

Shallow water circular hydraulic jump with high-order differential source terms

Final Project for AMCS-312 at KAUST

Carlos Muñoz-Moncayo
Instructor: Prof. David E. Keyes

May 12, 2025

1 Motivation

The Shallow Water Equations (SWE) or Saint-Venant Equations are a depth-averaged, first-order, and hyperbolic system of PDEs for free-surface water wave modeling. Since they are one of the most widely used models for geophysical studies, a large body of work has been devoted to determining their validity range. A particularly challenging problem is to model a standing shock wave that arises from the shallow water circular hydraulic jump (see Figure 1), where phenomena like carbuncles (more commonly observed in the study of gas dynamics) take place. Numerical evidence suggests that such a complex flow could be captured with the SWE through a carefully tailored numerical scheme [6]. Moreover, experimental results indicate that viscosity plays an important role in the behavior of marginally stable structures such as a standing hydraulic jump [11]. Motivated by this, the main goal of this course project is to formulate a shallow water-type model that includes a fourth-derivative "hyperdiffusive" term, and its a robust and scalable discretization.

2 Mathematical model

In their two-dimensional form, the SWE read

$$\partial_t \begin{pmatrix} h \\ hu \\ hv \end{pmatrix} + \partial_x \begin{pmatrix} hu \\ hu^2 + \frac{1}{2}gh \\ huv \end{pmatrix} + \partial_y \begin{pmatrix} hv \\ huv \\ hv^2 + \frac{1}{2}gh \end{pmatrix} + \begin{pmatrix} 0 \\ gh\partial_x z_b \\ gh\partial_y z_b \end{pmatrix} = \mathbf{0}, \quad (1)$$

where h denotes water depth, g gravitational force, z_b bottom topography, and u and v are the depth-averaged fluid velocity in the x and y directions respectively. In this work, we are concerned with the addition of a linear hyperviscosity source term of the form $(-\mu\Delta_{x,y}^2\eta, 0, 0)^T$ to the right-hand side of (1), where Δ^2 is the biharmonic operator, given by

$$\Delta_{x,y}^2\eta = \partial_x^4\eta + \partial_y^4\eta + 2\partial_x^2\partial_y^2\eta, \quad (2)$$

and $\eta(x, y, t) := \max\{h(x, y, t) + z_b(x, y), 0\}$ denotes the water surface elevation. Through a linear dispersion relation analysis (see Appendix A) it can be shown that $\mu > 0$ is necessary for the resulting system to be linearly stable. Since we want to solve the SWE in the annular domain $\Omega = \{(x, y) \in \mathbb{R}^2 : r_0 \leq \sqrt{x^2 + y^2} \leq r_1\}$, it is convenient to formulate them in polar coordinates. Then, the SWE with hyperviscosity may be written as

$$\partial_t \begin{pmatrix} h \\ hu \\ hv \end{pmatrix} + \partial_r \begin{pmatrix} hu \\ hu^2 + \frac{1}{2}gh \\ huv \end{pmatrix} + \frac{1}{r}\partial_\theta \begin{pmatrix} hv \\ huv \\ hv^2 + \frac{1}{2}gh \end{pmatrix} + \begin{pmatrix} 0 \\ gh\partial_r z_b \\ \frac{1}{r}gh\partial_\theta z_b \end{pmatrix} + \begin{pmatrix} \frac{h}{r}(u^2 - v^2) \\ \frac{2h}{r}uv \\ 0 \end{pmatrix} = \begin{pmatrix} -\mu\Delta_{r,\theta}^2\eta \\ 0 \\ 0 \end{pmatrix}, \quad (3)$$

where $r = \sqrt{x^2 + y^2}$, $\theta = \text{atan2}(y, x)$, and now u and v refer to radial and angular velocity, respectively. The biharmonic operator in polar coordinates becomes

$$\Delta_{r,\theta}^2\eta = \frac{r(r^2\partial_r^4\eta + 2r\partial_r^3\eta - \partial_r^2\eta + 2\partial_r^2\partial_\theta^2\eta) + \partial_r\eta - 2\partial_r\partial_\theta^2\eta + 4\partial_\theta^2\eta + \partial_\theta^4\eta}{r^4} \quad (4)$$

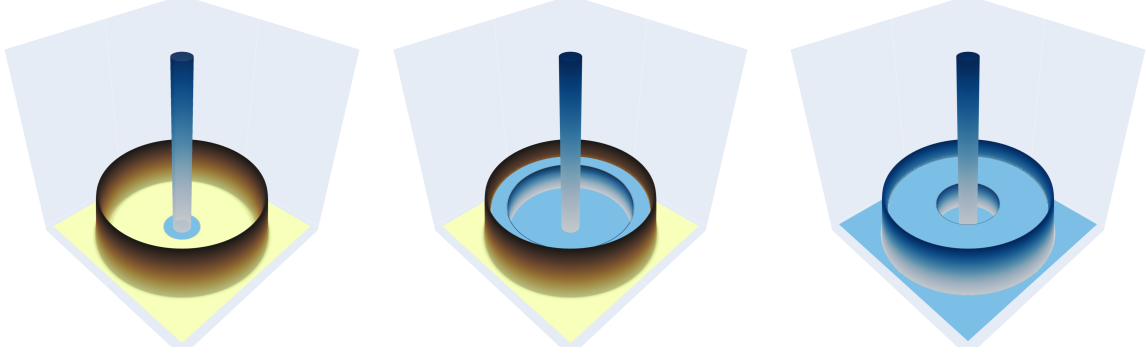


Figure 1: Circular hydraulic jump setup. The water jet hits the bottom topography at the center of the domain and flows outwards. The bathymetry rim forces water recirculation, which leads to the formation of a standing hydraulic jump.

