Finite Volume Methods for the 1D Shallow Water Equations

Théo VIDAL
theo.vidal@ensta.fr

Final report for MEC_4MF06_TA ENSTA Course – February-April 2025

Marica Pelanti <marica.pelanti@ensta.fr>, Pietro Congedo <pietro.congedo@inria.fr>

Abstract

In this study, different methods were numerically implemented to solve the 1D Shallow Water problem, for which no analytical solution can be found in the general case of a varying bottom topography. Performances of a Finite-Volume approach using Rusanov, and Roe schemes for Riemann problems, in terms of accuracy and efficiency are measured for three distinct problems and are discussed.

Contents

1 The 1D Shallow Water equations

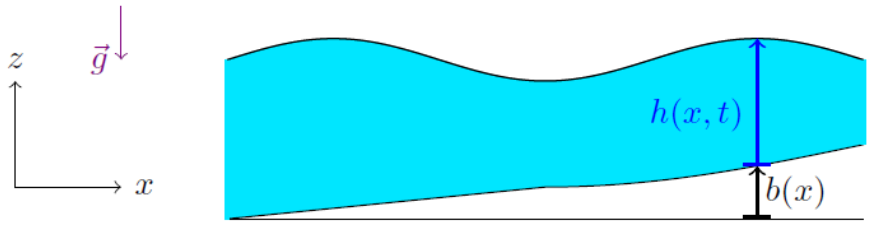


Figure 1: Illustration of the shallow water flow configuration used in the presented examples.

The shallow water equations model a free surface incompressible inviscid flow under the assumption $H \ll L$, with H a characteristic flow height and L a characteristic horizontal length. The equations are obtained by depth-averaging the Navier-Stokes equations under the considered hypothesis. Problems will here be solved by considering one-dimensional flows over a topography $b(x)$ as depicted on Figure 1. The equations can be written in the following conservative form:

$$\frac{\partial U}{\partial t} + \frac{\partial \mathcal{F}(U)}{\partial x} = \Psi \tag{1}$$

where

$$U = \begin{pmatrix} h \\ hu \end{pmatrix}, \quad \mathcal{F}(U) = \begin{pmatrix} hu \\ hu^2 + g\frac{h^2}{2} \end{pmatrix}, \quad \Psi = \begin{pmatrix} 0 \\ -gh\frac{\partial b}{\partial x} \end{pmatrix} \tag{2}$$

or in the quasi-linear conservative form:

$$\frac{\partial U}{\partial t} + A(U)\frac{\partial U}{\partial x} = \Psi \tag{3}$$

where $A(U)$ is the Jacobian matrix of flux $\mathcal{F}$ with respect of U, which is calculated by expressing $\mathcal{F}(U) = \begin{pmatrix} u_2 \\ \frac{u_2^2}{u_1} + g\frac{u_1^2}{2} \end{pmatrix}$:

$$A(U) = \frac{\partial \mathcal{F}}{\partial U} = \begin{pmatrix} 0 & 1 \\ -\frac{u_2^2}{u_1^2} + gu_1 & 2\frac{u_2}{u_1} \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ -u^2 + gh & 2u \end{pmatrix} \tag{4}$$

The characteristic polynomial of $A(U)$ is $\chi_{A(U)} = \det\begin{pmatrix} X & -1 \\ u^2-gh & X-2u \end{pmatrix} = X^2 - 2uX + (u^2 - gh) = (X - (u + \sqrt{gh}))(X - (u - \sqrt{gh}))$, hence the eigenvalues and eigenvectors of $A(U)$:

$$\lambda_1 = u - \sqrt{gh}, \ \lambda_2 = u + \sqrt{gh} \tag{5}$$

$$r_1 = \begin{pmatrix} 1 \\ u - \sqrt{gh} \end{pmatrix}, \ r_2 = \begin{pmatrix} 1 \\ u + \sqrt{gh} \end{pmatrix} \tag{6}$$

The two characteristic fields of the system are genuinely nonlinear, indeed, after noticing that $\lambda_1 = \frac{u_2}{u_1} + \sqrt{gu_1}$ and $\lambda_2 = \frac{u_2}{u_1} - \sqrt{gu_1}$,

$$\begin{aligned}\nabla\lambda_1 \cdot r_1 &= \begin{pmatrix} -\frac{u_2}{u_1^2} + \frac{1}{2}\sqrt{\frac{g}{u_1}} \\ \frac{1}{u_1} \end{pmatrix} \cdot r_1 = \begin{pmatrix} -\frac{u}{h} + \frac{1}{2}\sqrt{\frac{g}{h}} \\ \frac{1}{h} \end{pmatrix} \cdot \begin{pmatrix} 1 \\ u - \sqrt{gh} \end{pmatrix} \\ &= -\frac{u}{h} + \frac{1}{2}\sqrt{\frac{g}{h}} + \frac{u}{h} - \sqrt{\frac{g}{h}} = -\frac{1}{2}\sqrt{\frac{g}{h}} \neq 0 \end{aligned} \tag{7}$$

$$\text{and } \nabla\lambda_2 \cdot r_2 = \begin{pmatrix} -\frac{u}{h} - \frac{1}{2}\sqrt{\frac{g}{h}} \\ \frac{1}{h} \end{pmatrix} \cdot \begin{pmatrix} 1 \\ u + \sqrt{gh} \end{pmatrix} = \frac{1}{2}\sqrt{\frac{g}{h}} \neq 0 \tag{8}$$

Therefore, the wave structure of the solution of a Riemann problem for the shallow water equations can either be a rarefaction or a shock.

2 Dam-break problem without bottom topography

The problem aims at modeling the water flow from a sudden dam break, exhibiting two behaviors: a hydraulic jump propagating at the front of the dam, and a wave propagating from the back, as water fills the empty space.

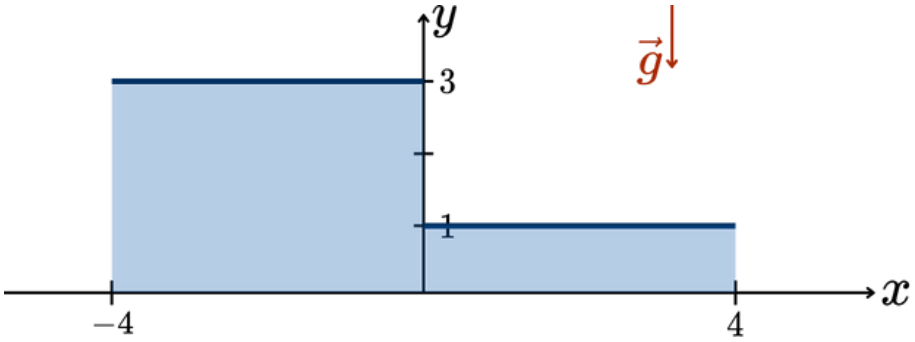


Figure 2: Illustration of the dam-break problem initial condition

The situation is modeled in a domain $[x_1, x_2] = [-4, 4]$ with transmissive boundary conditions, i.e. water can freely flow out of the domain. This problem corresponds to a Riemann problem, the solutions are either rarefactions or shocks.

2.1 Shallow Water equations without topography term

In order for an solution to be computed and the scheme to be validated for further experiments, the bottom is considered flat, hence $b(x) = 0$. The system (1) can be re-written as:

$$\frac{\partial U}{\partial t} + \frac{\partial F(U)}{\partial x} = 0 \tag{9}$$

The following initial conditions are considered on the space interval $[-x_1, x_2]$, representing two steady water levels in and out of the dam:

$$(h, u)(x, t = 0) = \begin{cases} (3, 0) \text{ if } x \leq \bar{x} \\ (1, 0) \text{ if } x > \bar{x} \end{cases} \tag{10}$$

where $\bar{x}$ is the position of the dam in the domain. For a simpler analysis, the value $\bar{x} = 0$ is taken, without loss of generality thanks to the chosen transmissive boundary conditions.

2.2 Numerical implementation

The previous system of equation can be solved using the Finite Volume Method, by taking a uniform discretization of the space interval $I = [-4, 4]$ with N_C cells. Discrete U vector for time n and cell i - centered at x_i and covering $\left[x_{i-1/2}, x_{i+1/2}\right]$ - is noted U_i^n, grid spacing is $\Delta x = x_{i+1} - x_i$ and Δt is the time step.

