PDE Lab 6: Wave Equation

Basile Mouret

December 2025

Collaborated with: Olesia Zaichenko, Jakub Maly

1 Introduction

This report presents the implementation and validation of a numerical solver for the 1D linear shallow water equations. The equations are discretized using 2nd-order centered finite differences in space and a 4th-order Runge-Kutta method in time. Additionally, an extension to nonlinear equations is implemented and tested.

2 Mathematical Formulation

2.1 Governing Equations

The 1D linear shallow water equations in first-order formulation are:

$$\frac{\partial h}{\partial t} = -\overline{h} \cdot \frac{\partial v}{\partial x}$$

$$\frac{\partial v}{\partial t} = -g \cdot \frac{\partial h}{\partial x}$$

where:
- h is the water height perturbation
- v is the velocity
- $\overline{h}$ is the average water depth
- g is the gravitational acceleration

2.2 Periodic Boundary Conditions

We consider a periodic domain $\Omega = [0, L]$ with:

$$h(x) = h(x + n \cdot L)$$

$$v(x) = v(x + n \cdot L)$$

for $n \in \mathbb{Z}$.

2.3 Initial Conditions

The initial conditions are:

$$h(x, 0) = \exp(-(x - L/2)^2)$$

$$v(x, 0) = 0$$

3 Numerical Methods

3.1 Spatial Discretization

The domain is discretized with N points at locations $x_i = i \cdot \Delta x$ where $\Delta x = \frac{L}{N}$.

The first derivative is approximated using 2nd-order centered differences:

$$\frac{\partial f}{\partial x} \approx \frac{f_{i+1} - f_{i-1}}{2\Delta x}$$

with periodic boundary conditions handled by wrapping indices.

3.2 Temporal Discretization

The classical 4th-order Runge-Kutta method is used:

$$k_1 = f(t_n, U_n)$$

$$k_2 = f\left(t_n + \Delta \frac{t}{2}, U_n + \Delta \frac{t}{2} \cdot k_1\right)$$

$$k_3 = f\left(t_n + \Delta \frac{t}{2}, U_n + \Delta \frac{t}{2} \cdot k_2\right)$$

$$k_4 = f(t_n + \Delta t, U_n + \Delta t \cdot k_3)$$

$$U_{n+1} = U_n + \Delta \frac{t}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

4 Validation

4.1 Analytical Solution

The wave speed for the linear shallow water equations is:

$$c = \sqrt{g \cdot |\overline{h}|}$$

For a periodic domain, the wave returns to its initial state after time:

$$T = \frac{L}{c}$$

This property allows us to use the initial condition as the reference solution at time T.

With the parameters:
- Domain size: $L = 150000$ m
- Gravity: $g = 9.81$ m/s²
- Average depth: $|\overline{h}| = 5000$ m

We obtain:
- Wave speed: $c = \sqrt{9.81 \times 5000} \approx 221.47$ m/s
- Period: $T = \frac{150000}{221.47} \approx 677.3$ s

4.2 Convergence Study

To verify the numerical accuracy, simulations were run with different grid resolutions. The L2 error is computed as:

$$\|e\|_{L^2} = \sqrt{\int_0^L (h(x,T) - h(x,0))^2 dx} \approx \sqrt{\sum_i (h_i^T - h_i^0)^2 \Delta x}$$

DOFs	Δx (m)	L2 error (h)	L2 error (v)	Order (h)
64	2343.75	3.94e+01	1.60e+00	—
128	1171.88	9.50e+00	7.66e-01	2.05
256	585.94	8.94e-01	2.27e-01	3.41
512	292.97	5.78e-02	5.90e-02	3.95
1024	146.48	3.62e-03	1.47e-02	4.00

