,t)$ above the bottom, and by its horizontal velocity $u(x,t)$. The vertical one is neglected. In one dimension, one has the following set of equations :

$$\begin{cases} \partial_t h + \partial_x \left(hu\right) = 0 \\ \partial_t \left(hu\right) + \partial_x \left(hu^2 + \frac{1}{2} g h^2\right) = -hg\partial_x z \end{cases} \tag{1}$$

Although this model is valid when the bottom is time dependent, in the sequel, we assume the bottom to be space dependent only.

The system (1) can be written in a more compact vector notation, as follow :

$$\partial_t \mathbf{U} + \partial_x \mathcal{F}(\mathbf{U}) = \mathbf{S} \tag{2}$$

with :

$$\mathbf{U} = \left(h, hu\right)^t$$
$$\mathcal{F}(\mathbf{U}) = \left(hu, hu^2 + \frac{1}{2} g h^2\right)^t$$
$$\mathbf{S} = \left(0, -hg\partial_x z\right)^t$$

1 The development

The proposed shallow water wave code uses the finite volume method. This method is known to handle properly shocks and contact discontinuities for a hyperbolic system of conservation laws. This code is based on a fast and stable well-balanced scheme with hydrostatic reconstruction, proposed by [Emmanuel Audusse et al., 2004]. General finite volumes schemes are inaccurate near steady states, that is why we needed this well-balanced scheme. Moreover, the proposed scheme preserved non-negativity of the water-height. In the SW code is implemented the second order scheme that they proposed. They reconstructed values $U_{i,r}$, $U_{i,l}$ are computed using a minmod function as follow :

$$U_{i,l} = U_i - 0.5\,\mathrm{minmod}\left(U_i - U_{i-1}, U_{i+1} - U_i\right) \tag{3}$$

$$U_{i,r} = U_i + 0.5\,\mathrm{minmod}\left(U_i - U_{i-1}, U_{i+1} - U_i\right) \tag{4}$$

where the minmod function returns the argument having the smallest modulus if the all have the same sign, zero otherwise. $U_{0,l}$ (and $U_{n,r}$) needs to be defined using an other definition ; a forward (and a backward) finite difference is used.

What about the numerical flux solver ? Here, we used the solver of Roe. This Riemann solver is perhaps the most well-known approximate solver. The idea of the method is the following one. Starting from a hyperbolic partial differential equation, representing a conservation law :

$$\partial_t \mathbf{U} + \partial_x \mathcal{F}\left(\mathbf{U}\right) = \mathbf{S}$$

one can use chain rule :

$$\partial_t \mathbf{U} + A\left(\mathbf{U}\right) \partial_x \mathbf{U} = \mathbf{S}$$

where $A\left(\mathbf{U}\right)$ is the Jacobian matrix of the flux function $\mathcal{F}\left(\mathbf{U}\right)$.

Then, for each cell of the discrete problem, one has to find a matrix $\tilde{A}\left(\mathbf{U}_i, \mathbf{U}_{i+1}\right)$, that is assumed constant and has some properties so that the local Riemann problem can be solved as a truly linear hyperbolic system. The needed properties are :

- $\tilde{A}$ must be diagonalizable with real eigenvalues and linearly independent ; ensuring the new linear system is truly hyperbolic.
- $\tilde{A}$ must be consistent with A ; $\mathbf{U}_i, \mathbf{U}_{i+1} \to \mathbf{U}$ we need $\tilde{A}\left(\mathbf{U}_i, \mathbf{U}_{i+1}\right) = A\left(\mathbf{U}\right)$.
- Conserving $\mathbf{F}_{i+1} - \mathbf{F}_i = \tilde{A}\left(\mathbf{U}_{i+1} - \mathbf{U}_i\right)$.

A method exists to find such a matrix. This method is fully explained in [Toro, 2009, pp. 351–353].

One drawback of the Roe solver is that it handles very badly the dry areas. Indeed, when $h = 0$, the two eigen values of $\tilde{A}$ are equal :

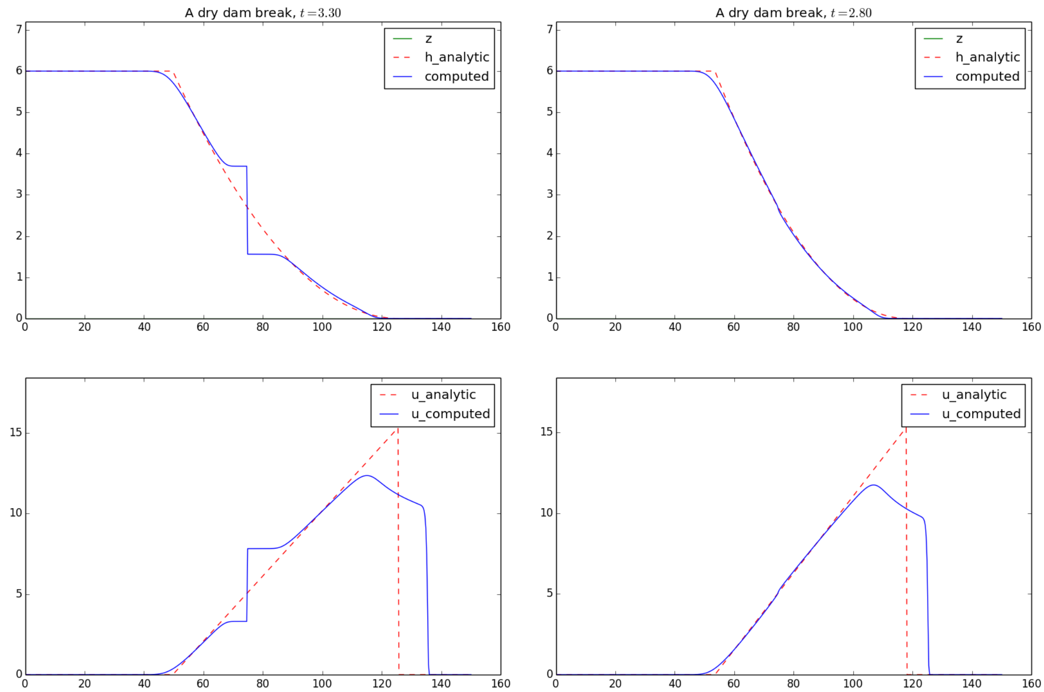
$$\lambda_1 = u + \sqrt{gh} \qquad\qquad \lambda_2 = -\sqrt{gh}$$

