Finite Volume Methods for the 1D Shallow Water Equations

Yasmina Elmore

March 10, 2025

1 1d Shallow Water Equations

We consider the following 1D Shallow water flow configuration :

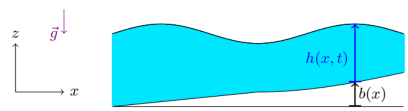


Figure 1: Shallow water flow configuration

The configuration verifies the Saint-Venant equations (Shallow water equations) that can be written in their conservative form :

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

where :

$$U = \begin{bmatrix} h \\ hu \end{bmatrix} = \begin{bmatrix} U_1 \\ U_2 \end{bmatrix}$$

$$\mathrm{F(U)} = \begin{bmatrix} hu \\ hu^2 + \frac{1}{2}gh^2 \end{bmatrix} = \begin{bmatrix} U_2 \\ \frac{U_2^2}{U_1} + \frac{1}{2}gU_1^2 \end{bmatrix} = \begin{bmatrix} \mathcal{F}_1 \\ \mathcal{F}_2 \end{bmatrix}$$

$$\Psi = \begin{bmatrix} 0 \\ -gh\dfrac{\partial b}{\partial t} \end{bmatrix}$$

Here, h refers to the flow depth, b to the bottom topography, and u the velocity along the x direction.

2 Homogeneous system

We assume that $b(x,t) = 0$, which means that the bottom doesn't present any "bumps." Therefore we can linearize the 1D Shallow water equations :

$$\frac{\partial \mathcal{U}}{\partial t} + \mathcal{A}(\mathcal{U})\frac{\partial \mathcal{U}}{\partial x} = 0 \tag{2}$$

Where $\mathcal{A}(\mathcal{U})$ is the Jacobian matrix. Thus,

$$\mathcal{A}(\mathcal{U}) = \begin{bmatrix} \dfrac{\partial \mathcal{F}_1}{\partial U_1} & \dfrac{\partial \mathcal{F}_1}{\partial U_2} \\ \dfrac{\partial \mathcal{F}_2}{\partial U_1} & \dfrac{\partial \mathcal{F}_2}{\partial U_2} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{U_2^2}{U_1^2} + gU_1 & 2\frac{U^2}{U^1} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -u^2 + gh & 2u \end{bmatrix}$$

We can easily deduce the eigenvalues of the Jacobian matrix. Indeed, we can show that $Tr(\mathcal{A}(\mathcal{U})) = 2u$ and $det(\mathcal{A}(\mathcal{U})) = u^2 - gh$. Thus the eigenvalues are the solutions of the polynomial:

$$\lambda^2 - Tr(\mathcal{A}(\mathcal{U}))\lambda + det(\mathcal{A}(\mathcal{U})) = 0$$

$$\Rightarrow \lambda^2 - 2u\lambda + (u^2 - gh) = 0$$

$$\Rightarrow \lambda^2 - ((u + \sqrt{gh}) + (u - \sqrt{gh}))\lambda + (u - \sqrt{gh})(u + \sqrt{gh}) = 0$$

We can deduce the two values of the eigenvalues :

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

And the corresponding eigenvectors :

$$r_1(\mathcal{U}) = \begin{bmatrix} 1 \\ u - \sqrt{gh} \end{bmatrix} \quad \text{and} \quad r_2(\mathcal{U}) = \begin{bmatrix} 1 \\ u + \sqrt{gh} \end{bmatrix}$$

That can be rewritten using U_1 and U_2 :

$$r_1(\mathcal{U}) = \begin{bmatrix} 1 \\ \frac{U_2}{U_1} - \sqrt{gU_1} \end{bmatrix} \quad \text{and} \quad r_2(\mathcal{U}) = \begin{bmatrix} 1 \\ \frac{U_2}{U_1} + \sqrt{gU_1} \end{bmatrix}$$

The eigenvalues are real for $h > 0$ and distinct. The eigenvectors associated with each eigenvalue are independent. Therefore, the linear system is strictly hyperbolic.

Now, we want to show that the characteristic fields are genuinely nonlinear, which means that $\nabla\lambda_k(\mathcal{U}) \cdot r_k(\mathcal{U}) \neq 0$. Thus, we need to calculate $\nabla\lambda_k(\mathcal{U})$.

$$\nabla\lambda_1(\mathcal{U}) = \begin{bmatrix} \dfrac{\partial\lambda_1}{\partial U_1} \\ \dfrac{\partial\lambda_1}{\partial U_2} \end{bmatrix} \text{ and } \nabla\lambda_2(\mathcal{U}) = \begin{bmatrix} \dfrac{\partial\lambda_2}{\partial U_1} \\ \dfrac{\partial\lambda_2}{\partial U_2} \end{bmatrix} \tag{3}$$