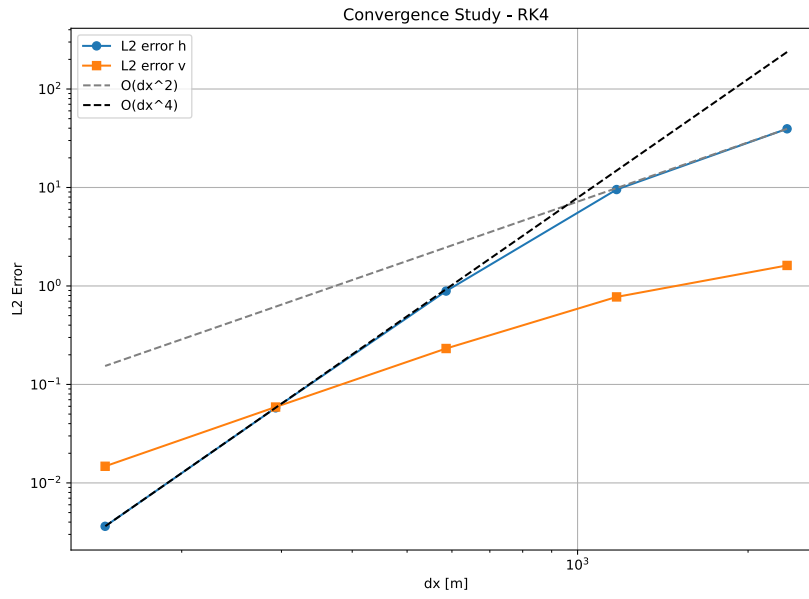Table 1: Convergence results for RK4 time integration

Figure 1: Convergence plot showing L2 error vs grid spacing

4.3 Discussion

The convergence study shows:

- At coarse resolutions, the spatial discretization error (2nd order) dominates.
- At fine resolutions, the convergence order approaches 4.
- The velocity error shows consistent 2nd order convergence, limited by the spatial discretization

The results confirm that the implementation is correct and achieves the expected order of accuracy.

5 Linear Wave Simulation

The linear wave simulation was run using the default parameters with RK4 time integration:

```
./out/main --timestepping-method rk4 --dt 0.1
```
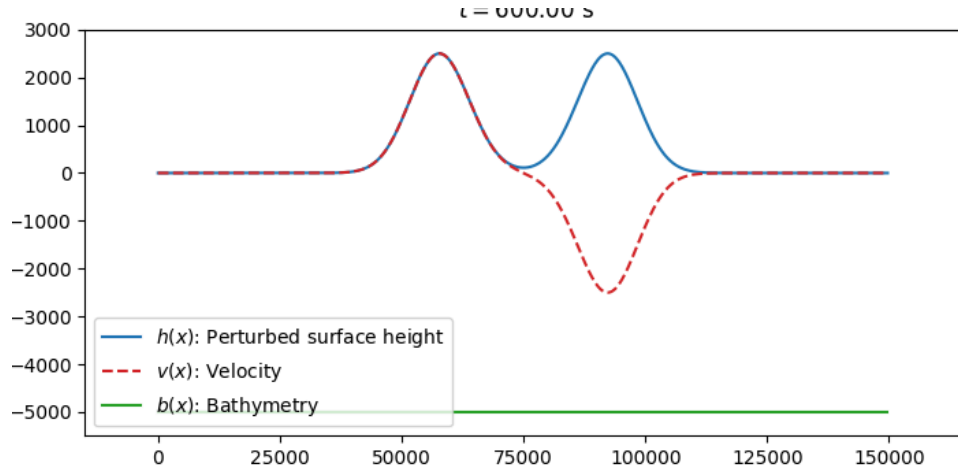
Figure 2: Linear wave propagation showing symmetric Gaussian waves traveling in both directions

The simulation shows two symmetric waves propagating to the left and right with wave speed $c = \sqrt{g \cdot |\overline{h}|}$. The waves maintain their shape due to the linear nature of the equations.