and it violates the needed properties of the solver. For that purpose, one needs to slightly modify it, and this modification is explained in [L. A. Monthe et al., 1998, pp. 218–219]. The figure 1a illustrates what would happened without this modification. This modification is called *an entropy fix*.

2 Comparisons between first and second order

2.1 First order

The figures 1a and 1b show the numerical improvements brought by the entropy fix of [L. A. Monthe et al., 1998].



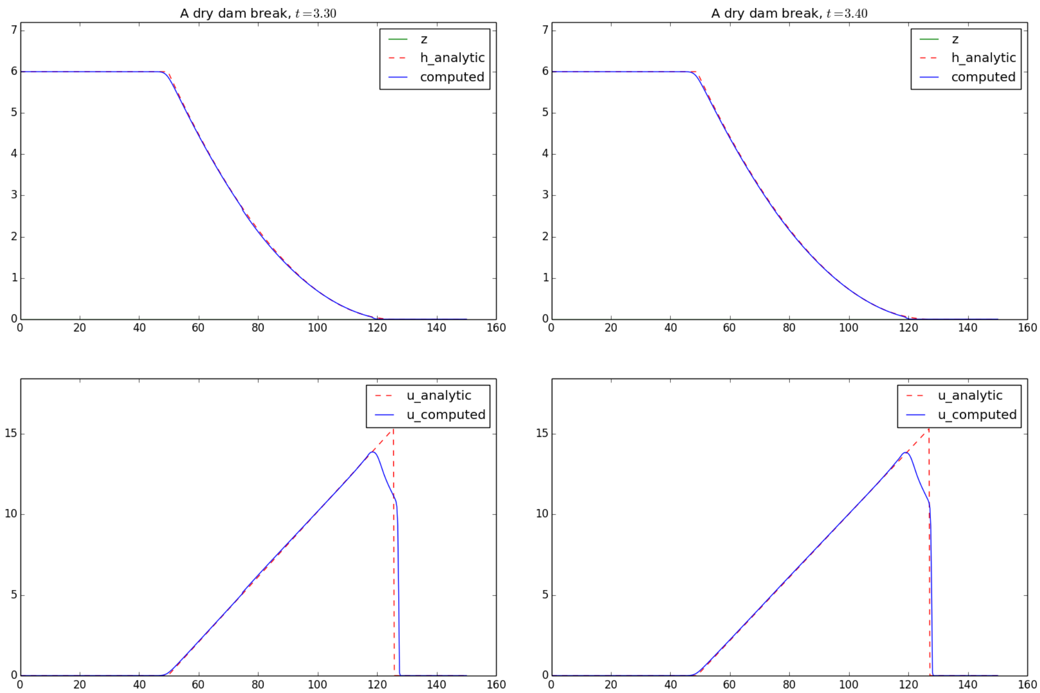
(a) without the entropy fix (b) with the entropy fix

Figure 1: A dry dam break, with a first order well balanced scheme and a Roe solver

One can notice that the unphysical shock has been almost eliminated. Nonetheless, there still is a small error at this position. The water height is almost correct, but not the velocity u, one can see there is a quite big error on the right front.

2.2 Second order

Using the second order corrects almost all those problems. As one can see, when using the second order method without the entropy fix, the water height is good, and the velocity is almost good also. But the results are even better with the entropy fix.



(a) without the entropy fix (b) with the entropy fix

Figure 2: A dry dam break, with a second order well balanced scheme and a Roe solver

3 Test cases for validation

This sections presents some test cases that have been implemented. The numerical computations are done with a 2nd order well-balanced scheme with hydrostatic reconstruction and with a Roe solver with an entropy fix. The numerical code is written in Fortran while the analytical solutions are computed by a Python code.

All the presented test cases are issued from [Delestre et al., 2013]. We have 5 test cases with bumps, 3 dam-breaks and 1 oscillating test-case.

