Pseudo-Transient Continuation for the Rapid Convergence of Shallow Water Equation Solvers

Internship Report

Jesse Buijs

Student number: 4892151
Project duration: September 2024 – December 2024
Supervisors: S. Westerbeek, Deltares
S. Kenjereš, TU Delft

Contents

Abstract

An implementation of a pseudo-transient continuation method was investigated for an existing shallow-water equations solver. Multiple problem cases - steady-state flow over a sloped bed, sloped wavy bed, and a weir - were tested and produce promising results. The method is capable of stabilizing simulations that would otherwise fail to converge. In addition to the pseudo-time step itself, a pseudo-mass matrix designed to dampen shorter modes was investigated. As shorter modes are associated with higher linearization errors through their higher spatial gradients, dampening them allows for larger pseudo-CFL numbers, resulting in an overall increase in convergence speed for the whole system. A speed-up of around seven times was achieved for the optimal combination of the pseudo-mass matrix and pseudo-time step with respect to an implementation without the pseudo-mass matrix. An initial implementation of a convergence-error dependent pseudo-time step was implemented. The result is an optimally scaled pseudo-time step that stabilizes the system while not inhibiting convergence speed.

1 Description of the company

Deltares is an independent institute for applied research in water and subsurface, offering both consultancy and software products. They specialize in areas like hydraulic engineering, water management, coastal protection, and geotechnical engineering. During this internship, I worked under the hydrodynamics and morphology software group.

Figure 1: Organogram of Deltares

2 Introduction

The shallow water equations are fundamental in modeling fluid dynamics, governing the behavior of flows where the horizontal length scales are much larger than the vertical depth. This makes them ideal for modeling shallow water phenomena like river flow, delta works, and wave propagation. In this project, an existing one-dimensional simulation model is extended with a pseudo-time stepping algorithm to test its increased robustness and possible speed-up capabilities.

In the model, the governing equations are integrated in time using a fully implicit time integration scheme. This has the advantage of improved robustness and the ability to simulate larger computational time steps. It is assumed that the equations can be nonlinear and need to be solved iteratively using Newton's method. This puts a time step restriction on the solution method, as the algorithm becomes unstable for very large solution corrections. To combat this, a pseudo-time stepping approach (also known as pseudo-transient continuation) is applied, which should limit the solution corrections to within a linearization criterion. The effect of this pseudo-time stepping algorithm is investigated, including a pseudo-mass matrix and its possible benefits in residual smoothing.

To decrease situation bias, testing is performed on three different problem cases, described in section 3. The method used to solve the shallow water equations is described in section 4 and the implementation of pseudo-time is described in section 5. The results are presented in section 6, followed by some concluding remarks and recommendations for future research in sections 7.

3 Problem cases

In order to gain insight into the effects of pseudo-transient continuation, it is tested for three problem cases. The first problem case involves steady-state flow over a sloped bed. Its straightforward geometry and known steady-state solution make it ideal for testing specific effects of pseudo-time. The second case introduces a sinusoidal variation to the sloped bed level of the first case. This increases convective effects, which are the main cause of linearization errors and should result in a more difficult problem. The third case involves steady-state flow over a weir. This setup represents a typical problem that fails to converge using Newton's method alone. The large gradients in bed level, and therefore flow velocity and water level, result in a scenario where pseudo-time is necessary to achieve convergence.

The geometry for each of these cases is sampled at equal intervals. Furthermore, regularization is applied to each geometry, smoothing out large changes in bed level, which could otherwise lead to instabilities.

For each case, a gravitational constant of $g = 10.0$ m s^{-2} and a Chézy coefficient (described in the following subsection) of $C_{\text{chezy}} = 50.0$ m$^{1/2}$ s^{-1} are chosen. The imposed (reflective) boundary conditions are a mass flux q at the west boundary and an imposed water surface level ζ at the east boundary.

3.1 Steady-state flow over sloped bed

The most basic test case is steady-state flow over a linearly sloped bed. The bed level decreases by 0.1 meters over a length of 1000 meters, visualized in figure 2. The modelled equations include convection and bed shear stress. This test case is easily solvable with Newton's method and does not require pseudo-time stepping to converge, as long as the initial state is chosen within reasonable bounds. While simple, this test is valuable in the fact that it has a known and simple steady-state solution, which satisfies the Chézy formula

$$u = C_{\text{chezy}} \sqrt{h S_0}, \tag{1}$$

where u is the flow velocity, C_{chezy} is the Chézy coefficient, h is the water column height, and S_0 is the slope of the bed. The Chézy coefficient determines the magnitude of the bed shear stress through the relationship: $c_f = g/C_{\text{chezy}}^2$. For the chosen parameters, the steady-state solution involves a water column height of $h = 4.0\,\text{m}$, and a uniform flow velocity of $u = 1.0\,\text{m}\,\text{s}^{-1}$. The simulation parameters can be found in table 1.

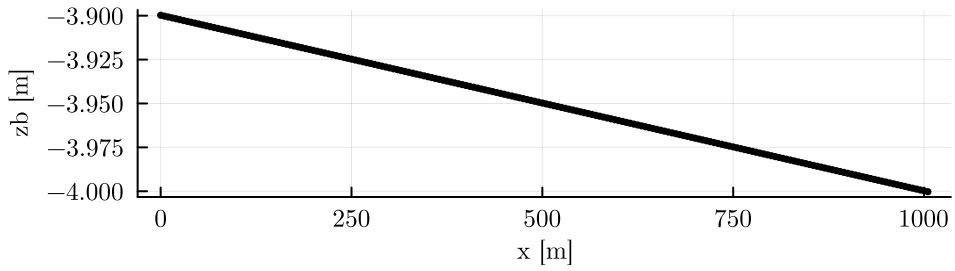

Figure 2: Bed level profile of the sloped bed case.

Table 1: Simulation parameters for the sloped bed case.

Parameter	Value	Unit
L_x	1000.0	m
Δx	5.0 & 2.5	m
depth	4.0	m
q_{west} (reflective)	4.0	$\text{m}^2\ \text{s}^{-1}$
ζ_{east} (reflective)	0.0	m
h_{init}	$A_p \sin(\pi \frac{x}{L_x}) \sin(2\pi N_p \frac{x}{L_x}) + 4.0$	m
A_p	3.0	m
N_p	13.0	-
q_{init}	4.0	$\text{m}\ \text{s}^{-1}$

3.2 Steady-state flow over sloped wavy bed

A more complex case is steady-state flow over a sloped wavy bed. The slope is identical to the first case, but a sinusoidal variation is added to the bed level. The modelled equations include convection and bed shear stress. The simulation parameters can be found in table 2.

Table 2: Simulation parameters for the sloped wavy bed case.

Parameter	Value	Unit
L_x	1000.0	m
Δx	5.0 & 2.5	m
depth	4.0	m
q_{west} (reflective)	4.0	$\text{m}^2\ \text{s}^{-1}$
ζ_{east} (reflective)	0.0	m
h_{init}	$A_p \sin(\pi \frac{x}{L_x}) \sin(2\pi N_p \frac{x}{L_x}) + 4.0$	m
A_p	3.0	m
N_p	13.0	-
q_{init}	4.0	$\text{m}\ \text{s}^{-1}$

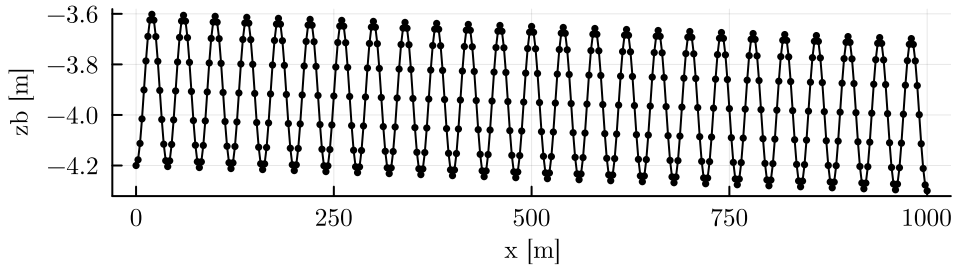

Figure 3: Bed level profile of the sloped wavy bed case.

3.3 Steady-state flow over weir

A case where Newton's method fails without pseudo-time stepping is steady-state flow over a weir. The large change in bed level results in a large change in flow velocity, which introduces a jump in water level. This leads to high nonlinearities, which are very difficult for traditional Newton to solve. The modelled equations include convection, bed shear stress, as well as artificial viscosity. The simulation parameters can be found in table 3.

Table 3: Simulation parameters for the weir case.

