POLITECNICO
MILANO 1863

# Finite element method for the Shallow Water Equations

Andrea Barletta (10665046), Enrico Tirri (10742316)

A.Y. 2025/2026

# Contents

# Introduction

*Shallow water equations* (SWEs) is a system of partial differential equations that can be used to study flow in fluid domains whose vertical length scale $H$ is much smaller than the characteristic horizontal direction length scale $L$ and in which the underwater topography does not change too fast.
The SWEs are widely used in applications where free-surface effects are dominant and vertical accelerations can be neglected. Typical use cases include river and coastal hydraulics, flood propagation and inundation modeling, dam-break problems, tsunami and storm-surge simulation, and large-scale oceanic and atmospheric flows.
From a numerical perspective, the shallow water equations present several challenges, for example the presence of non-linear terms, stability issues due to advection-dominated phenomena, and the accurate resolution of discontinuities without spurious oscillations.

# Mathematical formulation

## Incompressible Navier-Stokes Equations

The equations are derived from the *Incompressible Navier-Stokes equations* (INSE), which take the following form:

$$\begin{cases} \partial_t \rho \mathbf{u} + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u}) + \rho(f\hat{\mathbf{k}} \times \mathbf{u} - \mathbf{g}) + \nabla p - \nabla \cdot \bar{T} = 0 & \text{(ME-INS)} \\ \nabla \cdot \mathbf{u} = 0 & \text{(CE-INS)} \end{cases}$$

Where $\Omega \subseteq \mathbb{R}^3$ is the spatial domain, $t \in \mathbb{R}$ is the time, $\rho \in \mathbb{R}$ is the fluid density, $\mathbf{u} \in \mathbb{R}^3$ is the fluid velocity, $p \in \mathbb{R}$ is the fluid pressure, $f \in \mathbb{R}$ is the Coriolis parameter, $\mathbf{g} \in \mathbb{R}^3$ represents the acceleration due to gravity, $\bar{T}$ is the viscous stress tensor, and $\hat{\mathbf{k}}$ is a normal vector in the vertical direction.

## Shallow Water Equations

Given the INSEs, we can obtain our SWEs by depth-integrating ME-INS and CE-INS. The full derivation of the equations is beyond the scope of this project, and the full details can be found, for example in [3] and [10].
The top and bottom boundaries for the depth-integration are represented in the following diagram

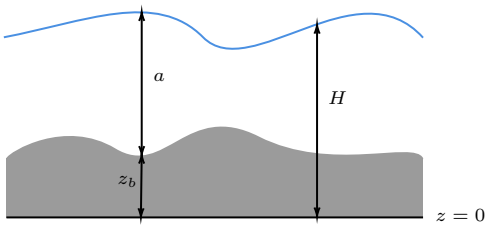


Figure 1: Top and bottom boundaries

### Assumptions

In order to simplify the mathematical formulation used in this project, we will use a slightly modified version of the SWEs, in particular we will assume:

- **Flat bottom topology**: $z_b(\vec{x}, t) = 0$, which implies $H = a$
- **Conservative form**: We will disregard lateral stress and driving forces. Moreover, we assume the simplest expression for the bottom stress, as done in Chapter 2 of [10]
- **Ignore Coriolis terms**: We will also disregard eventual Coriolis terms (equivalent to settings $f = 0$)

**Continuity Equation**

Given the depth-integration of CE-INS and flat bottom topology assumption, we obtain the Continuity Equation of the SWEs system, that is

$$\partial_t H + \nabla \cdot (H\mathbf{u}) = 0 \tag{CE-SW}$$

**Momentum Equation**

Given the depth integration of (ME-INS) and disregarding lateral stress and driving forces, we obtain the Momentum Equation of the SWEs system, that is

$$\partial_t \mathbf{u} + (\mathbf{u} \cdot \nabla)\mathbf{u} + g\nabla H + \frac{c_f}{H}||\mathbf{u}||\mathbf{u} = 0 \tag{ME-SW}$$

**Boundary Conditions**

Fluid dynamics boundary conditions typically take either Dirichlet or Neumann boundary conditions; however, for the Shallow water equations they can take various forms depending on whether boundaries are fixed, moving or open. For the sake of simplicity, on our numerical study, we will only consider fixed boundaries, moreover we will assume the so called *no-slip* condition for the velocity on the boundary:

$$\mathbf{u} = \mathbf{0} \qquad \text{on } \Gamma_D = \partial\Omega$$

**Note on the dimensions**

Although our starting Navier-Stokes Equations were in a spatial domain $\Omega \subseteq \mathbb{R}^3$, by depth-integrating we remove one spatial dimension. Therefore, from now on, we will consider $\Omega \subseteq \mathbb{R}^2$ and $\mathbf{u} \in \mathbb{R}^2$

# Numerical formulation

## Weak formulation

The weak formulation is obtained in the usual way by multiplying the momentum and continuity equations with the test functions $\boldsymbol{\varphi} \in V$ and $\psi \in W$ respectively, where $V \subseteq \mathcal{H}^1(\Omega) \times \mathcal{H}^1(\Omega)$ (with $\mathcal{H}^1(\Omega)$ the usual one-dimensional Sobolev space) and $W \subseteq L^2(\Omega)$.
In particular:

- **Continuity equation**:

$$(\partial_t H, \psi)_\Omega + (\nabla \cdot (H\mathbf{u}), \psi)_\Omega = 0$$

where $(\cdot, \cdot)_\Omega$ indicates the standard $L^2(\Omega)$inner product.
Using integration by parts, we can simplify the second term as follows

$$\int_\Omega \nabla \cdot (H\mathbf{u})\psi dV = -\int_\Omega H\mathbf{u}\nabla\psi dV + \int_{\partial\Omega} \psi H\mathbf{u} \cdot \hat{\mathbf{n}} dA$$

Due to the *no-slip* conditions, the second term vanishes. Therefore, we obtain the following formulation for the continuity equation

$$(\partial_t H, \psi)_\Omega - (H\mathbf{u}, \nabla\psi)_\Omega = 0$$

Hence, we obtain the final weak formulation for the continuity equation:

Find $\mathbf{u} \in V$, $H \in W$ s.t.

$$(\partial_t H, \psi)_\Omega - (H\mathbf{u}, \nabla\psi)_\Omega = 0 \qquad \text{(CE-SW-W)}$$

$\forall t \in (0, T)$, $\forall \psi \in W$

- **Momentum equation**:

$$(\partial_t \mathbf{u}, \boldsymbol{\varphi})_\Omega + ((\mathbf{u} \cdot \nabla)\mathbf{u}, \boldsymbol{\varphi})_\Omega - (g\nabla H, \boldsymbol{\varphi})_\Omega + \left(\frac{c_f}{H}||\mathbf{u}||\mathbf{u}, \boldsymbol{\varphi}\right)_\Omega = 0$$

Similarly to what was done in the continuity equation, we can use integration by parts on the fourth term to simplify it as follows

$$\int_\Omega g\nabla H \cdot \boldsymbol{\varphi} dV = -g\int_\Omega H\nabla \cdot \boldsymbol{\varphi} dV + g\int_{\partial\Omega} H\boldsymbol{\varphi} \cdot \hat{\mathbf{n}} dA$$

Due to the *no-slip* conditions, the second term vanishes. The final weak formulation for the momentum equation is therefore the following:

Find $\mathbf{u} \in V$, $H \in W$ s.t.

$$(\partial_t \mathbf{u}, \boldsymbol{\varphi})_\Omega + ((\mathbf{u} \cdot \nabla)\mathbf{u}, \boldsymbol{\varphi})_\Omega - (gH, \nabla \cdot \boldsymbol{\varphi})_\Omega + \left(\frac{c_f}{H}||\mathbf{u}||\mathbf{u}, \boldsymbol{\varphi}\right)_\Omega = 0 \qquad \text{(ME-SW-W)}$$

$\forall t \in (0, T)$, $\forall \boldsymbol{\varphi} \in V$

## Space discretization

The weak formulations CE-SW-W and ME-SW-W are then discretized in space using the Galerkin method.
Let $X_h^r$ be the space of finite elements defined by

$$X_h^r = \left\{ v_h \in C^0(\Omega) : v_h|_K \in \mathbb{P}^r(K), \quad \forall K \in \mathcal{T}_h \right\}$$

where $\mathbb{P}^r(K)$ is the space of polynomials of degree $q \leq r$ and $\mathcal{T}_h$ is a triangulation of $\Omega$.
The discrete spaces for velocity and water surface height are therefore defines as $V_h = [X_h^{r_\mathbf{u}}]^2 \cap V$ and $W_h = [X_h^{r_H}] \cap W$ respectively.
The space discretized weak formulation of the Shallow water equations is therefore:

Find $\mathbf{u}_h \in V_h$, $H_h \in W_h$ s.t.

$$\begin{cases} (\partial_t H_h, \psi_h)_\Omega - (H_h \mathbf{u}_h, \nabla \psi_h)_\Omega = 0 \\ (\partial_t \mathbf{u}_h, \boldsymbol{\varphi}_h)_\Omega + ((\mathbf{u}_h \cdot \nabla)\mathbf{u}_h, \boldsymbol{\varphi}_h)_\Omega - (gH_h, \nabla \cdot \boldsymbol{\varphi}_h)_\Omega + \left( \frac{c_f}{H_h} ||\mathbf{u}_h|| \mathbf{u}_h, \boldsymbol{\varphi}_h \right)_\Omega = 0 \end{cases} \tag{SW-H}$$

$\forall t \in (0,T)$, $\forall \boldsymbol{\varphi}_h \in V_h$, $\forall \psi_h \in W_h$

## Time discretization

Let $J_{\Delta t}$ be a discrete subset of $[0,T]$:

$$J_{\Delta t} = \{ t^n | t^n \in [0,T], t^n = n\Delta t, 0 \leq n \leq N, N\Delta t = T, \Delta t > 0 \}$$

Then $\mathbf{u}_h^n$ and $H_h^n$ are the discretized velocity and the water surface height at timestep $n$.
Using the Crank-Nicholson method for time discretization yields the following fully discretized problem:

Given $\mathbf{u}_h^0$, $H_h^0$, find $\mathbf{u}_h^{n+1} \in V_h$, $H_h^{n+1} \in W_h$ s.t.

$$\begin{cases} \left( \frac{H_h^{n+1} - H_h^n}{\Delta t}, \psi_h \right)_\Omega + \frac{1}{2} a(\mathbf{u}_h^{n+1}, H_h^{n+1}) + \frac{1}{2} a(\mathbf{u}_h^n, H_h^n) = 0 \\ \left( \frac{\mathbf{u}_h^{n+1} - \mathbf{u}_h^n}{\Delta t}, \boldsymbol{\varphi}_h \right)_\Omega + \frac{1}{2} b(\mathbf{u}_h^{n+1}, H_h^{n+1}) + \frac{1}{2} b(\mathbf{u}_h^n, H_h^n) = \frac{1}{2} f(H_h^{n+1}) + \frac{1}{2} f(H_h^n) \end{cases} \tag{SW-HT}$$