3.1 Bumps

All the bumps test cases use this bottom definition :

$$z(x) = \begin{cases} 0.2 - 0.05(\mathrm{x} - 10)^2 \text{ m} & \text{if } 8 \leqslant x \leqslant 12 \\ 0 \text{ m} & \text{else} \end{cases} \tag{5}$$

with $x \in [0 \text{ m}, 25 \text{ m}]$.

3.1.1 Lake at rest, bump immersed

This is the simplest test case one can imagine. A lake a rest, with the bottom below. That is to say one has :

$$\begin{aligned} h(x,t) + z(x) &= 0.5 \text{ m} & \forall x, t \\ u(x,t) &= 0 \text{ m/s} & \forall x, t \end{aligned}$$

See figure 3.

3.1.2 Lake at rest, bump emerged

Almost the same test case, bump with a dry area.

$$\begin{aligned} h(x,t) + z(x) &= \min(0.1 \text{ m}, z(x)) & \forall x, t \\ u(x,t) &= 0 \text{ m/s} & \forall x, t \end{aligned}$$

See figure 4.

3.1.3 Bumps with subcritical flow

A flow is subcritical when its Froude number is below 1, that is to say when the velocity of the flow is smaller than the wave velocity. The Froude number is defined by :

$$Fr = \frac{U}{\sqrt{gh}} \tag{6}$$

The flow is initially taken at rest, $h + z = \text{const}$. Two boundary conditions are set :

$$\begin{cases} h(x = L, t) & = 2.0 \text{ m} \\ q(x = 0, t) & = 4.42 \text{ m}^2/\text{s} \end{cases}$$