Parameter	Value	Unit
L_x	500.0	m
Δx	5.0	m
depth	12.0	m
q_{west} (reflective)	19.8656	$\text{m}^2\ \text{s}^{-1}$
ζ_{east} (reflective)	-3.0	m
ζ_{init}	0.0	m
q_{init}	19.8656	$\text{m}\ \text{s}^{-1}$

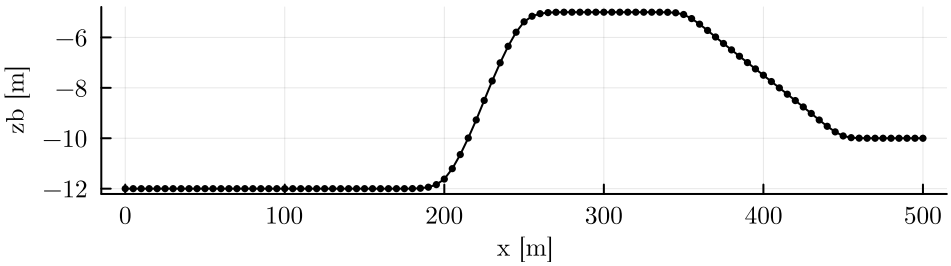

Figure 4: Bed level profile of the Weir case.

4 Methods

4.1 Shallow-water equations

The equations describing shallow water flow in one dimension, as used in the simulation, are shown in equations (2) and (3) [SMART numerics, 2024]. The continuity equation:

$$\frac{\partial h}{\partial t} + \frac{\partial q}{\partial x} = 0, \tag{2}$$

and the momentum equation:

$$\frac{\partial q}{\partial t} + \frac{\partial q^2/h}{\partial x} + gh\frac{\partial \zeta}{\partial x} + c_f \frac{q|q|}{h^2} - \frac{\partial}{\partial x}\left(\nu_{\text{art}} h \frac{\partial (q/h)}{\partial x}\right) = 0, \tag{3}$$

where h denotes the total water depth, q is the unit discharge in the x-direction, $\zeta = h + z_b$ is the free-surface level for bed level z_b, g is the gravitational constant, and $c_f = g\, C_{\text{chezy}}^{-2}$ a friction coefficient. These are derived by integrating the Navier-Stokes equations over the depth, in the case where the horizontal length scale is much larger than the vertical length scale [Vreugdenhil, 1986]. An artificial viscosity term with coefficient ν_{art} is included to handle discontinuities that would otherwise form [Von Neumann and Richtmyer, 1950].

4.2 Discretization

4.2.1 Time integration

Time is discretized into equal time steps. Time integration is implemented using Newton's method to compute the solution at the next time step, given a previous time step. Newton's method is a root-finding algorithm that can be used to solve nonlinear systems of equations. The shallow water equations are first written in the following form:

$$\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot \mathbf{F} - \mathbf{S} = 0 \tag{4}$$

where $\mathbf{U}$ represents the state vector, $\mathbf{F}$ represents the flux terms and $\mathbf{S}$ represents the source terms. This equation has to hold for each volume element. The volume integral is taken in combination with Gauss's

divergence theorem to form the following equation:

$$\int \frac{\partial \mathbf{U}}{\partial t} dV + \oint \mathbf{F} \cdot \mathbf{n} \, dS - \int \mathbf{S} \, dV = 0, \tag{5}$$

which can be condensed to:

$$f(\mathbf{U}) = 0, \tag{6}$$

with f some nonlinear function of $\mathbf{U}$. To apply Newton's method, the equation is linearized around state $\mathbf{U}_0$, giving:

$$f(\mathbf{U}) \approx f(\mathbf{U}_0) + \frac{\partial f(\mathbf{U}_0)}{\partial \mathbf{U}}(\mathbf{U} - \mathbf{U}_0). \tag{7}$$

Knowing that this expression should equal zero, it can be rewritten in the form:

$$\mathbf{U} = \mathbf{U}_0 - f(\mathbf{U}_0) \; / \; \frac{\partial f(\mathbf{U}_0)}{\partial \mathbf{U}}. \tag{8}$$

Using $\Delta\mathbf{U} = \mathbf{U} - \mathbf{U}_0$, the formula is transformed into delta notation,

$$\frac{\partial f(\mathbf{U}_0)}{\partial \mathbf{U}} \Delta\mathbf{U} = -f(\mathbf{U}_0), \tag{9}$$

which is a linear system of the form

$$\mathbf{A}\mathbf{x} = \mathbf{b}. \tag{10}$$

The system is solved for the solution correction $\Delta\mathbf{U}$, which, in turn, is used to update the state vector $\mathbf{U}$. This is done iteratively until the solution correction falls below some desired threshold, indicating that the algorithm has converged.

These steps are done for every subsequent time step and include an implicitness factor, θ, which determines the method's temporal characteristics:

- $\theta = 1$: Fully implicit method
- $\theta = 0$: Fully explicit method
- $\theta = 0.5$: Crank-Nicolson method (semi-implicit, second-order accurate)

The state vector used in the calculations is weighted between the previous time step and the current solution to the next time step

$$\mathbf{U}^{n+\theta,p} = \theta\mathbf{U}^{n+1,p} + [1-\theta]\mathbf{U}^{n}, \tag{11}$$

where n denotes the time step and p denotes the nonlinear (Newton) iteration. At each time step, Newton's method is employed to compute the solution for the next time level. For stationary, steady-state simulations, predominant in this project, the time derivative term is omitted and θ is set to 1, ensuring a fully implicit approach.

4.2.2 Spatial discretization

The applied spatial discretization is based on a collocated, vertex-centered, finite volume method called the Finite Volume Element method. This method is relatively insensitive to grid irregularities, due to the use of polynomial basis functions and a mass matrix for the time derivative [SMART numerics, 2024]. Additionally, orthogonal grids are not a requirement and thus the use of more intricate coordinate systems, like curvilinear coordinates, are made possible, useful for future extensions. In this project, simulations are done in 1D and space is discretized uniformly, i.e. grid cells are equal in size. As one-point Gauss quadrature rules are used to compute integrals for linear terms, the mass matrix for the time derivative is then defined as

$$\mathbf{M}_i = [\alpha, \, 1 - 2\alpha, \, \alpha] \tag{12}$$

with $\alpha = 0.125$ the determined optimal value. At the boundary, a parabolic interpolation is used, resulting in

$$\mathbf{M}_0 = [0.5 + 0.5\alpha_{\text{bnd}}, \, 0.5 - \alpha_{\text{bnd}}, \, 0.5\alpha_{\text{bnd}}] \tag{13}$$

for the west boundary (flipped for the east boundary), with $\alpha_{\text{bnd}} = -0.25$ the determined optimal value for natural boundary conditions [Borsboom, 2023].

5 Implementation

5.1 Pseudo-time stepping

As explained in section 4.2.1, to implement Newton's method, a linearization is made around the solution correction, neglecting terms that are second-order and higher in the solution correction. The purpose of this is to estimate the optimal direction of the solution correction given the solution at the previous iteration level p by solving a linear system of equations. This approximation only holds for sufficiently small solution corrections, which becomes clear when higher-order terms are considered:

$$\begin{aligned} f(\theta\mathbf{U}^{n+1,p+1} + [1-\theta]\mathbf{U}^n) = f(&\theta\mathbf{U}^{n+1,p} + [1-\theta]\mathbf{U}^n) \\ &+ \frac{\partial f(\theta\mathbf{U}^{n+1,p} + [1-\theta]\mathbf{U}^n)}{\partial \mathbf{U}^{n+1,p}}(\mathbf{U}^{n+1,p+1} - \mathbf{U}^{n+1,p}) \\ &+ \frac{1}{2}\frac{\partial^2 f(\theta\mathbf{U}^{n+1,p} + [1-\theta]\mathbf{U}^n)}{(\partial \mathbf{U}^{n+1,p})^2}(\mathbf{U}^{n+1,p+1} - \mathbf{U}^{n+1,p})^2 \\ &+ O([\mathbf{U}^{n+1,p+1} - \mathbf{U}^{n+1,p}]^3). \end{aligned} \tag{14}$$

For the linearization to be reliable, the effect of the higher-order terms must be small compared to the terms that are linear in the solution correction. For this to be the case, the following condition must hold [Borsboom, 2019, p.19]:

$$\left|\frac{1}{2}\frac{\partial^2 f(\theta\mathbf{U}^{n+1,p} + [1-\theta]\mathbf{U}^n)}{(\partial \mathbf{U}^{n+1,p})^2}\Delta\mathbf{U}^{n+1,p+1}\right| < O\left(\left|\frac{\partial f(\theta\mathbf{U}^{n+1,p} + [1-\theta]\mathbf{U}^n)}{\partial \mathbf{U}^{n+1,p}}\right|\right). \tag{15}$$

The idea behind pseudo-time stepping is to introduce a pseudo-time derivative term to the existing system of equations, to scale down the solution correction and satisfy the condition (15). Applying this, (6) becomes:

$$f(\mathbf{U}) = -\frac{\partial \mathbf{U}}{\partial t_{\text{pseu}}}, \tag{16}$$

turning it into a pseudo-time dependent equation. Following the same procedure implementing Newton as before results in

$$\mathbf{U} = \mathbf{U}_0 - \left(\mathbf{M}_{\text{pseu}}\mathbf{T}_{\text{pseu}} + \frac{\partial f(\mathbf{U}_0)}{\partial \mathbf{U}}\right)^{-1} f(\mathbf{U}_0), \tag{17}$$

and therefore,

$$\left(\mathbf{M}_{\text{pseu}}\mathbf{T}_{\text{pseu}} + \frac{\partial f(\mathbf{U}_0)}{\partial \mathbf{U}}\right)\Delta\mathbf{U} = -f(\mathbf{U}_0), \tag{18}$$

where $\mathbf{T}_{\text{pseu}}$ is a matrix with the reciprocal of the local pseudo-time step size Δt_{pseu} on the diagonal and $\mathbf{M}_{\text{pseu}}$ is the pseudo-mass matrix applied to the pseudo-time step. This is different from the normal mass matrix and is explained in the next section.