System (3) can be written as

$$\partial_t \mathbf{q} + \partial_r \mathbf{F}(\mathbf{q}) + \frac{1}{r} \partial_\theta \mathbf{G}(\mathbf{q}) + \mathbf{S}_b(\mathbf{q}) + \mathbf{S}_r(\mathbf{q}) = \mathbf{S}_v(\mathbf{q}). \quad (5)$$

3 Numerical methods

We consider the following Cauchy problem for System (5)

$$\begin{cases} \mathbf{q}(r, \theta, 0) = \mathbf{q}_{\text{initial}}(r, \theta), & \text{in } [r_0, r_1] \times [0, 2\pi[, \\ \mathbf{q}(r_0, \theta, t) = \mathbf{q}_{\text{inflow}}(\theta, t), & \text{in } [0, 2\pi[\times [0, T_{\text{final}}], \\ \mathbf{q}(r_1, \theta, t) = \mathbf{q}_{\text{outflow}}(\theta, t), & \text{in } [0, 2\pi[\times [0, T_{\text{final}}], \\ \mathbf{q}(r, 0, t) = \mathbf{q}(r, 2\pi, t) & \text{in } [r_0, r_1] \times [0, T_{\text{final}}]. \end{cases} \quad (6)$$

Let the tensor $\mathbf{q}^{[n]}$ be a discrete approximation of $\mathbf{q}(\cdot, \cdot, t_n)$. We evolve $\mathbf{q}^{[n]}$ to $\mathbf{q}^{[n+1]} \approx \mathbf{q}(\cdot, \cdot, t_n + \Delta t)$ through operator splitting and a hybrid finite volume-finite difference numerical scheme to deal with the advective and diffusive parts respectively.

Since we are interested in the behavior of the solution away from the center of the physical domain, we can safely use a logically rectangular grid in polar coordinates (Figure 3) mapped from a uniform rectangular grid (Figure 2) without worrying about singularities at $r = 0$. Then, we naturally impose the periodic boundary conditions in θ by identifying $C^{\text{comp}}_{i, \theta=0}$ with $C^{\text{comp}}_{i, \theta=2\pi}$, and the *inflow*(*outflow*) boundary conditions at r_0 (r_1) by using ghost cells.

3.1 Operator splitting

We split System (5) into two subproblems:

$$\partial_t \mathbf{q} + \partial_r \mathbf{F}(\mathbf{q}) + \frac{1}{r} \partial_\theta \mathbf{G}(\mathbf{q}) + \mathbf{S}_b(\mathbf{q}) + \mathbf{S}_r(\mathbf{q}) = \mathbf{0}, \quad (\text{ADV})$$

$$\partial_t \mathbf{q} = \mathbf{S}_v(\mathbf{q}), \quad (\text{DIFF})$$

which we solve through Lie-Trotter or Strang splitting [8]. For Lie-Trotter splitting, we

1. compute $\mathbf{q}^*$ as an approximation of the solution $\mathbf{q}(\cdot, \cdot, \Delta t)$ to (ADV) with initial data $\mathbf{q}^{[n]}$;
2. compute $\mathbf{q}^{[n+1]}$ as an approximation to the solution $\mathbf{q}(\cdot, \cdot, \Delta t)$ to (DIFF) with initial data $\mathbf{q}^*$.

For Strang splitting, we

1. approximately solve (DIFF) with initial data $\mathbf{q}^{[n]}$ over a half-time step $\frac{\Delta t}{2}$ to obtain $\mathbf{q}^*$;

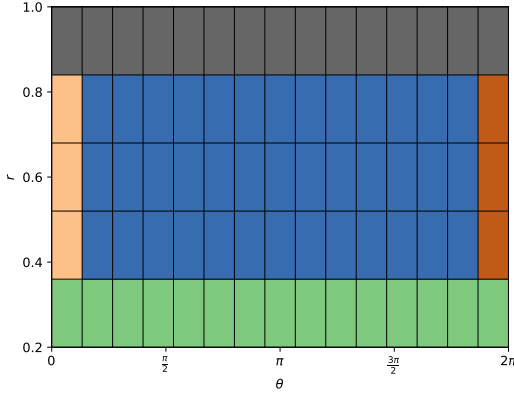


Figure 2: Grid in computational domain

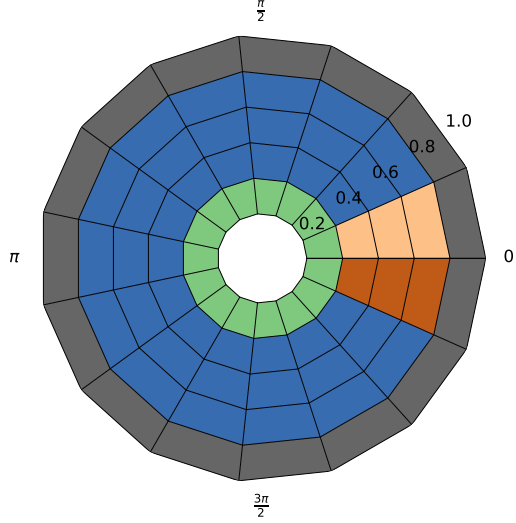


Figure 3: Grid in physical domain

2. compute $\mathbf{q}^{**}$ as an approximate solution to (ADV) over a full step Δt with initial data $\mathbf{q}^*$;
3. compute $\mathbf{q}^{[n+1]}$ as an approximate solution to (DIFF) over a half-time step $\frac{\Delta t}{2}$ with initial data $\mathbf{q}^{**}$.

3.2 Advective terms

A finite volume discretization is used for the purely hyperbolic part of the system. Following [7, Ch. 23.2], the updates to the conserved quantity $\mathbf{q}$, averaged over the physical cell C_{ij} , take the form

$$Q_{ij}^{n+1} = Q_{ij}^n - \frac{\Delta t}{\kappa_{ij} \Delta r} \left(\mathcal{A}^+ \Delta Q_{i-1/2,j}^n + \mathcal{A}^- \Delta Q_{i+1/2,j}^n \right) - \frac{\Delta t}{\kappa_{ij} \Delta \theta} \left(\mathcal{B}^+ \Delta Q_{i,j-1/2}^n + \mathcal{B}^- \Delta Q_{i,j+1/2}^n \right). \quad (7)$$

Here, the capacity function κ represents the ratio between the area of a cell in physical and computational domain, $\kappa_{ij} = \frac{|C_{ij}|}{\Delta r \Delta \theta}$. The fluctuations $\mathcal{A}^\pm \Delta Q_{i\pm 1/2,j}^n$ and $\mathcal{B}^\pm \Delta Q_{i,j\pm 1/2}^n$ represent the net contribution from the approximate Riemann solutions in the radial and angular directions, respectively. The Riemann problems are solved in a dimension-by-dimension basis. Therefore, we get a half CFL condition to estimate the time step [12]

$$\Delta t \leq \frac{1}{2} \min_{i,j} \left\{ \frac{1}{\frac{|s_r|}{\kappa_{ij} \Delta r} + \frac{|s_\theta|}{\kappa_{ij} \Delta \theta}} \right\}, \quad (8)$$

where s_r and s_θ are the speeds of the waves emanating from the Riemann solutions in the radial and angular directions. A detailed description of the one-dimensional Riemann solver for a flat bathymetry is given in [6]. For the test cases involving a variable topography, like the one depicted in Figure 1, we use a generalization of the solver presented in [6] based on hydrostatic reconstruction [2], which is being developed by the Numerical Mathematics Group at KAUST. The resulting scheme for the advective term:

- is second-order accurate for smooth solutions;
- preserves depth-positivity ;
- is entropy stable for flat bottoms;
- suppresses carbuncle instabilities.