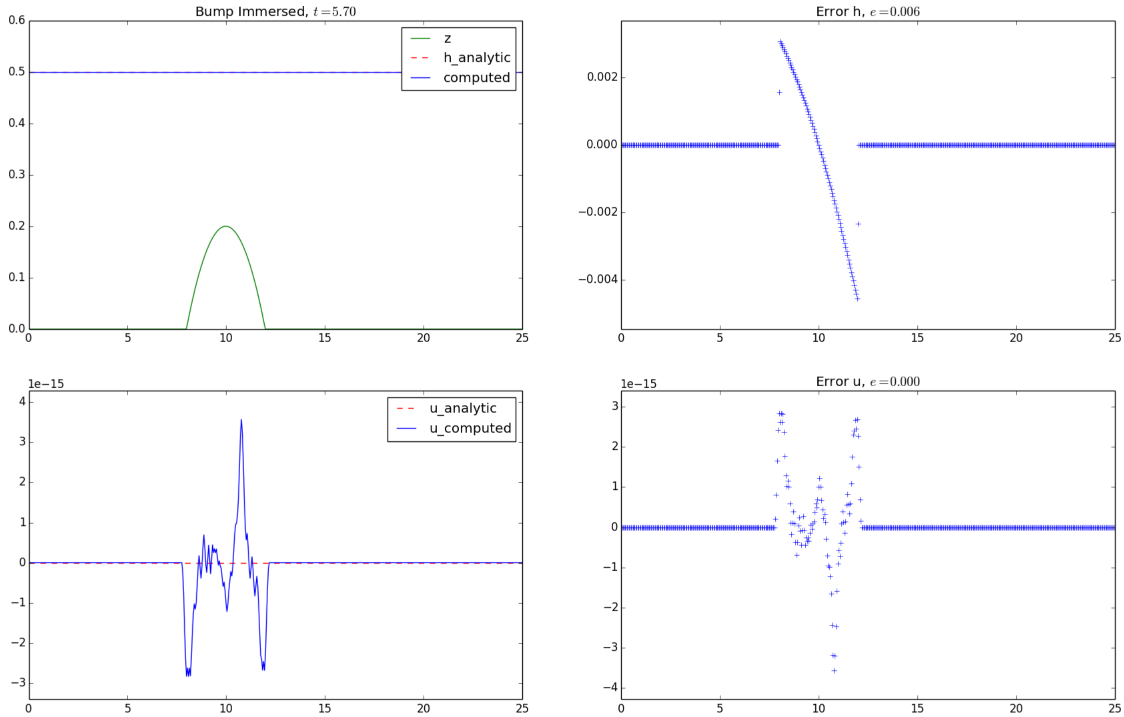


Figure 3: Lake at rest, bump immersed

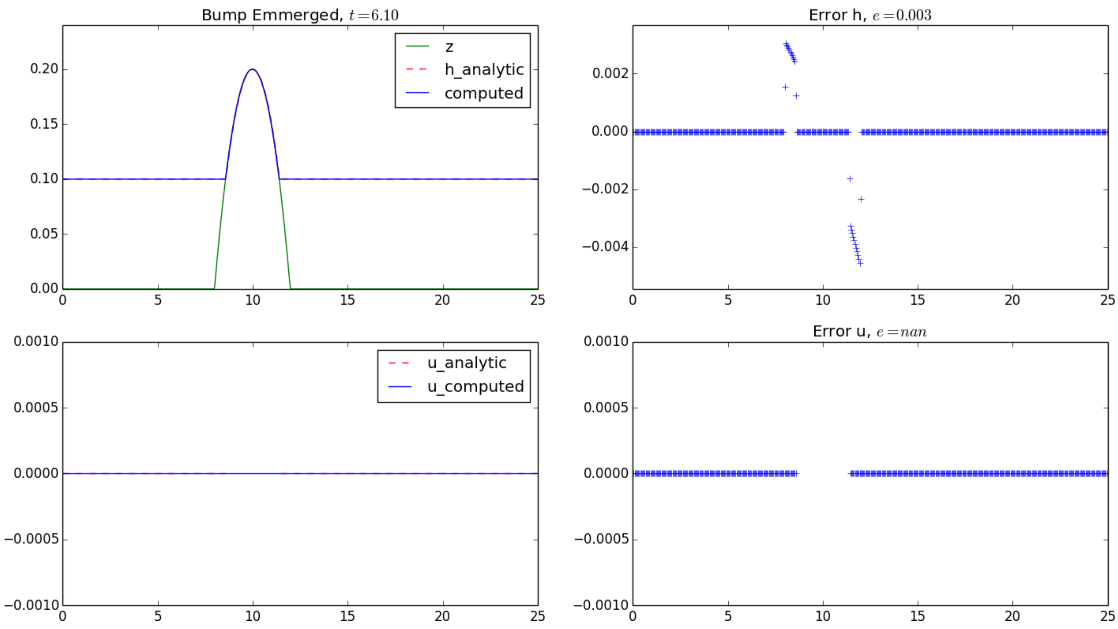


Figure 4: Lake at rest, bump emerged

where q is the flow rate, defined by $q(x,t) = h(x,t)u(x,t)$. The other parameters are free. This case converges to a steady state. The analytical solution of this steady state is found using the Bernouilli equation :

$$\frac{q^2}{2h^2} + g(h+z) = \text{const} \tag{7}$$

The water height being known at the position $x = x_p$, the constant can be estimated at this position, and one has :

$$\frac{q_0^2}{2h^2} + g(h+z) = \frac{q_0^2}{2h_p^2} + g(h_p + z_p)$$

where $h_p = h(x = x_p)$ and $z_p = z(x = x_p)$. This equation can be written as a third order polynomial of h :

$$h^3 + \left(z - \frac{q_0^2}{2gh_p^2} - h_p - z_p\right) h^2 + \frac{q_0^2}{2g} = 0 \tag{8}$$

which can be analytically solved, for instance, by the Cardano's method.

In this specific test case, the chosen position x_p is $x_p = L$, this gives $h_m = h(x = L) = 2.0$ m and $z_p = z(x = L) = 0$ m. The flow rate q_0 is equal to $q_0 = q = 4.42$ m^2/s, because in steady state the flow rate is the same everywhere.

3.1.4 Bumps transcritical with and without shock

Two other well known test cases are the transcritical flows with and without shock. The boundary conditions for the flow without shock are :

$$\begin{cases} h(x = L, t) & = 0.66 \text{ m} \\ q(x = 0, t) & = 1.53 \text{ m}^2/\text{s} \end{cases}$$

The boundary conditions with shock are :

$$\begin{cases} h(x = L, t) & = 0.33 \text{ m} \\ q(x = 0, t) & = 0.18 \text{ m}^2/\text{s} \end{cases}$$

The transcritical flow without shock is subcritical before the bump, and supercritical after (i.e. the flow velocity is larger than the wave velocity). The other test, with the shock, is subcritical everywhere excepted above the bump, where it is supercritical.

See figures 6 and 7.