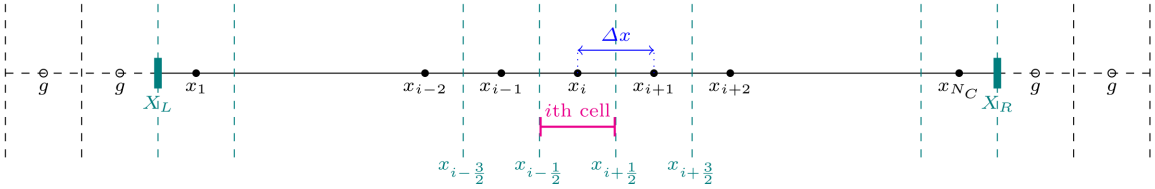


Figure 3: Representation of a uniform Finite Volume grid with N_c cells over an interval $[X_L, X_R]$. Two additional ghost cells are considered on the left and on the right of the interval limits, so as to enforce boundary conditions.

The hyperbolic system of laws (3) without bottom topography can be numerically approximated using a three-point finite volume conservative scheme:

$$U_i^{n+1} = U_i^n - \frac{\Delta t}{\Delta h}[F(U_i^n, U_{i+1}^n)) - F(U_{i-1}^n, U_i^n)] \tag{11}$$

where F is the *numerical flux*, which must satisfy the consistency condition $F(U,U) = \mathcal{F}(U)$. Two methods are implemented based on two different numerical fluxes.

2.2.1 Rusanov scheme

The *Rusanov scheme* defines the flux as:

$$F(U_i, U_{i+1}) = \frac{1}{2}\left[\mathcal{F}(U_i) + \mathcal{F}(U_{i+1}) - S_{i+1/2}(U_{i+1} - U_i)\right] \tag{12}$$

where $S_{i+1/2} = \max\left(|u_{i+1} + \sqrt{gh_{i+1}}|, |u_{i+1} - \sqrt{gh_{i+1}}|, |u_i + \sqrt{gh_i}|, |u_i - \sqrt{gh_i}|\right)$ denotes the local maximum absolute value of characteristic speeds.

2.2.2 Roe scheme

The *Roe scheme* is defined using an evaluation of variables in a special average state. The numerical flux between cells i and $i+1$ is defined as

$$F^{\text{Roe}}(U_i, U_{i+1}) = \mathcal{F}(U_i) + \sum_{k=1}^{2} \left(\hat{\lambda}_k^- \alpha_k \hat{r}_k\right)_{i+1/2} \tag{13}$$

For the shallow water equations (1), the Roe matrix is defined as $\hat{A} = A\left(\hat{U}_{\text{Roe}}\right)$, $\hat{U}_{\text{Roe}} = \hat{U}_{\text{Roe}(\hat{h},\hat{u})}$, with, for the interface $i+1/2$:

$$\hat{h}_{i+1/2} = \frac{1}{2}(h_i + h_{i+1}), \hat{u}_{i+1/2} = \frac{\sqrt{h_i}u_i + \sqrt{h_{i+1}}u_{i+1}}{\sqrt{h_i} + \sqrt{h_{i+1}}} \tag{14}$$

The Roe eigenvalues and Roe eigenvectors are

$$\hat{\lambda}_1 = \hat{u} - \sqrt{g\hat{h}}, \hat{\lambda}_2 = \hat{u} + \sqrt{g\hat{h}}, \hat{r}_1 = \begin{pmatrix} 1 \\ \hat{u} - \sqrt{g\hat{h}} \end{pmatrix}, \hat{r}_2 = \begin{pmatrix} 1 \\ \hat{u} + \sqrt{g\hat{h}} \end{pmatrix} \tag{15}$$

The coefficients are

$$\begin{aligned} \alpha_1 &= \frac{1}{2\sqrt{g\hat{h}}}\left(\left(\hat{u} + \sqrt{g\hat{h}}\right)\Delta U^{(1)} - \Delta U^{(2)}\right) \\ \alpha_2 &= \frac{1}{2\sqrt{g\hat{h}}}\left(-\left(\hat{u} - \sqrt{g\hat{h}}\right)\Delta U^{(1)} - \Delta U^{(2)}\right) \end{aligned} \tag{16}$$

where $\Delta U^{(\xi)}$ is the ξ^{th} component of the vector $\Delta U = U_{i+1} - U_i$.

2.2.3 Code implementation

The finite volume method and the two numerical schemes are implemented using the MATLAB language. In the main routine, a loop iterates over all time steps, updating the solution at each round according to the selected scheme. In order to enforce transmissive boundary conditions, values of the height and momentum in the ghost cells are set to the values at the adjacent cell of the physical domain.

For better readability, fluxes are defined in separate files, with particular care concerning cases where the water height is zero (a *dry state*). The numerical flux for the Rusanov method can be computed at one cell interface, while the Roe flux is computed at all cell interfaces at once for more efficiency.

```
for  istep=1:MaxStep
  w = qv; % store old qv

  % update solution qv
  switch method
  case 'Rusanov'
    % update solution at new time level
    for i=3:Ncp2
      fluip = fluxswRSn_templ(w(i,:), w(i+1,:));
      flui = fluxswRSn_templ(w(i-1,:), w(i,:));
      qv(i,:) = w(i,:) - dt/h*(fluip-flui);
    end

  case 'Roe'
    fRoe = fluxRoe(Nct, Ncp3, w);

    % update solution at new time level
    for i=3:Ncp2
      fluip = fRoe(i,:);      % first-order Roe flux at i+1/2 (row vector) - mod
2025
      flui  = fRoe(i-1,:);    % first-order Roe flux at i-1/2 - mod 2025
      qv(i,:) = w(i,:) - dt/h*(fluip-flui);
      end
  end

  % set Bondary conditions
  % left ghost cells
  for k = 1:2
    qv(2,k) =  qv(3,k); % zero-order extrapolation
    qv(1,k) =  qv(3,k);
  end

  [...]
  % right ghost cells
  for k = 1:2
    qv(Nct-1,k) =  qv(Ncp2,k); % zero-order extrapolation
    qv(Nct,k) =  qv(Ncp2,k);
  end

  [...]

  t=t+dt;
end  % end time loop
```

Listing 1: MATLAB implementation of the main loop for the Finite Volume method. Initialization and plotting parts were omitted.

```
function ff = fluxswRSn_templ(v,vp)
  [...]
  rl = v(1);   % water height
  rul = v(2);  % momentum

  if(rl>0)
   ul = rul/rl;   % velocity
  else
   ul=0;  % set zero velocity if dry
state
  end

  flul(1) = rul;
  flul(2) = rul*ul + grav*rl^2/2;

  rr = vp(1);   % water height
  rur = vp(2);  % momentum

  if(rr>0)
   ur = rur/rr;   % velocity
  else
   ur=0;  % set zero velocity if dry
state
  end

  flur(1) = rur;
  flur(2) = rur*ur + grav*rr^2/2;

  if ((rl ==0)&&(rr==0))  % flux is zero
if both left and right states are dry
    return
  end

  % velocities tied to kinetic energy
conservation
  cr = sqrt(grav * rr);
  cl = sqrt(grav * rl);

  S = max( ...
      max(abs(ur + cr), abs(ur - cr)),...
      max(abs(ul + cl), abs(ul - cl))...
  );

  ff = (flul + flur - S*(vp - v)) / 2;
```

Listing 2: MATLAB function to compute the Rusanov numerical flux at the interface between cells i and $i+1$.

```
function [fluxRoe] = fluxRoe(Nct, Ncp3,
w)
  [...]
  for i=1:Ncp3
    v = w(i, :);
    vp = w(i+1, :);
    [...]
    % set f(u_i)
    rl = v(1);     % water height
    rul = v(2);    % momentum
    ul = rul/rl;   % velocity
    flul(1) = rul;
    flul(2) = rul*ul + grav*rl^2/2;

    % set f(u_{i+1})
    rr = vp(1);    % water height
    rur = vp(2);   % momentum
    ur = rur/rr;   % velocity
    flur(1) = rur;
    flur(2) = rur*ur + grav*rr^2/2;

    hroe = 0.5 * (rr + rl); % water
height
    uroe = (sqrt(rl)*ul + sqrt(rr)*ur) /
(sqrt(rl) + sqrt(rr));  % velocity
    croe = sqrt(grav * hroe); % velocity
tied to kinetic energy conservation

    % Roe eigenvalues
    lambdav(1,i) = uroe - croe;
    lambdav(2,i) = uroe + croe;
    % Roe eigenvectors
    Rmatv(1,i,1) = 1.;
    Rmatv(2,i,1) = uroe - croe;
    Rmatv(1,i,2) = 1.;
    Rmatv(2,i,2) = uroe + croe;
    % coefficients projection Delta U
(analytical formulas)
    alphav(1, i) = ((uroe + croe) *
delta(1) - delta(2)) / (2 * croe);
    alphav(2, i) = (-(uroe - croe) *
delta(1) + delta(2)) / (2 * croe);

    ff=flul'; % column vector (consistent
with Rmatv)
    for k=1:2
        ff = ff + min(lambdav(k,i), 0.) *
alphav(k,i) * Rmatv(:,i,k);
    end

    fluxRoe(i,:) = ff';
  end % for i=1:Ncp3
```

