NTNU
Kunnskap for en bedre verden

Department of Mathematical Sciences

TMA4500 - Specialization Project

Scalar Conservation Laws and Shallow-Water Simulations with Vertical Obstacles

Author:
Martin Saue Winther

Supervisors:
Knut-Andreas Lie, Håvard Heitlo Holm and Kjetil Olsen Lye

December, 2024

Contents

1 Introduction

As climate change intensifies, floods are getting more frequent and severe. Such floods are not solely due to rivers overflowing in rural areas, but also due to intense rain. It has therefore become increasingly important with preventive measures such as levees and flood-walls. To that end, it is crucial to know where water will flood for the measures to have the greatest effects. One can simulate the flow using the shallow-water equations over a varying bottom topography with finite-volume methods (FVMs).

Two key properties are essential for these schemes. First, the lake-at-rest solution must represent a stable equilibrium state. Second, the water height must remain positive at all times. Schemes that satisfy these properties are referred to as well-balanced and positivity-preserving, respectively. However, only a limited number of proposed methods successfully achieve both properties simultaneously while maintaining computational efficiency. Kurganov and Petrova (2007) introduced a second-order scheme that fulfills these requirements, except in near-dry states. In such cases, the scheme remains positivity-preserving, but artifacts near dry beds become increasingly pronounced, particularly in regions with steep bathymetry.

Modelling floods in urban areas requires an accurate representation of buildings and other vertical boundaries within the computational domain. Representing these features through the bathymetry often results in numerous steep wet-dry transitions, which can negatively impact both the accuracy and computational efficiency of the simulation. Alternatively, these obstacles could be incorporated into the internal boundary conditions, treating them as immovable and impermeable walls. This approach ensures that the scheme retains its positivity-preserving and well-balanced properties for these cases.

The shallow-water equations represent a system of conservation laws in which mass and momentum are conserved. However, the theoretical understanding of such systems remains incomplete. Therefore, the first part of this project aims to develop intuition by examining the scalar case and its numerical approximations through finite-volume methods (FVMs), as outlined in the lecture notes by Mishra et al. (2017). Subsequently, the project will focus on the challenges posed by walls and buildings within the computational domain of shallow-water simulations. In particular, I will implement and evaluate an alternative method for handling such boundaries and compare its performance with the approach proposed by Kurganov and Petrova (2007).

The scheme described by Kurganov and Petrova (2007)has been implemented by SINTEF Digital within the SinFVM.jl package. This will be compared to my own implementation of a similar scheme, which extends the boundary condition to include interior cells. This extension enables the incorporation of an arbitrary number of walls within the computational domain.

2 Scalar conservation laws

The shallow-water equations constitute an example of a system of conservation laws. Assume that we want to model n quantities in a volume $\Omega \in \mathbb{R}^3$, whose densities are represented by the vector-valued function $\boldsymbol{U} : \Omega \times \mathbb{R}_+ \to \mathbb{R}^n$. For the case of the shallow-water equations, the vector of conserved quantities consist of the total mass and momentum in each axial direction. For any control-volume $\omega \subset \Omega$, the rate of change of these quantities is the amount created or destroyed and the inflow or flux through the boundary $\partial\omega$. Created mass could for instance be due to rainfall. The conservation law for the i-th quantity can be described mathematically by

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_\omega U^{(i)}\mathrm{d}\mathbf{x} = -\int_{\partial\omega} \boldsymbol{f}^{(i)}(\boldsymbol{U}) \cdot \boldsymbol{\nu}\mathrm{d}\sigma(\mathbf{x}) + \int_\omega \boldsymbol{S}^{(i)}\mathrm{d}\mathbf{x},$$

where $\boldsymbol{f}^{(i)}, S^{(i)}$ and $U^{(i)}$ is the flux, source and density of this quantity. For sufficiently smooth $\boldsymbol{U}, \boldsymbol{f}^{(i)}$ and ω, we can apply the Gauss divergence theorem and write

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_\omega U^{(i)}\mathrm{d}\mathbf{x} + \int_\omega \boldsymbol{\nabla} \cdot \boldsymbol{f}^{(i)}(\boldsymbol{U})\mathrm{d}\mathbf{x} = \int_\omega S^{(i)}\mathrm{d}\mathbf{x}.$$

Since this equation holds for all suitably smooth subdomains $\omega \subset \Omega$, we can write

$$U_t^{(i)} + \boldsymbol{\nabla} \cdot \boldsymbol{f}^{(i)}(\boldsymbol{U}) = S^{(i)} \quad \text{in } \Omega \times \mathbb{R}_+.$$

For all $1 \leq i \leq n$. This is an example of a first-order quasilinear[1] hyperbolic[2] equation. The solution of these equations are known to develop discontinuities in finite time, even with smooth initial conditions (LeVeque, 2012). Consider for example a plane breaking the sound barrier. The difference in air pressure is practically a shock. We therefore need to develop a special theory. For simplicity, we will present the theory for scalar conservation laws in one dimension,

$$U_t + f(U)_x = 0 \quad \text{in } \mathbb{R} \times \mathbb{R}_+, \tag{1}$$

where $U : \mathbb{R} \times \mathbb{R}_+ \to \mathbb{R}$ is the conserved variable and $f(U)$ its flux. The properties of these solutions are carried over in the discrete sense, motivating the development of finite-volume schemes.

As we will see, the solutions[3] of a hyperbolic conservation law may not be unique. However, one can narrow down the possible solutions by considering only physically possible solutions. For example, in thermodynamics, we know that the entropy increases with time. Mathematically, we can define an equivalent condition that ensures uniqueness and existence. We call this the entropy condition, and the solution satisfying it, the entropy solution.

These solutions satisfy a set of properties, such as stability in various norms. When developing finite-volume methods, it is important that these satisfy the discrete equivalents, and that they converge to the exact entropy solution. We will therefore start to look at these properties, before considering their discrete counterparts.

We will start assuming that f is strictly convex, and then state some results for smooth f.

2.1 The Rankine–Hugoniot condition

As mentioned in the previous section, the solutions may develop discontinuities, and are therefore not classical C^1-solutions. We therefore state the problem in its weak form: Find a function U, say in $L^1(\mathbb{R} \times \mathbb{R}_+)$ or $L^\infty(\mathbb{R} \times \mathbb{R}_+)$, such that

$$\int_{\mathbb{R}_+} \int_{\mathbb{R}} U\varphi_t + f(U)\varphi_x \mathrm{d}x \, \mathrm{d}t + \int_{\mathbb{R}} U_0(x)\varphi(x,0)\mathrm{d}x = 0 \tag{2}$$

for all test functions $\varphi \in C_c^\infty(\mathbb{R} \times \mathbb{R}_+)$.

This formulation imposes a condition on where these discontinuities may develop. Assume a situation similar to the previously described plane breaking the sound barrier. Mathematically, we have a weak solution U, which is a classic solution in two disjoint, connected domains Ω^+ and Ω^-, separated by the the C^1 shock line parametrized by $x = \gamma(t)$; see Figure 1. As this is a weak solution, it must satisfy

$$\begin{aligned} \int_{\Omega} U\varphi_t + f(U)\varphi_x \mathrm{d}\Omega = -\int_{\Omega^+} (U_t + f(U)_x)\varphi \mathrm{d}\Omega + \int_{\partial\Omega^+} \boldsymbol{G}(U^+) \cdot \boldsymbol{\nu}\varphi \mathrm{d}\Omega \\ - \int_{\Omega^-} (U_t + f(U)_x)\varphi \mathrm{d}\Omega + \int_{\partial\Omega^-} \boldsymbol{G}(U^-) \cdot \boldsymbol{\nu}\varphi \mathrm{d}\Omega = 0 \end{aligned}$$

for test functions $\varphi \in C_c^\infty(\Omega) \subset C_c^\infty(\mathbb{R} \times \mathbb{R}_+)$, where $\boldsymbol{G}(U) = (U, f(U))^T$ and $U^\pm$ is the value of U at $\partial\Omega^\pm$. The outward normal is given by $\boldsymbol{\nu} = (1, -s(t))$, up to normalization and orientation, where $s(t) = \frac{\mathrm{d}\gamma}{\mathrm{d}t}$ is the shock speed. The integrals over $\Omega^\pm$ vanish, so we are left with

$$\int_{\gamma} \big(s(t)\big(U^+ - U^-\big) - \big(f(U^+) - f(U^-)\big)\big)\varphi \mathrm{d}\Omega = 0.$$

[1] All first-order derivatives appear linearly.

[2] All the eigenvalues of the Jacobian matrix $\boldsymbol{f}'$ are real-valued.

[3] Solutions in the weak sense (2).

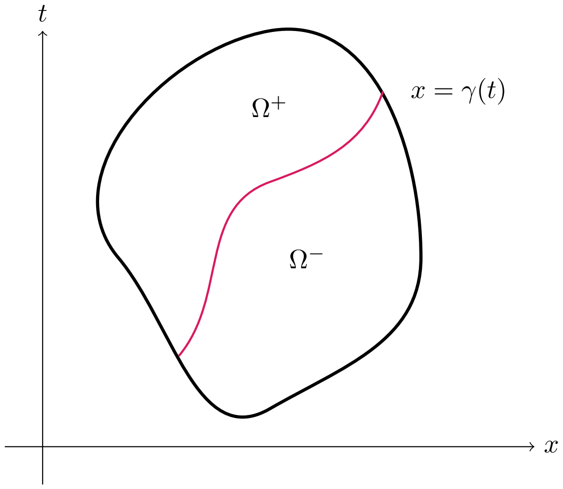


Figure 1: A domain $\Omega = \overline{\Omega^+ \sqcup \Omega^-}$ in the space-time plane divided by the red shock curve parametrized by $x = \gamma(t)$.

Denote by $[\![U]\!] = U^+ - U^-$ the jump along the shock line γ. As the simplified conservation law just derived holds for all $\varphi \in C_c^\infty(\Omega)$, we get the Rankine–Hugoniot condition

$$s = \frac{\mathrm{d}\gamma}{\mathrm{d}t} = \frac{[\![f(U)]\!]}{[\![U]\!]}. \tag{3}$$

These shocks should therefore satisfy this ODE.

