EPFL

Parallel and High Performance Computing

Dr. Pablo Antolín

Final Project

April 10 2025

Shallow water equations

1 Problem description

An oceanographic researcher has written a simple predictive tool to simulate tsunamis as they move over oceans and overflow land. The researcher's original MATLAB code was prohibitively slow! With the help of a friend, together they rewrote the code in C++, which was much faster. However, running a high-resolution simulation can take several hours running on a single CPU of a workstation.
You are asked to parallelize the code using the techniques that you have learned in MATH-454 and the computational resources available at SCITAS so to help the researcher.

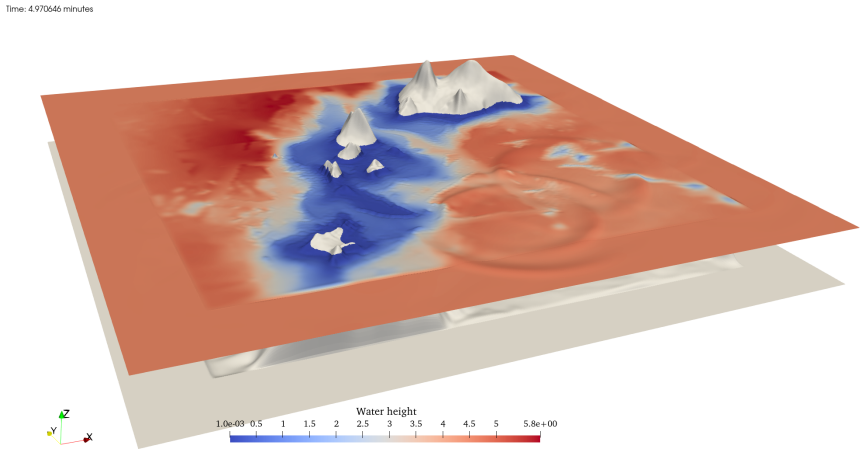

Figure 1: Simulation of tsunami using a shallow water equations model.

2 Shallow Water Equations

2.1 Mathematical model

The developed simulation tool is based on shallow water wave equations, a non-linear hyperbolic system of coupled partial differential equations often used to model various wave phenomenons. Simulation of ocean waves, river flows, hydraulic engineering, and atmospheric modeling are among the many areas of application.
The researcher has used a two-dimensional version of the equations along with a right-hand-side source term. The problem reads as follows: Find $h(x, y, t)$, $u(x, y, t)$ and $v(x, y, t)$ that at every point (x, y) and at every instant t satisfy:

$$h_t + (hu)_x + (hv)_y = 0 \tag{1a}$$

$$(hu)_t + \left(hu^2 + \frac{gh^2}{2}\right)_x + (huv)_y = -ghz_x \tag{1b}$$

$$(hv)_t + (huv)_x + \left(hv^2 + \frac{gh^2}{2}\right)_y = -ghz_y \tag{1c}$$

where h is the water height, u and v are the horizontal velocities at time t in the x and y directions respectively, g is the gravitational constant, and $z := z(x, y)$ is the topography of the ocean floor[1]. The notation a_b, with $b \in \{t, x, y\}$, denotes the partial derivative of a with respect to time t or spatial coordiantes x and y, respectively.

2.2 Numerical scheme

In order to approximate the solution of the problem (1), the researcher has discretized the equations using a finite volume method, a popular class of numerical methods for approximating the solution of hyperbolic equations.
For doing so, the researcher has used a structured rectangular uniform mesh in the x and y directions, such that $I_{i,j} = \left[x_{i-1/2}, x_{i+1/2}\right] \times \left[y_{j-1/2}, y_{j+1/2}\right]$ is the domain of a single cell. In a finite-volume scheme, one seeks to find the cell average $\bar{q}_{i,j}(t_n)$ that approximates a certain quantity $q(x_i, y_j, t_n)$ at every cell $I_{i,j}$ for a given time-step t_n in the sense

$$\bar{q}^n_{i,j} = \frac{1}{\Delta x \Delta y} \int_{I_{i,j}} q(x, y, t_n) \mathrm{d}x\mathrm{d}y \,, \tag{2}$$

where Δx and Δy are the cell sizes in the x and y directions, respectively.
The research has used the Lax-Friedrichs scheme to discretrize the problem. This is a first-order explicit scheme: the solution $\bar{h}^{n+1}_{i,j}$ at the next time-step t_{n+1} is computed from the solution $\bar{h}^n_{i,j}$ at the current time-step t_n, without the need to solve a system of equations. The same applies to the velocities $\overline{hu}^{n+1}_{i,j}$ and $\overline{hv}^{n+1}_{i,j}$

[1] The problem (1) has to be complemented with initial and boundary conditions for h, u, and v. These details have been omitted for the sake of brevity.