Adding the pseudo-time step enhances the diagonal dominance of matrix $\mathbf{A}$ from (10), improving robustness. Larger values on the diagonal of $\mathbf{A}$ leads to smaller solution corrections and thus smaller linearization errors. While counterintuitive, decreasing the size of the solution correction can lead to massive speedups, as overshooting is prevented.

5.1.1 Obtaining the pseudo-mass matrix

The linearization errors for the low-frequency modes are relatively small compared to those for the high-frequency modes. An implementation without the pseudo-mass matrix dampens all modes equally, even though the low-frequency modes do not necessarily need the same amount of dampening as the high-frequency modes, thus unnecessarily slowing down convergence. In contrast to the normal mass matrix for the physical time derivative defined in equation 12, the pseudo-mass matrix would be structured to effectively add a curvature dependence to the pseudo-time step:

$$\mathbf{M}_{\text{pseu},i} = [-\alpha_{\text{pseu}},\ 1 + 2\alpha_{\text{pseu}},\ -\alpha_{\text{pseu}}], \tag{19}$$

with $\alpha_{\text{pseu}} > 0$ a parameter to scale this effect. At the boundaries, an upwind pseudo-mass matrix is applied. For the west boundary:

$$\mathbf{M}_{\text{pseu},0} = [0.5 + 0.5\alpha_{\text{pseu,bnd}},\ 0.5 - 0.5\alpha_{\text{pseu,bnd}},\ 0], \tag{20}$$

flipped for the east boundary. For $\alpha_{\text{pseu,bnd}} = 1.0$, this corresponds to a fully upwind characteristic.

To gain insight into the effects on the convergence speed of different modes, a Fourier-mode analysis of this implementation is applied in appendix A. This analysis shows that shorter modes are dampened more strongly than longer modes. This would allow for a larger pseudo CFL number and consequently higher overall convergence speeds.

Since the Fourier-mode analysis does not incorporate nonlinear terms, the results do not take linearization errors into account. Empirical testing is therefore necessary in order to obtain realistic results and ideally find an optimal value for α_{pseu}. To this end, the sloped bed case is simulated with an initial condition equal to the steady-state result plus a perturbation in the water column height. The perturbation is chosen large enough to break traditional Newton and necessitate pseudo-time. The simulation is then run for different combinations of α_{pseu} and constant CFL_{pseu}. To compute the approximate convergence speed, the number of iterations needed to decrease $\log 10(\Delta h_{\max})$ from -3 to the convergence threshold is measured. The upper bound of -3 is chosen, as there is a region of rapid convergence in the first few iterations that could taint the results.

5.1.2 Computing the pseudo time step

The optimal magnitude of the pseudo-time step depends on the problem. It should be small enough so that convergence is guaranteed, but not too small to inhibit convergence speed. There are a number of options available, the simplest being a constant value. While this will help stabilize the system, it bottlenecks the convergence speed, especially when close to the solution. Ideally, the pseudo-time step size should increase as the solution converges, to prevent its effect from arriving at the solution to the original equations presented in (2) and (3). A technique that is often applied is the Switched Evolution Relaxation (SER) method [Mulder and Leer, 1985]. Here, the pseudo-time step size is updated based on the relative size of the residuals between iterations,

$$\Delta t_{\text{pseu}}^{p+1} = \Delta t_{\text{pseu}}^{p} \frac{||f(\mathbf{U}^{p})||}{||f(\mathbf{U}^{p+1})||}. \tag{21}$$

A disadvantage of this method is that it is applied ad hoc and globally. The former means that the magnitude is scaled between iterations, but is not optimized for the specific system it is applied to. The latter means that if a small pseudo-time step is required locally, it is applied everywhere, unnecessarily slowing down the entire iteration process [Borsboom, 2019].

Another technique that aims to solve this problem is to make the pseudo-time step size locally convergence-error dependent, as contrived by M. Borsboom for the shallow water equations [Borsboom, 1999]:

$$\frac{\Delta x}{\Delta t_{\text{pseu}}} = \max\left(0, \epsilon_{\text{pseu}}(|\Delta u| + |\Delta c|) - \frac{\Delta x}{\Delta t}\right), \tag{22}$$

where,

$$\begin{aligned} u &= \frac{q}{h}, & \Delta u &= \theta\left(\frac{1}{h}\Delta q - \frac{q}{h^2}\Delta h\right), \\ c &= \sqrt{gh}, & \Delta c &= \theta\frac{g}{2c}\Delta h, \end{aligned} \tag{23}$$

and parameter ϵ_{pseu} a scaling parameter of order $O(1)$. Larger values will result in smaller pseudo-time steps and consequently more robust but slower convergence. For difficult problems, it may be necessary to increase this value.

This is the close-to-ideal pseudo-time step to limit the solution correction just enough to be stable without sacrificing the convergence speed. However, to calculate Δu and Δc, the solution corrections Δh and Δq are needed, which are yet to be calculated. One way of dealing with this problem is to take the solution corrections from the previous iteration, and to under-relax the pseudo-time step to prevent possible instabilities. Since the solution correction for the first iteration does not yet exist, the initial pseudo-time step is conservatively set based on a local pseudo CFL number of around 2 [Borsboom, 2019, p.20]. Additionally, computing the sum of the magnitudes of the terms in Δu instead of $|\Delta u|$ itself may yield increased robustness (albeit at the expense of convergence speed). This implementation is the one applied in SOBEK-RE [Borsboom, 1999, p.6]. The derivation of equation 22 can be found in appendix B.

The typical behaviour of the resulting solution algorithm begins with a relatively small pseudo-time step. In this initial, so-called searching phase, the algorithm is looking for an approximate solution.

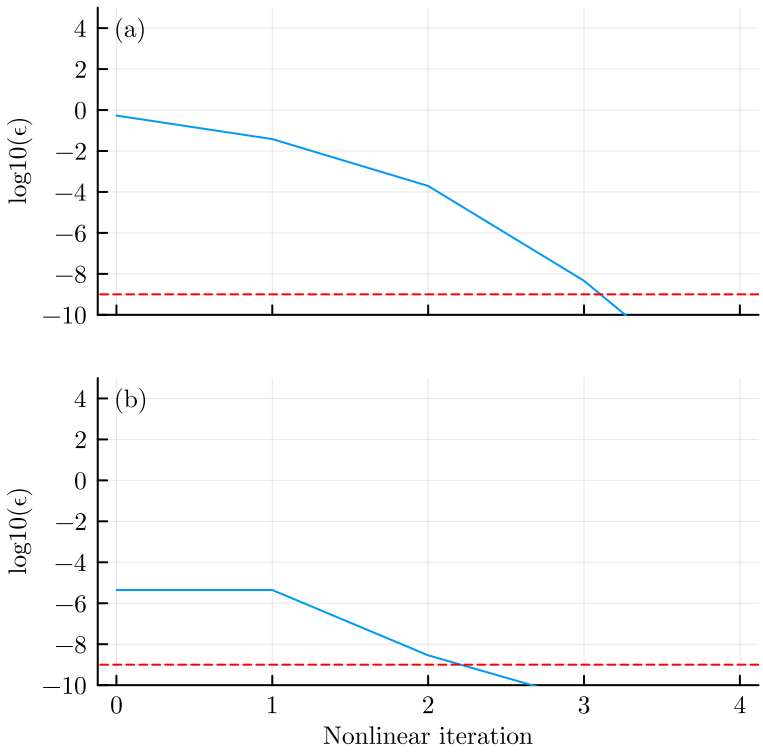

Figure 5: Solution corrections (blue) for $\epsilon = \Delta h_{\max}$ (a) and $\epsilon = \Delta q_{\max}$ (b) per nonlinear iteration for the sloped bed case with a small perturbation. The convergence criterion is depicted by the red dashed line.

Once this point is reached, the convergence errors become significantly smaller and the effect of the pseudo-time step rapidly decreases. This is called the converging phase, in which the algorithm behaves more like Newton's method, with rapid (close to quadratic) convergence.