$\forall t \in J_{\Delta t}$, $\forall \boldsymbol{\varphi}_h \in V_h$, $\forall \psi_h \in W_h$

where

$$a(\mathbf{u}_h^n, H_h^n) = -(H_h^n \mathbf{u}_h^n, \nabla \psi_h)_\Omega$$

$$b(\mathbf{u}_h^n, H_h^n) = ((\mathbf{u}_h^n \cdot \nabla)\mathbf{u}_h^n, \boldsymbol{\varphi}_h)_\Omega + \left( \frac{c_f}{H_h^n} ||\mathbf{u}_h^n|| \mathbf{u}_h^n, \boldsymbol{\varphi}_h \right)_\Omega$$

$$f(H_h^n) = (gH_h^n, \nabla \cdot \boldsymbol{\varphi}_h)_\Omega$$

## Dealing with nonlinear terms

The nonlinear terms are quasi-linearized using one term from the current time step and the other term as a linear extrapolation of the previous two time steps, as done in [6]. The linearly extrapolated terms are defined as follows:

$$\mathbf{u}^* = \mathbf{u}^{n+1/2} = \frac{3}{2}\mathbf{u}^n - \frac{1}{2}\mathbf{u}^{n-1} + \mathcal{O}(\Delta t^2)$$

which is a second-order explicit Adams–Bashforth extrapolation in time.
Therefore, we can modify the nonlinear terms as follows and obtain a linear system:

$$(\mathbf{u}_h^n \cdot \nabla)\mathbf{u}_h^n \longmapsto (\mathbf{u}_h^* \cdot \nabla)\mathbf{u}_h^n \qquad \frac{c_f}{H_h^n}||\mathbf{u}_h^n||\mathbf{u}_h^n \longmapsto \frac{c_f}{H_h^n}||\mathbf{u}_h^*||\mathbf{u}_h^n$$

Moreover, we can substitute $\mathbf{u}_h^n$ and $\mathbf{u}_h^{n+1}$ in the $H_h$ equation and obtain two independent equations to be solved in succession at each timestep, yielding the final fully discrete formulation:

Given $\mathbf{u}_h^0$, $H_h^0$

1. find $H_h^{n+1} \in W_h$ s.t.

$$\left(\frac{H_h^{n+1} - H_h^n}{\Delta t}, \psi_h\right)_\Omega + \frac{1}{2}a(\mathbf{u}_h^*, H_h^{n+1}) + \frac{1}{2}a(\mathbf{u}_h^*, H_h^n) = 0 \tag{SW-H-HT}$$

   $\forall \psi_h \in W_h$

2. find $\mathbf{u}_h^{n+1} \in V_h$ s.t.

$$\left(\frac{\mathbf{u}_h^{n+1} - \mathbf{u}_h^n}{\Delta t}, \boldsymbol{\varphi}_h\right)_\Omega + \frac{1}{2}b^*(\mathbf{u}_h^{n+1}, H_h^{n+1}) + \frac{1}{2}b^*(\mathbf{u}_h^n, H_h^{n+1}) = \frac{1}{2}f(H_h^{n+1}) + \frac{1}{2}f(H_h^n) \tag{SW-U-HT}$$

   $\forall \boldsymbol{\varphi}_h \in V_h$

$\forall t \in J_{\Delta t}$

where

$$b^*(\mathbf{u}_h^n, H_h^n) = ((\mathbf{u}_h^* \cdot \nabla)\mathbf{u}_h^n, \boldsymbol{\varphi}_h)_\Omega + \left(\frac{c_f}{H_h^n}||\mathbf{u}_h^*||\mathbf{u}_h^n, \boldsymbol{\varphi}_h\right)_\Omega$$

Note that AB2 is conditionally stable, therefore, even when using an unconditionally stable time-integrator (e.g. Crank-Nicolson), we may obtain a non-stable solver.
Moreover, the time-discretization error may be dominated by (AB2) and not (CN).

## Building the linear systems

Since the spaces $V_h$ and $W_h$ are finite-dimensional, we can write the discretized velocity and water surface height as a (finite) linear combination of appropriate basis functions $\boldsymbol{\varphi}_j$ and $\psi_j$. Therefore, $\mathbf{u}_h$ and $H_h$ can be expanded as follows:

$$\mathbf{u}_h(\vec{x},t) = \sum_{j=1}^{N_V} \mathbf{u}_j(t)\boldsymbol{\varphi}_j(\vec{x}) \qquad H_h(\vec{x},t) = \sum_{j=1}^{N_W} H_j(t)\psi_j(\vec{x})$$

moreover, since SW-H-HT and SW-U-HT hold for every test function, they must hold for all the basis functions $\psi_i$ and $\boldsymbol{\varphi}_i$. We can therefore rewrite the two equations as

$$\frac{M_H}{\Delta t}(H_h^{n+1} - H_h^n) + \frac{1}{2}A_H H_h^{n+1} + \frac{1}{2}A_H H_h^n = 0$$

$$\frac{M_U}{\Delta t}(\mathbf{u}_h^{n+1} - \mathbf{u}_h^n) + \frac{1}{2}A_U \mathbf{u}_h^{n+1} + \frac{1}{2}A_U \mathbf{u}_h^n = \frac{1}{2}F^{n+1} + \frac{1}{2}F^n$$

where:

- $M_H \in \mathbb{R}^{N_W \times N_W}$, $(M_H)_{ij} = \psi_i \psi_j$
- $A_H \in \mathbb{R}^{N_W \times N_W}$, $(A_H)_{ij} = -\mathbf{u}_h^* \cdot \nabla \psi_i \psi_j$
- $M_U \in \mathbb{R}^{N_V \times N_V}$, $(M_U)_{ij} = \boldsymbol{\varphi}_i \cdot \boldsymbol{\varphi}_j$
- $A_U \in \mathbb{R}^{N_V \times N_V}$, $(A_H)_{ij} = -\mathbf{u}_h^* \cdot \nabla \boldsymbol{\varphi}_i \boldsymbol{\varphi}_j + \frac{c_f}{H_h^{n+1}} ||\mathbf{u}_h^*|| \boldsymbol{\varphi}_i \cdot \boldsymbol{\varphi}_j$ (notice that $\nabla \boldsymbol{\varphi}_i \boldsymbol{\varphi}_j$ is a tensor contraction which yields a vector in $\mathbb{R}^{N_V}$)
- $F^n \in \mathbb{R}^{N_V}$, $(F^n)_i = g H_h^n \nabla \cdot \boldsymbol{\varphi}_i$

# Advection-dominated problem

The equations presented above cannot be solved reliably with the standard finite element method, especially the water height equation, which is an example of an advection-dominated problem. The main difficulty is that the solution does not have enough smoothness in directions perpendicular to the flow. While the solution is smooth along the streamlines defined by the velocity field $\mathbf{u}$, it can be non-smooth or even discontinuous in the transverse direction. These features cause numerical instabilities, making a stable solution with a standard continuous finite element discretization impossible.
A common way to overcome this issue is the streamline-upwind Petrov–Galerkin (SUPG) method, also known as the streamline diffusion method. A clear description of this method can be found in [5].

## SUPG stabilizer

The additional term introduced by the stabilizer in the water height equation is the following:

$$s(H, \psi) = \sum_{K \in \mathcal{T}_h} \tau_K \int_K \mathcal{L}(H) \mathcal{L}_{SS}(\psi)$$

where $\mathcal{L}(H)$ is the strong residual and $\mathcal{L}_{SS}(\psi)$ is its skew-symmetric part. In particular

$$\mathcal{L}(H) = \partial_t H + \mathbf{u} \cdot \nabla H$$

and, by computing the adjoint operator $\mathcal{L}^*$, we obtain

$$\mathcal{L}_{SS}(\psi) = \frac{\mathcal{L}(\psi) - \mathcal{L}^*(\psi)}{2} = \mathbf{u} \cdot \nabla \psi + \frac{1}{2}(\nabla \cdot \mathbf{u})\psi$$

Therefore, we can implement our SUPG stabilizer by adding the following terms to the mass and stiffness matrix:

$$M_{SUPG} = \frac{\psi_j \mathcal{L}_{SS}(\psi_i)}{\Delta t} \tag{SUPG-M}$$

$$A_{SUPG} = \left[\psi_j (\nabla \cdot \mathbf{u}_h^*) + \nabla \psi_j \cdot \mathbf{u}_h^*\right] \mathcal{L}_{SS}(\psi_i) \tag{SUPG-A}$$

Notice that, although the additional terms are not exactly mass or stiffness terms, we will use this division to group terms containing $\Delta t$.

## SUPG parameter

Among the various formulations for stabilization parameter available in literature, we chose the Tezduyar's UGN (Unit Gradient Norm) formulation [9].
The stabilization parameter can be defined as:

$$\tau_K = \left[ \left( \frac{2}{\Delta t} \right)^2 + \left( \frac{2||\mathbf{u}_h^*||}{h} \right)^2 \right]^{-1/2}$$

where $h$ is the cell diameter

## SUPG results

To assess the impact of SUPG stabilization, we performed dedicated tests in advection-dominated regimes. In particular, when the flow is characterized by relatively large velocities, the standard Galerkin formulation becomes unstable and exhibits nonphysical oscillations. In these cases, the SUPG term provides additional numerical diffusion aligned with the flow direction, effectively damping these oscillations while preserving the overall accuracy of the solution. We will therefore prescribe a fixed velocity field in the following form

$$\mathbf{u}(x, y, t) = \begin{bmatrix} 10 \sin(x) \sin(y) \\ 0 \end{bmatrix}$$

and an initial water height in the following form

$$H(x, y, 0) = 1 - 0.3x$$

The values for the $g$ and $c_f$ coefficients were based on standard values taken from the literature (see for example [10]), that is $g = 2.5 \times 10^{-4}$ and $c_f = 1 \times 10^{-2}$

After running our simulation up to final time $T = 0.2$ with timestep $\Delta t = 0.001$ on a mesh of size 50, we obtain the two following results