Thus,

$$\nabla\lambda_1(\mathcal{U}) = \begin{bmatrix} \frac{-U_2}{U_1^2} - \frac{1}{2}\sqrt{\frac{g}{U_1}} \\ \frac{1}{U_1} \end{bmatrix} = \begin{bmatrix} \frac{-u}{h} - \frac{1}{2}\sqrt{\frac{g}{h}} \\ \frac{1}{h} \end{bmatrix}$$

and

$$\nabla\lambda_2(\mathcal{U}) = \begin{bmatrix} \frac{-U_2}{U_1^2} + \frac{1}{2}\sqrt{\frac{g}{U_1}} \\ \frac{1}{U_1} \end{bmatrix} = \begin{bmatrix} \frac{-u}{h} + \frac{1}{2}\sqrt{\frac{g}{h}} \\ \frac{1}{h} \end{bmatrix}$$

From these equations, we can deduce that:

$$\nabla\lambda_1(\mathcal{U}) \cdot r_1(\mathcal{U}) = \begin{bmatrix} \frac{-u}{h} - \frac{1}{2}\sqrt{\frac{g}{h}} \\ \frac{1}{h} \end{bmatrix} \cdot \begin{bmatrix} 1 \\ u - \sqrt{gh} \end{bmatrix}$$

$$\Rightarrow \nabla\lambda_1(\mathcal{U}) \cdot r_1(\mathcal{U}) = \frac{-u}{h} - \frac{1}{2}\sqrt{\frac{g}{h}} + (\frac{u}{h} - \sqrt{\frac{g}{h}})$$

$$\Rightarrow \nabla\lambda_1(\mathcal{U}) \cdot r_1(\mathcal{U}) = -\frac{3}{2}\sqrt{\frac{g}{h}} \neq 0$$

and

$$
\nabla\lambda_1(\mathcal{U})\cdot r_1(\mathcal{U}) = \begin{bmatrix} \frac{-u}{h}+\frac{1}{2}\sqrt{\frac{g}{h}} \\ \frac{1}{h} \end{bmatrix} \cdot \begin{bmatrix} 1 \\ u+\sqrt{gh} \end{bmatrix}
$$

$$
\Rightarrow \nabla\lambda_1(\mathcal{U})\cdot r_1(\mathcal{U}) = \frac{-u}{h}+\frac{1}{2}\sqrt{\frac{g}{h}}+(\frac{u}{h}+\sqrt{\frac{g}{h}})
$$

$$
\Rightarrow \nabla\lambda_1(\mathcal{U})\cdot r_1(\mathcal{U}) = \frac{3}{2}\sqrt{\frac{g}{h}} \neq 0
$$

Thus, the two characteristic fluids can be considered nonlinear.
If we consider now that the shallow water equations have two states: the left state (U_l, h_l) and the right state (U_r, h_l), the possible solutions of this Riemann problem can be choc waves or expansion waves. There is a choc wave if $U_l > U_r$ and an expansion wave if $U_l < U_r$.

3 Dam-break problem

3.1 Discretization

We want to simulate the problem with Riemann problem initial conditions. For $b(x,t) = 0$, the hyperbolic system can be implemented using the following finite volume conservative scheme :

$$
\mathcal{U}_i^{n+1} = \mathcal{U}_i^n - \frac{\Delta t}{\Delta x}[F(\mathcal{U}_i^n, \mathcal{U}_{i+1}^n) - F(\mathcal{U}_{i-1}^n, \mathcal{U}_i^n)] \tag{4}
$$

3.2 Rusanov method

We first solve the dam-break problem by implementing the Rusanov method on MATLAB. The following equation gives the numerical flux of the Rusanov method schemequation :

$$
F(\mathcal{U}_i, \mathcal{U}_{i+1}) = \frac{1}{2}(\mathcal{F}(\mathcal{U}_i) + \mathcal{F}(\mathcal{U}_{i+1}) - S_{i+1/2}(\mathcal{U}_{i+1} - \mathcal{U}_i)) \tag{5}
$$

where $S_{i+1/2} = max(|\lambda_{1,i}|, |\lambda_{2,i}|, |\lambda_{1,i+1}|, |\lambda_{2,i+1}|)$

```
% set f(u_i)
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)*(rl)/2);
```