The evolution of the advective terms 7 is carried out in PetClaw [1], a PETSc-based massively parallelizable extension of the Clawpack library for the numerical solution of hyperbolic PDEs [10].

3.3 Hyperdiffusion term

The main concern of this work is the solution of the diffusive subproblem (DIFF). Since the first equation decouples from the rest for this problem, we are left with a scalar biharmonic heat-like equation

$$\partial_t h(r, \theta, t) = -\mu \Delta_{r,\theta}^2 \eta(r, \theta, t). \quad (9)$$

The choice of surface elevation on the right-hand side term is intended to avoid the introduction of spurious oscillations in scenarios where the surface elevation is (almost) constant and there are discontinuities in the water depth due to the bathymetry. With this approach, if (9) is solved exactly, a well-balanced solution to the advective subproblem would not be perturbed. Since the bathymetry has no time evolution, from (9) we get

$$\partial_t \eta(r, \theta, t) = -\mu \Delta_{r,\theta}^2 \eta(r, \theta, t). \quad (10)$$

We discretize (10) using finite differences in time and space. To maintain the directly proportional relation $\Delta t \propto \Delta r, \Delta \theta$ arising from the CFL condition (8) of the explicit scheme (7), we need to integrate (10) implicitly in time. We consider both Backward Euler

$$\frac{\eta^{[n+1]} - \eta^{[n]}}{\Delta t} = -\mu \Delta_{r,\theta}^2 \eta^{[n+1]}, \quad (11)$$

and Crank-Nicolson

$$\frac{\eta^{[n+1]} - \eta^{[n]}}{\Delta t} = -\mu \Delta_{r,\theta}^2 \frac{\eta^{[n+1]} + \eta^{[n]}}{2}. \quad (12)$$

In both cases, we have to solve a linear system of the form

$$(I + a\mu \Delta_{r,\theta}^2) \eta^{[n+1]} = B \eta^{[n]}, \quad (13)$$

for $\eta^{[n+1]}$. After getting $\eta^{[n+1]}$, we go back to the conservative variable $h^{[n+1]} = \max\{\eta^{[n+1]} - z_b, 0\}$. To construct an approximation of $\Delta_{r,\theta}^2$, we use second-order accurate centered finite difference approximations for each derivative in (4).

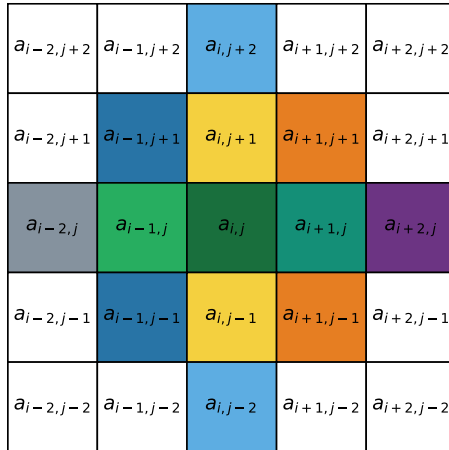


Figure 4: Second-order accurate centered finite difference stencil for $\Delta_{r,\theta}^2$

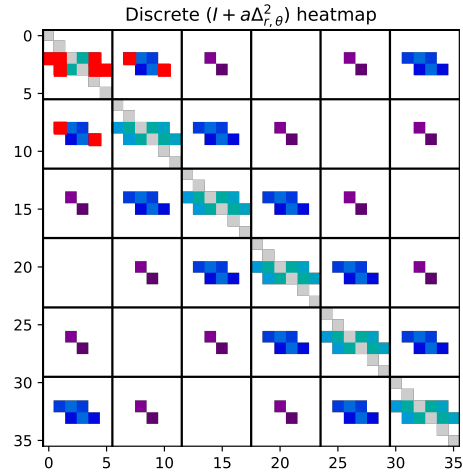


Figure 5: Heatmap for stencil (15) with $N_r = N_\theta = 6$

The resulting stencil is diamond-shaped, as depicted in Figure 4, and takes the form

$$\Delta_{r,\theta}^2 \eta_{i,j}^{[n+1]} \approx \sum_{k,l \in \{-2, \dots, 2\}} a_{i+k,j+l} \eta_{i+k,j+l}^{[n+1]}, \quad (14)$$

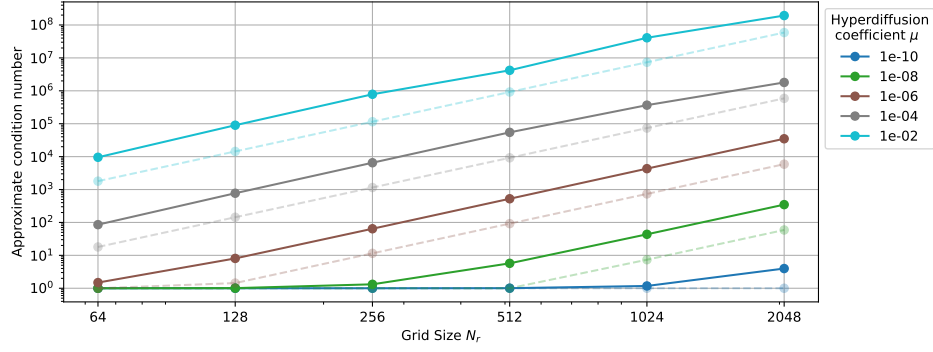


Figure 6: Approximate condition number of $A = (I + \Delta t \mu \Delta_{r,\theta}^2)$, $\kappa(A) = \|A^{-1}\| \|A\|$. For the solid lines, $\|A^{-1}\|$ is estimated by solving several problems $Ax = b$, with b random and unitary. Similarly, $\|A\|$ is approximated by $\|Ax\|$ with x random and unitary. The dashed lines are given by $\max\{1, \mu \Delta t \Delta r^{-4}\}$.