6 Results

As an example, in figure 5 the maximum solution corrections per nonlinear iteration are plotted for the sloped bed case with a small initial perturbation of 0.1 meters in the water column height. This type of plot is referred to as a convergence plot. The convergence threshold is indicated by the horizontal red dashed line. This case is easily solvable and Newton's rapid convergence is demonstrated. If the stationary problem is simple enough for standard Newton to solve, then any test with pseudo-time will result in slower convergence. Therefore, for the first two problem cases, the initial perturbation is chosen such that the algorithm is not able to achieve convergence without the use of pseudo-time, with the intention of testing its effectiveness.

6.1 Constant pseudo-time step

When the perturbation is large, Newton breaks down, producing nonphysical results and failing to converge to a solution. This is shown in figure 6 for the sloped bed case. Introducing a constant pseudo-time step resolves this issue by allowing the algorithm to more slowly approach the correct solution, as shown in figure 7 for a constant pseudo-CFL number of 0.5.

For subsequent analyses, the convergence speed of the algorithm is a good metric for its performance. For state variable h this is defined as the decrease in the solution error between iterations

$$\gamma_h = -\log_{10} \left| \frac{\delta h^{p+1}}{\delta h^p} \right|, \tag{24}$$

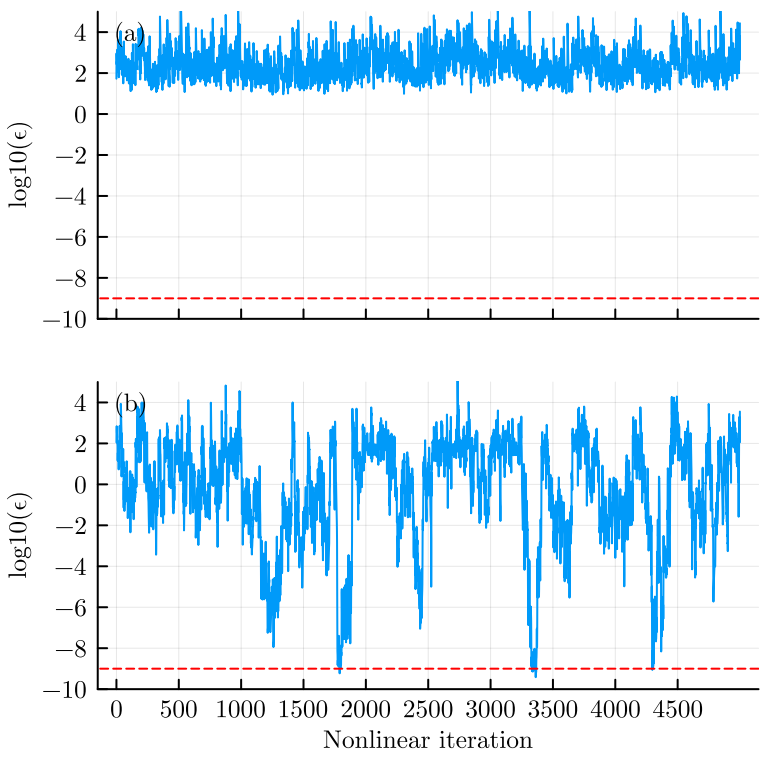

Figure 6: Solution corrections (blue) for $\epsilon = \Delta h_{\max}$ (a) and $\epsilon = \Delta q_{\max}$ (b) per nonlinear iteration for the sloped bed case without pseudo-time. The convergence criterion is depicted by the red dashed line.

where $\delta h^p = h^p - h^*$ with h^* the ideal solution. This can be measured empirically by

$$\gamma_h = -\log_{10} \left| \frac{\Delta h^{p+1}}{\Delta h^p} \right|. \tag{25}$$

That is, from the simulations, the convergence speed is empirically determined by computing the slope of the convergence plot. The same applies for q. By considering only the number of iterations needed for the maximum solution correction to drop from 10^{-3} to the desired convergence threshold, the section of rapid convergence in the first few iterations is neglected.

6.2 Pseudo-mass matrix

To analyze the effect of the pseudo-mass matrix, simulations are run with different combinations of α_{pseu} and CFL_{pseu}. The convergence speed for each simulation is then determined as defined in equation 25. The convergence speeds for the sloped bed case are shown in figure 8. Additionally, the same analysis is performed for a discretization that is twice as fine, whose results are shown in figure 9. A gray color corresponds to the algorithm exceeding the maximum allowed number of iterations (5000), but likely still converging if given enough time. A white color corresponds to the algorithm not achieving convergence.

From these figures, it becomes evident that the pseudo-mass matrix allows for higher pseudo CFL numbers that result in higher convergence speeds. Moreover, there is an optimal combination of these parameters to be found around $\alpha_{\text{pseu}} \approx 2$ and $\text{CFL}_{\text{pseu}} \approx 10$. The exact optimum differs between problem cases. Applying the same analysis for the sloped wavy bed case produces the results shown in figure 10 and again for a more finely discretized case in figure 11.

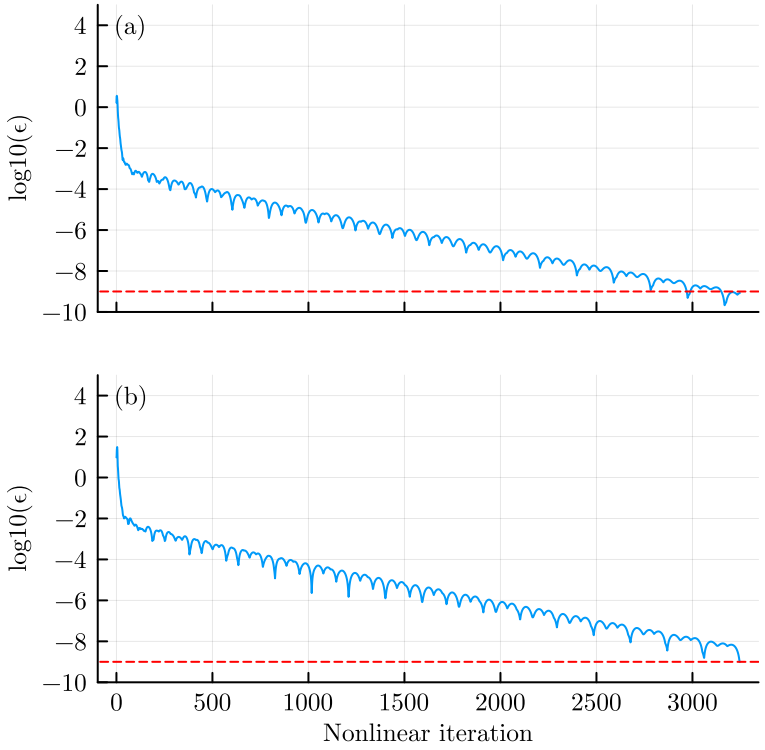

Figure 7: Solution corrections (blue) for $\epsilon = \Delta h_{\max}$ (a) and $\epsilon = \Delta q_{\max}$ (b) per nonlinear iteration for the sloped bed case with a constant pseudo-CFL number of 0.5. The convergence criterion is depicted by the red dashed line.

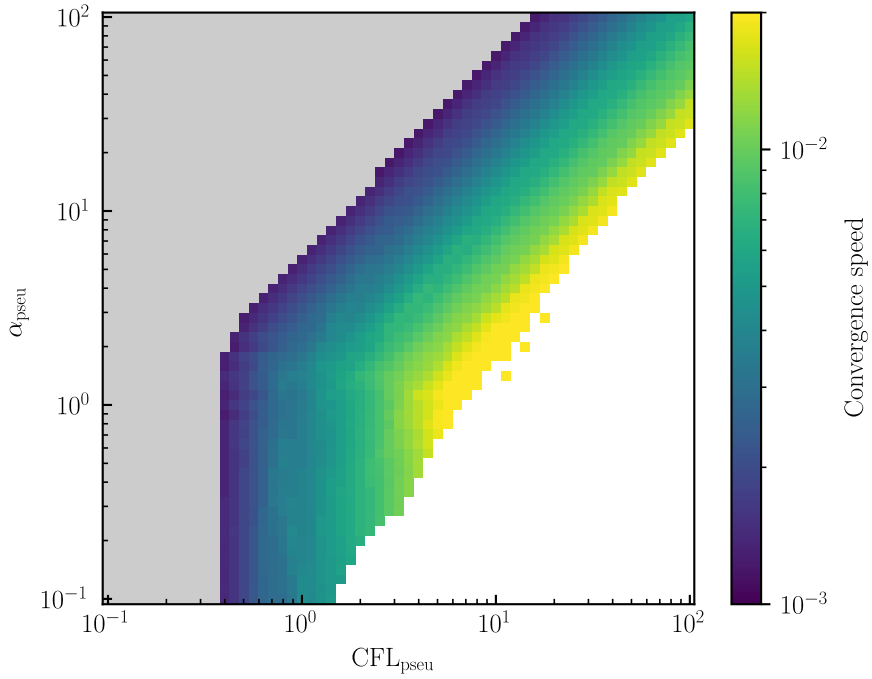

Figure 8: Convergence speed for the simulation of the sloped bed with initial perturbation in h according to table 1 for different values of α_{pseu} and CFL_{pseu}. ($\Delta x = 5.0$)