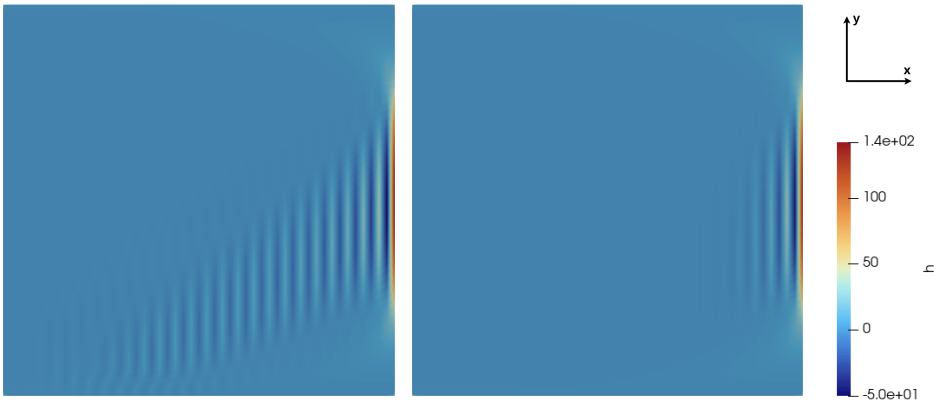


Figure 2: In order: w/o SUPG stabilizer, w/ SUPG stabilizer

As visible from the plots, the SUPG stabilizer managed to cancel (some) oscillations caused by the high-velocity advection field.

# Convergence and Stability

## Theoretical results

A key point is the choice of function spaces for the velocity and water height, which is critical to ensure stability, accuracy, and the proper representation of the wave dynamics. A common approach is to use mixed finite element spaces that satisfy an appropriate discrete inf–sup (Ladyzhenskaya–Babuška–Brezzi, LBB) condition to avoid spurious modes which, unlike the Navier–Stokes equations where oscillations typically appear only in the pressure field, can occur in both the surface elevation and the velocity fields.
Several studies (e.g. [7] for unstructured meshes) investigate mixed finite element pairs and their influence on the discrete inf–sup condition. Guided by these considerations, in this work we adopt the $P2$–$P1$ pair, using quadratic elements for the velocity and linear elements for the water height. This combination provides a good balance between accuracy and stability, while remaining computationally efficient. Based on the chosen numerical formulation and choice of finite element spaces, the expected (see for example [6]) convergence behaviour of the solver for the water height is second order in both space and time, that is

$$||H_{exact}(t^k) - H_h(t^k)||_{L^2(\Omega,T)} \leq C_1 h^2 + C_2 \Delta t^2$$

whereas it's third order in space and second order in time for the velocity field, that is

$$||\mathbf{u}_{exact}(t^k) - \mathbf{u}_h(t^k)||_{L^2(\Omega,T)} \leq C_3 h^3 + C_4 \Delta t^2$$

## Numerical results

To assess the convergence order of the solver in both space and time, we adopted a method of manufactured solutions. An analytical solution was prescribed a priori, and the corresponding source terms and boundary conditions were derived consistently and incorporated into the numerical model. The solver was then run on successively refined spatial meshes and with decreasing time step sizes, and the numerical solution was compared against the manufactured reference solution using appropriate error norms.
To test the convergence of our solver, we proposed the following manufactured solution:

$$H_{exact}(x, y, t) = 1 + xy\cos(\pi t)$$

$$\mathbf{u}_{exact}(x, y, t) = \begin{bmatrix} y\sin(\pi t)\sin(\pi x)\cos(\frac{\pi}{2}y) \\ \cos(\pi t)\sin(\pi x)\sin(\pi y) \end{bmatrix}$$

which satisfy the no-slip conditions. The values for the $g$ and $c_f$ coefficients were based on standard values taken from the literature as done in [2]

### Space convergence (Water height)

To test the water height convergence, we prescribe the velocity field from the exact (manufactured) solution. The final simulation time was set to $T = 0.1$, with a time step of $\Delta t = 0.001$ chosen to minimize the error introduced by the temporal discretization. A series of successively refined meshes with sizes 5, 10, 20, 50, 100, and 200 were used to evaluate the solver's spatial convergence.

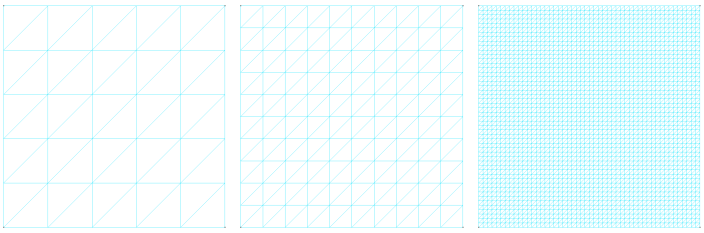

Figure 3: Example of meshes, namely size 5, 10 and 50

The error was computed by taking the $L^2$ norm of the difference between the numerical and manufactured solutions at each time step, and then combining these contributions across all time steps to obtain a single measure of the total error over the simulation. That is:

$$error_H = \sqrt{\sum_{k=0}^{T/\Delta t} ||H_{exact}(t^k) - H_h(t^k)||^2_{L^2(\Omega)}}$$

By plotting the errors against the mesh size, we get the following:

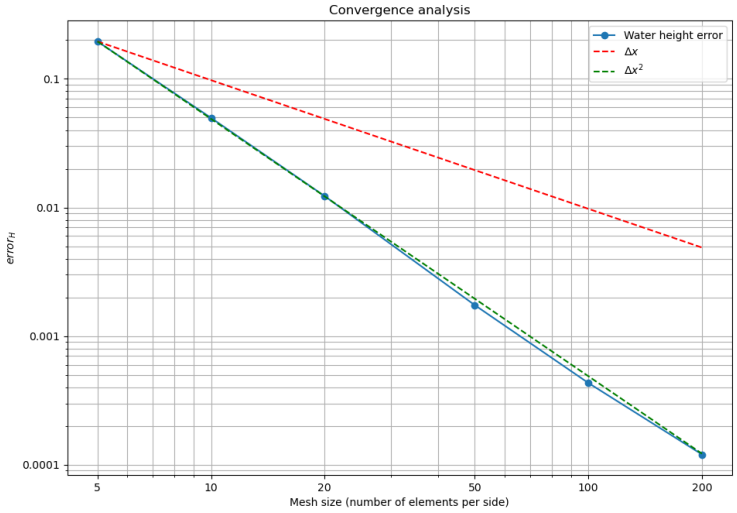


Figure 4: Space convergence of water height

**Time convergence (Water height)**

To assess the temporal convergence of the solver, we fixed a sufficiently fine mesh of size 200 to ensure that spatial discretization errors were negligible. The solver was then run with a series of decreasing timestep sizes $\Delta t \in \{0.01, 0.005, 0.0025, 0.00125\}$, up to a final time of $T = 1$. The error was evaluated at the final time step by comparing the numerical solution with the manufactured solution using the $L^2$ norm over the spatial domain.
By plotting the errors against the timestep size, we get the following:

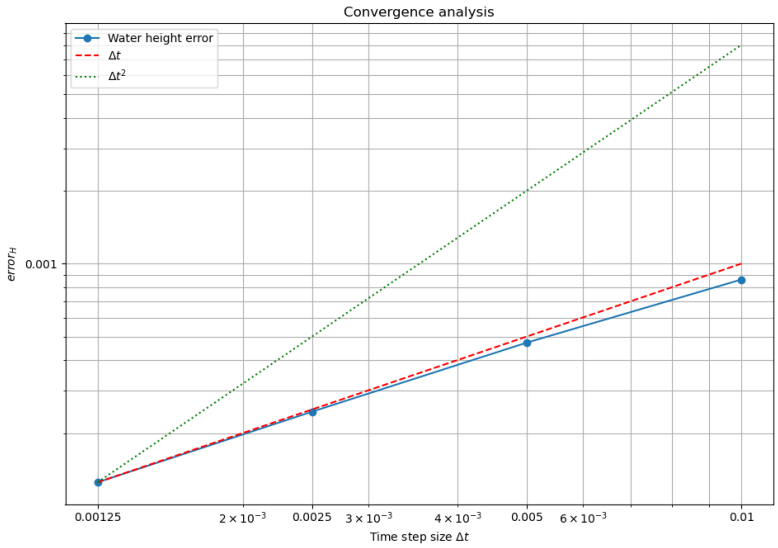


Figure 5: Time convergence of water height

Although the theoretical time accuracy of the (CN) + (AB2) scheme was expected to be second order, the convergence tests performed in this work exhibited an order of one. This deviation from the expected behaviour is likely not a numerical artifact but rather a consequence of the specific time discretization strategy adopted. In particular, it is well documented in the literature that IMEX-type schemes, where different terms are treated implicitly and explicitly, may suffer from order reduction even when the underlying implicit method is formally second order. In such settings, coupling errors and stiffness effects can dominate the temporal error, leading to first-order convergence in practice. For example in [11], the authors report second-order temporal convergence for the compressible euler equations problem when using a fourth-order Runge–Kutta IMEX scheme. Moreover, in [2] and references, they briefly discussed about possible causes for order reduction, which could stem from stiffness of the equations, lack of spatial smoothness for $\mathbf{u}$ or particular boundary conditions.

### Stability condition for velocity

Assessing the convergence of the velocity field requires special attention due to the conditional stability of the AB2 interpolator used for linearizing the system. The timestep must satisfy the CFL-type constraint

$$\frac{\Delta t}{\Delta x} \leq C$$

While this condition guarantees stable integration, timesteps near this limit can introduce local truncation errors that accumulate over multiple steps, increasing the global error. An additional complication arises from the AB2 interpolator, which requires the solution at time step $n-1$ to initialize the scheme. If this value is not available, for example in the first timestep, the interpolation error introduced at can dominate. In our implementation, this was handled by initializing the first two time steps from the exact solution, although more sophisticated startup procedures or one-step methods could be employed in a more robust setting.

### Space convergence (Velocity)

To observe the expected third-order convergence in space, we had to use extremely small timesteps (order of $10^{-5}$), as larger values caused the error to be dominated by the AB2 interpolation error. The final simulation time was set to $T = 10^{-3}$, and a series of successively refined meshed with size 5, 10, 20, and 100 were used to evaluate the solver's spatial convergence. The total error was computed in the same way as for the water height, namely

$$error_{\mathbf{u}} = \sqrt{\sum_{k=0}^{T/\Delta t} ||\mathbf{u}_{exact}(t^k) - \mathbf{u}(t^k)||^2_{L^2(\Omega)}}$$

By plotting the error against the mesh size, we get the following:

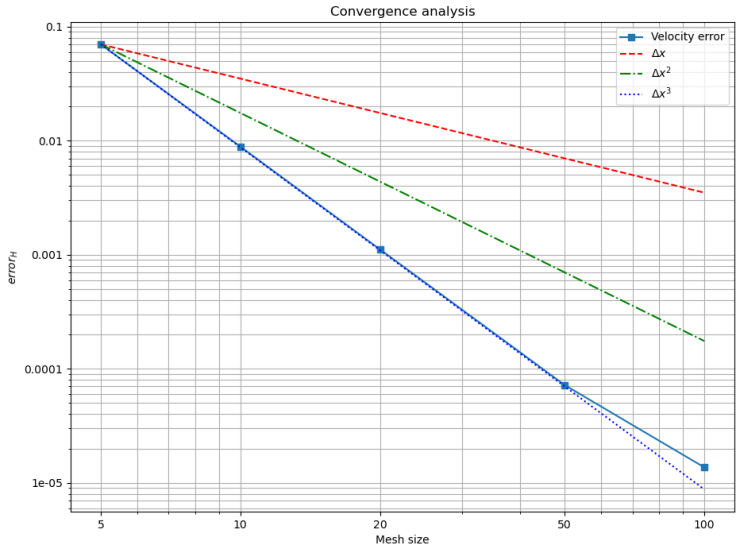


Figure 6: Space convergence of velocity field

**Time convergence (Velocity)**

Similar to what was done for the water height, we will fix a mesh sufficiently fine mesh of size 200 to ensure that spatial discretization error is negligible. We will then run the solver with a series of decreasing timesteps, keeping in mind that they should all be lower than the bound set by the CFL-type condition.
In practice, we were not able to identify a test configuration that clearly exhibited second-order convergence in time. To isolate temporal errors, the mesh had to be sufficiently fine so that spatial discretization errors were negligible; however, this required choosing time steps small enough to satisfy the stability constraint of the AB2 interpolator. At the same time, the dynamics of the solution needed to be sufficiently fast to prevent the initial GMRES guess from already satisfying the convergence tolerance, which would result in zero iterations and mask the true temporal error. The usable $(\Delta h, \Delta t)$ can be summarized using the following plot

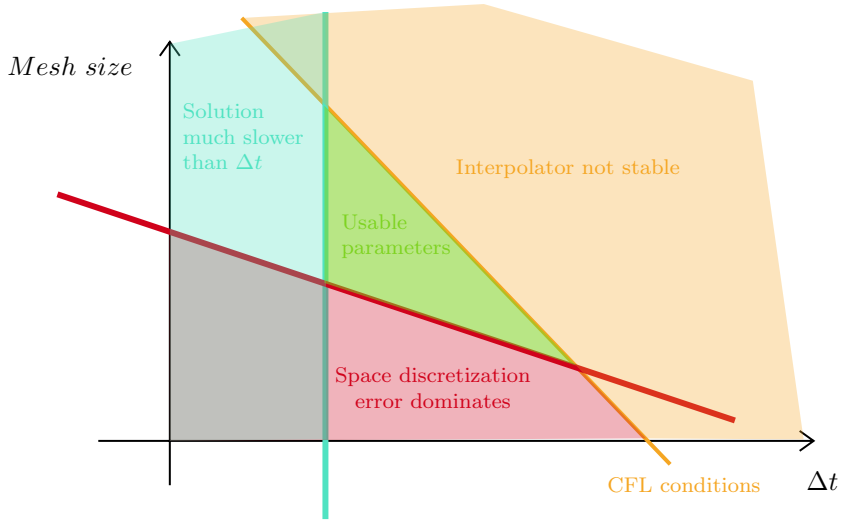


Figure 7: Parameter space for test set

These competing requirements made it difficult to define a regime in which temporal discretization errors dominated and could be measured reliably.

### Convergence (Both Height and Velocity)

The final benchmark of the project focused on testing the convergence of both the water height and velocity fields. Simulations were run using the manufactured solution to provide a reference for error evaluation. Surprisingly, the observed error remained essentially constant across mesh refinements, showing no clear decay. This behaviour is likely linked to the use of the AB2 interpolator for the velocity field: as a conditionally stable explicit method, AB2 can accumulate truncation and phase errors over time, which dominate the total error even when spatial and temporal resolutions are refined. These results highlight the practical impact of error accumulation in multistep explicit schemes and suggest that alternative time integration strategies may be necessary to fully realize the expected convergence rates.

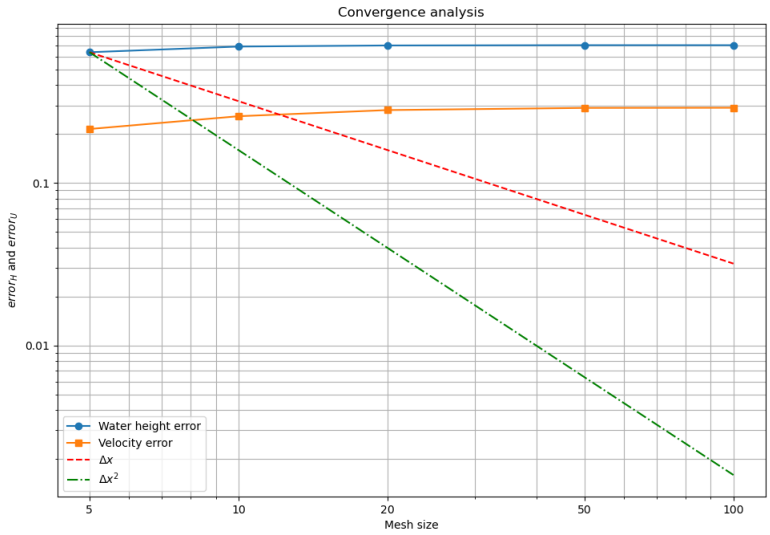


Figure 8: Space convergence of both water height and velocity field

Similar results were studied by [1], where they also obtained a plateau in the error when refining only the mesh size.
A similar scenario was encountered when attempting to evaluate the convergence order in time of the full coupled solver. As in the case of the velocity-only tests, the interaction between stability constraints, initialization errors, and solver tolerances prevented the identification of a testcase in which discretization errors could be cleanly isolated.