(a) Definition of $\mathcal{F}(\mathcal{U}_i)$

```
% set f(u_{i+1})
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)*(rr)/2);

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

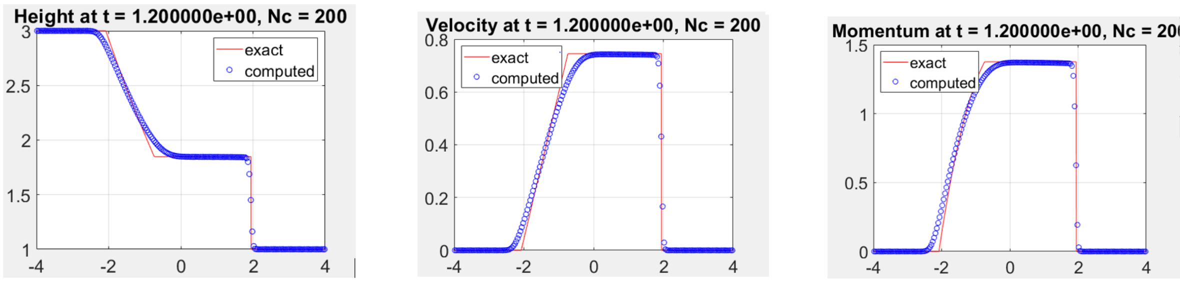
(b) Definition of $\mathcal{F}(\mathcal{U}_{i+1})$

Figure 2: Definition of the flux

```
% Rusanov numerical flux

lambda1r= abs(ur-sqrt(grav*rr));
lambda2r= abs(ur+sqrt(grav*rr));
srmax=max(lambda1r,lambda2r);
lambda1l=abs(ul-sqrt(grav*rl));
lambda2l= abs(ul+sqrt(grav*rl));
slmax=max(lambda1l,lambda2l);
S=max(slmax,srmax);

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

Figure 3: Rusanov numerical flux

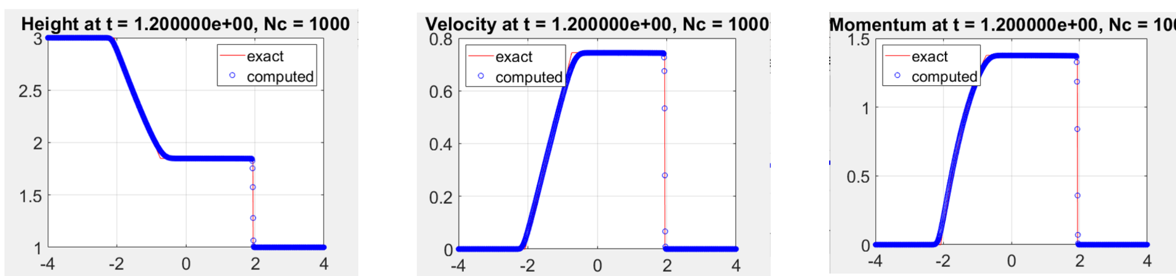
For a number of grid cells of 200, we have the following graphs of the exact and numerical solutions.



(a) Height of the solutions (b) Velocity of the solutions (c) Momemtum of the solutions

Figure 4: Graphics of the solutions

For a number of grid cells of 1000 :



(a) Height of the solutions (b) Velocity of the solutions (c) Momemtum of the solutions

Figure 5: Graphics of the solutions

We realize that the higher the number of grid cells is, the "closer" we are to the exact solution.

3.3 HLL method

We realize the same strategy but are using the HLL method this time. The numerical flow can be defined as follows:

$$F(U_i, U_{i+1}) = \begin{cases} \mathcal{F}(U_i) & \text{if } 0 \leq S^L_{i+1/2}, \\ \dfrac{S^R_{i+1/2}\mathcal{F}(U_i) - S^L_{i+1/2}\mathcal{F}(U_{i+1}) + S^L_{i+1/2}S^R_{i+1/2}(U_{i+1} - U_i)}{S^R_{i+1/2} - S^L_{i+1/2}} & \text{if } S^L_{i+1/2} < 0 < S^R_{i+1/2}, \\ \mathcal{F}(U_{i+1}) & \text{if } 0 \geq S^R_{i+1/2}, \end{cases}$$

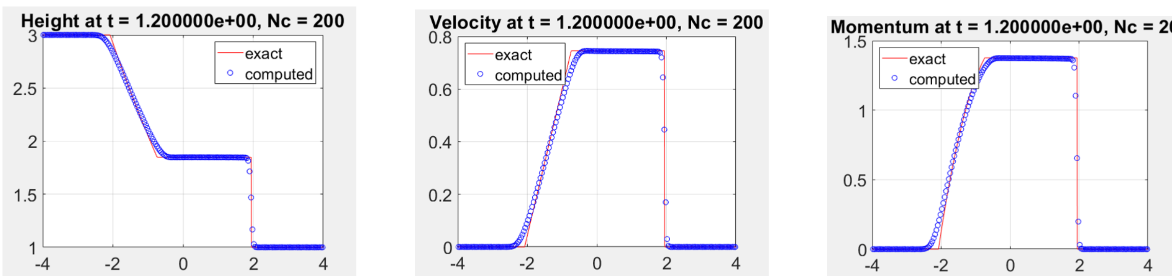
where $S^L_{i+1/2} = min(\lambda_{1,i}, \lambda_{1,i+1})$ and $S^R_{i+1/2} = min(\lambda_{2,i}, \lambda_{2,i+1})$