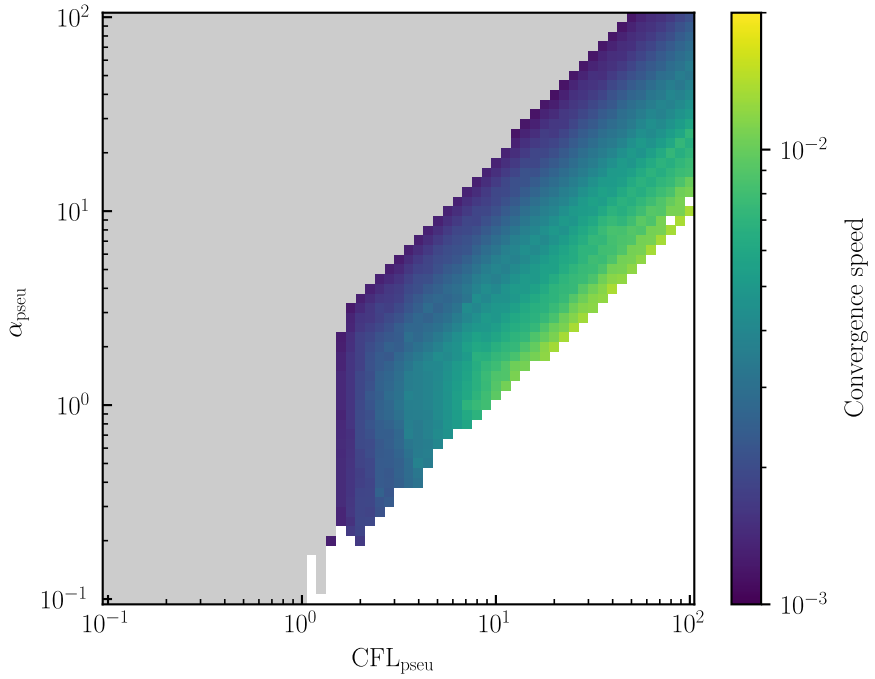

Figure 9: Convergence speed for the simulation of the sloped bed with initial perturbation in h according to table 1 for different values of α_{pseu} and CFL_{pseu}. ($\Delta x = 2.5$)

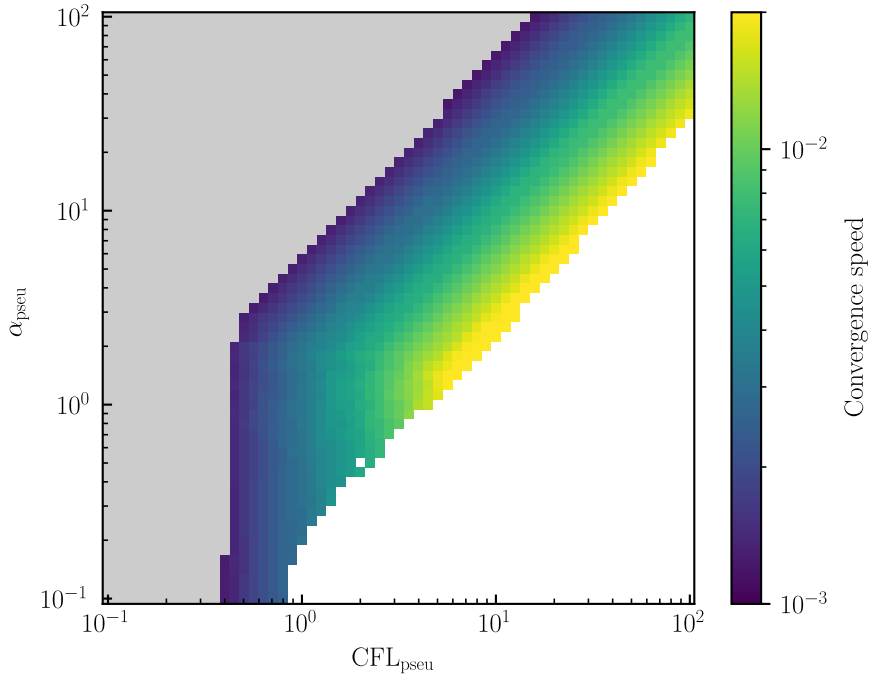

Figure 10: Convergence speed for the simulation of the sloped wavy bed with initial perturbation in h according to table 1 for different values of α_{pseu} and CFL_{pseu}. ($\Delta x = 5.0$)

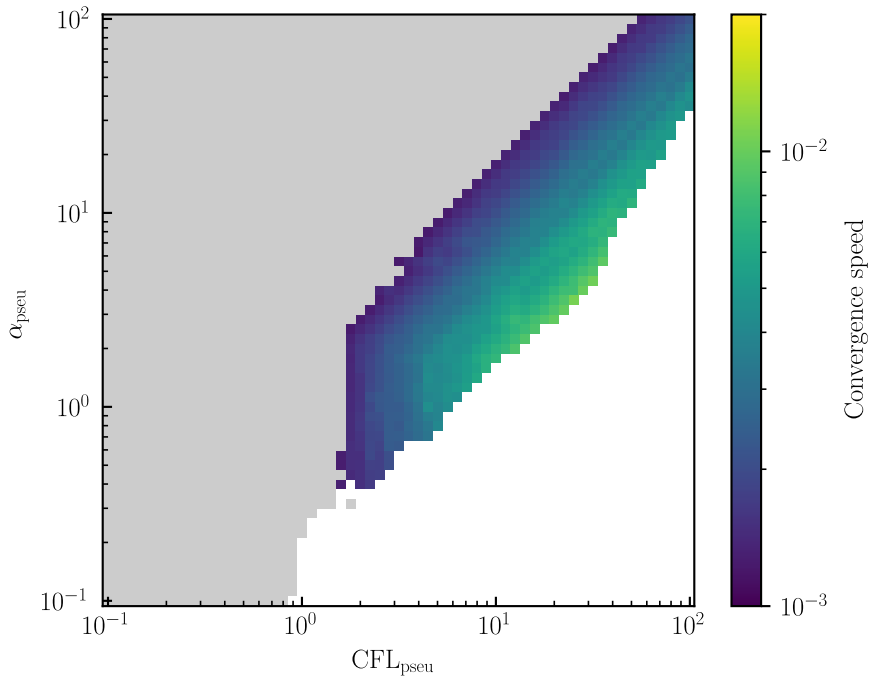

Figure 11: Convergence speed for the simulation of the sloped wavy bed with initial perturbation in h according to table 1 for different values of α_{pseu} and CFL_{pseu}. ($\Delta x = 2.5$)

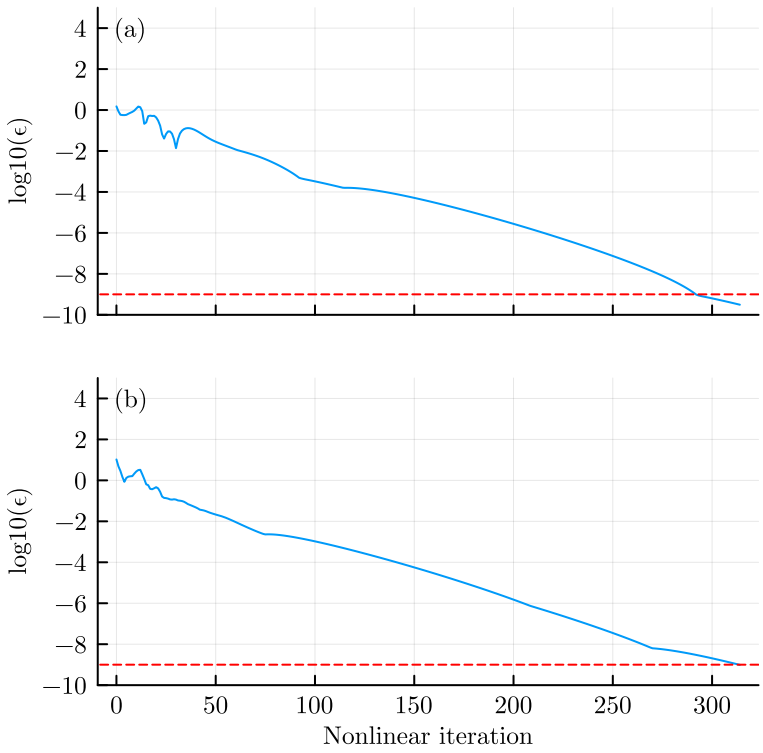

Figure 12: Solution corrections (blue) for $\epsilon = \Delta h_{\max}$ (a) and $\epsilon = \Delta q_{\max}$ (b) per nonlinear iteration for the weir case using a constant pseudo CFL number of 2.0. The convergence criterion is depicted by the red dashed line.