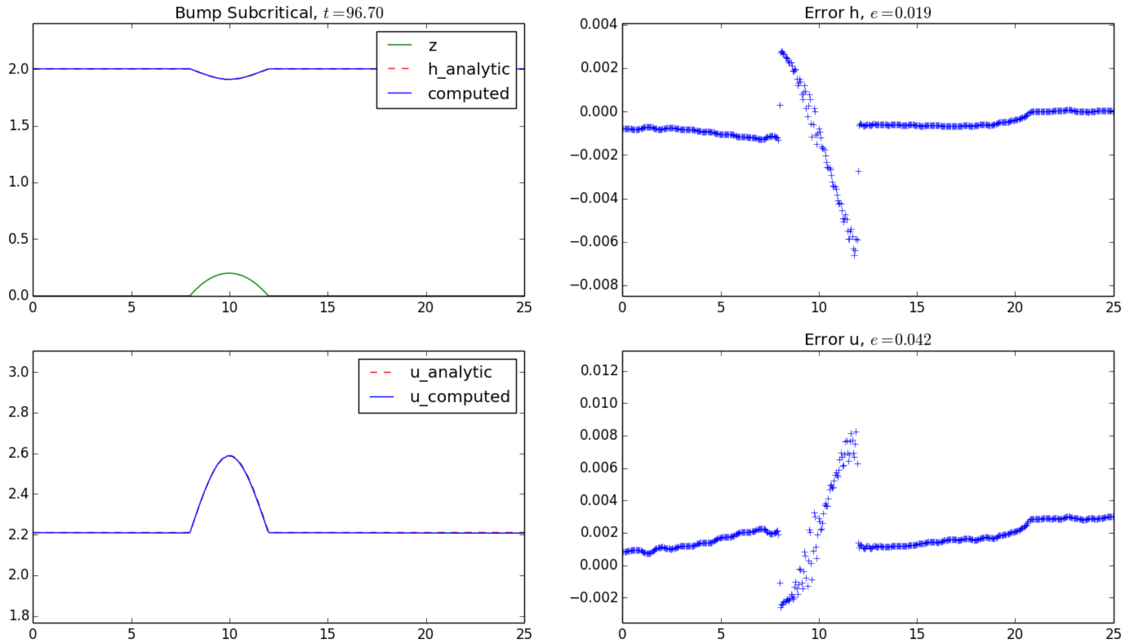


Figure 5: subcritical flow

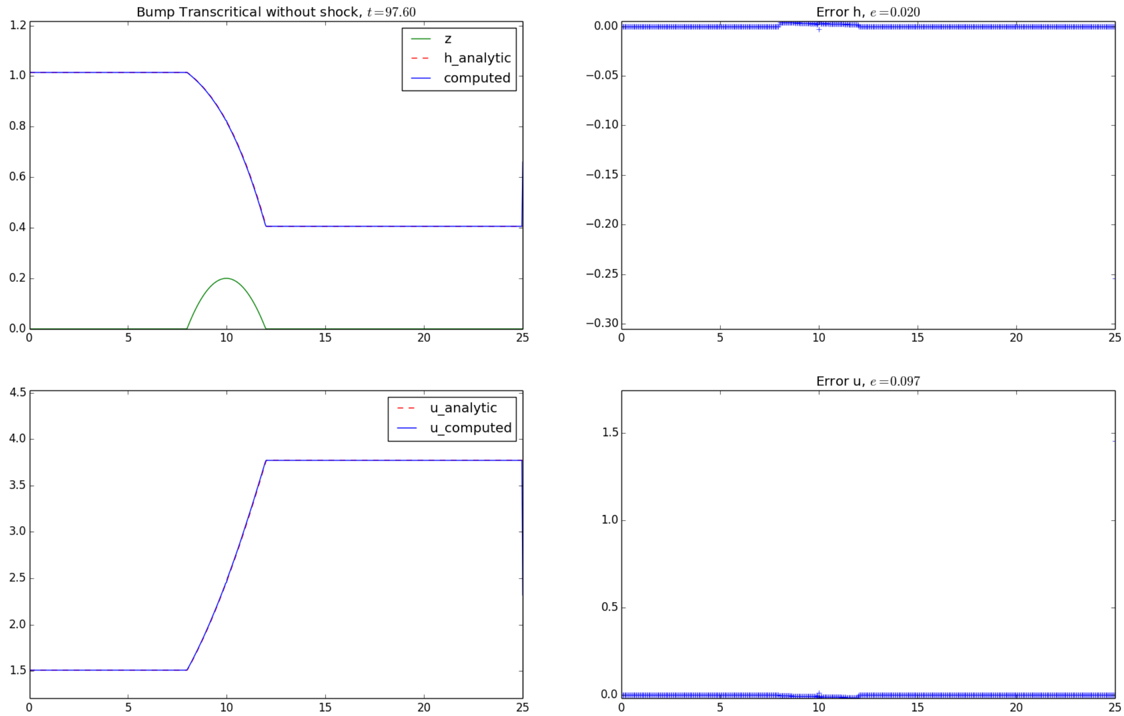


Figure 6: transcritical flow without shock

3.2 Dam breaks

The code has also been tested with three dam break test cases. A dam break test case consists in a initial configuration where the velocity is zero, and the left part of the water is at a given height h_L, while the right part is at a given height h_R. It is usually assume that $h_L > h_R$. The topography is taken flat.

As we said, three test cases are provided. The first is a wet dam break, i.e. we initially have $h_L > h_R > 0$. The second is a dry dam break, i.e. $h_L > h_R = 0$. And the last is a dry dam break with a Chézy's friction law.

3.2.1 A wet dam break

The initial conditions are :

$$\begin{cases} h(x,0) & = \begin{cases} 0.005 \text{ m} & \text{if } x < L/2 \\ 0.001 \text{ m} & \text{else} \end{cases} \\ u(x,0) & = 0.0 \text{ m/s} \end{cases}$$

with $L = 10$ m.

See figure 8.

3.2.2 A dry dam break

The initial conditions are :

$$\begin{cases} h(x,0) & = \begin{cases} 0.005 \text{ m} & \text{if } x < L/2 \\ 0.0 \text{ m} & \text{else} \end{cases} \\ u(x,0) & = 0.0 \text{ m/s} \end{cases}$$

with $L = 10$ m.

See figure 9.

3.2.3 A dry dam break with friction

This case involves a friction law. Friction is introduced by modifying the SW model as follow :

$$\begin{cases} \partial_t h + \partial_x (hu) = 0 \\ \partial_t (hu) + \partial_x \left(hu^2 + \frac{1}{2} g h^2\right) = -hg\partial_x z - S_f \end{cases} \tag{9}$$

where S_f is defined by a friction model. In our test case, this Chézy's friction is used. It is defined as follow :