The Lax-Friedrichs scheme is given by the following equations:

$$\begin{aligned}
\bar{h}_{i,j}^{n+1} &= \frac{1}{4}\left(\bar{h}_{i+1,j}^{n} + \bar{h}_{i-1,j}^{n} + \bar{h}_{i,j+1}^{n} + \bar{h}_{i,j-1}^{n}\right) \\
&\quad - \frac{\Delta t}{2\Delta x}\left(\overline{hu}_{i+1,j}^{n} - \overline{hu}_{i-1,j}^{n}\right) - \frac{\Delta t}{2\Delta y}\left(\overline{hv}_{i,j+1}^{n} - \overline{hv}_{i,j-1}^{n}\right), \qquad (3a) \\
\overline{hu}_{i,j}^{n+1} &= \frac{1}{4}\left(\overline{hu}_{i+1,j}^{n} + \overline{hu}_{i-1,j}^{n} + \overline{hu}_{i,j+1}^{n} + \overline{hu}_{i,j-1}^{n}\right) - \Delta t^n g z_x \bar{h}_{i,j}^{n+1} \\
&\quad - \frac{\Delta t^n}{2\Delta x}\left(\frac{\left(\overline{hu}_{i+1,j}^{n}\right)^2}{\bar{h}_{i+1,j}^{n}} - \frac{\left(\overline{hu}_{i-1,j}^{n}\right)^2}{\bar{h}_{i-1,j}^{n}} + \frac{1}{2} g\left(\left(\bar{h}_{i+1,j}^{n}\right)^2 - \left(\bar{h}_{i-1,j}^{n}\right)^2\right)\right) \qquad (3b) \\
&\quad - \frac{\Delta t^n}{2\Delta y}\left(\frac{\overline{hu}_{i,j+1}^{n}\overline{hv}_{i,j+1}^{n}}{\bar{h}_{i,j+1}^{n}} - \frac{\overline{hu}_{i,j-1}^{n}\overline{hv}_{i,j-1}^{n}}{\bar{h}_{i,j-1}^{n}}\right), \\
\overline{hv}_{i,j}^{n+1} &= \frac{1}{4}\left(\overline{hv}_{i+1,j}^{n} + \overline{hv}_{i-1,j}^{n} + \overline{hv}_{i,j+1}^{n} + \overline{hv}_{i,j-1}^{n}\right) - \Delta t^n g z_y \bar{h}_{i,j}^{n+1} \\
&\quad - \frac{\Delta t^n}{2\Delta y}\left(\frac{\left(\overline{hv}_{i,j+1}^{n}\right)^2}{\bar{h}_{i,j+1}^{n}} - \frac{\left(\overline{hv}_{i,j-1}^{n}\right)^2}{\bar{h}_{i,j-1}^{n}} + \frac{1}{2} g\left(\left(\bar{h}_{i,j+1}^{n}\right)^2 - \left(\bar{h}_{i,j-1}^{n}\right)^2\right)\right) \qquad (3c) \\
&\quad - \frac{\Delta t^n}{2\Delta x}\left(\frac{\overline{hu}_{i+1,j}^{n}\overline{hv}_{i+1,j}^{n}}{\bar{h}_{i+1,j}^{n}} - \frac{\overline{hu}_{i-1,j}^{n}\overline{hv}_{i-1,j}^{n}}{\bar{h}_{i-1,j}^{n}}\right).
\end{aligned}$$

Note the index n on Δt^n which indicates that the time-step Δt is not constant along the simulation. Indeed, to ensure stability of the scheme, the time-step Δt^n must satisfy the CFL condition:

$$\Delta t^n \leq \frac{\min(\Delta_x, \Delta_y)}{\sqrt{2}\max_{i,j} \nu_{i,j}^n}, \qquad (4)$$

where

$$\nu_{i,j}^n = \left[\left(\left|\frac{\overline{hu}_{i,j}^{n}}{\bar{h}_{i,j}^{n}}\right| + \sqrt{g\bar{h}_{i,j}^{n}}\right)^2 + \left(\left|\frac{\overline{hv}_{i,j}^{n}}{\bar{h}_{i,j}^{n}}\right| + \sqrt{g\bar{h}_{i,j}^{n}}\right)^2\right]^{1/2} \qquad (5)$$

is the wave speed at cell $I_{i,j}$ at time-step t_n.
After computing a new time-step $\bar{h}_{i,j}^{n+1}$ from $\bar{h}_{i,j}^{n}$, cells that are below a certain water height threshold are considered dry, they are set as inactive and their velocity is set to zero:

$$\begin{aligned}
\bar{h}_{i,j}^{n+1} &= \begin{cases} \bar{h}_{i,j}^{n+1} & \text{if } \bar{h}_{i,j}^{n+1} > 0 \\ 10^{-5} & \text{otherwise} \end{cases}, \\
\overline{hu}_{i,j}^{n+1} &= \begin{cases} \overline{hu}_{i,j}^{n+1} & \text{if } \bar{h}_{i,j}^{n+1} > 10^{-4} \\ 0 & \text{otherwise} \end{cases}, \\
\overline{hv}_{i,j}^{n+1} &= \begin{cases} \overline{hv}_{i,j}^{n+1} & \text{if } \bar{h}_{i,j}^{n+1} > 10^{-4} \\ 0 & \text{otherwise} \end{cases}.
\end{aligned}$$

3 The project

The serial C++ code can be found in the folder `project_swe` of the repository `math454-phpc/exercises-2025`. Study it carefully!