6.3 Convergence-error dependent pseudo-time step

In and of itself, a constant pseudo-time step can already lead to massive speed-ups. For example, the weir case is not solvable as a stationary simulation without pseudo-time and needs to be computed as a time-dependent system with absorbing boundary conditions. This necessitates many smaller time steps, which takes a significant amount of time to simulate. With pseudo-time, the simulation is solvable in a stationary way in a few hundred iterations (see figure 12) and the runtime is decreased from minutes to seconds. The initial implementation of the convergence-error dependent pseudo-time step enhances this efficiency even further, taking just tens of iterations (see figure 13).

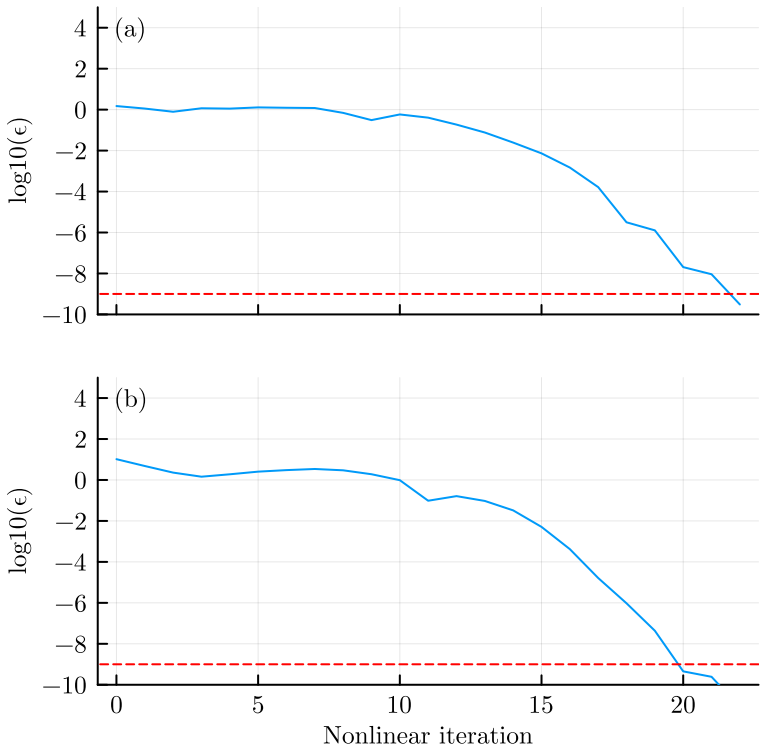

Figure 13: Solution corrections (blue) for $\epsilon = \Delta h_{\max}$ (a) and $\epsilon = \Delta h_{\max}$ (b) per nonlinear iteration for the weir case using a convergence-error dependent pseudo time step with $\epsilon_{\text{pseu}} = 2.0$. The convergence criterion is depicted by the red dashed line.

7 Concluding remarks and recommendations for future research

The implementation of pseudo-transient continuation in this framework has shown promising results. The algorithm is capable of stabilizing simulations that would otherwise fail to converge. A pseudo-mass matrix designed to dampen shorter modes was investigated and appears to produce favorable results, with multiple simulations showing a speed-up of around seven times. It allows for higher pseudo-CFL numbers to increase convergence speed of the whole system, without being bottlenecked by shorter modes that result in high linearization errors. Further investigation is needed to refine and expand on this method, in order to make it applicable to a broad range of problem cases.

While the current implementation is done for a one-dimensional problem, the extension to two dimensions is a logical next step. This would require inserting the pseudo-time into the linear problem in a more intricate way, and using a compatible two-dimensional pseudo-mass matrix.

The handling of the pseudo-mass matrix at the boundaries can be improved. Similar to the normal mass matrix, suitable values may allow for the dampening of spurious modes that may enter the domain.

Finally, an initial application of a convergence-error dependent pseudo-time step, as contrived by M. Borsboom [Borsboom, 1999], was implemented. The result is an optimally scaled pseudo-time step that stabilizes the system while not inhibiting convergence speed. A possible improvement, that was proposed for SUPSUB, is to base the pseudo-time step on both the residual and the maximum allowed solution correction. This technique should yield more rapid convergence, as it does not rely on the solution correction of the previous iteration, but rather only information of the current iteration. This needs to be investigated further, as it has not yet been extensively tested. Additionally, including a mode-dampening pseudo-mass matrix in combination with a convergence-error dependent pseudo-time step needs to be researched. The effect of the pseudo-mass matrix may, for example, allow for a less constricting choice of ϵ_{pseu}. This could be investigated by generating heatmaps similar to figure 8 but with ϵ_{pseu} on the x-axis, as opposed to CFL_{pseu}.

8 Skills learned

Before starting this internship, I was very much looking forward to working in a team within a larger company. This proved to be very enjoyable, as well as being a great learning experience. During the internship, I was able to discuss daily problems of my own as well as problems that team members might have been facing. An outside perspective is often essential for finding solutions that one might not have thought of themselves. As the team was quite small, communication between team members was natural and uninhibited. Additionally, I learned the importance of interdisciplinary communication. Discussing the problems I ran into with colleagues gave insight into alternative perspectives and approaches that I hadn't considered. This process not only improved my understanding of the challenges at hand, but also advanced my problem-solving skills in general.

The second learning objective was to enhance my time management skills. At the start of the internship, a general plan was made for the trajectory of the project. This helped shape the distribution of tasks that needed to be done throughout the project. Despite some uncertainty within the team about which exact specifications to focus on within the limited time of the project, adapting to the evolving plan while staying aligned with the overarching goals worked out successfully.

Moving to more subject-related skills, I learned about the existence and importance of the shallow water equations and numerous techniques for solving them. These techniques are general and can be applied to many fields of physics. A handy tool to have for future projects. Furthermore, programming in Julia was something that had piqued my interests before, and this internship allowed me to dive into its intricacies and computational advantages.

References

Borsboom, M. (1999). SOBEK, betere schaling van cflipse. Memo.

Borsboom, M. (2019). Numerical design of a fast, robust and accurate 1d shallow-water solver for pipe flows with large time scales – proposal (work in progress). Memo.

Borsboom, M. (2020). SUPSUB pseudo-time stepping. Memo.

Borsboom, M. (2023). Numerical implementation of (absorbing) boundary conditions in 1d (and in 2d). Internal document.

Mulder, W. and Leer, B. V. (1985). Experiments with implicit upwind methods for the euler equations. *Journal of Computational Physics*.

SMART numerics (2024). An introduction to SMART numerics. Internal document.

Von Neumann, J. and Richtmyer, R. D. (1950). A method for the numerical calculation of hydrodynamic shocks. *Journal of Applied Physics*, 21(3):232–237.

Vreugdenhil, C. B. (1986). *Numerical Methods for Shallow-Water Flow*. Springer Dordrecht.

A Fourier-mode analysis for the pseudo-mass matrix

A Fourier-mode analysis is performed on the shallow water equations in order to determine the convergence speed for different Fourier modes. To achieve this, the ratio of the solution error between two iterations $g_{k,h} = \frac{\delta h_k^{p+1}}{\delta h_k^p}$ and $g_{k,q} = \frac{\delta q_k^{p+1}}{\delta q_k^p}$, needs to be found. Here, $\delta h^p = h^p - h^*$ is the difference between the solution at iteration p and the ideal solution h^* (likewise for q).

To start, the shallow water equations are rewritten for clarity:

$$\frac{\partial h}{\partial t} + \frac{\partial q}{\partial x} = 0, \tag{26}$$

$$\frac{\partial q}{\partial t} + \frac{\partial q^2/h}{\partial x} + gh\frac{\partial \zeta}{\partial x} + c_f\frac{q|q|}{h^2} - \frac{\partial}{\partial x}\left(\nu h\frac{\partial (q/h)}{\partial x}\right) = 0, \tag{27}$$

and linearized in the same way as in the solution algorithm:

$$\frac{\partial \Delta h}{\partial t} + \frac{\partial \Delta q}{\partial x} = -\frac{\partial q}{\partial x}, \tag{28}$$

$$\begin{aligned}
&\frac{\partial \Delta q}{\partial t} + \frac{\partial}{\partial x}\left(\frac{2q}{h}\Delta q - \frac{q^2}{h^2}\Delta h\right) \\
&\quad + g\left(h\frac{\partial \Delta h}{\partial x} + \Delta h\frac{\partial \zeta}{\partial x}\right) \\
&\quad + c_f\left(\frac{2|q|}{h^2}\Delta q - \frac{2q|q|}{h^3}\Delta h\right) \\
&\quad - \nu\frac{\partial}{\partial x}\left(\frac{\partial \Delta q}{\partial x} - (\frac{1}{h}\Delta q - \frac{q}{h^2}\Delta h)\frac{\partial h}{\partial x} - \frac{q}{h}\frac{\partial \Delta h}{\partial x}\right) \\
&\quad = -\frac{\partial q^2/h}{\partial x} - gh\frac{\partial \zeta}{\partial x} - c_f\frac{q|q|}{h^2} + \nu\frac{\partial}{\partial x}\left(h\frac{\partial (q/h)}{\partial x}\right).
\end{aligned} \tag{29}$$