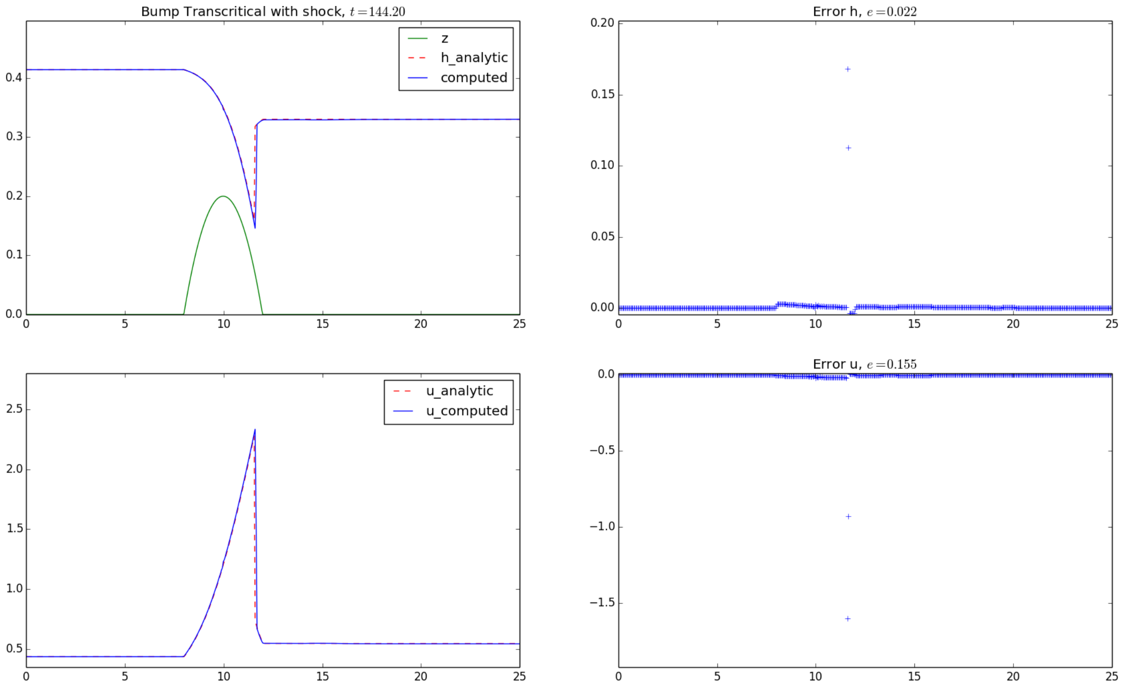


Figure 7: transcritical flow with shock

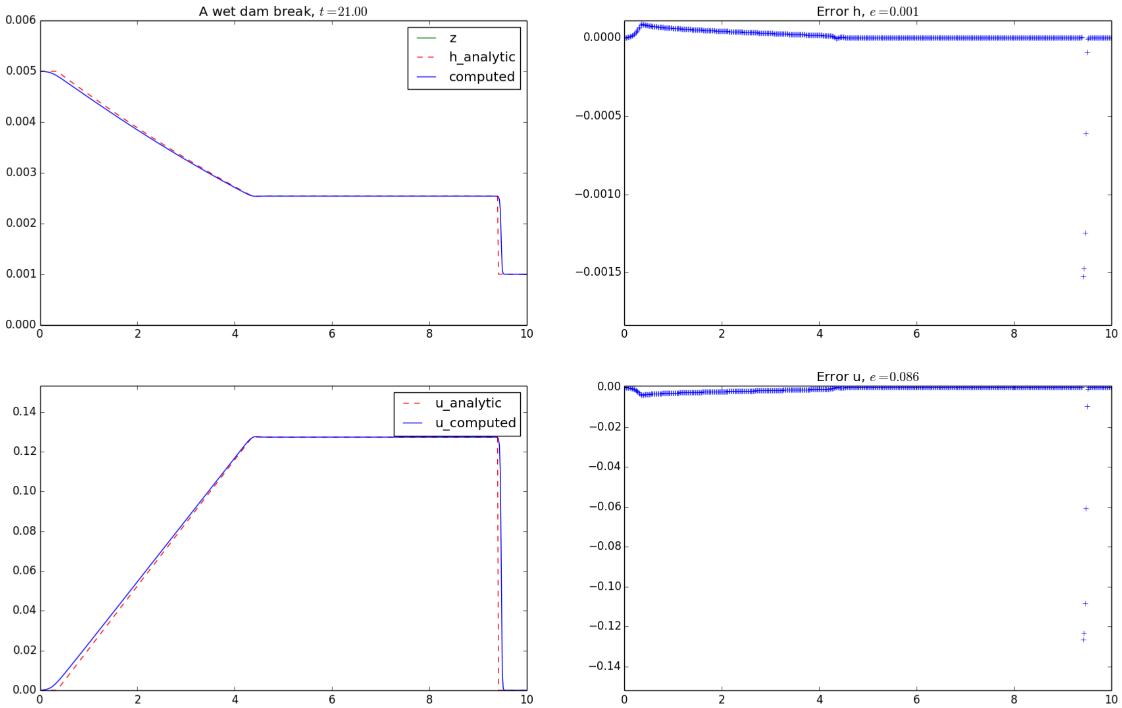


Figure 8: A wet dam break

$$S_f = C_f \frac{U\,|U|}{h} \tag{10}$$

where C_f is a friction coefficient. This coefficient is related to the Chézy's coefficient C by : $C_f = 1/C^2$.