2.2 The Lax entropy condition

However, the condition (3) alone is not enough to ensure uniqueness of weak solutions. Consider for example the non-viscous Burgers' equation, $f(U) = \frac{1}{2}U^2$, with so-called Riemann initial data

$$U_0(x) = \begin{cases} U_L, & x < 0, \\ U_R, & x > 0. \end{cases} \tag{4}$$

As an example, we pick $U_L = 0, U_R = 1$. The characteristics are given by

$$x(t) = \begin{cases} x_0, & x < 0, \\ x_0 + t, & x > 0, \end{cases}$$

so they do not cover the whole plane; see Figure 2. Nevertheless, a valid weak solution is given by preserving the discontinuity along the shock line $\gamma(t) = \frac{1}{2}t$ given by (3)

$$U(x,t) = U_0\big(x - \gamma(t)\big). \tag{5}$$

To demonstrate the non-uniqueness of the solutions, we can also introduce a third intermediate state, creating two new shock lines satisfying (3):

$$U(x,t) = \begin{cases} 0, & x < \frac{1}{4}t, \\ \frac{1}{2}, & \frac{1}{4}t < x < \frac{3}{4}t, \\ 1, & x > \frac{3}{4}t. \end{cases} \tag{6}$$

One way to constraint our solutions is to consider the flow of information. We would like the solution to depend uniquely on the initial data. However, as just demonstrated, one can construct infinitely many weak solutions using intermediate states satisfying (3). Few of these have any dependence on the initial data. Intuitively, we want the characteristics of the solution

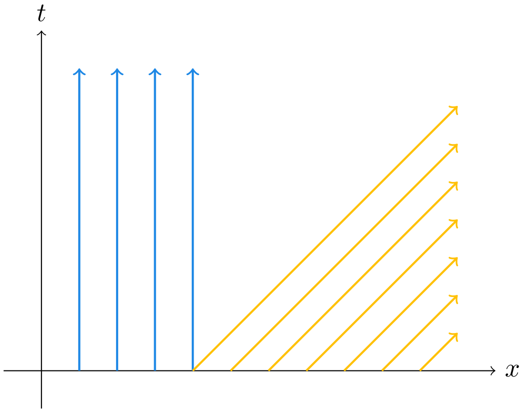
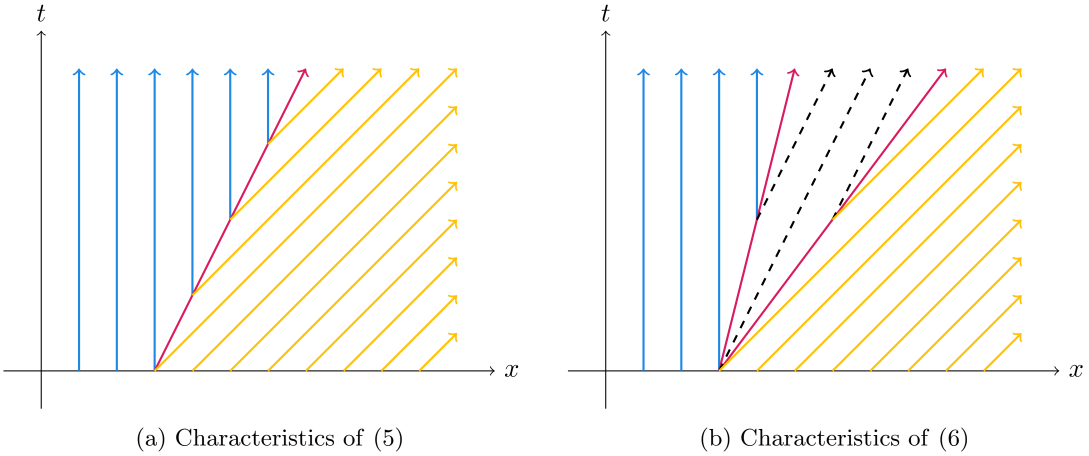


Figure 2: In some cases, the characteristics emanating from the initial values do not cover the whole pane.



(a) Characteristics of (5)

(b) Characteristics of (6)

Figure 3: The characteristics of the solutions to the Riemann problem (4) with $(U_L, U_R) = (0, 1)$ are illustrated as follows: blue vertical arrows to the left correspond to $U = 0$, yellow arrows pointing to the right correspond to $U = 1$, and dashed black arrows represent $U = \frac{1}{2}$. In both cases, the characteristics originate from the shock line(s).

to flow into the shock lines. Otherwise, it would violate causality, as the shock would create the left and right states, not the other way around. The speeds of the characteristics on the left and right are given by $f(U^+)$ and $f(U^-)$, respectively, while the shock speed is denoted by s. This leads to the *Lax entropy condition*

$$f'(U^+) \geq s \geq f'(U^-). \tag{7}$$

Neither of the last two solutions satisfy this condition. In fact, it is impossible to construct such a shock solution for this initial data. This can be seen visually in Figure 2 by considering where the characteristics of such a solution would flow. Eventually, some would start from a shock curve.

Therefore, one should look for continuous solutions. Consider again the undetermined part of the plane in Figure 2. Perhaps one could construct a solution whose characteristic speeds vary continuously from the left to the right states, thus creating a fan covering the whole plane. Notice that (1) is invariant under scaling of the variables $(x, t) \mapsto (\lambda x, \lambda t)$. A solution in this part of the plane could therefore be of the form $U(x, t) = V(\xi)$, with $\xi = \frac{x}{t}$ if starting at $(0, 0)$. We immediately see that this solution gives characteristics of the type just mentioned; $x = \xi t$. Inserting into (1), we get

$$f'(V) = \xi.$$

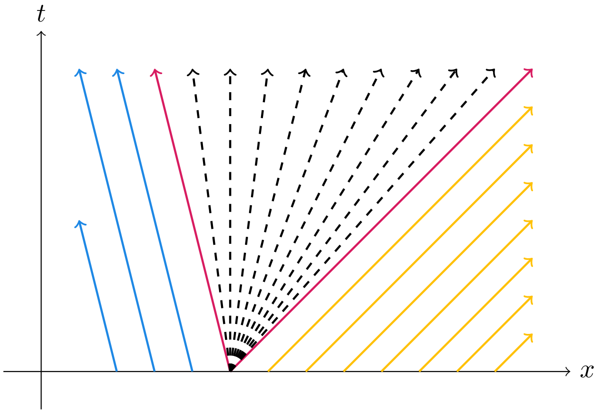


Figure 4: The characteristics of the rarefaction solution (8). The solution have the value $V(\xi)$ along the characteristics $x = \xi t$.

Since f is strictly convex, the inverse of f' exists and we get the solution

$$V(\xi) = (f')^{-1}(\xi).$$

Notice that inserting the speeds $f'(U_L)$ and $f'(U_R)$ for ξ gives us the values U_L and U_R. We now have the continuous solution

$$U(x,t) = \begin{cases} U_L, & x \leq f'(U_L)t, \\ V(x/t), & f'(U_L) < x/t < f'(U_R), \\ U_R, & f'(U_R)t \leq x, \end{cases} \tag{8}$$

which we call a *rarefaction wave*. To summarize, we choose the shock wave whenever it satisfies the Lax entropy condition, and a rarefaction wave otherwise.

2.3 The entropy condition

In practice, (7) is difficult to apply globally in proofs, and we have yet to show uniqueness and existence for such solutions. Instead we can introduce another condition, which we will later show is equivalent. Again, we are interested in physically feasible solutions to conservation laws from nature. Many of these have a small, almost negligible, diffusion term, yielding the convection-diffusion equation

$$U_t^\varepsilon + f(U^\varepsilon)_x = \varepsilon U_{xx}^\varepsilon. \tag{9}$$

Therefore, we want the solution of (1) to behave similarly to to the viscosity solution U^ε. Hence, the goal is to derive an inequality and show that it carries over to the vanishing viscosity solution as $\varepsilon \to 0$, if it exists.

The solutions of (9) are well studied and known to be C^∞ for $t > 0$, even with a discontinuous initial condition (LeVeque, 2012). We choose any strictly convex function $\eta : \mathbb{R} \to \mathbb{R}$ and define

$$q(U) = \int_0^U f'(v)\eta'(v)\mathrm{d}U.$$

We call (η, q) an entropy pair. Next, multiplying (9) by $\eta'(U^\varepsilon)$ yields

$$\eta(U^\varepsilon)_t + q(U^\varepsilon)_x = \varepsilon\eta(U^\varepsilon)_{xx} - \varepsilon\eta''(U^\varepsilon)(U_x^\varepsilon)^2.$$

Now, the last term is non-negative, so we get the inequality

$$\eta(U^\varepsilon)_t + q(U^\varepsilon)_x \leq \varepsilon\eta(U^\varepsilon)_{xx}.$$

One can show that the right hand side vanishes with ε (LeVeque, 2012). We thus get the entropy condition

$$\eta(U)_t + q(U)_x \leq 0, \tag{10}$$

which must hold for the vanishing viscosity solution $U = \lim_{\varepsilon\to 0} U^\varepsilon$. Again, this must be interpreted in the sense of distributions: for all non-negative test functions $\varphi \in C_c^\infty(\mathbb{R}\times\mathbb{R}_+)$,

$$\int_{\mathbb{R}_+}\int_{\mathbb{R}} \eta(U)\varphi_t + q(U)\varphi_x \mathrm{d}x\mathrm{d}t + \int_{\mathbb{R}} \eta(U_0(x))\varphi(x,0)\mathrm{d}x \le 0. \tag{11}$$

Definition 1 (Entropy solution)**.** We call a function $U \in L^\infty(\mathbb{R}\times\mathbb{R}^+)$ an entropy solution of (1) if it is a weak solution in the sense of (2) and satisfies (11) for all entropy pairs (η, q).

2.3.1 The Kruzkov entropy condition

Of special importance is the family of entropy pairs

$$\eta(U;c) = |U-c|, \qquad q(U;c) = \operatorname{sign}(U-c)\big[f(U)-f(c)\big], \tag{12}$$

for $c \in \mathbb{R}$. A solution U satisfies the *Kruzkov entropy condition* if

$$|U-c|_t + \operatorname{sign}(U-c)[f(U)-f(c)]_x \le 0 \tag{13}$$

in the weak sense (11) for all $c \in \mathbb{R}$. With the right choices of c, one can show that (13) implies that U is a weak solution. Additionally, arguing that convex functions are limits of linear combinations of functions of the form $\eta(u;k)$, one can show that (13) implies (10) (Holden and Risebro, 2002). To conclude, U is an entropy solution if and only if it satisfies the Kruzkov entropy condition (13).

With this tool, Mishra et al. (2017, Theorem 3.8) draw the connection to Lax entropy condition (7):

Theorem 1. *Let $\gamma \in C^1(\mathbb{R}_+)$ be a curve and $s = \gamma'$ be its speed. Assume $U \in L^\infty(\mathbb{R}\times\mathbb{R}_+)$ is a weak solution of* (1) *of the form*

$$U(x,t) = \begin{cases} U^+(x,t), & x < \gamma(t), \\ U^-(x,t), & x > \gamma(t), \end{cases}$$

where $U^\pm \in C^1$ are classical solutions in each their domain. Then, the following are equivalent:

1. *U is an entropy solution.*
2. *If f is convex/concave, then*
$$f'(U^+) \ge s \ge f'(U^-)$$
along $x = \gamma(t)$.

Therefore, for convex or concave flux functions, Lax entropy condition (7) is often sufficient to guarantee the uniqueness of shock solutions, assuming the entropy solution is unique. Furthermore, the discrete counterpart of the Kruzkov entropy condition will be crucial for establishing discrete stability estimates.

2.4 Stability estimates

We can now use these tools to show the existence of the entropy solution. Further, deriving L^1-stability, one can easily show that this solution is unique in L^1 in the sense that

$$\|U(\cdot,t) - V(\cdot,t)\|_{L^1} = 0$$

whenever U and V are entropy solutions with the same initial condition. Additionally, one can introduce a bound on the oscillation. In other words, the solution should not develop oscillations with time. Mathematically, one can describe this oscillation as the total variation of a function g

$$\|g\|_{TV([a,b])} := \sup_{\mathcal{P}} \sum_j |g(x_{j+1}) - g(x_j)|,$$

where $\mathcal{P}$ is the set of all partitions $\{x_j\}$ of $[a, b]$. For differentiable g, this can be simplified to the L^1-norm of its derivative. This is only a semi-norm. Nevertheless, we can define the bounded variation

$$\|g\|_{BV([a,b])} := \|g\|_{L^1([a,b])} + \|g\|_{TV([a,b])},$$

which forms the Banach space of functions of bounded variation on $\mathbb{R}$:

$$BV(\mathbb{R}) = \Big\{ g \in L^1(\mathbb{R}) : \|g\|_{BV(\mathbb{R})} < \infty \Big\}.$$

As previously mentioned, these estimates will be important when developing numerical schemes, as they should satisfy discrete equivalents.