Your task is to develop two parallel versions of the code: one using MPI and another one with CUDA.
In order to parallelize the proper parts of the code, you should first profile the sequential code and determine which part needs to be parallelized. Based on this decision, you should be able to predict the sequential fraction of the code and thus apply Amdhal's and Gustafson's laws.

3.1 Building and running the code

For building the provided serial code, you need to load the following modules (in `jed` or `izar`):

```
module load gcc hdf5
```

or

```
module load intel hdf5
```

and then invoke `make` in the folder `project_swe`. Then you can run the executable `swe`.

3.2 Test cases

Three different test cases are provided right now in the code:

a) Two drops of water falling (local Gaussian elevations) in a square domain with reflective boundaries (see Figure 2 (left)).

b) A simplified tsunami case built through analytical functions (see Figure 2 (right)).

c) A realistic tsunami (see Figure 1).

The different alternatives can be selected within the file `main.cc`.

The topography and initial conditions for the first two cases are automatically built in an analytical way. Therefore the user can set the number of cells per direction in an arbitrary way.
However, the setup of the realistic tsunami case requires to load the topography and initial conditions from external HDF5 data files. Files are provided in the code folder for small number of cells (`Data_nx*_500km.h5`), while for larger number of cells can be found here.

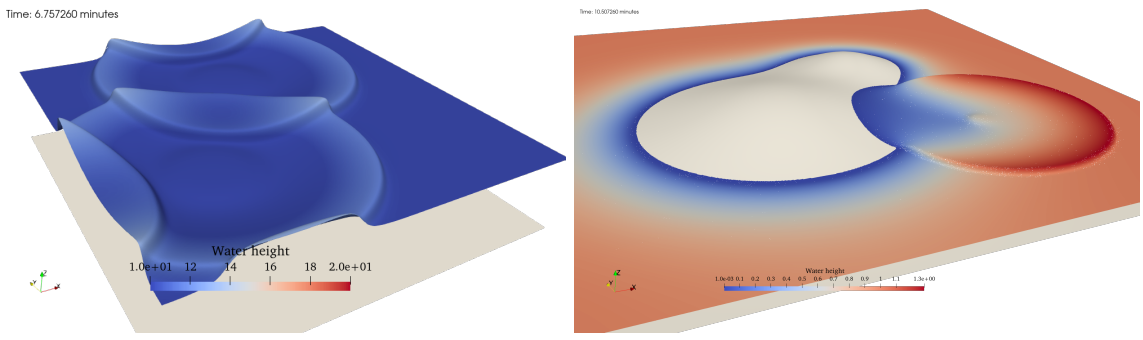

Figure 2: Two simple simulation cases: two Gaussian (water drops) falling (left) and analytical tsunami (right).

4 Output

The code can be configured to produced XDMF files (HDF5 format) for visualization in ParaView (see the options in `main.cc`). State files (`pv_*.pvsm`) are provided to ease the visualization of the results in ParaView (load them in the `File > Load State` menu).

However, be aware that the output files can be very large and a large number of files may be produced. Set the variable `output_n` in `main.cc` carefully (`output_n=0` deactivates the output).

Once the generated, the output files can be visualized in ParaView directly from the server (follow the instructions here), or the files can be downloaded to your local machine and visualized there.

IMPORTANT: In principle, the parallelization of the input (reading the data files) and output (writing the data files) is not required. But if you want to do it (expecially for the output), you can use the HDF5 library in parallel mode.

5 Submitting the project and oral presentation

The evaluation of the project will be based on a report, the source code, and an oral presentation. The report should be no longer than 4 pages. You should submit your code as and archive (tarball): Please provide the source code for the MPI and the CUDA versions in two different folders. You must also prepare a presentation and submit your slides before the exam. Check the deadlines below. The presentation must have 3 slides:

1 Strong scaling with Amdhal's prediction (MPI).

2 Weak scaling with Gustafson's prediction (MPI).

3 The influence of grid and block sizes on the performance of the CUDA version.

Important deadlines:

- The code, report, and slides have to be submitted through Moodle before June 8 (2025) at 23h59 CEST.
- The oral presentation will be less than 5 minutes + 5 minutes of questions about the project and the course in general.
- The oral presentations will be held according to EPFL's exams calendar. The precise schedule will be published on Moodle.

Access to computers: For doing the project, you will have access to the `jed` and `izar` computers. For submitting your jobs, use the following options:

- `--account=math-454`
- `--qos=math-454`