The test case we implemented uses the following initial conditions :

$$\begin{cases} h(x,0) & = \begin{cases} 6.0 \text{ m} & \text{if } x < L/2 \\ 0.0 \text{ m} & \text{else} \end{cases} \\ u(x,0) & = 0.0 \text{ m/s} \end{cases}$$

with $L = 2000$ m, and $C_f = 1/C^2$, with $C = 40$ m$^{1/2}$/s.

See figure 10.

3.3 Oscillations

This last test case is the evolution of a planar surface of water in a parabola without friction. The parabola is defined as follow :

$$z(x) = h_0 \left(\frac{1}{a^2} \left(x - \frac{L}{2} \right)^2 - 1 \right) \tag{11}$$

and the initial condition on the water surface is :

$$h(x) = \begin{cases} -h_0 \left(\left(\frac{1}{a} \left(x - \frac{L}{2} \right) + \frac{B}{\sqrt{2gh_0}} \right)^2 - 1 \right) & \text{if } x_0 < x < x_1 \\ 0 \text{ m} & \text{else} \end{cases} \tag{12}$$

with :

$$x_0 = -\frac{1}{2} - a + \frac{L}{2} \qquad\qquad x_1 = -\frac{1}{2} + a + \frac{L}{2}$$

In our simulation, $a = 1$ m, $h_0 = 0.5$ m, $L = 4$ m. See figure 11.

4 Get the code

One can have a glance to the code at this address : `https://chabotsi.fr/hg/unice/sw`, and launch quickly all the test cases (one by one) typing `make test_all`. Any comment is welcomed.