where the non-zero coefficients are given by

$$a_{i-2,j} = \Delta r^{-4} - r^{-1} \Delta r^{-3}, \quad (15a)$$

$$a_{i-1,j+1} = 2r^{-2}(\Delta r \Delta \theta)^{-2} + r^{-3} \Delta r^{-1} \Delta \theta^{-2}, \quad (15b)$$

$$a_{i-1,j} = -4\Delta r^{-4} - 4r^{-2}(\Delta r \Delta \theta)^{-2} - 2r^{-3} \Delta r^{-1} \Delta \theta^{-2} + 2r^{-1} \Delta r^{-3} - r^{-2} \Delta r^{-2} - \frac{1}{2} r^{-3} \Delta r^{-1}, \quad (15c)$$

$$a_{i-1,j-1} = a_{i-1,j+1} \quad (15d)$$

$$a_{i,j+2} = r^{-4} \Delta \theta^{-4}, \quad (15e)$$

$$a_{i,j+1} = -4r^{-4} \Delta \theta^{-4} - 4r^{-2}(\Delta r \Delta \theta)^{-2} + 4r^{-4} \Delta \theta^{-2}, \quad (15f)$$

$$a_{i,j} = 6\Delta r^{-4} + 6r^{-4} \Delta \theta^{-4} + 8r^{-2}(\Delta r \Delta \theta)^{-2} + 2r^{-2} \Delta r^{-2} - 8r^{-4} \Delta \theta^{-2}, \quad (15g)$$

$$a_{i,j-1} = a_{i,j+1} \quad (15h)$$

$$a_{i,j-2} = a_{i,j+2} \quad (15i)$$

$$a_{i+1,j+1} = 2r^{-2}(\Delta r \Delta \theta)^{-2} - r^{-3} \Delta r^{-1} \Delta \theta^{-2}, \quad (15j)$$

$$a_{i+1,j} = -4\Delta r^{-4} - 4r^{-2}(\Delta r \Delta \theta)^{-2} + 2r^{-3} \Delta r^{-1} \Delta \theta^{-2} - 2r^{-1} \Delta r^{-3} - r^{-2} \Delta r^{-2} + \frac{1}{2} r^{-3} \Delta r^{-1}, \quad (15k)$$

$$a_{i+1,j-1} = a_{i+1,j+1} \quad (15l)$$

$$a_{i+2,j} = \Delta r^{-4} + r^{-1} \Delta r^{-3}, \quad (15m)$$

and r denotes $r_{i,j}$ for clarity. A heatmap of the stencil (15) is depicted in Figure 5, where a block-circulant structure arises as a consequence of the periodic boundary conditions in θ . As illustrated in Figure 6, the condition number of the matrix $A = I + \Delta t \mu \Delta_{r,\theta}^2$ grows proportionally to Δr^{-4} and, thus, the problem (13) becomes severely ill-conditioned as the grid is refined.

The ghost cell-based boundary conditions in r are simply imposed by artificially setting to 1 the center entry and zeroing-out the other entries of the stencil that correspond to a boundary in r . A conditioning improvement that could be done, following [3], is to appropriately move the entries that correspond to ghost cells (e.g. red cells in the first block of the heatmap 5) to the right-hand side of the system. Then, a block preconditioner that exploits the resulting good structure of the submatrices could be used. The construction of the necessary DMDAs, the matrix assembly, and the preconditioning-solution (BoomerAMG+LGMRES with $rtol=1.e-12$) of the linear problems (13) has been implemented using *petsc4py* [4].

4 Results

The solvers for the advective and diffusive subproblems communicate through PyClaw [5], a Python interface to Clawpack. Since instabilities were observed in problems with discontinuous solutions with Crank-Nicolson (we suspect this behavior is related to its lack of L-stability), we present only results obtained with backward Euler in this section.

4.1 Quasilinear advection with hyperdiffusion

To qualitatively verify the behavior of the discrete radial biharmonic operator without the effect of nonlinear interactions, we consider the quasilinear scalar advection equation in 2D. The equation is given by

$$\partial_t u + a(x, y) \partial_x u + b(x, y) \partial_y u = -\mu(\partial_x^4 u + \partial_y^4 u + \partial_y^2 \partial_x^2 u), \quad (16)$$

in $\Omega = \{(x, y) \in \mathbb{R}^2 : \frac{1}{5} \leq r(x, y) \leq 1\}$, with constant extrapolation at $r = \frac{1}{5}$ and $r = 1$ and periodic boundary conditions in θ . The advection speeds are given by $a(x, y) = \cos(\theta(x, y))$ and $b(x, y) = \sin(\theta(x, y))$. Thus, for $\mu = 0$, the initial conditions

$$u(x, y, 0) = \mathbb{1}_{\{\frac{3}{5} \leq r(x, y) \leq 1 \wedge \theta(x, y) < \frac{1}{2}\}} + \mathbb{1}_{\{\frac{2}{5} \leq r(x, y) \leq \frac{1}{2} \wedge \pi < \theta(x, y) < \pi + \frac{1}{2}\}}, \quad (17)$$

will simply rotate in the angular direction. A colormap of the numerical solutions after a full rotation is shown in Figure (7) for different values of μ . We can see that both the desired diffusive and boundary behavior are attained.