6 Individual Assignment: Extension to Nonlinearities

6.1 Nonlinear Shallow Water Equations

The full nonlinear shallow water equations include additional terms that were neglected in the linear formulation. The implemented nonlinear equations are:

$$\frac{\partial h}{\partial t} = -\overline{h} \cdot \frac{\partial v}{\partial x} - h \cdot \frac{\partial v}{\partial x} - v \cdot \frac{\partial h}{\partial x}$$

$$\frac{\partial v}{\partial t} = -g \cdot \frac{\partial h}{\partial x} - v \cdot \frac{\partial v}{\partial x}$$

The additional nonlinear terms are:

- $-h \cdot \frac{\partial v}{\partial x}$: advection of velocity by height perturbation
- $-v \cdot \frac{\partial h}{\partial x}$: advection of height by velocity
- $-v \cdot \frac{\partial v}{\partial x}$: self-advection of velocity

6.2 Implementation

The nonlinear terms were added to the `df_dt` function in `TimeStepperBase.hpp`. A command line flag `--nonlinear-equation` was added to enable the nonlinear equations.

6.3 Results

The simulation was run with the nonlinear equations enabled using:

```
./out/main --timestepping-method rk4 --dt 0.1 --nonlinear-equation 1 --simtime 30000
--bavg 100
```

With a shallower water depth of 100 m (instead of 5000 m), nonlinear effects become clearly visible. The ratio of perturbation to water depth is now:

$$\frac{h}{\overline{h}} \approx \frac{1}{100} = 0.01$$
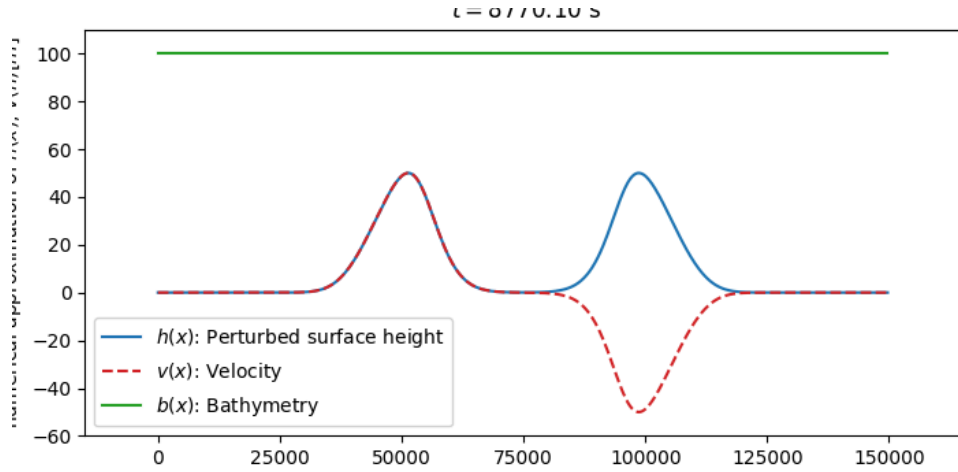
which is 50 times larger than with the default depth.

Figure 3: Wave profile with nonlinear equations showing wave steepening

The simulation shows clear wave steepening: the front of the wave becomes steeper than the back. This asymmetry is characteristic of nonlinear wave propagation and is the precursor to shock formation.

6.4 Discussion

The nonlinear effects are visible when the water depth is reduced, increasing the ratio $\frac{h}{h}$. With the default depth of 5000 m, no visible difference was observed between linear and nonlinear simulations. With 100 m depth and a longer simulation time of 30000 s, wave steepening is clearly visible.

The steepening occurs because in nonlinear waves, the wave speed depends on the local amplitude. Higher parts of the wave travel faster than lower parts, causing the wave front to steepen over time.

7 Conclusion

A numerical solver for the 1D linear shallow water equations was successfully implemented and validated. The 2nd-order centered difference scheme combined with the 4th-order Runge-Kutta method provides accurate and stable simulations. The convergence study confirms the expected order of accuracy.

The extension to nonlinear equations was implemented successfully. With shallow water depth (100 m instead of 5000 m), wave steepening characteristic of nonlinear wave propagation was observed.