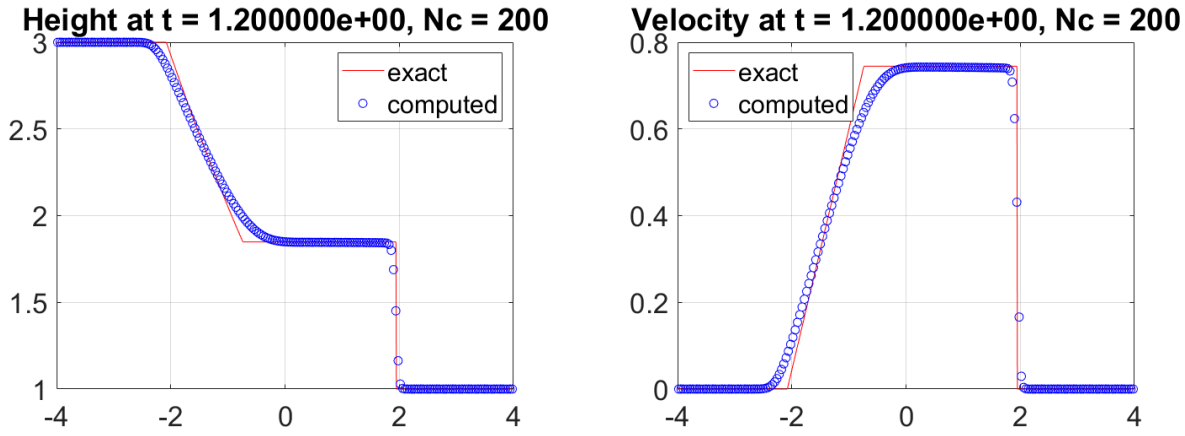
Listing 3: MATLAB function to compute the Roe numerical flux for all cell interfaces.

2.3 Experimental results

The numerical solution is computed using the routine described in Listing 1, and compared to the exact solution of the Riemann problem, solved using MATLAB. The two schemes listed above, Rusanov and Roe, are used for comparison on grids of size $N_c = 200$ and $N_c = 1000$, with a fixed ratio $\frac{\Delta t}{\Delta x} = 0.4$. The computed solution is directly plotted on MATLAB, and all results are shown in the next figures. Additional joint plots of the three solutions are provided for better comparison. According to the analytical solution, a rarefaction wave propagating from the left is expected, hence a positive velocity on this domain, as water has to move to the right to fill the newly empty space, a phenomenon tied to the mass conservation.

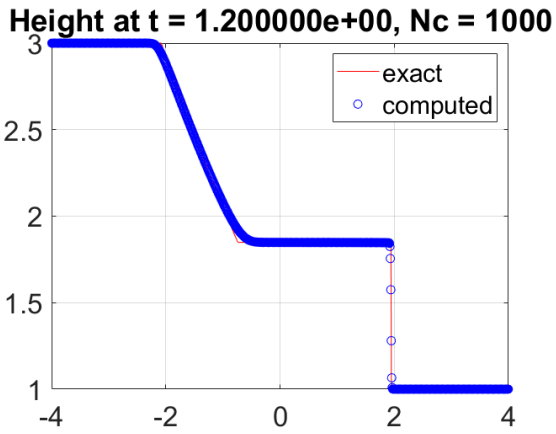
The maximum velocity computed for the analytical model is $u = 0.745$ and the maximum height is $h = 3$ – water cannot go upper than its initial state, as no energy source is present. The Courant number is $\sigma = |\lambda_{\max}|\frac{\Delta t}{\Delta x} = 0.991 \leq 1$, so the *Courant-Friedrichs-Lewy* (CFL) Condition is verified, providing a necessary (but not sufficient) condition for stability.

A benchmark of the main routine Listing 1 was also performed in order to compare the computing time of the implementations. Each problem was solved 25 times to calculate mean durations and their variance, and results are listed in Table 1. Note that the absolute duration value may vary from one computer to another, for that reason only the differences will be analyzed. The average duration of one loop step is taken by dividing the total duration by the number of steps, which equals to $\left\lceil \frac{t_f}{\Delta t} \right\rceil$, so 75 for $N_c = 200$ and 375 for $N_c = 1000$.

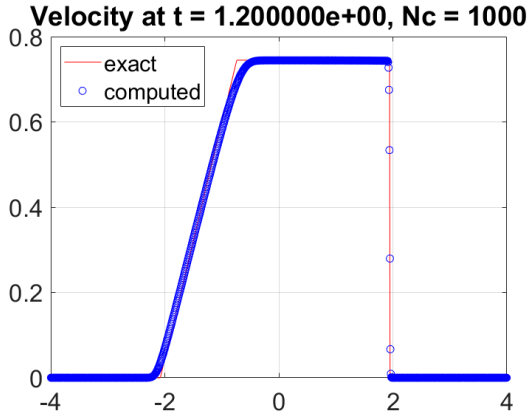


(a) Water height h (b) Water velocity u

Figure 4: Comparison between the exact solution of the dam-break problem, and the computed solution using Rusanov scheme for $N_c = 200$ grid cells and $\frac{\Delta t}{\Delta x} = 0.4$, plotted at time $t = 1.2$.

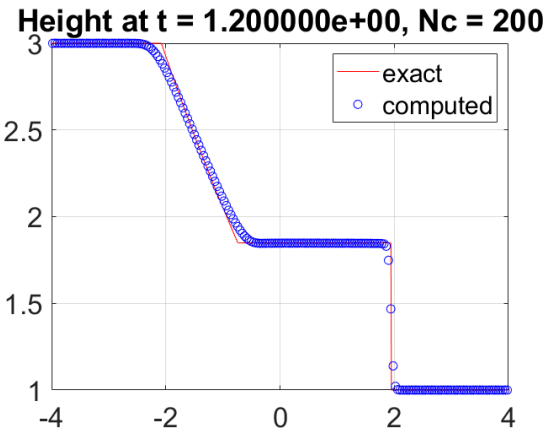


(a) Water height h

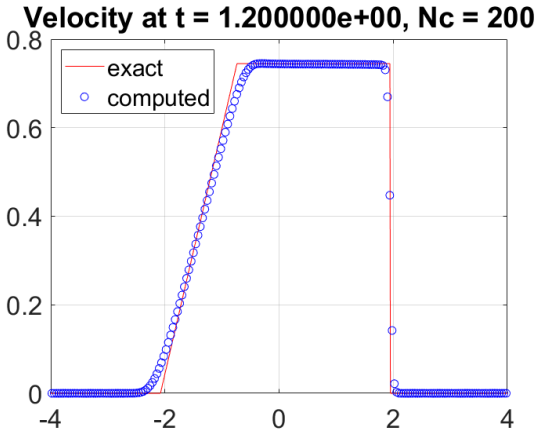


(b) Water velocity u

Figure 5: Comparison between the exact solution of the dam-break problem, and the computed solution using Rusanov scheme for $N_c = 1000$ grid cells and $\frac{\Delta t}{\Delta x} = 0.4$, plotted at time $t = 1.2$.