By implementing it on MATLAB :

```
% HLL numerical flux
lambda1l=ul-sqrt(grav*rl);
lambda1r=ur-sqrt(grav*rr);
lambda2l=ul+sqrt(grav*rl);
lambda2r=ur+sqrt(grav*rr);
sl = min(lambda1l,lambda1r);
sr = max(lambda2l,lambda2r);

if (sl >=0)
   ff = flul;
elseif (sr <=0)
   ff = flur;
else
 ff = (sr*flul -sl*flur +sl*sr*(vp-v))/(sr-sl);
```

Figure 6: HLL numerical flux

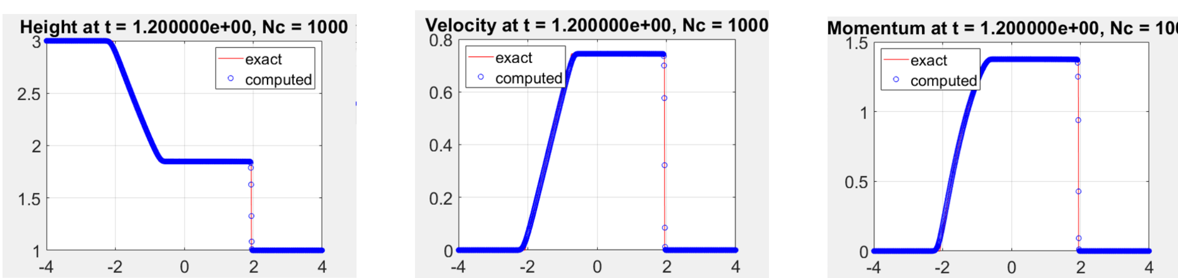
For a number of grid cells of 200, we have the following graphs of the exact and numerical solutions.



(a) Height of the solutions (b) Velocity of the solutions (c) Momemtum of the solutions

Figure 7: Graphics of the solutions

For a number of grid cells of 1000 :



(a) Height of the solutions (b) Velocity of the solutions (c) Momemtum of the solutions

Figure 8: Graphics of the solutions

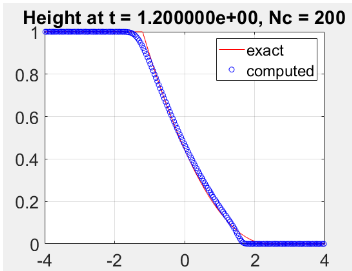
Just like in the case of the Rusanov method, we realize that the higher the number of grid cells is, the "closer" we are to the exact solution.

3.4 Dam-break Problem with right dry state

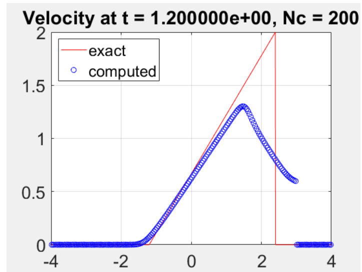
Likewise, we use the HLL and Rusanov methods to solve this dam-break problem.

Rusanov method:

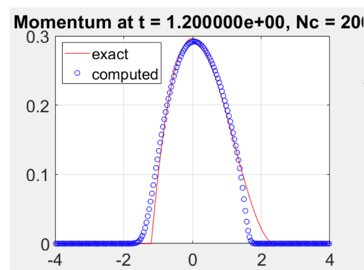
For a number of grid cells of 200, we have the following graphs of the exact and numerical solutions.



(a) Height of the solutions



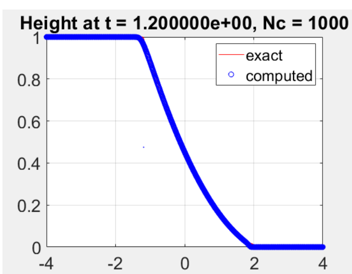
(b) Velocity of the solutions



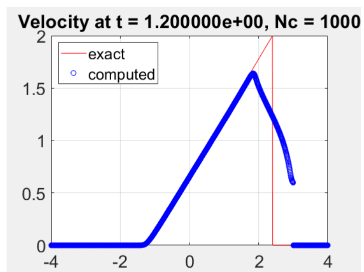
(c) Momemtum of the solutions

Figure 9: Graphics of the solutions

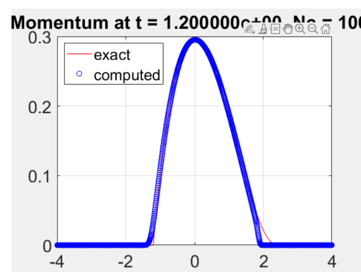
For a number of grid cells of 1000 :