Using again (13), the following theorem can be proven (Holden and Risebro, 2002, Theorem 2.14)

Theorem 2. *Assume f is Lipschitz continuous and $U_0, V_0 \in BV(\mathbb{R})$. Then, there exists unique entropy solutions U and V of* (1) *with initial condition U_0 and V_0, respectively, with the following properties:*

1. *L^1-bound:*
$$\|U(\cdot, t)\|_{L^1(\mathbb{R})} \leq \|U_0\|_{L^1(\mathbb{R})} \quad \forall t > 0.$$
2. *L^∞-bound:*
$$\|U(\cdot, t)\|_{L^\infty(\mathbb{R})} \leq \|U_0\|_{L^\infty(\mathbb{R})} \quad \forall t > 0.$$
3. *Total variation bound:*
$$\|U(\cdot, t)\|_{TV(\mathbb{R})} \leq \|U_0\|_{TV(\mathbb{R})}.$$
4. *Time continuity:*
$$\|U(\cdot, t) - U(\cdot, s)\|_{L^1(\mathbb{R})} \leq |t - s| \|f\|_{Lip} \|U_0\|_{TV(\mathbb{R})}.$$
5. *L^1-stability:*
$$\|U(\cdot, t) - V(\cdot, t)\|_{L^1(\mathbb{R})} \leq \|U_0 - V_0\|_{L^1(\mathbb{R})} \quad \forall t \geq 0.$$
6. *Monotonicity: If $U_0 \leq V_0$ almost everywhere in $\mathbb{R}$, then*
$$U(\cdot, t) \leq V(\cdot, t) \quad a.e. \ in \ \mathbb{R}.$$

Remark 1. The solutions of Riemann problems are of special interest. Since step functions are dense in L^1, we can approximate any $U_0 \in L^1$ to arbitrary accuracy with a superposition of Riemann problems, say $U_{0,\Delta x}$. Further, due to the L^1 stability estimate in Theorem 2, the entropy solution $U_{\Delta x}$ of the superposition of Riemann problems converge to U, in the sense that

$$\|U_{\Delta x}(\cdot, t) - U(\cdot, t)\|_{L^1(\mathbb{R})} \leq \|U_{0,\Delta x} - U_0\|_{L^1(\mathbb{R})} \xrightarrow{\Delta x \to 0} 0 \quad \forall t > 0.$$

△

3 Finite-volume methods

As previously discussed, we want our numerical schemes to exhibit solutions that satisfy discrete versions of the stability estimates in Theorem 2. To do this, we first need to discretize the domain.

3.1 Discretization

Recall that the entropy solution is a weak solution, so it is not everywhere pointwise defined. Therefore, it does not make much sense to approximate point values. Instead, we can approximate its averages in smaller and smaller finite volumes, which we call *cells* or *control volumes.*

For simplicity, we use a uniform grid of N cells, whose widths are $\Delta x = \frac{x_R - x_L}{N}$. We start by discretizing the grid into the cell interfaces

$$x_{j+1/2} = x_L + j\Delta x, \quad j = 0, \dots, N$$

and cell centers

$$x_j = x_{j+1/2} - \frac{1}{2}\Delta x, \quad j = 1, \dots, N.$$

Then, the cells are given by

$$\mathcal{C}_j = \big[x_{j-1/2}, x_{j+1/2}\big), \quad j = 1, \dots N.$$

Throughout this section, we will ignore the boundary conditions, which would require extra care.

3.2 Deriving the finite volume scheme

Denote by U_j^n the cell average in $\mathcal{C}_j$ at time t^n. As argued in Remark 1, the entropy solution of the Riemann problem

$$U_0(x) = U_j^0, \quad x \in \mathcal{C}_j$$

converges to the exact entropy solution as $\Delta x \to 0$. Therefore, one may try to solve this Riemann problem instead. We start by integrating (1) over the square $\mathcal{C}_j \times [t^n, t^{n+1})$:

$$\int_{t^n}^{t^{n+1}} \int_{\mathcal{C}_j} U_t + f(U)_x \mathrm{d}x \, \mathrm{d}t = 0.$$

The solution of this problem is a combination of rarefaction waves, shocks and constant states, so we can apply Leibniz integral rule, resulting in

$$\int_{\mathcal{C}_j} U \mathrm{d}x \Bigg|_{t^n}^{t^{n+1}} + \int_{t^n}^{t^{n+1}} f(U) \mathrm{d}t \Bigg|_{x_{j-1/2}}^{x_{j+1/2}} = 0. \tag{14}$$

Further, introduce the time-averaged flux $\overline{F}_{j\pm 1/2}^n$ along the line $x = x_{j+1/2}$:

$$\overline{F}_{j\pm 1/2}^n = \frac{1}{\Delta t} \int_{t^n}^{t^{n+1}} f\Big(U\big(x_{j\pm 1/2}, t\big)\Big) \mathrm{d}t.$$

Then, we can rewrite (14) as

$$U_j^{n+1} = U_j^n - \frac{\Delta t}{\Delta x}\Big(\overline{F}_{j+1/2}^n - \overline{F}_{j-1/2}^n\Big). \tag{15}$$

Notice that the flux at the interface $x = x_{j+1/2}$ remains constant as long as only the characteristics emanating from $\mathcal{C}_j$ and $\mathcal{C}_{j+1}$ intersect with the interface. The speeds of these characteristics are given by $f'(U)$. To prevent intersection at all interfaces, the resulting CFL condition is

$$\frac{\Delta x}{\Delta t} > \max_j \big|f'(U_j^n)\big|,$$

see Figure 5. With these new averages, we repeat the steps. This is an *exact Riemann solver.* However, this method is often impractical, as computing the flux $\overline{F}_{j\pm 1/2}^n$ often requires an optimization step (LeVeque, 2012). Instead, one can approximate it using a *numerical flux*

$$\overline{F}_{j+1/2}^n \approx F_{j+1/2}^n = F\big(U_j^n, U_{j+1}^n\big),$$

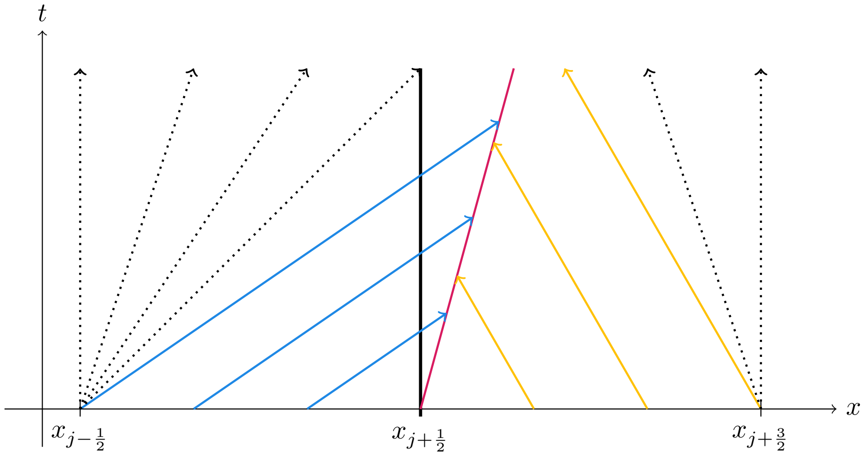


Figure 5: For a superposition of Riemann-problems, the flux at the interface $x = x_{j+1/2}$ is constant for sufficiently small times t. In the case of a shock solution, the flux is determined by the shock curve. The flux could change again if characteristics from other cells intersects with the interface.

for example the *Lax-Friedrichs flux*

$$F^{\mathrm{LxF}}(U_j^n, U_{j+1}^n) = \frac{f(U_j^n) + f(U_{j+1}^n)}{2} + \frac{\Delta x}{2\Delta t}(U_{j+1}^n - U_j^n),$$

resulting in the *approximate Riemann solver*

$$U_j^{n+1} = U_j^n - \frac{\Delta t}{\Delta x}\Big(F_{j+1/2}^n - F_{j-1/2}^n\Big). \tag{16}$$

This is an example of a 3-point stencil FVM. As previously mentioned, we want (16) to exhibit approximations that converge to the entropy solution and behave similar to it. The next section will therefore discuss the discrete versions of the stability estimates in Theorem 2 and convergence.

Remark 2. Although the theory for conservation laws in multiple spatial dimensions lies beyond the scope of this project, the scheme is largely analogous to the one-dimensional case. Integrating over a d-dimensional cell $\mathcal{C}_{\boldsymbol{j}}$

$$0 = \int_{\mathcal{C}_{\boldsymbol{j}}} \int_{t^n}^{t^{n+1}} U_t + \boldsymbol{\nabla} \cdot \boldsymbol{f}(U) \mathrm{d}t \, \mathrm{d}x,$$

and applying arguments similar to those used in the one-dimensional case, we can derive the following update formula:

$$\begin{aligned} U_{\boldsymbol{j}}^{n+1} &= U_{\boldsymbol{j}}^n - \frac{1}{\Delta V} \int_{\partial \mathcal{C}_{\boldsymbol{j}}} \int_{t^n}^{t^{n+1}} \boldsymbol{f}(U) \cdot \boldsymbol{\nu} \mathrm{d}t \mathrm{d}\sigma \\ &= U_{\boldsymbol{j}}^n - \sum_{i=1}^{d} \frac{\Delta t}{\Delta x_i} \Big(\overline{F}_{i,j_i+1/2}^n - \overline{F}_{i,j_i-1/2}^n\Big). \end{aligned}$$

Here, to compute the numerical approximation of the exact average fluxes, one divides the problem into d one-dimensional problems, computing the contribution from each direction. This is called *dimensional splitting*. As long as the scheme for every dimension is monotone[4], the solution will converge to the entropy solution (LeVeque, 2012, Theorem 18.2). △

[4] See Definition 4

3.3 Discrete analogues of the continuous properties

To develop suitable numerical fluxes, we first give three definitions that are important for guaranteeing convergence and other stability estimates. For this analysis, we introduce the more general $(2p+1)$-point stencil update function H:

$$U_j^{n+1} = H(U_{j-p}^n, \dots, U_{j+p}^n). \tag{17}$$

3.3.1 Conservative, consistent and monotone schemes

In the absence of sources or sinks, the total amount of U should not change. Mathematically, this is formulated as

$$\frac{\mathrm{d}}{\mathrm{d}t} \int_{\mathbb{R}} U \mathrm{d}x = 0.$$

Therefore, the integral

$$\int_{\mathbb{R}} U(x,t)\mathrm{d}x = \sum_j \int_{\mathcal{C}_j} U\big(x, t^{n+1}\big)\mathrm{d}x = \sum_j \int_{\mathcal{C}_j} U(x, t^n)\mathrm{d}x$$

is time-independent and the sum of the cell averages are constant. This leads to the following definition:

Definition 2 (Conservative scheme)**.** A numerical scheme (17) satisfying

$$\sum_j U_j^{n+1} = \sum_j U_j^n \tag{18}$$

for all n is called *conservative.*

Summing (16) over all j, the right-hand side forms a vanishing telescoping sum of numerical fluxes, leading to (18). Hence, the finite volume method is conservative.

Consider again the Riemann problem with $U_L = U_R = U$. Then, the flux across the interface is $f(U)$, and we expect that $F(U,U) = f(U)$. Additionally, if all the cells in the stencil are U, the solution should not spuriously change.

Definition 3 (Consistent scheme)**.** A numerical scheme (17) with $H(U, \dots, U) = U$ is called *consistent.* Similarly, a numerical flux with $F(U, \dots, U) = f(U)$ is consistent.