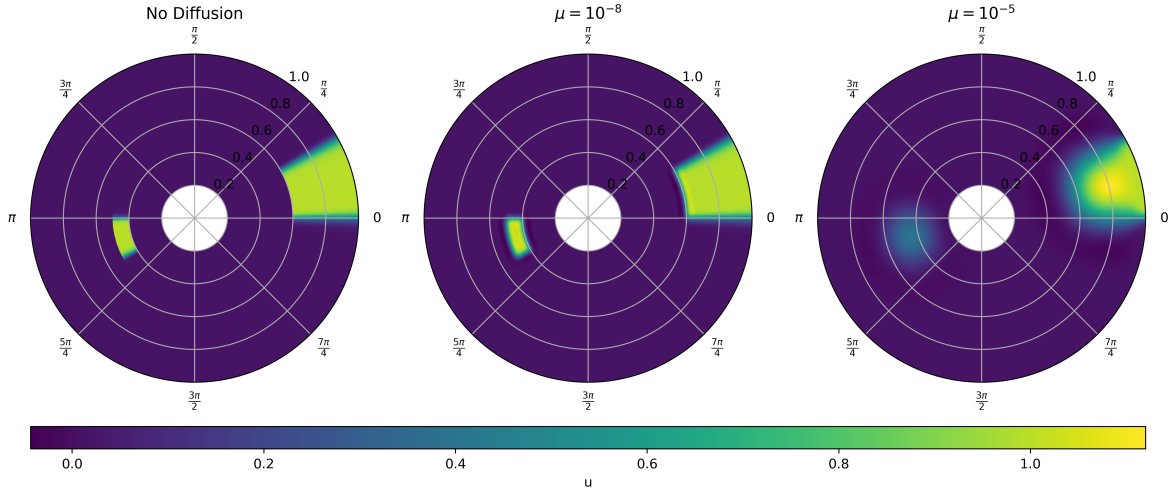


Figure 7: Hyperdiffusion applied to the 2D advection equation on a radial domain.

4.2 SWE lake at rest

A common requirement for shallow water solvers is well-balancing. As noted in Section 3.3, at the analytical level, the diffusive subproblem does not perturb the well-balancing of the advective one. However, at the numerical level, we will have perturbations larger than machine precision arising from the iterative solution of the linear problem (13). Nevertheless, these errors do not accumulate fast, as illustrated in Figure 8 for $\mu \in \{0, 10^{-10}, 10^{-8}, 10^{-6}, 10^{-4}\}$, $r_0 = 0.1$, $r_1 = 1$, and 64 grid points by dimension. We can see that the perturbations of the lake at rest solution remain within $2.5 \cdot 10^{-5}m$ of the baseline (not exactly well-balanced) solution without diffusion.

4.3 SWE radial dam-break problem

Another verification test that we consider is a radial dam-break problem, for which we set the radial and angular velocities to zero everywhere and the depth is a radial step function of the form

$$h(x, y, 0) = \mathbb{1}_{\{\frac{1}{10} \leq r(x, y) \leq \frac{3}{10}\}} + \frac{1}{2} \mathbb{1}_{\{\frac{3}{10} < r(x, y) \leq 3\}}, \quad (18)$$

in $\Omega = \{\frac{1}{10} \leq r \leq 3\}$. Snapshots of the solution at different times are shown in the top row and bottom left entry of Figure 9 for the purely hyperbolic problem $\mu = 0$.

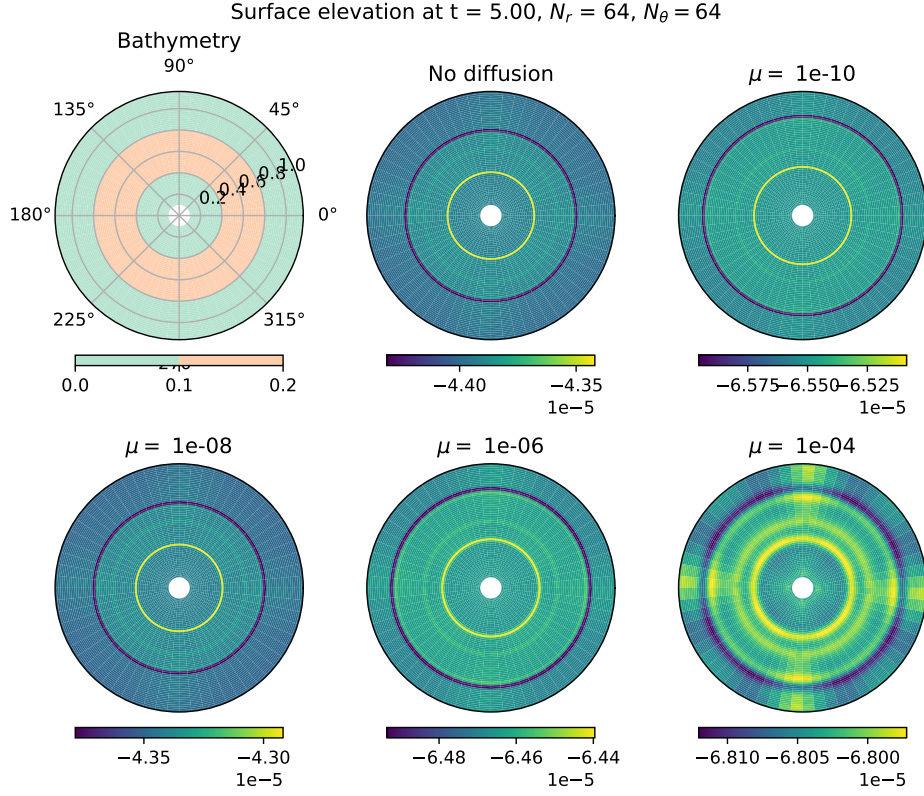


Figure 8: Well-balancing preservation test. Zero (up to machine precision) surface elevation at $t = 0$.

Due to the symmetry of the problem in the angular direction, the behavior of the exact solution is effectively one-dimensional. Indeed, if we take $\partial_\theta \equiv 0$ and $v = 0$ (recalling that v denotes angular velocity in polar coordinates) in (3), we get

$$\partial_t \begin{pmatrix} h \\ hu \end{pmatrix} + \partial_r \begin{pmatrix} hu \\ hu^2 + \frac{1}{2}gh \end{pmatrix} + \begin{pmatrix} 0 \\ gh\partial_r z_b \end{pmatrix} + \begin{pmatrix} \frac{h}{r}u \\ \frac{h}{r}(u^2) \end{pmatrix} = \begin{pmatrix} -\mu\tilde{\Delta}_r^2\eta \\ 0 \end{pmatrix}, \quad (19)$$

where

$$\tilde{\Delta}_r^2\eta = r^{-3} \left(r \left(r^2 \partial_r^4 \eta + 2r \partial_r^3 \eta - \partial_r^2 \eta \right) + \partial_r \eta \right). \quad (20)$$

Therefore, we can discretize (19) with a hybrid finite volume-finite difference scheme as the one described in Section 3. A snapshot of the dam-break solution of (19) for different values of μ is depicted in the bottom right entry of Figure 9.