(a) Height of the solutions



(b) Velocity of the solutions

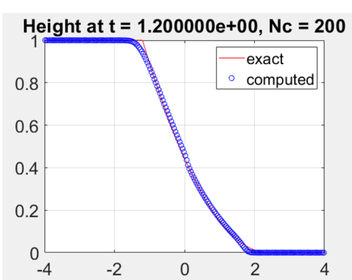


(c) Momemtum of the solutions

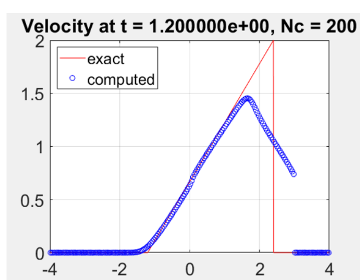
Figure 10: Graphics of the solutions

HLL method:

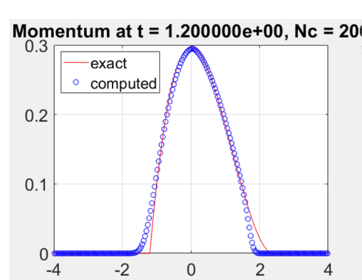
For a number of grid cells of 200, we have the following graphs of the exact and numerical solutions.



(a) Height of the solutions



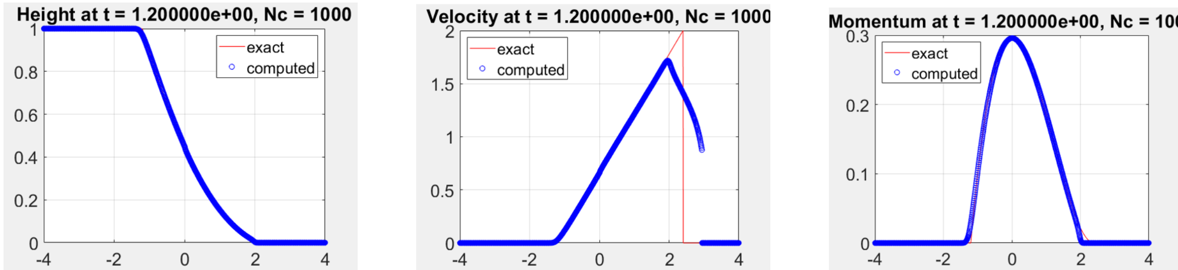
(b) Velocity of the solutions



(c) Momemtum of the solutions

Figure 11: Graphics of the solutions

For a number of grid cells of 1000 :



(a) Height of the solutions (b) Velocity of the solutions (c) Momemtum of the solutions

Figure 12: Graphics of the solutions

Like the last question, we realize that the higher the number of grid cells, the closer we are to the exact solution. However, we easily realize also that, in this case, the HLL method is more adapted than the Rusanov method by looking at the Velocity graphs. Indeed, the numerical solution based on the HLL method "follows" more the real solution of the equations. However, due to the discontinuity of the real solution, both solutions can't "reach" the top for the velocity, and there is a jump. So in both cases, the numerical solution calculated is very different from the real solution.

4 System with topography source term

In this first section, we want to show that the stationary states at rest imply the condition: $h+b = C_{ste}$.

Indeed, we consider the 1D shallow water equations :

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

$\Rightarrow \dfrac{\partial \mathcal{F}(\mathcal{U})}{\partial x} = \Psi$ because $\dfrac{\partial (.)}{\partial t} = 0$

$\Rightarrow \frac{\partial}{\partial x}\begin{bmatrix} hu \\ hu^2 + \frac{1}{2}gh^2 \end{bmatrix} = \begin{bmatrix} 0 \\ -gh\dfrac{\partial b}{\partial x} \end{bmatrix}$

$\Rightarrow \frac{\partial}{\partial x}\begin{bmatrix} 0 \\ \frac{1}{2}gh^2 \end{bmatrix} = \begin{bmatrix} 0 \\ -gh\dfrac{\partial b}{\partial x} \end{bmatrix}$ because $u = 0$

$\Rightarrow \dfrac{\partial}{\partial x}(\frac{gh^2}{2}) = -gh\dfrac{\partial b}{\partial x}$

$\Rightarrow \frac{h}{2} \times 2g\dfrac{\partial h}{\partial x} = -gh\dfrac{\partial b}{\partial x}$

$\Rightarrow \dfrac{\partial (h+b)}{\partial x} = 0$

$\Rightarrow h + b = C_{ste}$

Thus, we can say that :

$$h + b = C_{ste} \tag{6}$$

We assume now that $u_L = u_R = 0$, $b^* = max(b_L, b_R)$ and $h_L + b_L = h_R + b_R$.