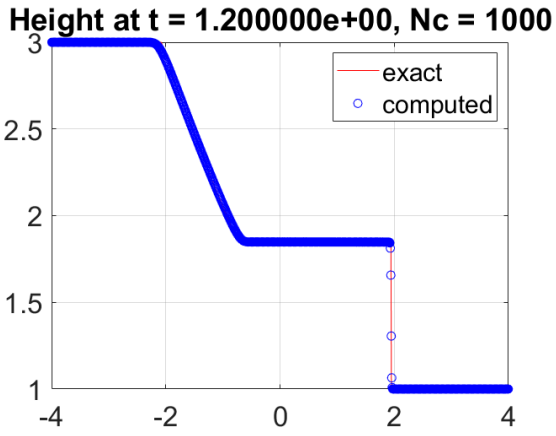


(a) Water height h

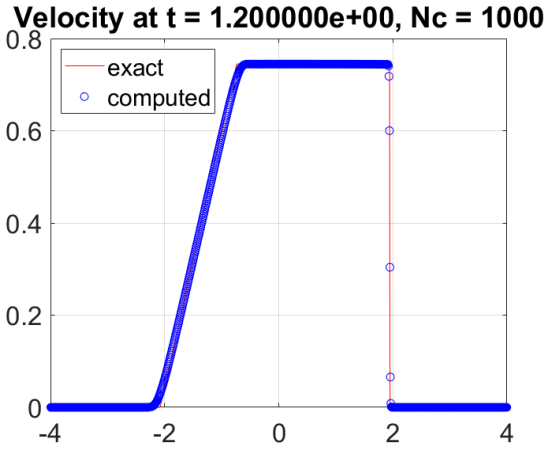


(b) Water velocity u

Figure 6: Comparison between the exact solution of the dam-break problem, and the computed solution using Roe scheme for $N_c = 200$ grid cells and $\frac{\Delta t}{\Delta x} = 0.4$, plotted at time $t = 1.2$.

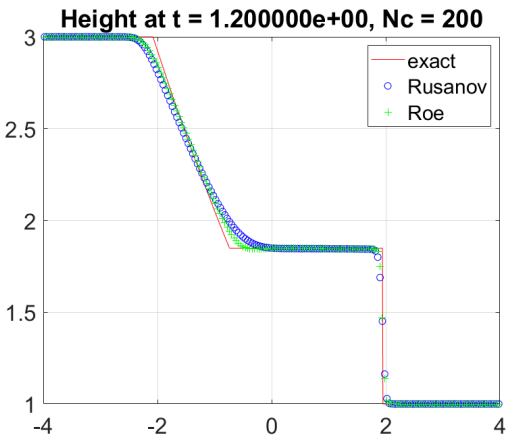


(a) Water height h

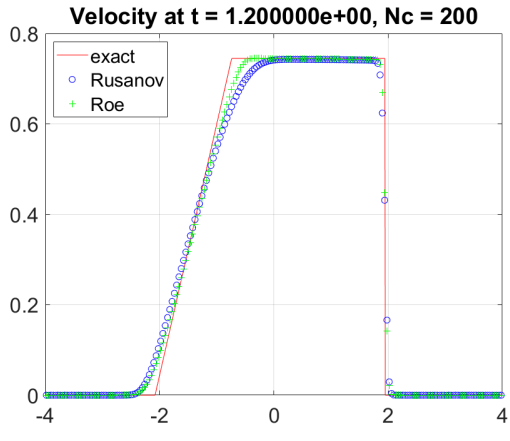


(b) Water velocity u

Figure 7: Comparison between the exact solution of the dam-break problem, and the computed solution using Roe scheme for $N_c = 1000$ grid cells and $\frac{\Delta t}{\Delta x} = 0.4$, plotted at time $t = 1.2$.

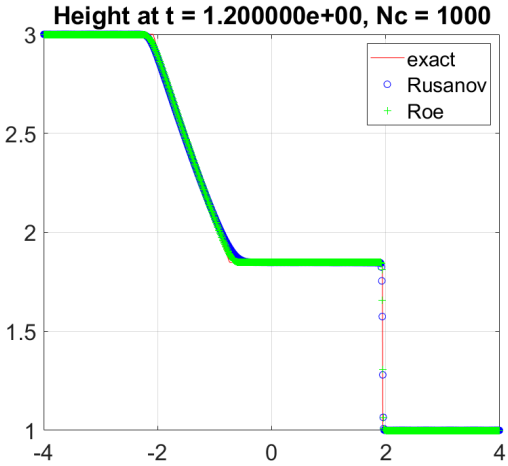


(a) Water height h

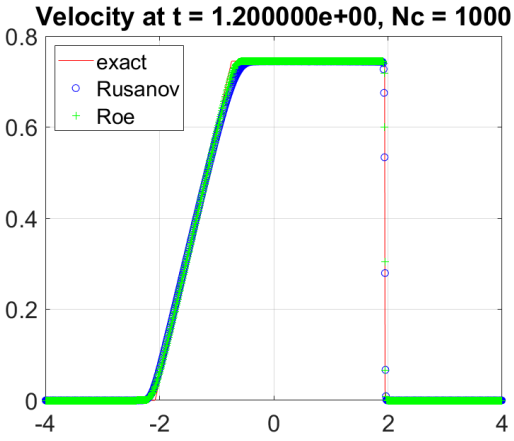


(b) Water velocity u

Figure 8: Comparison between the exact solution of the dam-break problem, and the computed solution using Rusanov and Roe schemes for $N_c = 200$ grid cells and $\frac{\Delta t}{\Delta x} = 0.4$, plotted at time $t = 1.2$.



(a) Water height h



(b) Water velocity u

Figure 9: Comparison between the exact solution of the dam-break problem, and the computed solution using Rusanov and Roe schemes for $N_c = 1000$ grid cells and $\frac{\Delta t}{\Delta x} = 0.4$, plotted at time $t = 1.2$.

Scheme	N_c	**Duration for $t_f = 1.2$ s**	**Average duration for one step**
Rusanov	200	162.82 ± 12.83 ms	2.17 ± 0.17 ms
Rusanov	1000	2.67 ± 0.093 s	7.12 ± 0.25 ms
Roe	200	166.44 ± 13.88 ms	2.22 ± 0.19 ms
Roe	1000	2.43 ± 0.083 s	6.48 ± 0.22 ms

Table 1: Benchmarks for the computed solution for the dam-break problem. Mean and variance are calculated on 25 measured durations.

2.4 Discussion

Looking at the graphs of the analytical and numerical solutions, using the Roe scheme seems slightly more precise than using Rusanov's scheme. Indeed, even if angular points (transitions between the rarefaction wave and the constant parts) are not well represented, the Roe scheme still provides

better accuracy. These boundaries are closer for the numerical height h as well as for the velocity u, indicating a better convergence and less diffusivity. Overall, both schemes approximate well the speeds of the wave, contrary to traditional, non-Godunov schemes such as Upwind, FTCS or Lax-Friedrichs, for which the wave speed can be different because they may not converge to a weak solution.

Both schemes approximate well the shock wave: only four points are located in the middle of the constant parts for the height, and four to five for the velocity, indicating that the discontinuity is mostly respected. In addition, when increasing the grid resolution N_c, the boundaries of this shock are straighter so the transition is less smooth, which is an intended behavior in order to better model a discontinuity.

These two numerical phenomena, a good approximation of the shock wave and diffusivity of the smooth parts, are tied to the first-order nature of the Rusanov and Roe schemes, whereas a second-order scheme might create oscillations around the discontinuity.

The selected numerical methods should always guarantee the conservation of involved physical quantities h and hu: indeed, the finite volume method states that, for the system (3) with no source term,

$$\int_{x_1}^{x_2} U(x,t_f)dx = \int_{x_1}^{x_2} U(x,0)dx + \int_0^{t_f} \mathcal{F}(U(x_1,t))dt - \int_0^{t_f} \mathcal{F}(U(x_2,t))dt \tag{17}$$

Consequently, numerical integrals $\Delta x \sum_{i=1}^{N_c} h_i$ and $\Delta x \sum_{i=1}^{N_c} (hu)_i$ should be equal to the analytical integrals (17). For the considered dam-break problem with initial conditions (10), for $t_f = 1.2$, both rarefaction and shock waves have not reached the boundaries of the physical domain $[-4,4]$, thus $U(x_i,t) = U(x_i,0)$ for $i = 1,2$. The two vector components of the integral can be calculated using provided numerical values:

$$\begin{aligned}
\int_{-4}^{4} h(x,1.2)dx &= \int_{-4}^{4} h(x,0)dx + \int_0^{1.2} (hu)(-4,t)dt - \int_0^{1.2} (hu)(4,t)dt \\
&= 4 \times 3 + 4 \times 1 + \int_0^{1.2} \cancel{(hu)(-4,0)dt} - \int_0^{1.2} \cancel{(hu)(4,0)dt} \\
&= 16
\end{aligned} \tag{18}$$

$$\begin{aligned}
\int_{-4}^{4} (hu)(x,1.2)dx &= \int_{-4}^{4} (hu)(x,0)dx + \int_0^{1.2} \left(hu^2 + g\frac{h^2}{2}\right)(-4,t)dt - \int_0^{1.2} \left(hu^2 + g\frac{h^2}{2}\right)(4,t)dt \\
&= 1.2 \times \left(\cancel{hu^2} + g\frac{h^2}{2}\right)(-4,0) - 1.2 \times \left(\cancel{hu^2} + g\frac{h^2}{2}\right)(4,0) \\
&= 1.2 \times \frac{g}{2}(h^2(-4,0) - h^2(4,0)) \\
&= 1.2 \times \frac{1}{2}(3^2 - 1^2) \text{ (considering } g = 1 \text{ in the problem)} \\
&= 4.8
\end{aligned} \tag{19}$$

These two values are found back using provided scripts and discrete formulas above, for $N_c = 200$ and $N_c = 1000$ for both Rusanov and Roe schemes. The numerical solution can accordingly be considered as physically conserving quantities h and hu.