Recall the monotonicity estimate in Theorem 2. We want a similar estimate to hold for our numerical scheme. This follows from the following definition:

Definition 4 (Monotone scheme)**.** We say that the numerical scheme (17) is *monotone* if the update function H is non-decreasing in each of its arguments.

3.3.2 Stability estimates

From the monotonicity, consistency and conservativity, one can now derive the estimates in Theorem 2.

Discrete Maximum Principle The maximum principle follows directly from the consistency and monotonicity of the scheme. We get that

$$\min\left\{U_{j-p}^n, \dots, U_{j+p}^n\right\} \le U_j^{n+1} \le \max\left\{U_{j-p}^n, \dots, U_{j+p}^n\right\},$$

and propagating through time, we get

$$\max_j \left|U_j^n\right| \le \max_j \left|U_j^0\right| \quad \forall\, n \tag{19}$$

L^p-bounds Inspired by the analytical case, one can define a discrete version of the entropy condition. One can then use this to derive L^p-bounds, and importantly, convergence to the entropy solution. The *Crandall–Majda numerical flux* is given by

$$Q^n_{j+1/2} = Q(U^n_j, U^n_{j+1}) = F(U^n_j \vee k, U^n_{j+1} \vee k) - F(U^n_j \wedge k, U^n_{j+1} \wedge k),$$

where $a \vee b = \max\{a, b\}$ and $a \wedge b = \min\{a, b\}$. For consistent numerical fluxes, we see that this entropy flux is consistent with the Kruskov entropy flux (12)

$$\begin{aligned} Q(U,U) &= f(U \vee k) - f(U \wedge k) \\ &= \operatorname{sign}(U-k)[f(U) - f(k)] \\ &= q(U;k). \end{aligned}$$

With this flux, Crandall and Majda (1980) show that the solution U^n_j computed by a consistent, conservative and monotone three-point scheme satisfy the *discrete entropy inequality*

$$\left|U^{n+1}_j - k\right| - \left|U^n_j - k\right| + \frac{\Delta t}{\Delta x}\Big(Q^n_{j+1/2} - Q^n_{j-1/2}\Big) \leq 0 \quad \forall n, j. \tag{20}$$

Setting $k = 0$ and summing over all j and n, we get the L^1-bound

$$\Delta x \sum_j \left|U^n_j\right| \leq \|U_0\|_{L^1},$$

assuming $U_0 \in L_1$. As argued previously with the Kruzkov entropy flux (12), we similarly get

$$\Delta x^p \sum_j \left|U^n_j\right|^p \leq \|U_0\|^p_{L^p}$$

for all $1 \leq p < \infty$.

TV-Bound Interpreting the cell averages as a step function, the total variation reduces to

$$\|U^n\|_{TV} = \sum_j \left|U^n_{j+1} - U^n_j\right|.$$

By reformulating (16), Harten (1983, Lemma 2.2) shows that any conservative 3-point stencil scheme with locally Lipschitz continuous numerical flux $F(a, b)$ is TVD

$$\left\|U^{n+1}\right\|_{TV} \leq \|U^n\|_{TV},$$

under the CFL-condition

$$\left|\frac{\partial F}{\partial a}(v,w)\right| + \left|\frac{\partial F}{\partial b}(u,v)\right| \leq \frac{\Delta x}{\Delta t} \quad \forall u, v, w. \tag{21}$$

Time continuity The last property remaining in Theorem 2 is time continuity. This can be derived starting from (16) and using the Lipschitz continuity of F. Let C_F be the Lipschitz constant of F. Then, following the steps hinted at in (Mishra et al., 2017, Lemma 4.14), we have

$$\begin{aligned} \Delta x \sum_j \left|U^{n+1}_j - U^n_j\right| &= \Delta t \sum_j \left|F^n_{j+1/2} - F^n_{j-1/2}\right| \\ &\leq \Delta t \sum_j \left|F(U^n_j, U^n_{j+1}) - F(U^n_j, U^n_j)\right| \\ &\quad + \Delta t \sum_j \left|F(U^n_j, U^n_j) - F(U^n_{j-1}, U^n_j)\right| \\ &\leq \Delta t \sum_j C_F \left|U^n_{j+1} - U^n_j\right| + \Delta t \sum_j C_F \left|U^n_j - U^n_{j-1}\right| \\ &= 2C_F \Delta t \sum_j \left|U^n_{j+1} - U^n_j\right|. \end{aligned}$$

Next, with repeated use of the triangle inequality and that the scheme is TVD, we obtain the desired result:

$$\Delta x \sum_j \left|U^n_j - U^m_j\right| \leq 2C_F |t^n - t^m| \|U_0\|_{L^1}.$$

3.3.3 Convergence to the entropy solution

It now remains to show that the numerical solution converges to the entropy solution. This is done in three steps. Assuming the approximation converges to some function U it is a weak solution. Further, it satisfies the entropy condition. Lastly, one need to show such a function exists. To simplify notation, we will use the piecewise constant function

$$U^{\Delta x}(x,t) = U_j^n \quad \text{for } (x,t) \in \mathcal{C}_j \times [t^n, t^{n+1}).$$

Theorem 3 (Lax-Wendroff)**.** *Let U_j^n be the approximate solution of* (1) *using a conservative, consistent, monotone scheme whose numerical flux is differentiable. Assume that*

- $U_0 \in L^\infty(\mathbb{R})$*,*
- $U^{\Delta x}$ *is uniformly bounded in* Δx *and*
- *there exists a function* $U \in L^1_{loc}(\mathbb{R} \times R_+)$ *such that* $\left\|U^{\Delta x} - U\right\|_{L^1_{loc}(\mathbb{R})} \to 0$ *as* $\Delta x, \Delta t \to 0$*.*

Then, U is a weak solution of (1) *with initial data* U_0*.*

LeVeque, 2012, Theorem 12.1 shows this starting from the update formula (16), multiplying by a test function and summing over n and j. If $U_0 \in L^\infty(\mathbb{R})$, it follows from the discrete maximum principle (19) that $U^\Delta x$ is uniformly bounded in Δx. We therefore only need convergence to the entropy solution and existence. Following the same steps in the proof of Theorem 3, starting with the discrete entropy inequality (20), it is shown that this solution must be the entropy solution. If additionally, $U_0 \in BV(\mathbb{R})$ and f has finitely many inflection points, LeFloch (2002, Theorem IV-2.1) guarantees that the limit exists.

3.4 Higher-order TVD schemes

The bound on total variation was only shown to hold for two-point numerical fluxes. This therefore limits the size of the stencil and hence the order of convergence of the scheme. The Reconstruct-Evolve-Average (REA) algorithm combines variation-diminishing steps with a two-point numerical flux to create a TVD scheme on a larger stencil.

3.4.1 TVD reconstructions

Assuming the entropy solution U is piecewise smooth in x, one may try to use the neighbouring cell averages to reconstruct $U(\cdot, t^n)$ in each cell $\mathcal{C}_j$, for instance as polynomials p_j^n. We then obtain a new set of Riemann problems at the interfaces, defined by the reconstructed values at the interfaces. These can then be solved using an exact (15) or approximate (16) Riemann solver evaluating the numerical flux on the reconstruction on both sides of the interfaces.

$$F_{j+1/2}^n = F\big(p_j^n(x_{j+1/2}), p_{j+1}^n(x_{j+1/2})\big).$$

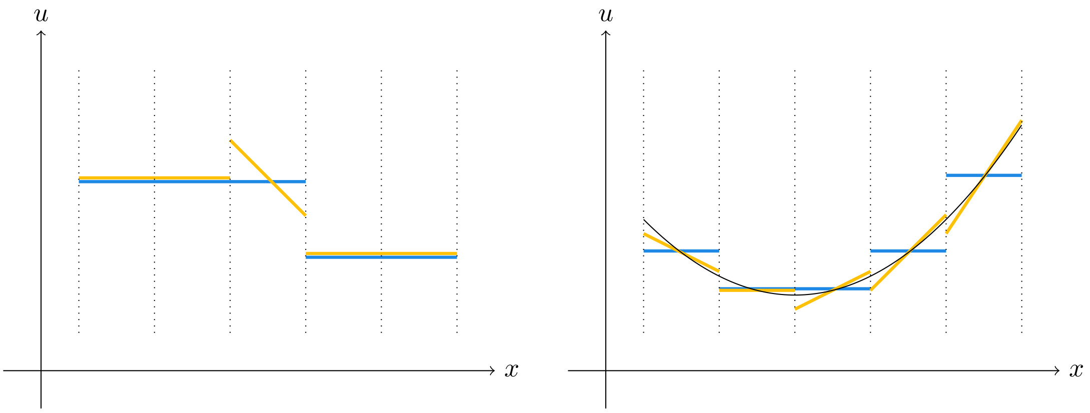
In order to remain conservative, these reconstructions must preserve the original averages. Therefore, linear reconstructions take the form

$$p_j^n(x) = U_j^n + \sigma_j^n(x - x_j).$$

However, one must be careful when choosing the slopes σ_j^n. For instance, one may approximate the slopes using forward differences:

$$\sigma_j^n = \frac{U_{j+1} - U_j}{\Delta x}.$$

Now, consider the Riemann problem (4). The total variation of U_0 is $\|U_0\|_{TV} = |U_R - U_L|$, while that of the reconstruction is $2|U_R - U_L|$. This reconstruction is clearly not TVD, which can also be seen in Figure 6. Similar results holds for the backwards and central slopes.



(a) Reconstruction of the Riemann problem (4).

(b) Reconstruction of a quadratic function.

Figure 6: Constant and linear reconstruction using the forward difference slope. The latter is a much better approximation than the first. However, it introduces new oscillations not present in the original function.

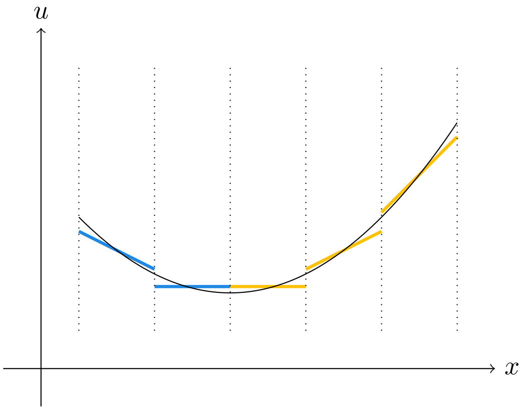


Figure 7: Linear reconstruction using the minmod-operator. This also yields better spatial approximations and is non-oscillatory. In blue: backward slope, in yellow: forward slope.

The minmod slope The problem above arise for example when the backward and forward slopes have opposite signs or the difference in magnitude is large, as seen in Figure 6. One solution is therefore to fall back to no reconstruction whenever the signs are different, and otherwise chose the slope with smallest modulus. This results in the minmod limiter defined by

$$\sigma_j^n = \operatorname{minmod}\left(\frac{U_{j+1}^n - U_j^n}{\Delta x}, \frac{U_j^n - U_{j-1}^n}{\Delta x}\right),$$

where

$$\operatorname{minmod}\{a_k\}_k = \begin{cases} \operatorname{sign}(a_1)\min_k |a_k|, & \operatorname{sign}(a_1) = \cdots = \operatorname{sign}(a_n) \\ 0, & \text{otherwise.} \end{cases}$$

This can be shown to be TVD (LeVeque, 2012), and an example can be seen in Figure 7.