First, we will show that $U_L^* = U_R^*$

$h_R^* = max(h_L + b_L - b^*)$

$\Rightarrow h_R^* = max(h_R + b_R - max(b_L, b_R))$

$\Rightarrow h_R^* = max(h_L + b_L - max(b_L, b_R))$ because $h_L + b_L = h_R + b_R$

$\underline{\Rightarrow h_R^* = h_L^*}$

And, $u_R = u_L = 0 \;\Rightarrow\; h_L^* u_L = h_R^* u_R = 0$

Thus, $U_L^* = (h_L^*, h_L^* u_L)$

$\Rightarrow U_L^* = (h_L^*, 0)$

$\Rightarrow U_L^* = (h_R^*, 0)$

$\Rightarrow U_L^* = (h_R^*, h_R^* u_R)$

$\underline{\Rightarrow U_L^* = U_R^*}$

We want to show now that: $\tilde{F_l} = \mathcal{F}(U_L)$ and $\tilde{F_r} = \mathcal{F}(U_R)$. We will show how to demonstrate the first equality. The second can be deduced using the same method, U_L* and U_R* playing symmetrical roles.

$$\tilde{F_l} = F(U_L^*, U_R^*) + \begin{bmatrix} 0 \\ \frac{1}{2}g(h_L^2 - h_L^{*2}) \end{bmatrix}$$

$$\Rightarrow \tilde{F_l} = F(U_L^*, U_L^*) + \begin{bmatrix} 0 \\ \frac{1}{2}g(h_L^2 - h_L^{*2}) \end{bmatrix}$$

$$\Rightarrow \tilde{F_l} = \mathcal{F}(U_L^*) + \begin{bmatrix} 0 \\ \frac{1}{2}g(h_L^2 - h_L^{*2}) \end{bmatrix}$$

$$\Rightarrow \tilde{F_l} = \begin{bmatrix} h_L^* u_L \\ h_L^* u_L^2 + \frac{1}{2}g h_L^{*2} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{2}g(h_L^2 - h_L^{*2}) \end{bmatrix}$$

$$\Rightarrow \tilde{F_l} = \begin{bmatrix} 0 \\ \frac{1}{2}g h_L^{*2} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{2}g(h_L^2 - h_L^{*2}) \end{bmatrix} \text{ because } u_L = 0$$

$$\Rightarrow \tilde{F_l} = \begin{bmatrix} 0 \\ \frac{1}{2}g h_L^2 \end{bmatrix}$$

$$\Rightarrow \tilde{F_l} = \begin{bmatrix} h_L u_L \\ h_L u_L^2 + \frac{1}{2}g(h_L^2 - h_L^{*2}) \end{bmatrix} \text{ because } u_L = 0$$

$$\Rightarrow \tilde{F_L} = \mathcal{F}(U_L)$$

Thus,

$$\tilde{F_L} = \mathcal{F}(U_L) \quad \text{and} \quad \tilde{F_R} = \mathcal{F}(U_R) \tag{7}$$

Therefore, the scheme preserves the stationary states with zero velocity u=0.

We can implement the scheme using the Rusanov method for different final times.

```
for i=3:Ncp2
   hRp=w(i+1,1);
   hLp=w(i,1);
   bRp=bot(i+1);
   bLp=bot(i);
   b_etoilep=max(bLp,bRp);
   hR_etoilep=max(hRp+bRp-b_etoilep,0);
   hL_etoilep=max(hLp+bLp-b_etoilep,0);

   hR=w(i,1);
   hL=w(i-1,1);
   bR=bot(i);
   bL=bot(i-1);
   b_etoile=max(bL,bR);
   hR_etoile=max(hR+bR-b_etoile,0);
   hL_etoile=max(hL+bL-b_etoile,0);

   %we want to define UL_etoile and UR_etoile
   %for that, we need to calculate uL and uR

   %we initialize the zero velocity for dry state everywhere
   uLp=0;
   uRp=0;

   uL=0;
   uR=0;
```