The solution correction Δh and Δq are at iteration $p+1$ while all other terms are at iteration p. Additionally, the artificial viscosity coefficient ν is chosen constant for simplicity. For this analysis, a steady-state simulation is assumed and the time derivative term is dropped. Instead, the pseudo-time term is added, containing the pseudo mass matrix to be analyzed. Next, the state variables are assumed to be equal to the steady-state solution plus some perturbation $h^p = h^* + \delta h^p$. Filling this in for the continuity equation leads to,

$$\mathbf{M}_{\text{pseu}}\frac{\delta h^{p+1} - \delta h^p}{\Delta t_{\text{pseu}}} + \frac{\partial}{\partial x}(\delta q^{p+1} - \delta q^p) = -\frac{\partial}{\partial x}(q^* + \delta q^p), \tag{30}$$

simplified to,

$$\mathbf{M}_{\text{pseu}}\frac{\delta h^{p+1} - \delta h^p}{\Delta t_{\text{pseu}}} + \frac{\partial}{\partial x}\delta q^{p+1} = -\frac{\partial}{\partial x}q^* = 0. \tag{31}$$

Applying the same steps to the momentum equation is less straightforward. A number of approximations are made. The perturbation is assumed small, and its quadratic terms are therefore dropped. Furthermore, terms with h in the denominator are expanded using binomial expansion,

$$(h^* + \delta h^p)^n = h^{*n}\left(1 + \frac{\delta h^p}{h^*}\right)^n = h^{*n}\left(1 + n\frac{\delta h^p}{h^*} + \frac{n(n-1)}{2!}\frac{\delta h^{p2}}{h^{*2}} + \cdots\right), \tag{32}$$

and the quadratic (and higher-order) terms are dropped as well. The result for the momentum equation

is,

$$
\begin{aligned}
&\mathbf{M}_{\text{pseu}} \frac{\delta q^{p+1} - \delta q^{p}}{\Delta t_{\text{pseu}}} \\
&+ \frac{\partial}{\partial x} \left(\frac{2q^*}{h^*} \delta q^{p+1} - \frac{{q^*}^2}{{h^*}^2} \delta h^{p+1} \right) \\
&+ g \left(h^* \frac{\partial \delta h^{p+1}}{\partial x} + \delta h^{p+1} \frac{\partial \zeta^*}{\partial x} \right) \\
&+ c_f \left(\frac{2|q^*|}{{h^*}^2} \delta q^{p+1} - \frac{2q^*|q^*|}{{h^*}^3} \delta h^{p+1} \right) \\
&- \nu \frac{\partial}{\partial x} \left(\frac{\partial \delta q^{p+1}}{\partial x} - \left(\frac{1}{h^*} \delta q^{p+1} - \frac{q^*}{{h^*}^2} \delta h^{p+1} \right) \frac{\partial h^*}{\partial x} - \frac{q^*}{h^*} \frac{\partial \delta h^{p+1}}{\partial x} \right) \\
&= 0.
\end{aligned}
\tag{33}
$$

Moving the terms with δh^p and δq^p to the right-hand side for both the continuity equation and the momentum equation gives the following system of equations:

$$
\mathbf{G}_{\text{impl}} \delta \mathbf{U}^{p+1} = \mathbf{G}_{\text{expl}} \delta \mathbf{U}^{p}, \tag{34}
$$

with,

$$
\mathbf{G}_{\text{impl}} = \begin{pmatrix} \frac{\mathbf{M}_{\text{pseu}}}{\Delta t_{\text{pseu}}} & \frac{\partial}{\partial x} \\ -\frac{\partial}{\partial x} \frac{{q^*}^2}{{h^*}^2} + g[\frac{\partial \zeta^*}{\partial x}] + g h^* \frac{\partial \cdot}{\partial x} & \frac{\partial}{\partial x} \frac{2q^*}{h^*} + c_f \frac{2q^*}{{h^*}^2} \\ -c_f \frac{2{q^*}^2}{{h^*}^3} - \nu \frac{\partial}{\partial x} (\frac{q^*}{{h^*}^2} [\frac{\partial h^*}{\partial x}] - \frac{q^*}{h^*} \frac{\partial}{\partial x}) & -\nu \frac{\partial}{\partial x} (\frac{\partial \cdot}{\partial x} - \frac{1}{h^*} [\frac{\partial h^*}{\partial x}]) + \frac{\mathbf{M}_{\text{pseu}}}{\Delta t_{\text{pseu}}} \end{pmatrix}, \tag{35}
$$

$$
\mathbf{G}_{\text{expl}} = \begin{pmatrix} \frac{\mathbf{M}_{\text{pseu}}}{\Delta t_{\text{pseu}}} & 0 \\ 0 & \frac{\mathbf{M}_{\text{pseu}}}{\Delta t_{\text{pseu}}} \end{pmatrix}, \tag{36}
$$

and,

$$
\delta \mathbf{U}^{p} = \begin{pmatrix} \delta h^{p} \\ \delta q^{p} \end{pmatrix}. \tag{37}
$$

The terms within the square brackets are evaluated, i.e. the operators within the square brackets do not operate on the perturbations $\delta \mathbf{U}$. To find the Fourier-mode dependence, the Fourier transform over space is applied to this system. Considering Fourier-modes

$$
\delta h^{p} = \sum_{k} \delta h_k^{p} e^{jkx}, \tag{38}
$$

the discretized operators $\mathbf{M}_{\text{pseu}}$, $\frac{\partial}{\partial x}$, and $\frac{\partial^2}{\partial x^2}$ become

$$
\mathbf{M}_{\text{pseu}} \rightarrow 1 + 2\alpha_{\text{pseu}} - 2\alpha_{\text{pseu}} \cos(k\Delta x), \tag{39}
$$

$$
\frac{\partial}{\partial x} \rightarrow \frac{j \sin(k\Delta x)}{\Delta x}, \tag{40}
$$

$$
\frac{\partial^2}{\partial x^2} \rightarrow \frac{-2 + 2\cos(k\Delta x)}{\Delta x^2}, \tag{41}
$$

in k-space with the imaginary number $j = \sqrt{-1}$. To find the amplification factors, the eigenvalues g_k of $\mathbf{G}_k = \mathbf{G}_{\text{impl},k}^{-1} \mathbf{G}_{\text{expl},k}$ are calculated. The convergence speed is then obtained through

$$
\gamma_k = -\log_{10} |g_k| . \tag{42}
$$

The calculated analytical convergence speeds for different values of α_{pseu} and multiple Fourier-modes for the sloped bed case are shown in figure 14. Only the convergence speed for h is shown, as the values for q are almost identical. Consistent with the theory, increasing α_{pseu} reduces the convergence speed, with a more pronounced effect on shorter modes.

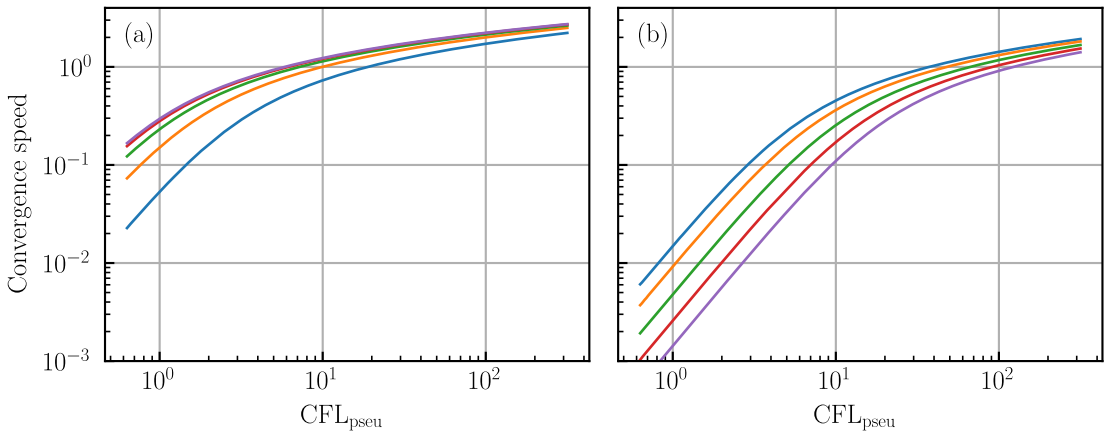

Figure 14: Analytical convergence speed for Fourier-modes $k\Delta x = 0.1\pi$ (blue), 0.2π (orange), 0.3π (green), 0.4π (red), 0.5π (purple) for $\alpha_{\mathrm{pseu}} = 0$ (a) and 10 (b).