In the 1D and 2D scenarios, we consider reflecting boundary conditions for the radial velocity and constant extrapolation for the other variables at $r_0 = \frac{1}{10}$. Constant extrapolation is used at the outer radial boundary $r_1 = 3$. A comparison between the numerical solution to the 2D hyperdiffusive problem across different radial transects, and the corresponding 1D solution is displayed in Figure 10 for different values of μ . The hybrid 2D scheme preserves correctly the symmetry of the solution and yields a good approximation of the radial 1D solution.

4.4 SWE circular hydraulic jump

Finally, we consider the hydraulic jump test, which has been the main motivation for the development of this work. We use two approaches to generate it:

- Imposing a constant depth and inflow radial velocity u_{jet} at the interior boundary r_0 , and a constant depth and outflow radial velocity at the exterior boundary r_1 . This can be thought of as a water jet flowing into a water tank, such that the introduced volume of water is negligible compared to the existing volume in the tank.

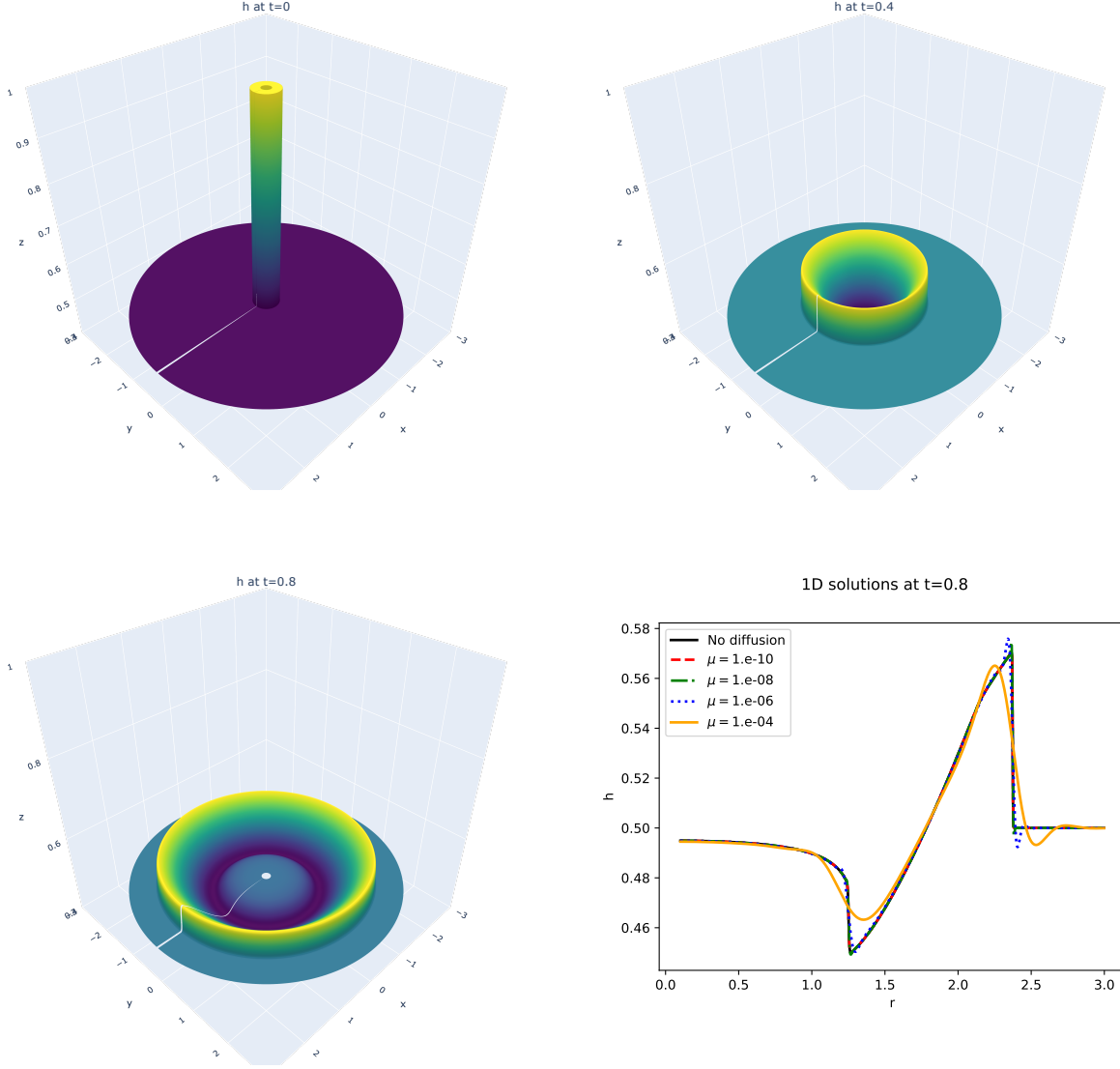


Figure 9: Radial dam break problem and . Top row and bottom left entry: $\mu = 0$ and $N_r = N_\theta = 384$. Bottom left entry: 1D reference radial solutions with forcing source term for different values of μ , $N_r = 1000$.

- Imposing a constant depth and inflow radial velocity u_{jet} at the interior boundary r_0 and using a rim-like bottom topography that forces water recirculation, as illustrated in Figure 1.

From the Rankine-Hugoniot conditions, the location of the jump can be predicted from the boundary conditions, and the exterior boundary conditions can be computed from the interior boundary conditions and the location of the jump [6]. This is the procedure we follow to impose initial conditions and exterior boundary conditions.

4.4.1 Subcritical flow with flat bottom

We consider $r_0 = 0.1$, $r_1 = 1$, $h_{jet} = 0.3$, and $u_{jet} = 0.75$, $h_{out} = 3.44$ and run a simulation with different values of μ on a 384×384 grid to evaluate the effect of hyperdiffusion on a regime with laminar flow. As can be seen in Figures 11 and 12, the nature of the standing front radically changes as the value of μ increases.