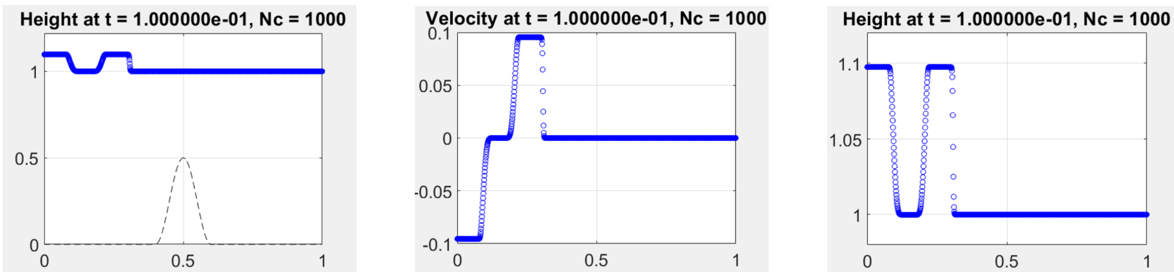
Figure 13: MATLAB code for the numerical solution

```
%and we run the test to know if we are really in the dry state
%if we are not (h>0), we deduce the value from the second column of
%w
if (hRp>0)
    uRp=w(i+1,2)/hRp;
end
if (hLp>0)
    uLp=w(i,2)/hLp;
end

if (hR>0)
    uR=w(i,2)/hR;
end
if (hL>0)
    uL=w(i-1,2)/hL;
end

%we now can write the expression of UL_etoile et UR_etoile
UL_etoilep=[hL_etoilep, uLp*hL_etoilep];
UR_etoilep=[hR_etoilep, uRp*hR_etoilep];

UL_etoile=[hL_etoile, uL*hL_etoile];
UR_etoile=[hR_etoile, uR*hR_etoile];

%we can now calculate fluip and flui
fluip = fluxswRSn(UL_etoilep,UR_etoilep)+[0,0.5*grav*(hLp*hLp-hL_etoilep*hL_etoilep)];
flui = fluxswRSn(UL_etoile,UR_etoile)+[0,0.5*grav*(hR*hR-hR_etoile*hR_etoile)];
```

Figure 14: MATLAB code for the numerical solution

For $t = 0.1$:

For a number of grid cells of 1000, we have the following graphs of the exact and numerical solutions.

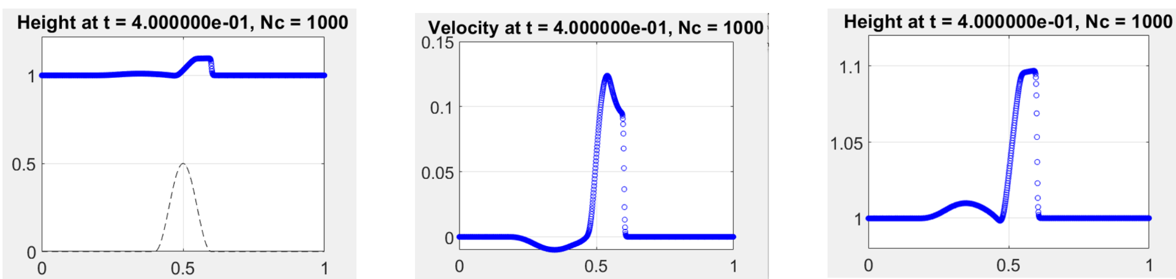


(a) Graph of the phenomenon (b) Velocity of the solutions (c) Height of the solutions

Figure 15: Graphics of the solutions

For $t = 0.4$:

For a number of grid cells of 1000, we have the following graphs of the exact and numerical solutions.

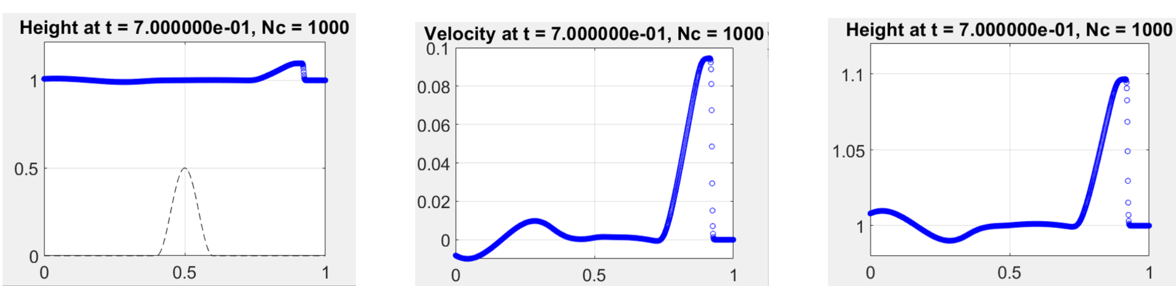


(a) Graph of the phenomenon (b) Velocity of the solutions (c) Height of the solutions

Figure 16: Graphics of the solutions

For $t = 0.7$:

For a number of grid cells of 1000, we have the following graphs of the exact and numerical solutions.

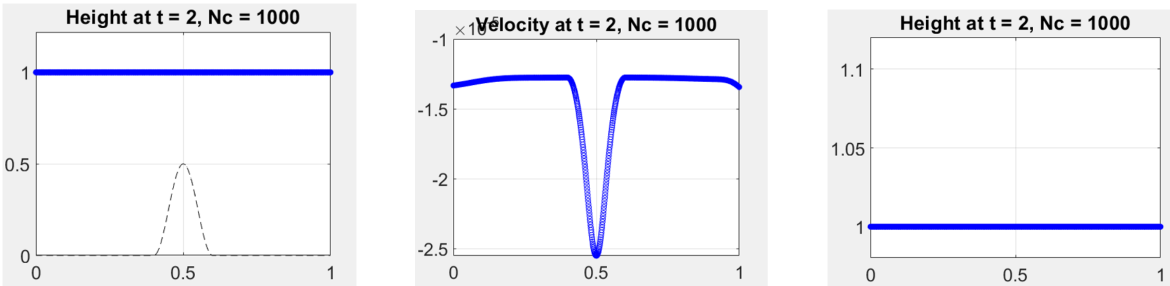


(a) Graph of the phenomenon (b) Velocity of the solutions (c) Height of the solutions

Figure 17: Graphics of the solutions

For $t = 2$:

For a number of grid cells of 1000, we have the following graphs of the exact and numerical solutions.