B Derivation of the convergence-error dependent pseudo-time step

To start, the system of equations (2) and (3) is discretized in time and linearized into the following form:

$$\frac{\Delta h}{\Delta t_{\mathrm{pseu}}} + \frac{\Delta h}{\Delta t} + \theta \frac{\partial \Delta q}{\partial x} = \mathrm{res}_{\mathrm{cont}}^{n+1,p}, \tag{43}$$

$$\begin{aligned} \frac{\Delta q}{\Delta t_{\mathrm{pseu}}} + \frac{\Delta q}{\Delta t} &+ \theta \frac{\partial}{\partial x}\left(\frac{2q}{h}\Delta q - \frac{q^2}{h^2}\Delta h\right) \\ &+ \theta g\left(\Delta h \frac{\partial \zeta}{\partial x} + h\frac{\partial \Delta h}{\partial x}\right) \\ &+ \theta c_f\left(\frac{2|q|}{h^2}\Delta q - \frac{2q|q|}{h^3}\Delta h\right) \\ &- \nu\theta \frac{\partial}{\partial x}\left(\frac{\partial \Delta q}{\partial x} - \left[\frac{\Delta q}{h} - \frac{q\Delta h}{h^2}\right]\frac{\partial h}{\partial x} - \frac{q}{h}\frac{\partial \Delta h}{\partial x}\right) \\ &= \mathrm{res}_{\mathrm{mom}}^{n+1,p}, \end{aligned} \tag{44}$$

with $\mathrm{res}_{\mathrm{cont}}^{n+1,p}$ and $\mathrm{res}_{\mathrm{mom}}^{n+1,p}$ the residuals of the continuity equation and momentum equation respectively. This is the system that is solved iteratively using the Newton method. Note that the indices n and p have been omitted for readability. The solution correction terms Δh and Δq are at index $n+1, p+1$ while all other terms are at index $n+\theta, p$.

Before continuing the derivation of the convergence-error dependent pseudo-time step, the following assumptions are made [Borsboom, 2020, p.5]:

- Newton's method should converge when all the terms in the governing equations are smooth in space. As this should be the case for the solution h, q, the convergence problems can only occur when the solution correction Δh, Δq is not smooth in space.
- The bottom friction term is free of derivatives in the solution correction Δh, Δq and tends to dampen perturbations. It is therefore not a primary cause for convergence problems due to linearization errors.
- The artificial viscosity term is primarily a smoothing term for Δh, Δq, and therefore does not cause convergence problems due to linearization errors.

Taking these assumptions into account, the bottom friction and artificial viscosity terms are dropped, and the following system is obtained:

$$\frac{\Delta h}{\Delta t_{\text{pseu}}} + \frac{\Delta h}{\Delta t} + \theta \frac{\partial \Delta q}{\partial x} = \text{res}_{\text{cont}}^{n+1,p}, \tag{45}$$

$$\begin{aligned} \frac{\Delta q}{\Delta t_{\text{pseu}}} + \frac{\Delta q}{\Delta t} + \theta \frac{\partial}{\partial x} \left(\frac{2q}{h} \Delta q - \frac{q^2}{h^2} \Delta h \right) \\ + \theta g \left(\Delta h \frac{\partial \zeta}{\partial x} + h \frac{\partial \Delta h}{\partial x} \right) \\ = \text{res}_{\text{mom}}^{n+1,p}, \end{aligned} \tag{46}$$

While this clearly removes physical information from the system, it is sufficient for the derivation of the convergence-error dependent pseudo-time step. Next, the second-order Taylor terms, which determine the linearization error, are added. Additionally, since h, q and ζ are smooth in space, their spatial derivatives are negligible and are dropped as well, resulting in the following:

$$\frac{\Delta h}{\Delta t_{\text{pseu}}} + \frac{\Delta h}{\Delta t} + \theta \frac{\partial \Delta q}{\partial x} = \text{res}_{\text{cont}}^{n+1,p}, \tag{47}$$

$$\begin{aligned} \frac{\Delta q}{\Delta t_{\text{pseu}}} + \frac{\Delta q}{\Delta t} + 2\theta \left(\frac{q}{h} + \theta \left[\frac{1}{h} \Delta q - \frac{q}{h^2} \Delta h \right] \right) \frac{\partial \Delta q}{\partial x} \\ - \theta \left(\frac{q^2}{h^2} + \theta \left[\frac{2q}{h^2} \Delta q - \frac{2q^2}{h^3} \Delta h \right] \right) \frac{\partial \Delta h}{\partial x} \\ + \theta g \left(h + \theta \Delta h \right) \frac{\partial \Delta h}{\partial x} \\ = \text{res}_{\text{mom}}^{n+1,p}. \end{aligned} \tag{48}$$

To condense this system into matrix form, the following variables are introduced:

$$\begin{aligned} u &= \frac{q}{h}, & \Delta u &= \theta \left(\frac{1}{h} \Delta q - \frac{q}{h^2} \Delta h \right), \\ c &= \sqrt{gh}, & \Delta c &= \theta \frac{g}{2c} \Delta h, \end{aligned} \tag{49}$$

to arrive at

$$\mathbf{N} = \theta \begin{pmatrix} 0 & 1 \\ -u^2 - 2u\Delta u & 2u + 2\Delta u \\ +c^2 + 2c\Delta c & \end{pmatrix}. \tag{50}$$

The system can now be written in compact form, namely:

$$\left(\frac{1}{\Delta t_{\text{pseu}}} + \frac{1}{\Delta t} \right) \begin{pmatrix} \Delta h \\ \Delta q \end{pmatrix} + \mathbf{N} \frac{\partial}{\partial x} \begin{pmatrix} \Delta h \\ \Delta q \end{pmatrix} = \begin{pmatrix} \text{res}_{\text{cont}}^{n+1,p} \\ \text{res}_{\text{mom}}^{n+1,p} \end{pmatrix}. \tag{51}$$

The eigenvalues of the matrix $\mathbf{N}$ give insight into the relative importance of the nonlinear terms.

$$\lambda = u + \Delta u \pm (c + \Delta c) \tag{52}$$

The effect of the linearization errors on the convergence behavior can be expected to be small when:

$$|\Delta u|, |\Delta c| \ll |u|, |c| \tag{53}$$

The solution corrections Δh, Δq and subsequently Δu, Δc can be made arbitrarily small by ensuring the diagonal $1/\Delta t_{\text{pseu}} + 1/\Delta t$, and thus $1/\Delta t_{\text{pseu}}$, is made sufficiently large. To this end, the inverse pseudo-CFL number is introduced as:

$$\frac{1}{CFL_{\text{pseu}}} = \max \left(0, \epsilon_{\text{pseu}} \frac{|\Delta u| + |\Delta c|}{|u| + |c|} - \frac{1}{CFL} \right), \tag{54}$$

with parameter ϵ_{pseu} a scaling parameter of order $O(1)$. Larger values will result in smaller pseudo-time steps and consequently more robust but slower convergence. For difficult problems, it may be necessary to increase this value. Considering $1/CFL = \Delta x/((|u| + |c|)\Delta t)$, (54) can also be written as:

$$\frac{\Delta x}{\Delta t_{\text{pseu}}} = \max\left(0, \epsilon_{\text{pseu}}(|\Delta u| + |\Delta c|) - \frac{\Delta x}{\Delta t}\right). \tag{55}$$

This is the close-to-ideal pseudo-time step to limit the solution correction just enough to be stable without sacrificing the convergence speed. However, to calculate Δu and Δc, the solution corrections Δh and Δq are needed, which are yet to be calculated. One way of dealing with this problem is to take the solution corrections from the previous iteration. To prevent possible instabilities, the pseudo-time step is underrelaxed. Since the solution correction for the first iteration does not yet exist, the initial pseudo-time step is conservatively set based on a local pseudo CFL number of around 2 [Borsboom, 2019, p.20]. Additionally, computing the sum of the magnitudes of the terms in Δu instead of $|\Delta u|$ itself may yield increased robustness (albeit at the expense of convergence speed). This implementation is the one applied in SOBEK-RE [Borsboom, 1999, p.6].

Another solution is to take the maximum allowed values for Δh and Δq, assume them to be fairly smooth in space, and plug them into (51) to obtain:

$$\frac{\Delta x}{\Delta t_{\text{pseu}}} = \max\left(\frac{\Delta x}{\Delta t}, \frac{|\text{res}_{\text{cont}}^{n+1,p}|}{\Delta h_{\max}}, \frac{|\text{res}_{\text{mom}}^{n+1,p}|}{\Delta q_{\max}}\right) - \frac{\Delta x}{\Delta t}, \tag{56}$$

as derived in [Borsboom, 2020, p.11]. Notice that while the solution correction Δh, Δq (and thus the linearization error) can be scaled by choosing the pseudo-time step, it also scales with the size of the residual. As the residual vanishes, so does the solution correction (and thus the linearization error). It is important to note that this technique has not yet been extensively tested.