3.4.2 TVD time steppers

The update formula (16) is equivalent to applying forward Euler on the semi-discretization

$$\frac{1}{\Delta x}\frac{\mathrm{d}}{\mathrm{d}t}\int_{\mathcal{C}_j} U \mathrm{d}x\bigg|_{t=t^n} = -\frac{1}{\Delta x}\left(F_{j+1/2}^n - F_{j-1/2}^n\right) =: \mathcal{L}(U_j^n).$$

For smooth functions, this is known to be first-order accurate. Therefore, the scheme will not achieve higher accuracy despite using higher order reconstructions. Naively, one could therefore apply a higher-order integrator or *time-stepper* such as the standard second-order Runge–Kutta integrator to achieve higher accuracy. However, it can be shown that this integrator is not TVD (Gottlieb et al., 2001). Nevertheless, in Subsection 3.3.2, we have in practice shown that forward Euler is TVD [5]. We can therefore construct higher-order time steppers with repeated use of forward Euler. For example, one may take two steps forward with Euler and take the average with the current value:

$$\begin{aligned} k_j^* &= U_j^n + \Delta t \mathcal{L}(U_j^n), \\ k_j^{**} &= k_j^* + \Delta t \mathcal{L}(k_j^*), \\ U_j^{n+1} &= \frac{1}{2}\big(U_j^n + k_j^{**}\big). \end{aligned}$$

This is the second-order *strong-stability-preserving* (SSP) Runge–Kutta method.

One can now combine various TVD reconstructions, time-steppers and numerical fluxes, constructing various schemes. The order of convergence is not discussed in this project. Intuitively, however, combining higher order reconstructions and time-steppers should result in better schemes. This is discussed in more detail in for example in (LeVeque, 2012)

4 The shallow-water simulations

In this project, a finite-volume scheme is implemented in Julia[6], for solving the shallow-water equations in one dimension and constant bottom topography. The aim is to study how we efficiently can treat vertical walls within the domain. This is done by treating the interior walls in the same manner as the boundary condition. In one dimension with $M-1$ interior walls, this is equivalent to just solving M independent problems. It might therefore seem redundant and inefficient to simulate these independent problems as one. However, this is not the case for the two-dimensional case. Further, it is easier to analyse the one-dimensional case to explore some of the issues one face in two dimensions.

The project wishes to address the problems that arise when (almost) vertical obstacles emerge from water in the simulation domain. Here, the simulations may produce spurious momentum or negative water heights, so they may not be both well balanced and positivity-preserving simultaneously. A theory for well balanced and positivity-preserving schemes with an arbitrary bottom topography except at near-dry states is developed in (Kurganov and Petrova, 2007). With this scheme, one can model walls as the bottom topography protruding from the water. SINTEF Digital implemented this scheme in package SinFVM.jl[7], which serves as the basis for comparison.

4.1 The shallow-water equations

The free-surface flow of water is inherently three-dimensional and is fundamentally governed by the Navier–Stokes equations. However, the high computational costs makes it highly impractical to use these equations for numerical simulations (Bridson, 2008). Instead, one can make a series of simplifications to reduce the complexity of the simulations. By considering only the water height h of the water surface above the seabed, the problem reduces to the two horizontal dimensions. This can be done by depth-integrating the quantities of interest. Further, assuming that the water is shallow enough, one can neglect vertical variations in the horizontal velocities. Then, we can approximate the depth-averaged momentum-flux along a horizontal direction $\boldsymbol{\nu}$. Let $\boldsymbol{u}$ be the velocity and $\overline{\boldsymbol{u}}$ be its depth-average. Then, we can write

$$\int_0^h (\boldsymbol{u} \cdot \boldsymbol{\nu}) \boldsymbol{u} \, \mathrm{d}z \approx h(\overline{\boldsymbol{u}} \cdot \boldsymbol{\nu}) \overline{\boldsymbol{u}}.$$

[5] For locally Lipschitz-continuous 2-point numerical flux F under the CFL-condition (21)

[6] https://github.com/martinsw01/homemade-conslaws

[7] https://github.com/sintefmath/SinFVM.jl

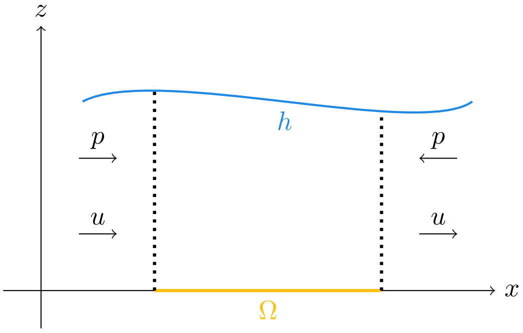


Figure 8: A cross-section of the control volume in the zx-plane. The mass and momentum hu are carried with velocity u resulting in the influxes hu and hu^2 across the dotted boundary. The dominating force acting on the volume is due to the hydrostatic pressure p.

Further, assuming water is incompressible, the depth-averaged mass-density is proportional to the height h. Additionally, we assume that the vertical acceleration of the fluid is negligible. Then, the horizontal forces cancel each other out, and only the hydrostatic pressure $p(z) = g(h - z)$ remains.

From now on, we will let $\boldsymbol{u} = (u, v)^T$ be the depth-averaged velocity in the horizontal plane. Next, we can consider the conservation of mass and momentum in a control-volume with height h and base area Ω; see Figure 8. The rate of change of water in the volume is equal to the incoming flux $\boldsymbol{u}h$:

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_\Omega h \mathrm{d}\Omega = -\int_{\partial\Omega} (h\boldsymbol{u}) \cdot \boldsymbol{\nu} \mathrm{d}\sigma = -\int_\Omega \boldsymbol{\nabla} \cdot (h\boldsymbol{u}) \mathrm{d}\Omega.$$

The choice of Ω is arbitrary, so we get the differential form

$$h_t + \boldsymbol{\nabla} \cdot (h\boldsymbol{u}) = 0.$$

The rate of change of momentum is the total force acting on the volume together with the influx of momentum. Ignoring friction and viscosity, the dominating force is due to pressure. Again, assuming the fluid is hydrostatic, the pressure only acts horizontally. Integrating the pressure over the depth, we get $\frac{1}{2}gh^2$. Consider first the momentum along $\hat{\boldsymbol{x}}$. This gives

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}\int_\Omega hu \mathrm{d}\Omega &= -\int_{\partial\Omega} hu\boldsymbol{u} \cdot \boldsymbol{\nu} \mathrm{d}\sigma - \int_{\partial\Omega} \frac{1}{2}gh^2 \hat{\boldsymbol{x}} \cdot \boldsymbol{\nu} \mathrm{d}\sigma \\ &= -\int_\Omega \boldsymbol{\nabla} \cdot (hu\boldsymbol{u}) \mathrm{d}\Omega - \int_\Omega \left(\frac{1}{2}gh^2\right)_x \mathrm{d}\Omega.\end{aligned}$$

In differential form, the above becomes

$$(hu)_t + \left(hu^2 + \frac{1}{2}gh^2\right)_x + (huv)_y = 0.$$

Similarly holds in the other direction. These three equations are called the *shallow-water equations*, and can be summarized as

$$\begin{pmatrix} h \\ hu \\ hv \end{pmatrix}_t + \begin{pmatrix} hu \\ hu^2 + \frac{1}{2}gh^2 \\ huv \end{pmatrix}_x + \begin{pmatrix} hv \\ huv \\ hv^2 + \frac{1}{2}gh^2 \end{pmatrix}_y = \boldsymbol{0}.$$

In one dimension, these equations reduce to

$$\boldsymbol{U}_t + \boldsymbol{f}(\boldsymbol{U})_x = \boldsymbol{0},$$

where $\boldsymbol{U} = (h, hu)^T$ and the flux is given by

$$\boldsymbol{f}(\boldsymbol{U}) = \begin{pmatrix} hu \\ hu^2 + \frac{1}{2}gh^2 \end{pmatrix}.$$

4.2 Numerical scheme

we have exclusively considered the theory for scalar conservation laws. However, the theory for systems is less developed, and many of the results do not directly transfer. For example, solutions are not always TVD. For instance, when two water waves meet, they may interfere constructively, leading to an increased total variation of h. Therefore, it might also seem inaccurate to require our schemes to be TVD. However, this still prevents unphysical oscillations near shocks. Further, as the finite volume methods are easily generalized to systems in multiple space dimensions and in a lack of better alternatives, these methods are still used.

Again, the numerical schemes described above is easily generalized for systems of conservation laws. Each conserved variable is independently reconstructed. Further, we now get a system of semi-discretizations

$$\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{U}_j^n = -\frac{1}{\Delta x}\Big(\boldsymbol{F}_{j+1/2}^n - \boldsymbol{F}_{j-1/2}^n\Big),$$

where $\boldsymbol{F}_{j\pm1/2}^n$ are the vectors of numerical fluxes. Then, one use a suitable time-stepper like forward Euler or SSP-RK2.

4.2.1 The central-upwind numerical flux

One numerical flux that has been shown to be well balanced and positivity-preserving in the absence of dry states is the central-upwind flux (Kurganov and Petrova, 2007), given by

$$\boldsymbol{F}(\boldsymbol{U}_{j+1},\boldsymbol{U}_j) = \frac{a^+\boldsymbol{f}(\boldsymbol{U}_j^n) - a^-\boldsymbol{f}(\boldsymbol{U}_{j+1})}{a^+ - a^-} + \frac{a^+a^-}{a^+ - a^-}(\boldsymbol{U}_{j+1} - \boldsymbol{U}_j),$$

where

$$a^+ = \max\{|\lambda(\boldsymbol{U}_j)|, |\lambda(\boldsymbol{U}_{j+1})|, 0\}, \qquad a^- = \min\{|\lambda(\boldsymbol{U}_j)|, |\lambda(\boldsymbol{U}_{j+1})|, 0\}$$

are the largest and smallest of the absolute value of the eigenvalues λ of the Jacobian matrix of $\boldsymbol{f}$ evaluated at $\boldsymbol{U}_j$ and $\boldsymbol{U}_{j+1}$. For the shallow-water equations, these eigenvalues are

$$\lambda(\boldsymbol{U}) = \Big\{u - \sqrt{gh}, u + \sqrt{gh}\Big\}.$$

Kurganov and Petrova (2007) further use the CFL-condition

$$\frac{\Delta x}{\Delta t} \geq 2a, \tag{22}$$

where $a = \max_j \Big\{\max\Big\{a_{j+1/2}^+, a_{j+1/2}^-\Big\}\Big\}$.