In terms of efficiency, calculating the solution using the Roe scheme is faster than using Rusanov's for $N_c = 200$. A surprising result is the decreasing of duration when using a finer grid with $N_c = 1000$: this can either be due to a specific performance variation of the used computer (which should not be the case using the statistics methodology) or to the optimization depicted on Listing 3. Calculating the flux for all cells in the same routine for the Roe scheme may have optimized calculation for higher grid resolutions. Overall, using a more precise grid leads to an increase of the calculation time: for five times more grid cells, the total duration is multiplied by 15 in average, and the duration for one step increases by a factor of 3 in average. The calculation time is not linear with N_c, which should be taken into account when increasing the resolution.

3 System with topography source term

The shallow water problem (3) now contains a bottom topography term, i.e $\exists x \in I, b(x) \neq 0$. The previous scheme cannot be implemented as-is, otherwise physical aspects of water behavior would be omitted.

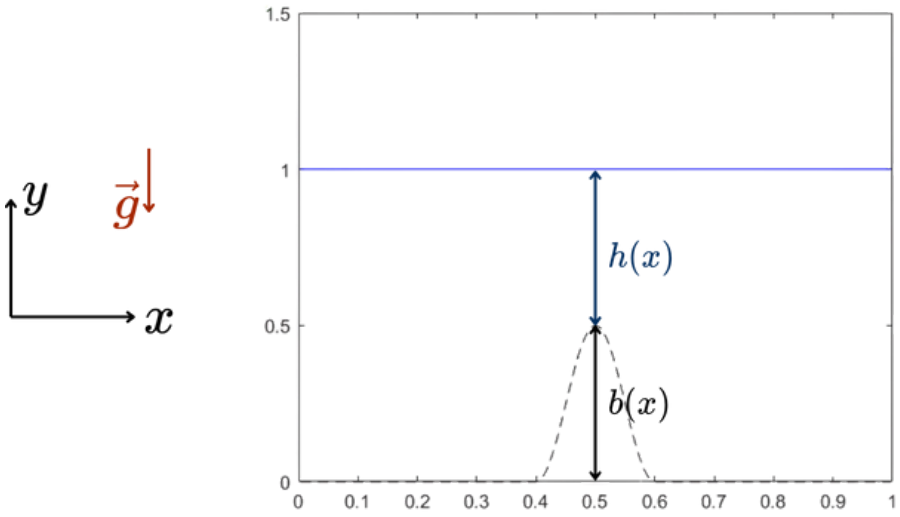


Figure 10: Illustration of the shallow water problem with a bottom topography.

The *lake at rest* condition must be satisfied with the bottom source term: physically, in a stationary state at rest ($\frac{\partial \cdot}{\partial t} = 0, u = 0$), the water level should be horizontal. System (3) becomes:

$$\cancel{\frac{\partial U}{\partial t}} + \begin{pmatrix} 0 & 1 \\ \cancel{-u^2} + gh & \cancel{2u} \end{pmatrix} \frac{\partial}{\partial x} \begin{pmatrix} h \\ \cancel{hu} \end{pmatrix} = \begin{pmatrix} 0 \\ -gh\frac{\partial b}{\partial x} \end{pmatrix} \Rightarrow gh\frac{\partial h}{\partial x} = -gh\frac{\partial b}{\partial x} \tag{20}$$

By simplifying the previous equation assuming $h \neq 0$ (if $h = 0$, no water is present so the "lake" is effectively at rest) and integrating over x, the lake at rest condition can be mathematically stated as:

$$\left(u = 0, \frac{\partial \cdot}{\partial t} = 0\right) \Rightarrow (h + b = \text{constant}) \tag{21}$$

For a non-stationary solution, i.e. $\frac{\partial \cdot}{\partial t} \neq 0$, the bottom topography is expected to impact water at the surface. This effect can be quantified by the Froude number $F_r = \frac{U}{\sqrt{gL}}$, which compares the effects of inertia and gravity, with U and L characteristic speeds and lengths of the system. If $F_r < 1$, the flow is *subcritical*, otherwise it is *supercritical*. In the following study, the perturbations of the flow are chosen so that $F_r < 1$: in this case, a positive bottom topography creates a bump in the water level above when a wave passes by.

3.1 Numerical implementation

To ensure the *lake at rest* condition is satisfied at the discrete level, a well-balanced scheme should be employed, as depicted in [1]. Property (21) corresponds to the discrete formulation: $(\forall i, u_i^n = 0, h_i^n + b_i = \text{constant}) \Rightarrow (\forall i, u_i^{n+1} = 0, h_i^{n+1} + b_i = \text{constant})$. For this purpose, the Hydrostatic Reconstruction scheme described in [2] and [1] is used: for a cell $\left[x_{i-1/2}, x_{i+1/2}\right]$ centered in x_i,

$$U_i^{n+1} = U_i^n - \frac{\Delta t}{\Delta x}\left[\tilde{F}_l(U_i^n, U_{i+1}^n, b_i, b_{i+1}) - \tilde{F}_r(U_{i-1}^n, U_i^n, b_{i-1}, b_i)\right] \tag{22}$$

where the numerical fluxes are defined as:

$$\tilde{F}_l(U_L, U_R, b_L, b_R) = F(U_L^*, U_R^*) + \begin{pmatrix} 0 \\ \frac{1}{2}g(h_L^2 - h_L^{*2}) \end{pmatrix} \tag{23}$$

$$\tilde{F}_r(U_L, U_R, b_L, b_R) = F(U_L^*, U_R^*) + \begin{pmatrix} 0 \\ \frac{1}{2}g(h_R^2 - h_R^{*2}) \end{pmatrix} \tag{24}$$

using

$$\begin{aligned} U_{L/R}^* &= \begin{pmatrix} h_{L/R}^* \\ h_{L/R}^* u_{L/R} \end{pmatrix}, \\ h_{L/R}^* &= \left(h_{L/R} + b_{L/R} - b^*\right)_+ = \max\left(h_{L/R} + b_{L/R} - b^*, 0\right) \\ \text{and } b^* &= \max(b_L, b_R) \end{aligned} \tag{25}$$

F is a consistent numerical flux, that is $F(U,U) = \mathcal{F}(U)\ \forall U$, for the system with no bottom source term ($b(x) = 0\ \forall x \in I$). F can be for instance the Rusanov flux implemented in Section 2.2.1, or the Roe flux depicted in Section 2.2.2. Other values are

This scheme should preserve the stationary states with zero velocity $u = 0$ and $h + b =$ constant. By calculating flux quantities of (22) at the interface between states U_L and U_R at time n (powers are omitted for clarity), using $u_R = u_L = 0$ and $h_L + b_L = h_R + b_L$,

$$\begin{aligned} & h_L^* = (h_L + b_L - b^*)_+ = (h_R + b_R - b^*)_+ = h_R^* \text{ and } u_R = u_L = 0 \\ & \Rightarrow U_L^* = U_R^* \\ & \Rightarrow F(U_L^*, U_R^*) = F(U_R^*, U_R^*) = \mathcal{F}(U_R^*) = \mathcal{F}(U_L^*) \text{ (consistent numerical scheme)} \\ & \Rightarrow \tilde{F}_l(U_L, U_R, b_L, b_R) = \mathcal{F}(U_L^*) + \begin{pmatrix} 0 \\ \frac{1}{2}g(h_L^2 - h_L^{*2}) \end{pmatrix} \\ & \quad = \begin{pmatrix} \cancel{h_L^* u_L} \\ \cancel{h_L^* u_L^2} + g\frac{h_L^{*2}}{2} \end{pmatrix} + \begin{pmatrix} 0 \\ \frac{1}{2}g(h_L^2 - h_L^{*2}) \end{pmatrix} = \begin{pmatrix} 0 \\ g\frac{h_R^2}{2} \end{pmatrix} = \begin{pmatrix} \cancel{h_L u_L} \\ \cancel{h_L u_L^2} + g\frac{h_L^2}{2} \end{pmatrix} = \mathcal{F}(U_L) \\ & \text{and } \tilde{F}_r(U_L, U_R, b_L, b_R) = \mathcal{F}(U_R) = \mathcal{F}(U_L) = \tilde{F}_l(U_L, U_R, b_L, b_R) \\ & \text{thus } U_{i+1}^n = U_i^n \end{aligned} \tag{26}$$

The Hydrostatic Reconstruction scheme is preserving the stationary states with zero velocity, under the assumption that F is a consistent numerical flux for the system with no bottom source term.