# Scalability

## DealII and MPI

The parallelization strategy of the program is based on the native MPI support provided by the deal.II library, which enables scalable distributed-memory execution through high-level abstractions. In particular, the use of **parallel::distributed::Triangulation** allows the mesh to be automatically partitioned across MPI processes, while distributed vectors and matrices from the **TrilinosWrappers::MPI** namespace manage ownership, ghost entries, and communication transparently. Degree-of-freedom distribution, assembly, and synchronization are handled internally by deal.II, significantly reducing the need for explicit MPI calls in the application code.

## Results

The scalability tests were performed using a **CPU with 6 physical cores** on a mesh with **50 triangles per side**, with measured runtime averaged over **20 timesteps** of **1 ms** each.
Although the program demonstrates clear speedup with increasing MPI processes, it is not optimal (i.e. linear). This is partly due to communication overhead between processes and the relatively small problem size used in the tests, which limits the amount of work that can be effectively distributed. Under these conditions, the computation per process is small, so communication and synchronization costs are the most significant portion of the total runtime. For larger meshes the parallel efficiency is expected to improve. Moreover, we tested different preconditioners to see how they would impact performance. Our results are shown in the following plot.

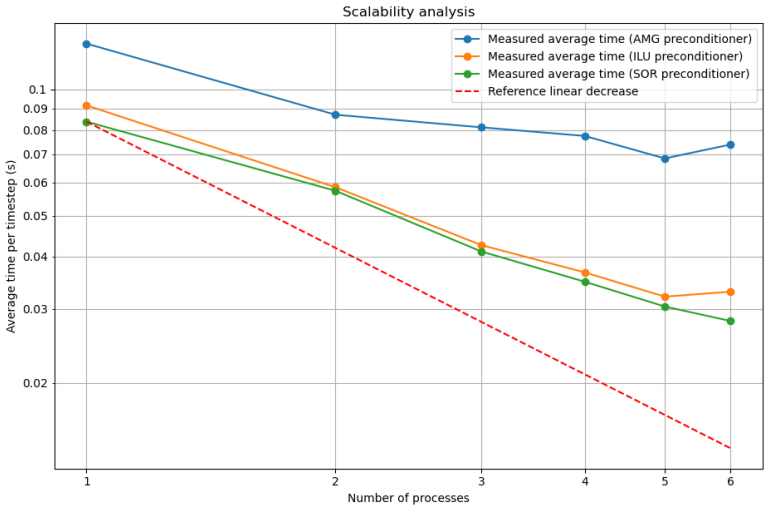


Figure 9: Scalability results

## Conclusions and Possible future works

In this project, we developed a finite element solver for the shallow water equations. Nonlinearities were addressed using an AB2 interpolation scheme, which allowed the system to be split and solved sequentially in a cascade approach. While this strategy proved promising when applied to individual equations, the combined scheme exhibited a noticeable degradation in overall accuracy, highlighting the challenges of coupling explicit and implicit components. Stabilization via SUPG was necessary to control spurious oscillations, particularly in the water height. Scalability tests showed that the implementation performed reasonably well in parallel, although the speedup was not optimal. Future work could explore alternative time integration strategies or fully implicit approaches to improve accuracy and stability. Other promising directions include semi-Lagrangian schemes (see for example [8]), or Discontinuous Galerkin (DG) formulations (see for example [4]), which offer increased flexibility for handling complex flows, sharp gradients, and local adaptivity while maintaining high-order accuracy.

# References


[1] S. Bishnu, M. Peterson, and B. Quaife. On the spatial and temporal order of convergence of hyperbolic pdes. 05 2021.

[2] K. C. B. H. e. a. Carpenter, M.H. Fourth-order runge–kutta schemes for fluid mechanics applications. 2005.

[3] C. Dawson and C. Mirabito. The shallow water equations. `https://users.oden.utexas.edu/~arbogast/cam397/dawson_v2.pdf`, 2008.

[4] C. Dawson and J. Proft. Discontinuous and coupled continuous/discontinuous galerkin methods for the shallow water equations. *Computer Methods in Applied Mechanics and Engineering*, 191(41):4721–4746, 2002.

[5] H. Elman, D. Silvester, and A. Wathen. *Finite Elements and Fast Iterative Solvers: with Applications in Incompressible Fluid Dynamics*. Oxford University Press, 2014.

[6] I. Navon. Finite-element simulation of the shallow-water equations model on a limited-area domain. *Applied Mathematical Modelling*, 1979.

[7] D. Y. L. Roux, A. Staniforth, and C. A. Lin. Finite elements for shallow-water equation ocean models. *Monthly Weather Review*, 126(7):1931 – 1951, 1998.

[8] A. Staniforth and J. Côté. Semi-lagrangian integration schemes for atmospheric models—a review. *Monthly Weather Review*, 119(9):2206 – 2223, 1991.

[9] T. Tezduyar and S. Sathe. Stabilization parameters in supg and pspg formulations. *Journal of Computational and Applied Mechanics*, 4:71–88, 01 2003.

[10] C. Vreugdenhil. *Numerical Methods for Shallow-Water Flow*. Springer Dordrecht, 1994.

[11] H. Zhang, A. Sandu, and S. Blaise. High order implicit-explicit general linear methods with optimized stability regions. *SIAM Journal on Scientific Computing*, 38(3):A1430–A1453, 2016.