4.2.2 Boundary conditions

For this project, we will assume that we have immovable and impermeable walls at the boundaries so that any incoming wave will inevitably be reflected. This is equivalent to having an equal wave travelling in the opposite direction, mirrored at the wall interface; see Figure 9. These waves will meet, resulting in constructive superposition in height and destructive superposition in momentum. For a wall starting at the interface $x_{j-1/2}$, this imaginary wave $\tilde{\boldsymbol{U}}$ on the other side of the wall is thus given by

$$\tilde{\boldsymbol{U}}(x_{j-1/2}+x) = \begin{pmatrix} \tilde{h}(x_{j-1/2}+x) \\ (\tilde{hu})(x_{j-1/2}+x) \end{pmatrix} = \begin{pmatrix} h(x_{j-1/2}-x) \\ -(hu)(x_{j-1/2}-x) \end{pmatrix}. \tag{23}$$

For simplicity, we assume that the walls start and stop at cell interfaces. To update the cell average $\boldsymbol{U}_j$ at a cell right of a wall, we need to compute the incoming numerical fluxes from the left and the right. These make use of the reconstructions $\tilde{\boldsymbol{p}}_{j-1}, \boldsymbol{p}_j$ and $\boldsymbol{p}_{j+1}$

$$\begin{aligned}\boldsymbol{F}_{j-1/2} &= \boldsymbol{F}\big(\tilde{\boldsymbol{p}}_{j-1}(x_{j-1/2}), \boldsymbol{p}_j(x_{j-1/2})\big),\\ \boldsymbol{F}_{j+1/2} &= \boldsymbol{F}\big(\boldsymbol{p}_j(x_{j+1/2}), \boldsymbol{p}_{j+1}(x_{j+1/2})\big).\end{aligned}$$

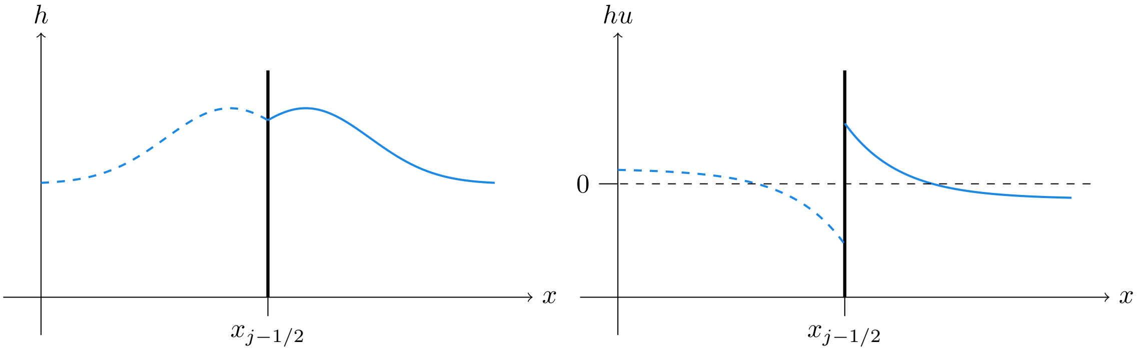


Figure 9: Height and momentum of a wave and its complement wave acting as a wall.

However, on the left, we have a wall and therefore no meaningful cell average we can use to compute the reconstruction $\boldsymbol{p}_j$. To solve this, we treat $\tilde{\boldsymbol{U}}_{j-1}$ in the same manner as $\tilde{\boldsymbol{U}}$ in (23), but using $\boldsymbol{U}_j$ instead of $\boldsymbol{U}$, mirroring the height and flipping the sign of the momentum. We do the same for the reconstruction $\tilde{\boldsymbol{p}}_{j-1}$:

$$\tilde{\boldsymbol{U}}_{j-1} = \begin{pmatrix} \tilde{h}_{j-1} \\ \tilde{hu}_{j-1} \end{pmatrix} = \begin{pmatrix} h_j \\ -hu_j \end{pmatrix}, \tag{24}$$

$$\tilde{\boldsymbol{p}}_{j-1}(x_{j-1/2} + x) = \begin{pmatrix} \tilde{p}^h_{j-1}(x_{j-1/2} + x) \\ \tilde{p}^{hu}_{j-1}(x_{j-1/2} + x) \end{pmatrix} = \begin{pmatrix} p^h_j(x_{j-1/2} - x) \\ -p^{hu}_j(x_{j-1/2} - x) \end{pmatrix}. \tag{25}$$

Similarly is done for a cell to the left of a wall.

4.2.3 Implementation

The code implemented in Julia for this project utilizes the RAE algorithm described in Subsection 3.4 and is inspired by SinFVM.jl. The flux, numerical flux, reconstruction, time-stepper and boundary conditions can be arbitrarily changed. It deviates from SinFVM.jl for example in how it handles cells near the boundaries, and therefore the reconstruction and time-stepper. The reconstruction may require cell averages that may be outside the computational domain. To solve this, SinFVM.jl use so-called *ghost cells* representing these cells and take on the value (24). Similarly, ghost cells represents left and right reconstructions.

This works fine when the boundary condition is only enforced on the boundary of the computational domain. However, it requires even more ghost cells if we want to enforce the boundary condition in the interior. This works fine in one dimension, where the grid can be represented as a vector of cell-averages. However, in two dimensions, this becomes significantly more difficult to implement. The grid is often represented as a matrix, where the columns and rows represent the location of each cell. In this case, it is more difficult to arbitrarily insert ghost cells where there may be walls.

Alternatively, one could do multiple sweeps over the grid, one for each type of cell. This is done in my implementation. First, one deals with the interior cells as normal. Then, one use modified methods when sweeping over the cells that otherwise would use ghost cells. These are described in algorithm 2 and algorithm 3 in the case of a reconstruction using the two closest neighbours. The array of cell averages $\boldsymbol{U}$, left and right reconstruction $\boldsymbol{U}^l, \boldsymbol{U}^r$ and semi-discretizations $\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{U}$ are preallocated. This approach avoids any additional difficulties with ghost cells when extending to the two-dimensional case.

Algorithm 1: Compute ghost cell

```
Input: $\boldsymbol{U}_j$
$h_j, hu_j \leftarrow \boldsymbol{U}_j$ ;
$\tilde{\boldsymbol{U}}_j \leftarrow (h_j, -hu_j)$ ;
return $\tilde{\boldsymbol{U}}_j$
```

Algorithm 2: Reconstruct boundary cells

```
Input: Cell averages $\boldsymbol{U}$
for $j \in$ cell index left of walls do
    $\tilde{\boldsymbol{U}}_{j+1} \leftarrow$ compute_ghost_cell($\boldsymbol{U}_j$) ;
    $\boldsymbol{U}^l_j, \boldsymbol{U}^r_j \leftarrow$ reconstruct_cell($\boldsymbol{U}_{j-1}, \boldsymbol{U}_j, \tilde{\boldsymbol{U}}_{j+1}$) ;
end
for $j \in$ cell index right of walls do
    $\tilde{\boldsymbol{U}}_{j-1} \leftarrow$ compute_ghost_cell($\boldsymbol{U}_j$) ;
    $\boldsymbol{U}^l_j, \boldsymbol{U}^r_j \leftarrow$ reconstruct_cell($\tilde{\boldsymbol{U}}_{j-1}, \boldsymbol{U}_j, \boldsymbol{U}_{j+1}$) ;
end
```

Algorithm 3: Compute semi-discretization for boundary cells

```
Input: Left and right reconstruction $\boldsymbol{U}^l, \boldsymbol{U}^r$
for $j \in$ cell index left of walls do
    $\tilde{\boldsymbol{U}}^l_{j+1} \leftarrow$ compute_ghost_cell($\boldsymbol{U}^r_j$) ;
    $\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{U}_j \leftarrow -\frac{1}{\Delta x}\Big(\boldsymbol{F}(\boldsymbol{U}^r_j, \tilde{\boldsymbol{U}}^l_{j+1}) - \boldsymbol{F}(\boldsymbol{U}^r_{j-1}, \boldsymbol{U}^l_j)\Big)$ ;
end
for $j \in$ cell index right of walls do
    $\tilde{\boldsymbol{U}}^r_{j-1} \leftarrow$ compute_ghost_cell($\boldsymbol{U}^l_j$) ;
    $\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{U}_j \leftarrow -\frac{1}{\Delta x}\Big(\boldsymbol{F}(\boldsymbol{U}^r_j, \boldsymbol{U}^l_{j+1}) - \boldsymbol{F}(\tilde{\boldsymbol{U}}^r_{j-1}, \boldsymbol{U}^l_j)\Big)$ ;
end
```

From now on, this method will be referred to as the *boundary condition (BC) method.*

4.2.4 Bottom topography

Another solution is to include the bottom topography in the shallow-water equations by introducing the source term $-ghB_x$ to the momentum, where B denotes the bottom topography (Kurganov and Petrova, 2007). This results in the conservation law

$$\begin{pmatrix} h \\ hu \end{pmatrix}_t + \begin{pmatrix} hu \\ hu^2 + \frac{1}{2}gh^2 \end{pmatrix}_x = -\begin{pmatrix} 0 \\ ghB_x \end{pmatrix}, \tag{26}$$

where h is the water depth measured from the bottom topography B, see Figure 10.

With the introduction of the source term, we get some additional complexity. As explained in the introduction, we expect that a lake at rest, given by $hu = 0, h + B =$ constant, remains still. Additionally, one must make sure that the water depth h above the bottom B is always positive, unless we happen to be on dry land. This can be challenging, for example in the reconstruction step. Here, one may risk reconstructing a negative value of h. This is discussed in detail (Kurganov and Petrova, 2007), which proposes a solution dealing with this. In short, the topography is linearly interpolated between the cell interfaces. Further, after each reconstruction step, if any reconstructed water height is negative, it is adjusted to match the topography, while still making sure that mass is conserved. Now, a wall can be modelled as the bottom topography rising almost vertically from the water. This method will be referred to as the *bottom topography (TOP) method.*

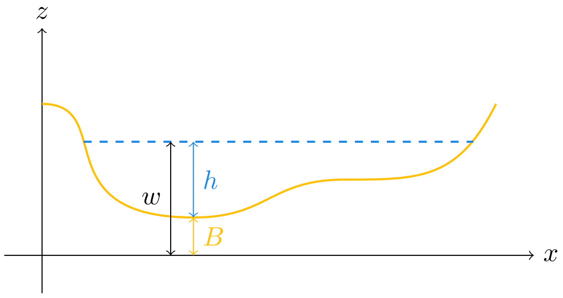
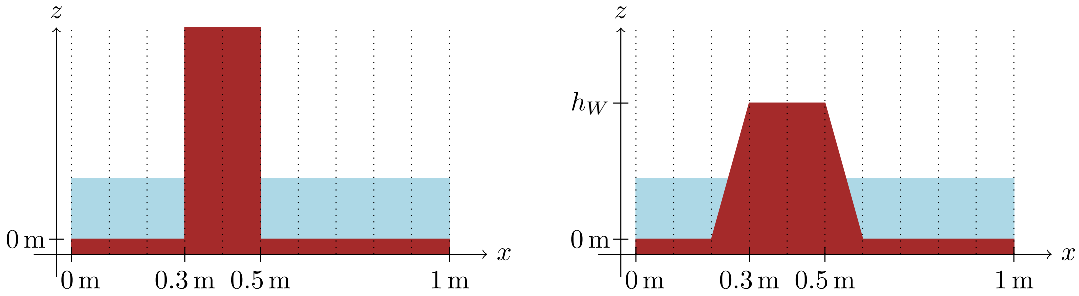


Figure 10: The height h above the bottom topography B, and the relative height w.



(a) Using the BC method, the wall is infinitely high.

(b) Using the TOP method, the wall has finite height, and is linearly interpolated from the neighbouring cell interfaces.

Figure 11: The average water height in each cell for Case 1 with $N = 10$. Water is shown in blue, wall/bathymetry in brown, and cell interfaces are represented by dotted lines.

5 Numerical results

For the comparison, we will consider two cases exploring the non-physical artifacts of the TOP method and the computational costs of the schemes.

5.1 Lake at rest

We will start by introducing *Case 1*. Consider a one meter wide domain discretized into N cells. It is initially filled with water at rest, to a depth of one meter. A wall, 0.2 meters wide, begins at $x = 0.3\,\mathrm{m}$ with a height h_W, as shown in Figure 11. We will assume that we have reached a near-steady state in ten seconds.