The same Finite Volume method as in Section 2 is used, and implemented in MATLAB following the same logic. The numerical flux is defined in a separate function for more readability, and called in the Finite Volume method loop. Codes are listed in Listing 4 and Listing 5.

```
[...]
for  istep=1:MaxStep
  w = qv;  % store old qv

  % update solution qv
  for i=3:Ncp2
    fluip = flux_hydrost(w(i,:), w(i+1,:), bot(i), bot(i+1), 'left'); % Left flux
    flui = flux_hydrost(w(i-1,:), w(i,:), bot(i-1), bot(i), 'right'); % Right flux
    qv(i,:) = w(i,:) - dt/h*(fluip-flui);
  end
  [boundary conditions]
end
[...]
```

Listing 4: MATLAB implementation of the main loop for the Finite Volume method. Initialization and plotting parts were omitted.

```
function [ff] = flux_hydrost(Ul, Ur, bl, br, side)
  % side = 'left' (flux calculation between i and i+1) or 'right' (flux calculation
between i-1 and i)
  grav = 1.;
  ff = zeros(1, 2);
  b_star = max(bl, br);
  hl = Ul(1);   % water height
  hul = Ul(2);  % momentum
  if(hl>0)
   ul = hul/hl;   % velocity
  else
   ul=0;  % set zero velocity if dry state
  end
  hl_star = max(hl + bl - b_star, 0);
  Ul_star = [hl_star, ul * hl_star];

  hr = Ur(1);   % water height
  hur = Ur(2);  % momentum
  if(hr>0)
   ur = hur/hr;   % velocity
  else
   ur=0;  % set zero velocity if dry state
  end
  hr_star = max(hr + br - b_star, 0);
  Ur_star = [hr_star, ur * hr_star];

  F_star = fluxswRSn_templ(Ul_star, Ur_star);
  if strcmp(side, 'left') % left flux
      ff = F_star + [0, 0.5*grav*(hl^2 - hl_star^2)];
  else % right flux
      ff = F_star + [0, 0.5*grav*(hr^2 - hr_star^2)];
  end
```

Listing 5: MATLAB function to compute the Hydrostatic Reconstruction numerical flux at the interface between cells i and $i+1$, using the Rusanov flux implemented on Listing 2.

3.2 Experimental results

The solution was computed using the Rusanov scheme with $N_c = 200$ grid cells, $\frac{\Delta t}{\Delta x} = 0.8$ and the topography depicted on Figure 10:

$$b(x) = \begin{cases} \frac{1}{4}\left(1 + \cos 10\pi\left(x - \frac{1}{2}\right)\right) & \text{if } \left|x - \frac{1}{2}\right| < \frac{1}{10} \\ 0 & \text{else} \end{cases} \tag{27}$$

The initial condition is taken as the lake at rest defined by $h + b = 1, u = 0$. A perturbation of the water height $h(x, 0) = 1 + \varepsilon$ is introduced for $0.1 < x < 0.2$ with $\varepsilon = 0.2$, and transmissive boundary conditions are applied on both sides. This problem was initially defined in [3], expected results at time $t = 0.7$ are shown on Figure 11. The left wave is expected to exit the domain by this time, while the right wave should stretch out and be mitigated, influenced by the bottom topography at $x = 0.5$.

Computed solutions are directly plotted on MATLAB and shown on Figure 12 and Figure 13. To get an insight at the influence of the grid size N_c, a calculation is performed by setting $N_c = 1000$ and shown on Figure 15, all other parameters remaining the same. Benchmarks of these configurations can be found in Table 2, with average durations per step calculated on 175 steps for $N_c = 200$ and 875 steps for $N_c = 1000$. As a comparison, numerical solutions computed for both Rusanov and Roe schemes are plotted jointly on Figure 14.

The velocity of the computed solution is comprised between 0.02 and 0.12, while water height can't exceed 1.3 due to the conservation of energy and the lack of a shore or a solid wall (the height of the perturbation being 1.2). Using $\frac{\Delta t}{\Delta x} = 0.8$, the Courant number is $\sigma 1 + 8 \times 10^{-3}$, so the CFL condition should be verified for the numerical solution considering that σ was calculated for a rough approximation with large boundaries.

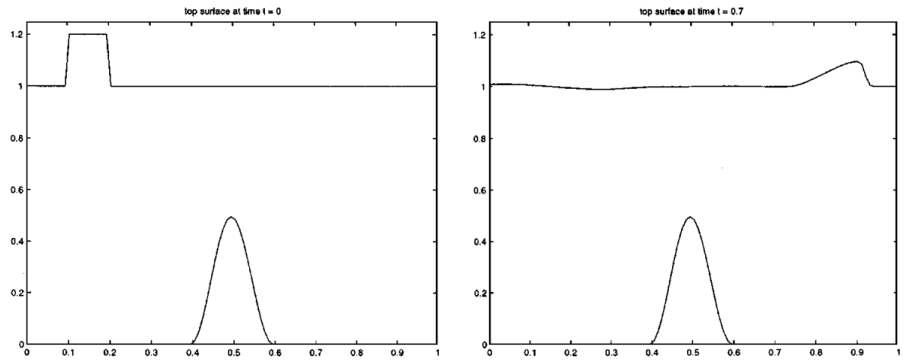
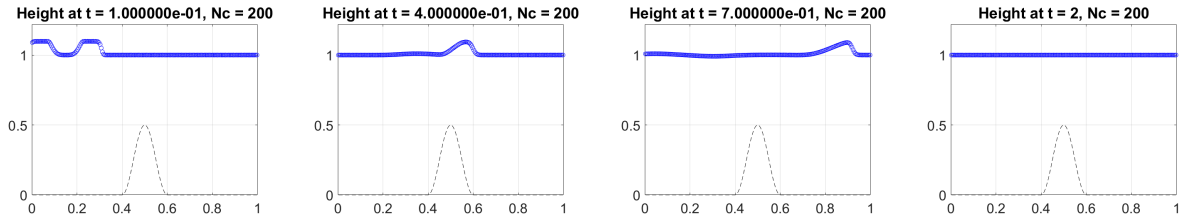
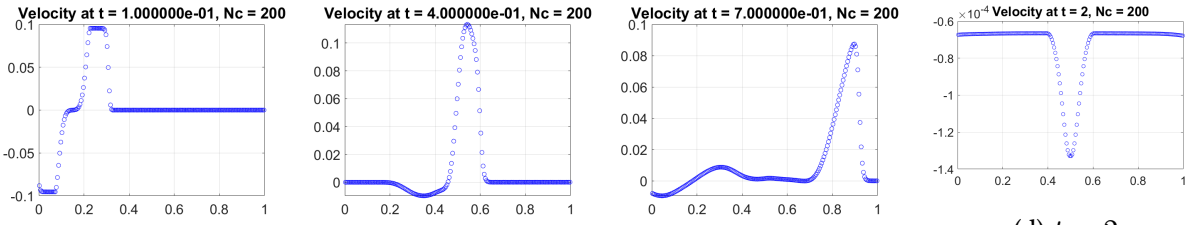


Figure 11: Figure from [3]: "Bottom topography and top surface for the one-dimensional shallow water equations in the case $\varepsilon = 0.2$. At time $t = 0.7$ the right-going portion of the pulse has moved past the hump."