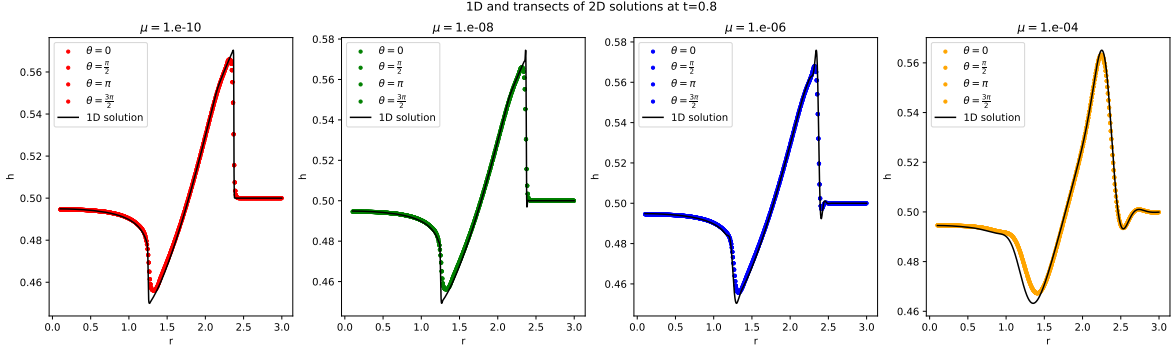


Figure 10: Scatter plots of two-dimensional solution to the dam-break problem across different radial slices and 1D reference solutions. 2D solutions computed with $N_r = N_\theta = 384$, 1D solution computed with $N_r = 1000$.

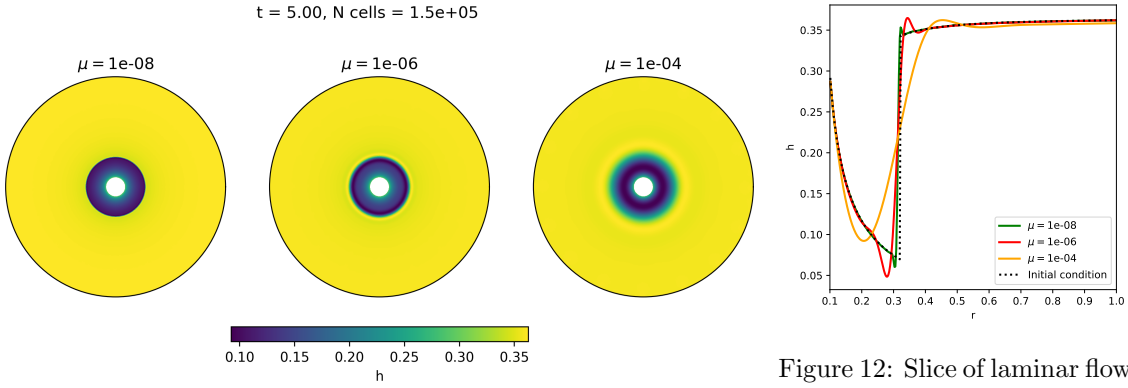


Figure 11: Circular hydraulic jump with laminar flow

Figure 12: Slice of laminar flow circular hydraulic jump across $\theta = 0$

4.4.2 Transition to turbulent flow

A potential application of System (3), and its discretization presented herein, is the study of the transition from laminar to turbulent flow as the Froude number $Fr = |u|/\sqrt{gh}$ increases. Experimental evidence suggests that turbulence is triggered suddenly as certain threshold in Fr at the jet is crossed. However, once it settles, it does not go away immediately as Fr goes below the threshold, but decreases smoothly as function of Fr . Capturing this behavior with a purely hyperbolic model becomes challenging, even with an entropy stable discretization [6], due to the apparent marginal stability of the standing jump. A high-order diffusive term can be used as a handle to numerically capture this behavior in a very fine grid, where numerical diffusion plays less of a stabilizer role.

While we have not reproduced the complete phenomenon, Figure 13 illustrates the transition to turbulent flow. On a 384×384 discretized domain with flat bathymetry, we take $h_{jet} = 0.3$, fix $\mu = 10^{-8}$, and set the location of the jump at $r = 0.35$. Then, we allow u_{jet} to increase until turbulence is triggered (outflow boundary conditions will change depending on u_{jet}).

4.4.3 Weak scalability study with a rim-generated jump

For the final test, we consider a similar setup to the previous test. However, now we generate the hydraulic jump with a rim in the topography as depicted in Figure 14. We fix the diffusion coefficient to $\mu = 10^{-8}$, and consider the grid sizes $N_r = N_\theta \in \{32, 64, 128, 256, 512, 1024\}$ and the number of cores $N_{cores} = (N_r/32)^2 \in \{1, 4, 16, 64, 256, 1024\}$. The weak scalability test is run on IBEX's AMD Rome compute nodes with 128 2Ghz cores (I have not been able to successfully get Clawpack to work with PETSc on Shaheen yet).

The results are displayed in Figures 15 and 16. For the computation of the fractions of execution time, we've simply computed the difference in execution time with and without diffusion. Thus, it's an

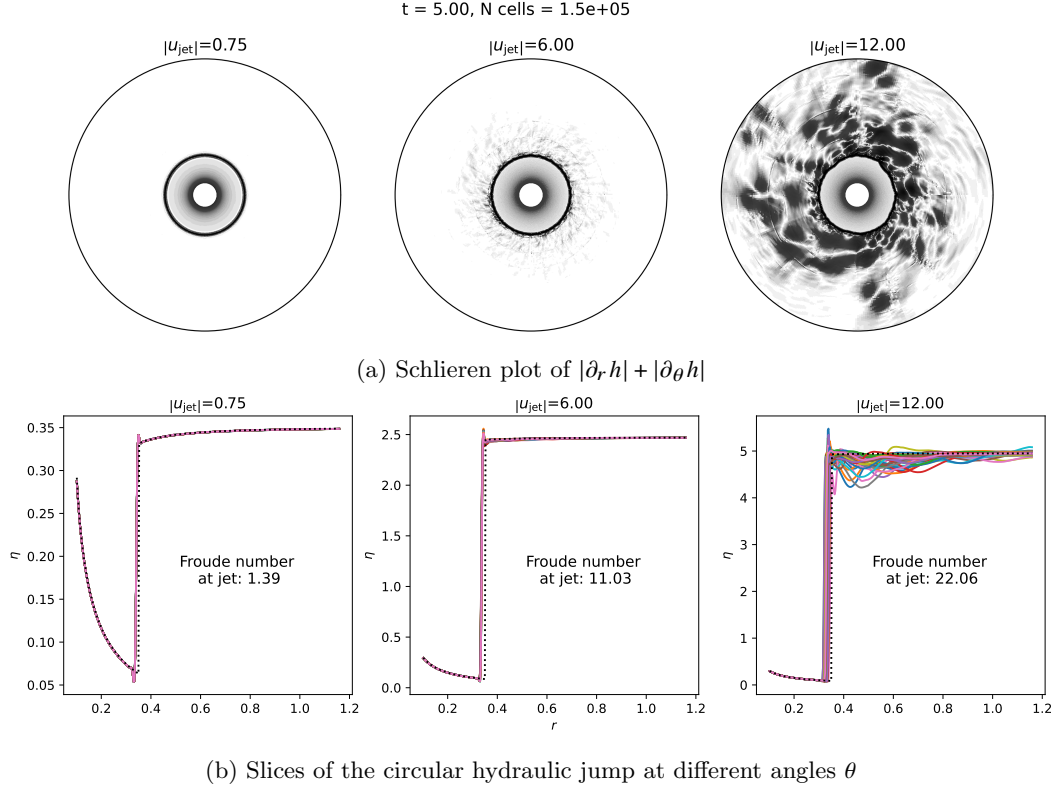


Figure 13: Transition from laminar to turbulent flow in the circular hydraulic jump