Consider first the TOP method. While this scheme is always positivity-preserving, it is only well balanced in the absence of dry land. Consider now Case 1 with $h_W > 1\,\mathrm{m}$. To preserve positivity, the reconstruction interfacing the wall is corrected to the height of the wall. This generates momentum away from the wall. In Figure 12, the steady states developed from Case 1 is plotted for various wall heights. As expected, for $h_W < 1\,\mathrm{m}$, we still have a lake-at-rest. This is clearly not the case for the other wall heights. Additionally, as shown in Figure 13, the moment generated by the reconstruction in the TOP method increases as the wall height increases.

Nevertheless, the deviation from the lake-at-rest solution seems to only affect the cells closest to the wall, and the largest errors appears to be independent of how steep the wall is. Furthermore, the L^1-error seems to vanish with finer grids; see Figure 14. On the other side, when modelling

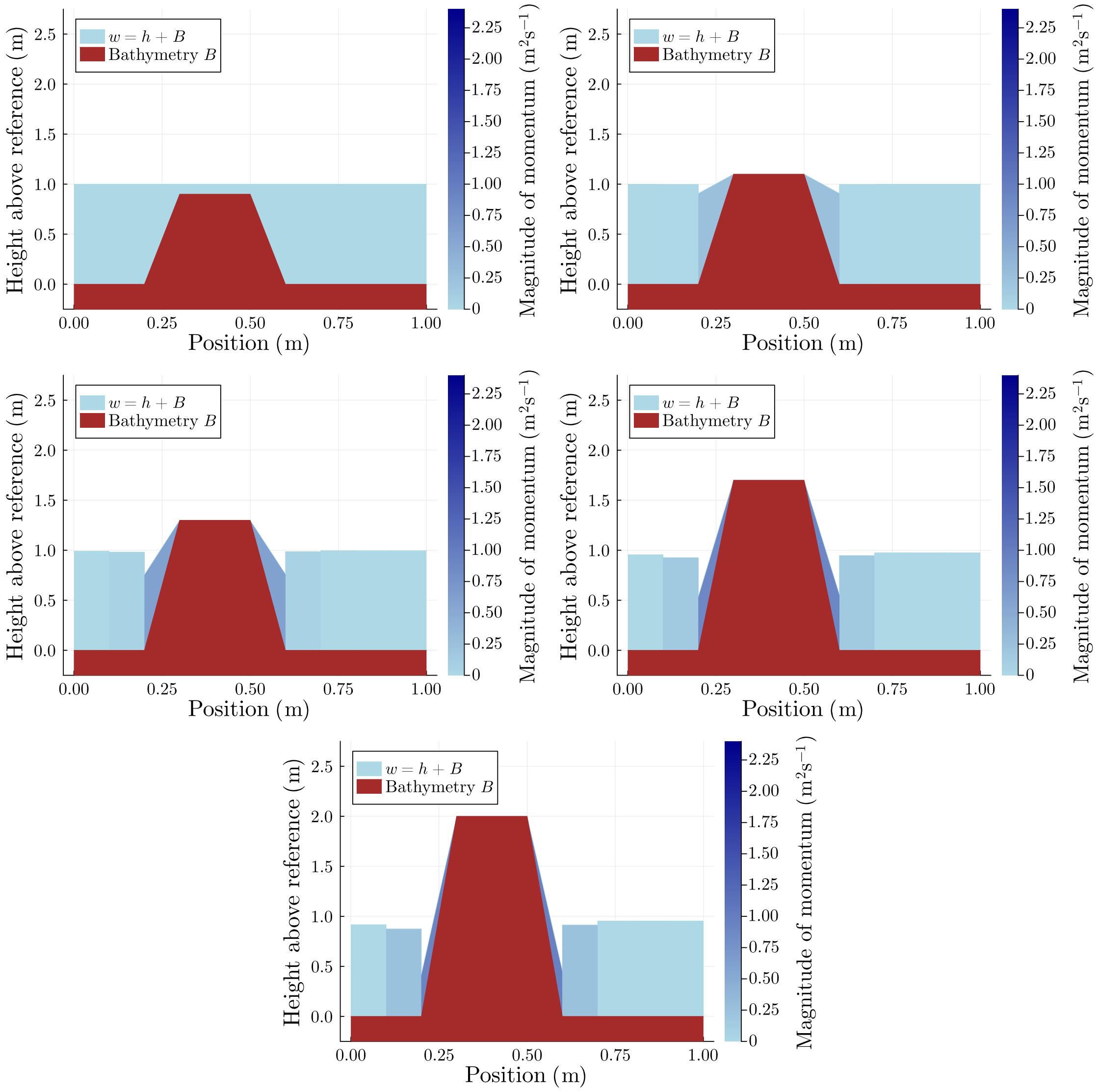


Figure 12: The reconstructed steady-state solutions developed from the lake-at-rest solution in Case 1 with $N = 10$ and wall height h_W of 0.9, 1.1, 1.3, 1.7 and 2 meters. The magnitude of the momentum of the water is coloured in shades of blue. For $h_W > 1\,\mathrm{m}$, we clearly do not have a lake at rest.

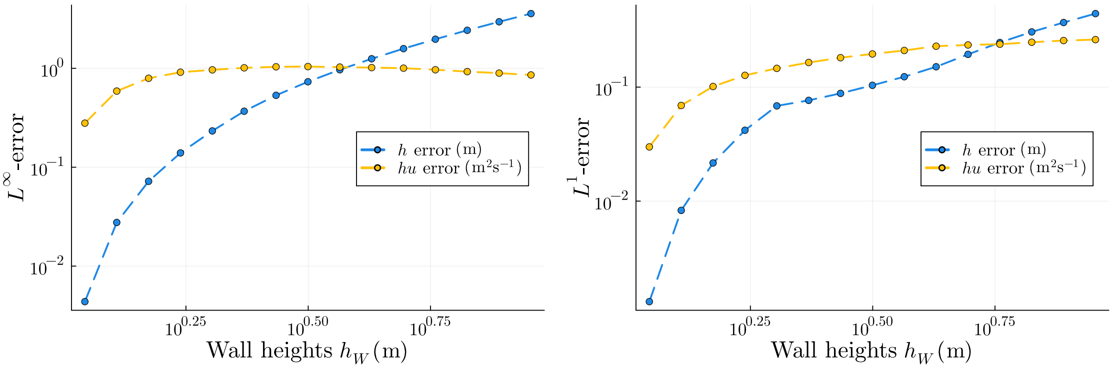
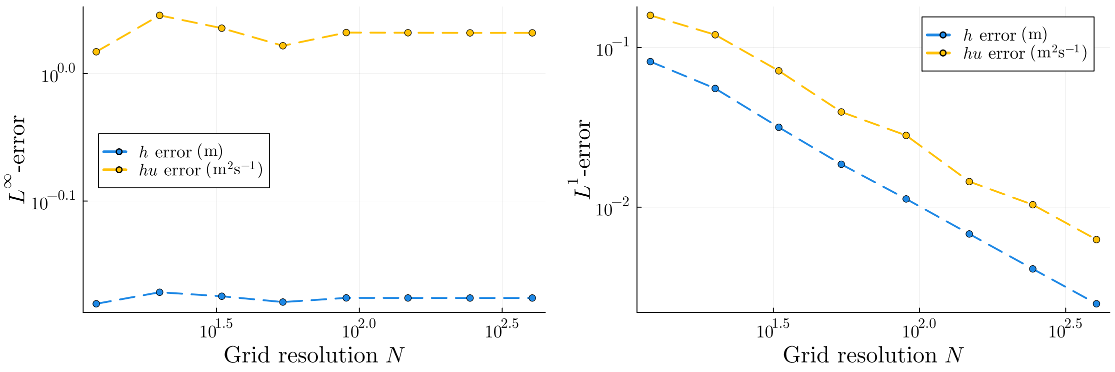


Figure 13: The L^∞ and L^1 deviations of the cell averages from the lake-at-rest solution are shown, with the steady-state solution developed from Case 1 with $N = 10$ using the TOP method, and wall heights h_W ranging from 1.1 to 10 meters.



(a) The L^{∞}-error is seems to be independent of how steep the wall is.

(b) The L^1-error seems to vanish for finer grids.

Figure 14: The L^{∞} and L^1 deviations of the cell averages from the lake-at-rest solution are shown, where the steady-state solution is developed from Case 1 using the TOP method with various grid resolutions N and wall height $h_W = 3\,\text{m}$.

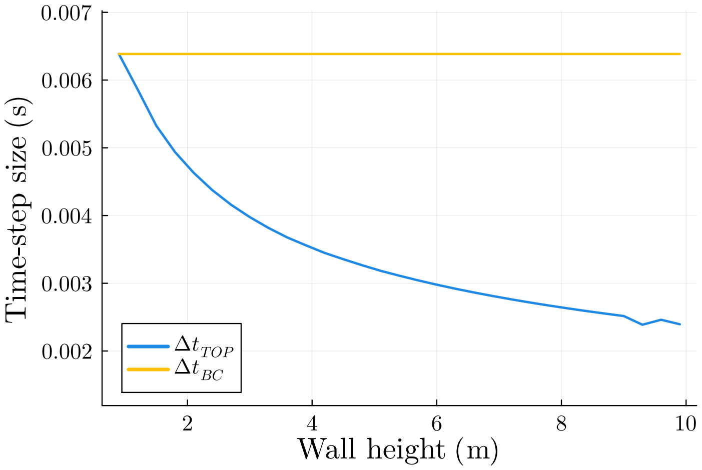
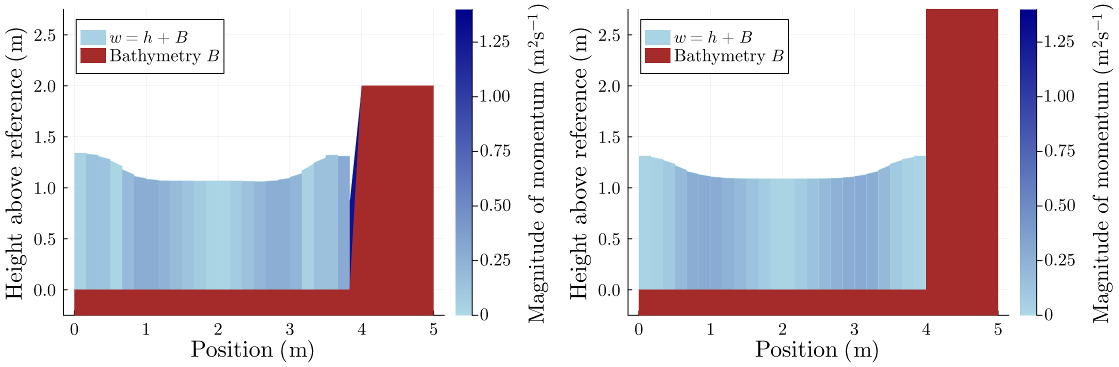


Figure 15: The maximum time-step Δt_{TOP} determined by the CFL-condition for different wall-heights h_W ranging from $0.9\,\text{m}$ to $10\,\text{m}$ for a steady-state solution developed from Case 1 with $N = 10$ using the TOP method. The straight line in yellow is the time-step Δt_{BC} when using the BC method, and is by construction independent of the height of the wall. When $h_W < 1\,\text{m}$, the TOP method is also well balanced, so $\Delta t_{TOP} = \Delta t_{BC}$.

floods in urban areas, the computational domain is much larger, and the cell-widths could be on the order of one meter. Therefore the error may still be significant.

In addition to being non-physical, the artifacts just described also affects the time-steps, which are restricted by the CFL-condition (22), which depends on the height and momentum of the water. As described, a taller wall in Case 1 will result in larger momentum. As illustrated in Figure 15, the time-steps therefore become significantly smaller, resulting in longer computation times. Therefore, even if the BC method could introduce some extra computations for the cells close to the walls, it is negligible compared to how costly the TOP method becomes.