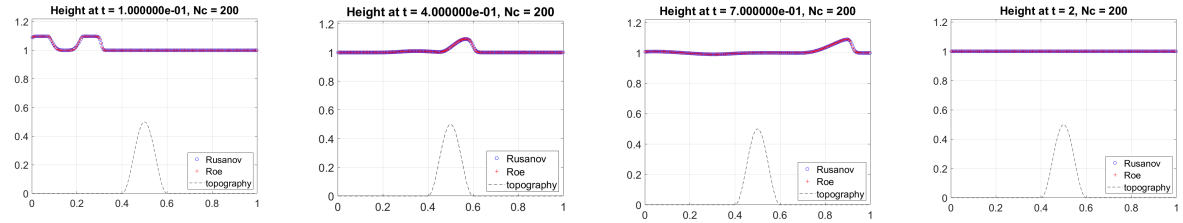
(a) $t = 0.1$ (b) $t = 0.4$ (c) $t = 0.7$ (d) $t = 2$

Figure 12: Computed water height h for the lake with topography problem using the Rusanov scheme, $N_c = 200$ grid cells and $\frac{\Delta t}{\Delta x} = 0.8$, plotted at different time steps. The dotted line represents the topography $b(x)$.



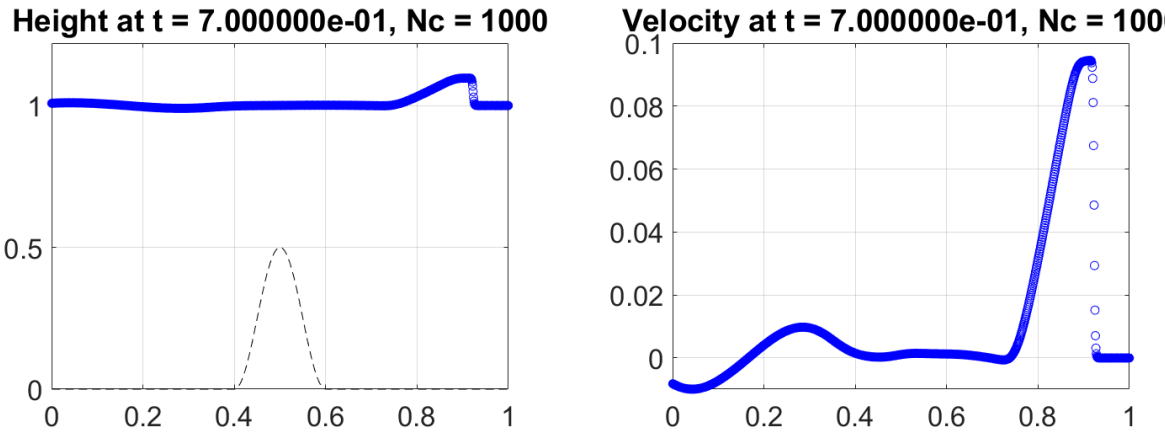
(a) $t = 0.1$ (b) $t = 0.4$ (c) $t = 0.7$ (d) $t = 2$

Figure 13: Computed water velocity v for the lake with topography problem using the Rusanov scheme, $N_c = 200$ grid cells and $\frac{\Delta t}{\Delta x} = 0.8$, plotted at different time steps. The dotted line represents the topography $b(x)$.



(a) $t = 0.1$ (b) $t = 0.4$ (c) $t = 0.7$ (d) $t = 2$

Figure 14: Computed water height h for the lake with topography problem using the Rusanov and Roe schemes, $N_c = 200$ grid cells and $\frac{\Delta t}{\Delta x} = 0.8$, plotted at different time steps. The dotted line represents the topography $b(x)$.



(a) Water height h (b) Water velocity u

Figure 15: Computed solution for the lake with topography problem using the Rusanov scheme, $N_c = 1000$ grid cells and $\frac{\Delta t}{\Delta x} = 0.8$, plotted at time $t = 0.7$. The dotted line represents the topography $b(x)$.

N_c	Duration for $t_f = 0.7$	Average duration per step
200	402.42 ± 22.29 ms	2.3 ± 0.13 ms
1000	7.67 ± 2.10^{-4} s	8.77 ± 2.10^{-4} ms

Table 2: Benchmarks for the computed solution for the lake with topography problem using the Rusanov Scheme. Mean and variance are calculated on 25 measured durations.

3.3 Discussion

As expected in [3], two symmetrical waves are first issued from the perturbation $1 + \varepsilon$. The left one quickly disappears to the side: at $t = 0.4$, it is no longer noticeable. No prints are left, due to the transmissive boundary condition: water can freely flow out of the domain, and as the topography remains flat without additional flow perturbation, these is no reason for the trajectory of the wave to be modified.

After passing the bottom hill, the geometry of the right wave is modified and appears more elongated: a mitigation due to the topography is added to the Shallow Water equations mitigation (tied to the rarefaction wave dynamics), so the speed of the wave goes progressively down as depicted on Figure 13. Moreover, a small reflected wave propagates to the left, hinting at the momentum conservation when the bottom water hits the topography. This can be seen on Figure 13 .d by the negative velocity. The forehead of the transmitted wave becomes straighter over time, i.e. it is being put upright after passing by the bottom topography. Intuitively, if the bottom topography goes high enough (for instance near a shore), the forehead would become vertical, leading to a discontinuity in the numerical model – even a non-entropic, un-physical solution – and to a wave break in the reality.

Finally, for $t = 2$, both transmitted and reflected waves have left the domain, leading to a lake almost at rest: a negligeable velocity can still be measured above the topography hill, as depicted on the last plot of Figure 13. The perturbation $1 + \varepsilon$ is still considered stable for long-enough times, as the lake tends to reach its initial equilibrium position $h + b = 1, u = 0$.

For this problem, the Roe scheme gives similar results as the Rusanov scheme and the differences are not noticeable, as depicted on Figure 14: both of them can be chosen without impact. Increasing the grid size by a factor of 5 produces a calculation time per step that is 3.8 times more important, for

a total duration multiplied by 19. However, with a grid size of $N_c = 1000$, the forehead of the right wave is best represented as seen on Figure 15 and is more accurate to the expected solution depicted on Figure 11 from [3].

4 Oscillating lake problem with dry regions

The numerical method implemented earlier is now tested on a new category of problems, which is a free-surface water mass constrained between two physical boundaries - all content stays within the domain. One may have experienced this problem in the daily life, for instance by shaking filled buckets or bowls. This problem can be stated as a free-surface problem with steady solid interfaces: pressure and velocity are varying along sloshing modes, which are linear combinations of sinusoidal functions.

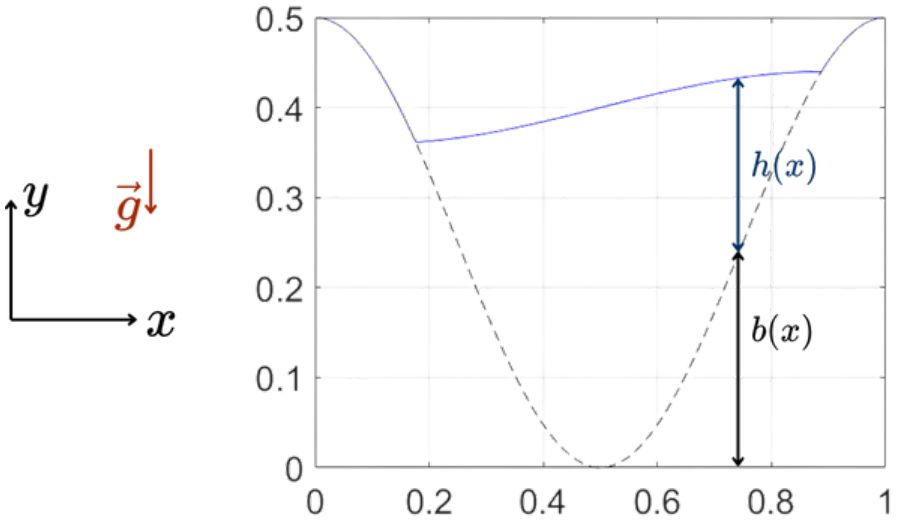


Figure 16: Illustration of the shallow water problem in a lake ; the water surface is expected to oscillate due to the mass conservation.

This problem contains dry regions in order to represent the shores, i.e. $h = 0$ at some points, a property that the Roe scheme might not handle well, as it can compute negative values of the flow height. The same numerical implementation as for Section 3 can therefore be used, but by only enabling the Rusanov scheme.