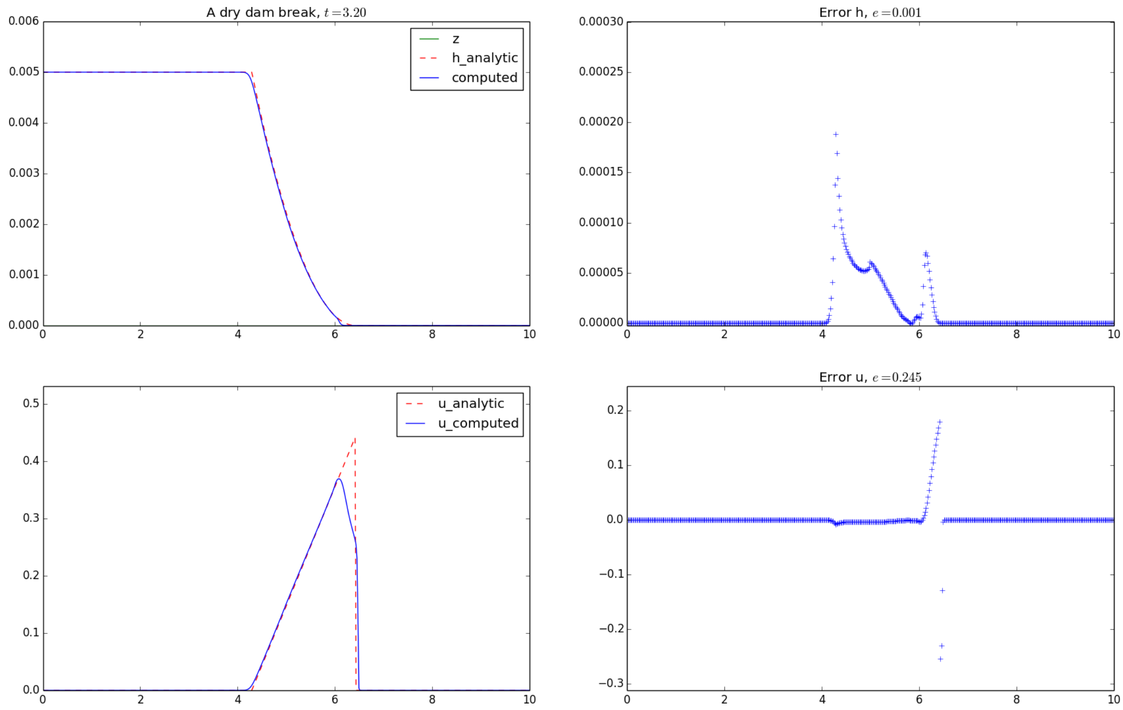


Figure 9: A dry dam break

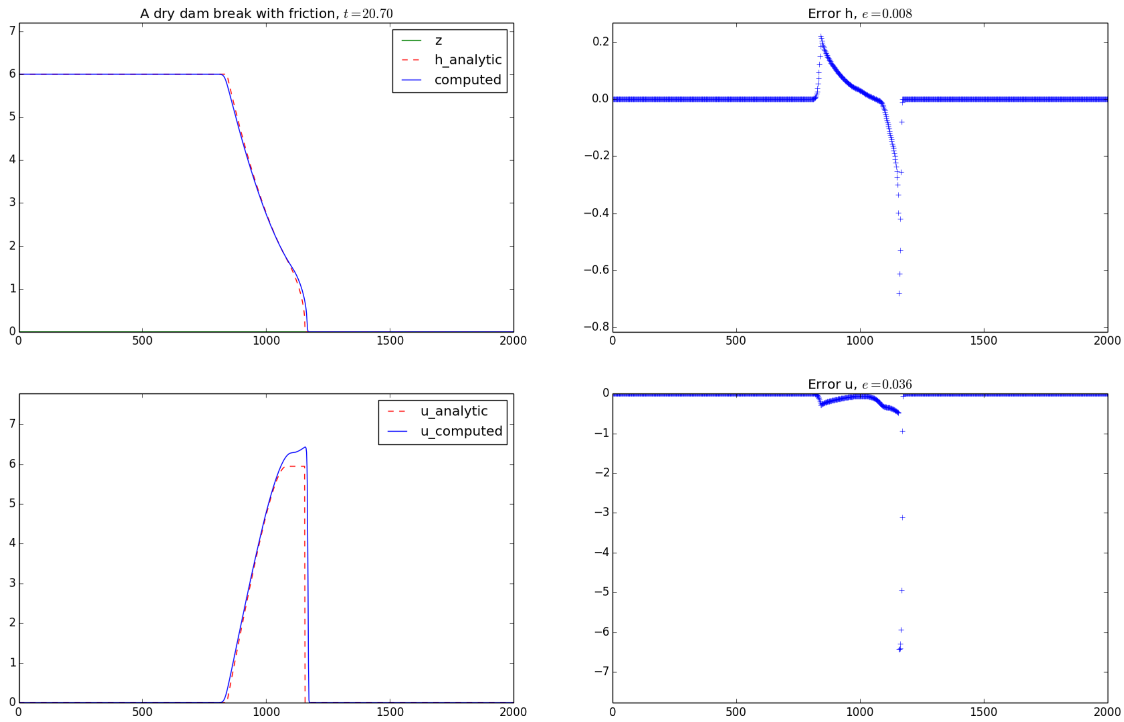


Figure 10: A dry dam break with friction

References

[Delestre et al., 2013] Delestre, O., Lucas, C., Ksinant, P.-A., Darboux, F., Laguerre, C., Vo, T.-N.-T., James, F., and Cordier, S. (2013). SWASHES: a compilation of shallow water analytic solutions for hydraulic and environmental studies. *International Journal for Numerical Methods in Fluids*, 72(3):269–300.

[Emmanuel Audusse et al., 2004] Emmanuel Audusse, François Bouchut, Marie-Odile Bristeau, Rupert Klein, and Benoît Perthame (2004). A fast and stable well-balanced scheme with hydrostatic reconstruction for shallow water flows. *Society for Industrial and Applied Mathematics.*

[L. A. Monthe et al., 1998] L. A. Monthe, F. Benkhaldoun, and I. Emahi (1998). Positivity preserving finite volume Roe schemes for transport-diffusion equations. *Computer methods in applied mathematics and engineering.*

[Toro, 2009] Toro, E. F. (2009). *Riemann Solvers and Numerical Methods for Fluid Dynamics.* Springer Berlin Heidelberg, Berlin, Heidelberg.

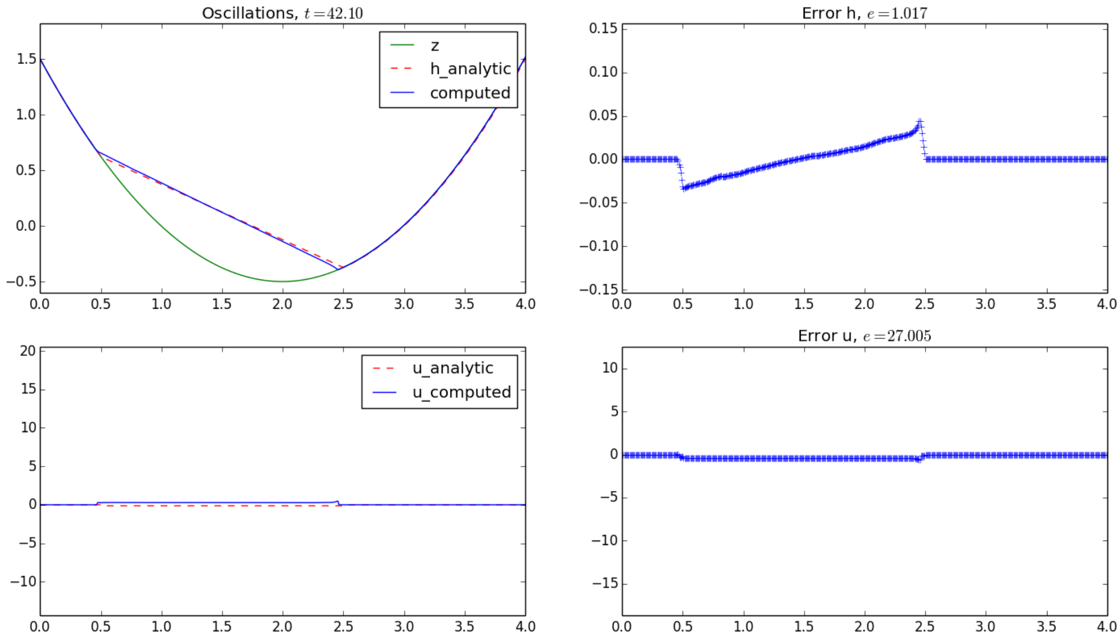


Figure 11: Oscillating test case, after 21 periods