(a) Graph of the phenomenon (b) Velocity of the solutions (c) Height of the solutions

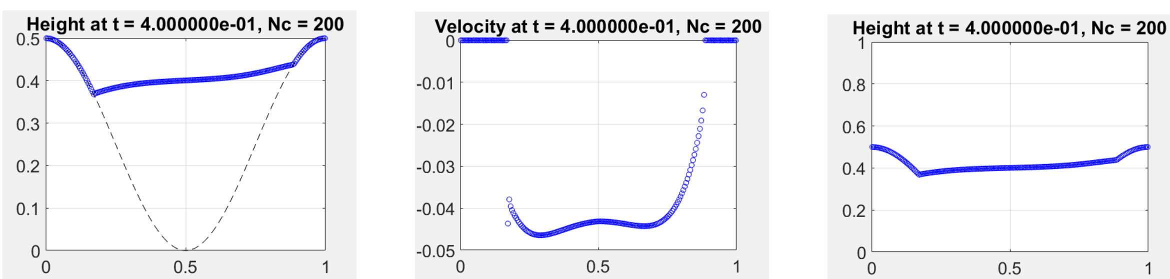
Figure 18: Graphics of the solutions

We can see that, in the end, the solution stabilizes itself and that the conditions are steady.

We now change the conditions and apply again the Rusanov method to simulate the wave propagation.

For $t = 0.4$:

For a number of grid cells of 200, we have the following graphs of the exact and numerical solutions.

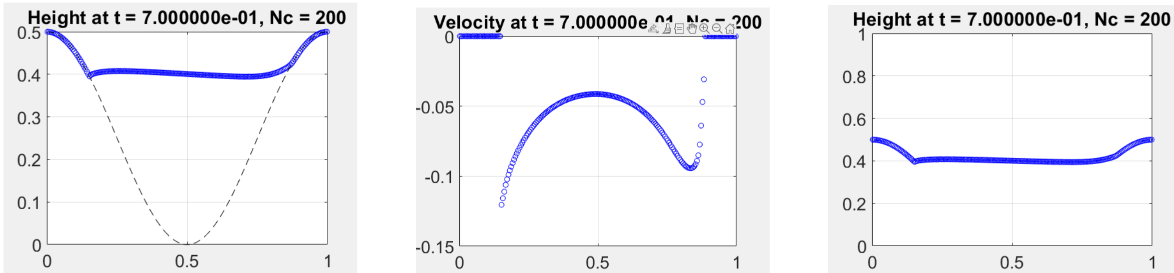


(a) Graph of the phenomenon (b) Velocity of the solutions (c) Height of the solutions

Figure 19: Graphics of the solutions

For $t = 0.7$:

For a number of grid cells of 200, we have the following graphs of the exact and numerical solutions.

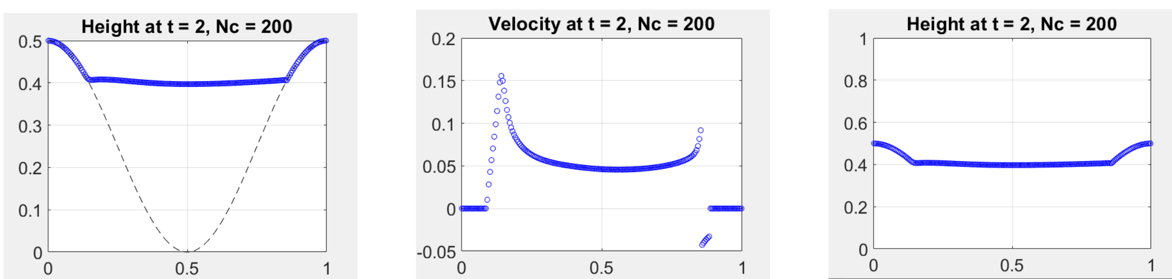


(a) Graph of the phenomenon (b) Velocity of the solutions (c) Height of the solutions

Figure 20: Graphics of the solutions

<u>For $t = 2$:</u>

For a number of grid cells of 200, we have the following graphs of the exact and numerical solutions.



(a) Graph of the phenomenon (b) Velocity of the solutions (c) Height of the solutions

Figure 21: Graphics of the solutions