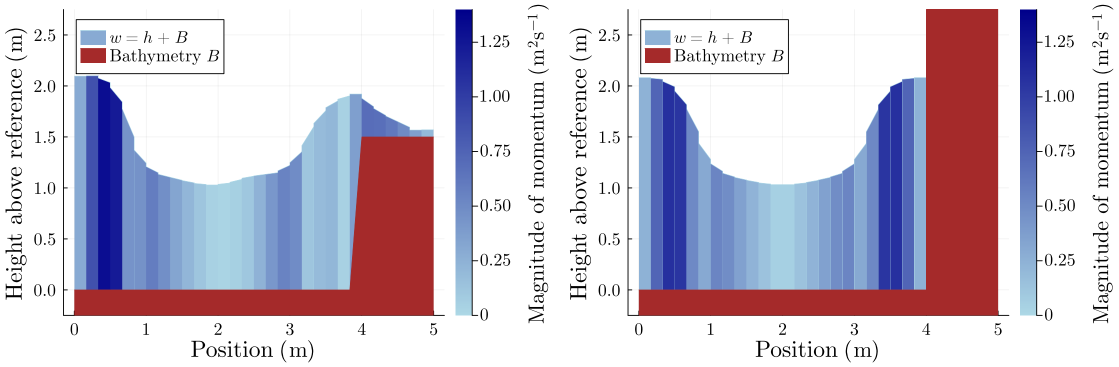
One could solve this by not using the exact heights of buildings, but rather a height comparable to that of the water. However, this height may vary and it defeats the purpose of the simulation, as this is precisely what the simulation aims to discover. Additionally, one may risk the building being submerged in water, allowing it to flow where it should not.



(a) Simulation using the TOP method. The solution is still somewhat symmetric.

(b) Simulation using the BC method.

Figure 16: Simulation of Case 2 after 2.8 s. Since the initial wave was symmetric, it should stay symmetric.



(a) Simulation using the TOP method.

(b) Simulation using the BC method.

Figure 17: A 1.5 m tall wall submerged in water after 0.5 s using the TOP method. This cannot be simulated with the BC method.

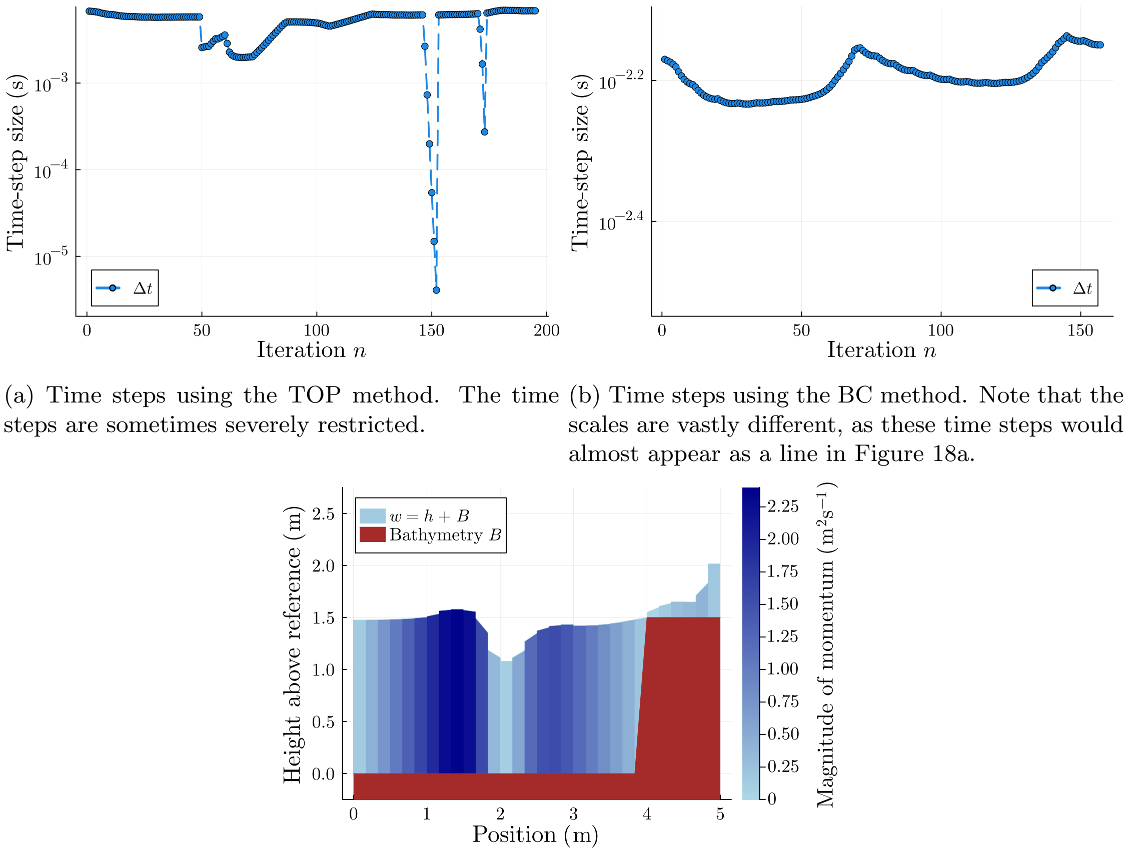
5.2 Submersible wall

Next, for *Case 2*, we will consider a one-dimensional five meter long "lake". The computational domain is given by $(x_L, x_R) = (0\,\text{m}, 5\,\text{m})$ on a grid with $N = 30$ cells. A wall starts at $x = 4\,\text{m}$ and covers the rest of the domain. This should therefore be equivalent to a four meter wide lake. For this case, we want to study how a standing wave shaped like a bell curve centred at $x = 2\,\text{m}$ develops in time. When using the TOP method, the initial height and momentum is adjusted to be a steady state close to the wall.

This wave should be symmetric around $x = 2\,\text{m}$ at all times. This holds true for the BC method, as $h(2 - x) = h(x + 2)$ and $uh(2 - x) = -uh(2 + x)$ and can be seen in Figure 16. This seems to be the case for for the TOP method too, except near the wall.

To demonstrate an example where the TOP method is more applicable than the BC method, we will lower the wall and use a taller initial standing wave such that the wave can flow above the wall; see Figure 17. This cannot be simulated using the BC method.

However, we still have some issues with the CFL condition. Plotting the time-steps that were used in the simulation using the TOP method, we see in Figure 18a that the time-steps can get even smaller than those discussed in Subsection 5.2. For comparison, the same situation was simulated using the BC method indicating that it may not be due to large momentum. This seems to be the case when the water on top of the wall is disconnected from the rest of the water as seen in Figure 18c. This could be caused by floating point errors in the computation of the



(a) Time steps using the TOP method. The time steps are sometimes severely restricted.

(b) Time steps using the BC method. Note that the scales are vastly different, as these time steps would almost appear as a line in Figure 18a.

(c) Simulation is performed until $t \approx 0.73\,\mathrm{s}$, corresponding to $n = 152$, which represents the smallest time step in Figure 18a.

Figure 18: Simulation of Case 2 over one second with a large wave capable of submerging the wall.

eigenvalues $\lambda(\boldsymbol{U}) = \frac{hu}{h} \pm \sqrt{gh}$ when h is small. Kurganov and Petrova (2007) presents some de-singularization methods, one of which is used in SinFVM.jl. Perhaps an alternative de-singularization method tailored to these cases could improve the results. Brodtkorb and Holm (2021) presents a method that may be more applicable to these problems. This is also implemented in SinFVM.jl, but the author was not aware of this.

6 Conclusion

This project report is divided into two parts. First, the theory for scalar hyperbolic conservation laws were reviewed to build some intuition. Some stability results were explored and carried over into the discrete sense motivating the behaviour of finite volume methods.

The second part consisted of simulation of urban flooding using the shallow-water equations. It aimed to find a solution to the problems that arise when vertical obstacles such as buildings and walls are included within the computational domain. The TOP method models such obstacles using the bottom topography. Alternatively, one can enforce the boundary condition on these obstacles, which is done in the BC method.

One problem with the TOP method is that it generates a momentum on the sides of the walls modelled using the bottom topography. This is both non-physical and can significantly reduce the time steps, making the simulations more computationally expensive. This gets worse with taller obstacles. However, the physical artifacts gets smaller for finer grids. Nevertheless, such fine grids are impractical for large-scale simulations, such as those covering entire cities. The BC method faces neither of these issues.

On the other hand, the TOP method is more general, as it can model smaller obstacles that risk being submerged in water. Therefore, a possible solution is to combine both methods. Tall walls and buildings that are guaranteed not to be submerged in water could be modelled with the BC method and everything else with the TOP method. This could improve accuracy and reduce computation time. However, there are still issues with this combined method. When water barely submerge walls in water as in Subsection 5.2, we still risk time steps several orders of magnitude smaller than the usual time steps.

This may be due to the de-singularization of u when computing the eigenvalues, and could therefore be explored further in future work. Additional topics could be to test other methods from the literature or implement the BC method in two dimensions. Furthermore, it would be interesting to make the simulator differentiable, enabling the optimization of wall placement for flood prevention.

Bibliography

Bridson, Robert (2008). *Fluid Simulation for Computer Graphics.* Natick, MA: A K Peters/CRC Press. ISBN: 9781568813264.

Brodtkorb, André R. and Håvard Heitlo Holm (Jan. 2021). 'Coastal ocean forecasting on the GPU using a two-dimensional finite-volume scheme'. In: *Tellus A: Dynamic Meteorology and Oceanography.* DOI: 10.1080/16000870.2021.1876341.

Crandall, Michael G. and Andrew Majda (Jan. 1980). 'Monotone difference approximations for scalar conservation laws'. English (US). In: *Mathematics of Computation* 34.149, pp. 1–21. ISSN: 0025-5718. DOI: 10.1090/S0025-5718-1980-0551288-3.

Gottlieb, Sigal, Chi-Wang Shu and Eitan Tadmor (2001). 'Strong Stability-Preserving High-Order Time Discretization Methods'. In: *SIAM Review* 43.1, pp. 89–112. DOI: 10.1137/S003614450036757X. eprint: https://doi.org/10.1137/S003614450036757X. URL: https://doi.org/10.1137/S003614450036757X.

Harten, Ami (1983). 'High resolution schemes for hyperbolic conservation laws'. In: *Journal of Computational Physics* 49.3, pp. 357–393. ISSN: 0021-9991. DOI: https://doi.org/10.1016/0021-9991(83)90136-5. URL: https://www.sciencedirect.com/science/article/pii/0021999183901365.

Holden, Helge and Nils Henrik Risebro (2002). *Front Tracking for Hyperbolic Conservation Laws.* Springer Berlin, Heidelberg.

Kurganov, Alexander and Guergana Petrova (2007). 'A Second-Order Well-Balanced Positivity Preserving Central-Upwind Scheme for the Saint-Venant System'. In: *Communications in Mathematical Sciences* 5.1, pp. 133–160.

LeFloch, Philippe G. (2002). *Hyperbolic Systems of Conservation Laws.* Birkhäuser Basel.

LeVeque, Randall J. (2012). *Numerical Methods for Conservation Laws.* Birkhäuser Basel.

Mishra, Siddhartha, Ulrik Skre Fjordholm and Rémi Abgrall (2017). *Numerical methods for conservation laws and related equations.* https://www.uio.no/studier/emner/matnat/math/MAT-IN9240/h17/pensumliste/numcl_notes.pdf.