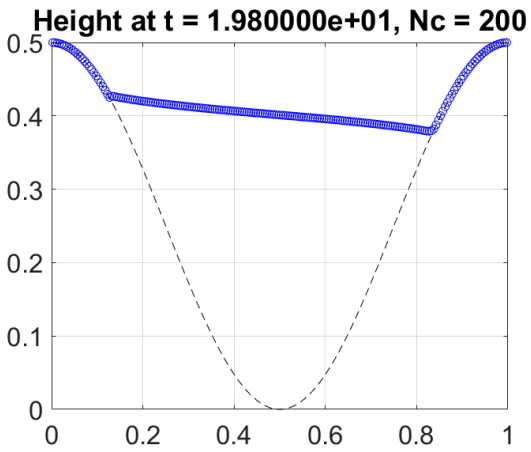
4.1 Experimental results

The problem is solved using the new topography $b(x) = \frac{1}{2}\left(1 - \frac{1}{2}\left(1 + \cos 2\pi\left(x - \frac{1}{2}\right)\right)\right)$ over the interval $[0, 1]$, which depicts the lake shown on Figure 16 and presents a smooth transition between the shore and the bottom. An initial sinusoidal perturbation of the free surface is taken, $\eta(x) = \frac{1}{25}\sin\left(\frac{x-0.5}{25}\right)$ centered at $(0, 0.4)$. The Hydrostatic Reconstruction scheme implementation Listing 5 with the Rusanov numerical flux is used on $N_c = 200$ grid cells and for $\frac{\Delta t}{\Delta x} = 0.8$.

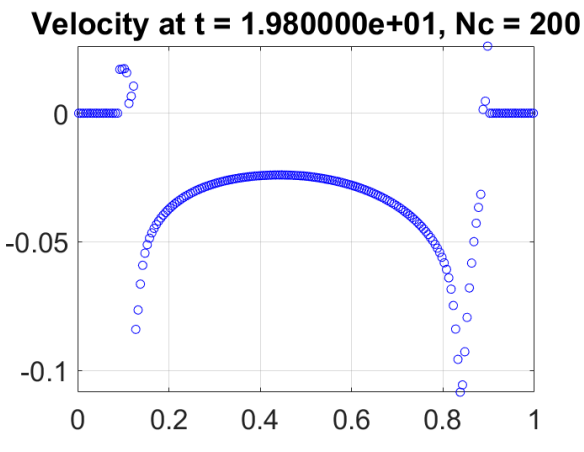
Dry regions on the sides are represented by a height of zero, numerical points $h + b$ thus follow the topography, and velocity is expected to be zero. The Rusanov scheme is theoretically able to handle data with vacuum, preserving the non-negativity of the flow height at the discrete level and ensuring the modelling of shores.

Results are shown on Figure 17. The total calculation time for $t_f = 19.8$, averaged on 25 measurements, is 8.69 ± 2.10^{-4} s, for an mean of 1.76 ± 2.10^{-4} ms per loop step, calculated for 4950 steps. A video film is also available to further investigate the dynamics of the free surface.

This problem can't be solved using the Roe scheme: indeed, when trying to compute the solution, values of `NaN` appear on the shores, and "propagate" to the whole domain due to the involved calculations, as depicted on Figure 18.



(a) Water height h



(b) Water velocity u

Figure 17: Computed solution for the oscillating lake problem, using the Rusanov scheme, $N_c = 200$ grid cells and $\frac{\Delta t}{\Delta x} = 0.8$, plotted at time $t = 19.8$. The dotted line represents the topography $b(x)$.

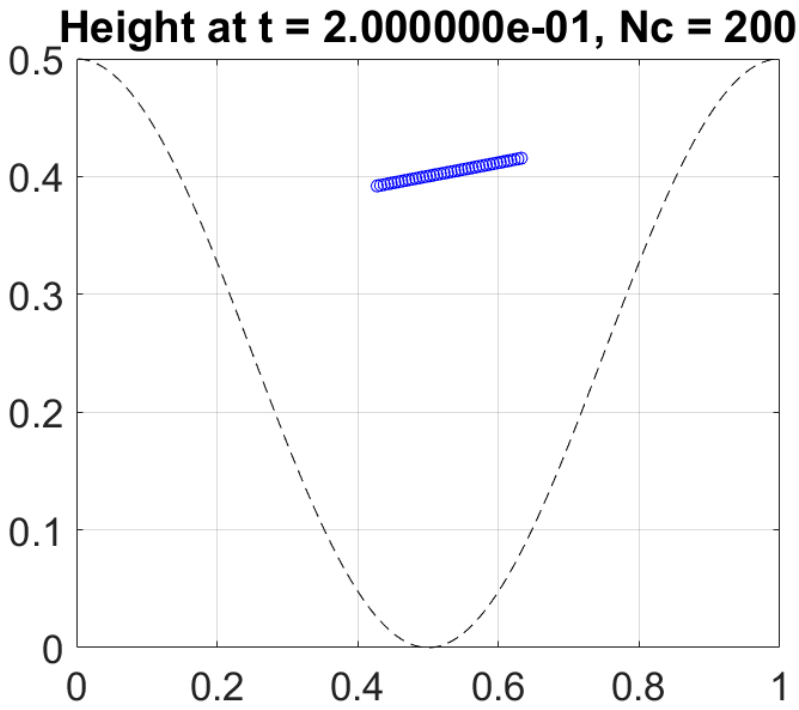


Figure 18: Computed solution for the oscillating lake problem, using the Roe scheme, $N_c = 200$ grid cells and $\frac{\Delta t}{\Delta x} = 0.8$, plotted at time $t = 0.2$. The dotted line represents the topography $b(x)$. Other values of $h(x)$ than showed on the plot are `NaN`.

4.2 Discussion

During the time interval $[0, t_f]$, the free surface oscillates in response to the initial perturbation and is not mitigated in time - oscillations continue over and over - thus is it unstable. These oscillations don't follow a sinusoidal form, as variations are driven by an infinite amount of sloshing modes. For the first times, before $t = 6$, the free surface is strongly oscillating, reaching height peaks on the shores and even creating surface depressions as the wave retracts before surging upward again. The oscillations are then mitigated over time, and the surface depression phenomenon disappears. This dynamics is coherent with the physical intuition for this problem and the everyday experience.

The maximum water height on the left shore is expected to be reached at time $t = 19.8$. However this is not observed in the numerical experiments: right after, water continues to rise on the shore, and higher levels are even reached before this time. This could be due to the numerical scheme implementation, that accumulates errors over time.

The velocity on the shore is mostly zero, excepted for a small region around the actual water flow. It even seems discontinuous on the left shore, brutally transitioning from a negative to a positive one. This could be due to the momentum conservation on the shores, or to values that are not correctly handled by the numerical scheme.

It is worth noticing that in the reality, waves may break on the shore, a dynamic that can't be represented using the hyperbolic system (3). Indeed, this model assumes that the fluid domain is made of full water columns, filled from 0 to h without intermediary holes.

5 Conclusion

In this study, the Saint-Venant equations for shallow water were solved numerically using a Finite Volume method: the conservative hyperbolic form allowed for the use of the Rusanov scheme, and the Roe scheme for Riemann problems. The Hydrostratic Reconstruction scheme ensured that the physical behavior of water took into account a varying bottom topography, in order to solve a wide range of problems.

The studied parameters were the grid resolution and the choice of scheme, and they were compared using qualitative observations as well as quantitative benchmarks of their performances. In general, Roe scheme is slightly more accurate than Rusanov's, at the price of a higher computation time. Moreover, refining the grid leads to a higher increase of duration, as it does not follow a linear law. Consequently, one should adjust the two parameters in order to find a balance between accuracy and efficiency, depending on the problem to solve and the practical use of the computed solution. One must also ensure that the convergence and stability criteria are met, which are not sufficient but still necessary to achieve a good numerical resolution.

Finally, as depicted in Section 2, one should always take a step back when interpreting a computed solution. Numerical schemes may fail to describe parts of the dynamics, and a code should always be tested and on a simpler problem with a known analytical solution, in order to precisely quantify sources of uncertainty and use these programs with full awareness.

References

[1] F. Bouchut, *Nonlinear Stability of Finite Volume Methods for Hyperbolic Conservation Laws and Well-Balanced Schemes for Sources*. 2004, p. . doi: 10.1007/b93802.

[2] E. Audusse, F. Bouchut, M.-O. Bristeau, R. Klein, and B. Perthame, "A Fast and Stable Well-Balanced Scheme with Hydrostatic Reconstruction for Shallow Water Flows," *SIAM Journal on Scientific Computing*, vol. 25, no. 6, pp. 2050–2065, 2004, doi: 10.1137/S1064827503431090.

[3] R. J. LeVeque, "Balancing Source Terms and Flux Gradients in High-Resolution Godunov Methods: The Quasi-Steady Wave-Propagation Algorithm," *Journal of Computational Physics*, vol. 146, no. 1, pp. 346–365, 1998, doi: https://doi.org/10.1006/jcph.1998.6058.