unfair comparison for the solver of the advective subproblem, since its fraction includes initialization routines and handling of DMDAs which are being used for the diffusive subproblem. Nevertheless, we can appreciate a clear pattern: as the grid is refined the linear solves take most of the computational cost. This can be attributed to the condition number of the matrix growing as Δr^{-4} , requiring more iterations for the numerical solution. We can see a fast drop in parallel efficiency as the grid size increases. The main reason for this is that, despite each core handling a constant number of grid points, the CFL condition (8) forces us to take more time steps per processor as we refine, and the problem size per processor does not remain constant.

5 Future work

- The improvements to the structure of the stiffness matrix discussed in Section 3.3 are a clear next step to improve the robustness and efficiency of the solver.

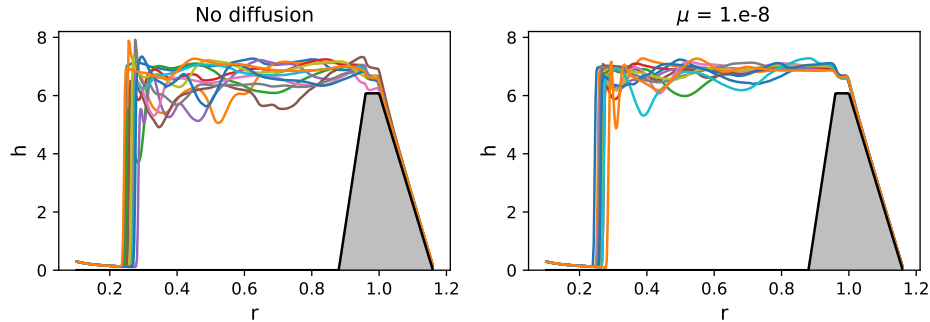


Figure 14: Setup for weak scalability test

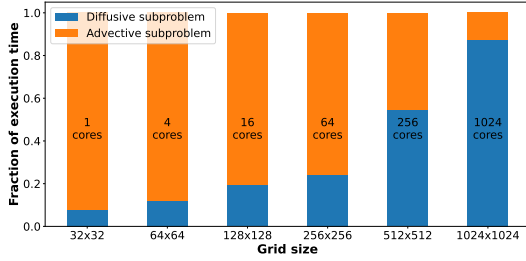


Figure 15: Fraction of execution time taken by each subproblem

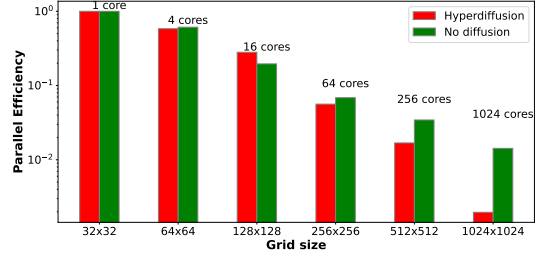


Figure 16: Parallel efficiency of the solver with/without hyperdiffusion in a weak scalability test

- A detailed characterization of turbulence, as discussed in Section 4.4.2 would be an interesting research direction in which this solver can be used.
- Eventually, we would like to incorporate this solver (and perhaps some of these tests as examples) to the Clawpack/PyClaw repository.
- Hyperviscosity could be used to search for polygonal-shaped hydraulic jumps, as the ones found experimentally in [11].
- Future work could include the addition of differential source terms that account for dispersive behavior. A particularly interesting one is a linear third-order dispersive source term analogous to Nwogu’s Boussinesq-type equation [9, Eq. (1.1), $b = c = 0$].

6 Acknowledgments

I would like to thank Prof. David Ketcheson for suggesting this course project to me. The starting point for this project has been a solver for the shallow water equations on an annular grid with a second-order differential source term, developed by Dr. Manuel Quezada de Luna, Dr. Manvendra Rajvanshi, and Prof. David Ketcheson. I would like to thank them for their invaluable advice.

References

- [1] Amal Alghamdi, Aron Ahmadi, David I Ketcheson, Matthew Knepley, Kyle Mandli, and Lisandro Dalcin. Petclaw: A scalable parallel nonlinear wave propagation solver for python. 19th High Performance Computing Symposium (HPC 2011), 2011.
- [2] Emmanuel Audusse, François Bouchut, Marie-Odile Bristeau, Rupert Klein, and Benoît Perthame. A fast and stable well-balanced scheme with hydrostatic reconstruction for shallow water flows. *SIAM Journal on Scientific Computing*, 25(6):2050–2065, 2004.
- [3] Edward L. Bueler. *PETSc for partial differential equations: numerical solutions in C and Python*. Number 31 in Software, environments, and tools. Society for Industrial and Applied Mathematics, Philadelphia, 2021.
- [4] Lisandro D Dalcin, Rodrigo R Paz, Pablo A Kler, and Alejandro Cosimo. Parallel distributed computing using python. *Advances in Water Resources*, 34(9):1124–1139, 2011.
- [5] David I. Ketcheson, Kyle Mandli, Aron J. Ahmadi, Amal Alghamdi, Manuel Quezada De Luna, Matteo Parsani, Matthew G. Knepley, and Matthew Emmett. PyClaw: Accessible, Extensible, Scalable Tools for Wave Propagation Problems. *SIAM Journal on Scientific Computing*, 34(4):C210–C231, January 2012.
- [6] David I Ketcheson and Manuel Quezada de Luna. Numerical simulation and entropy dissipative cure of the carbuncle instability for the shallow water circular hydraulic jump. *International Journal for Numerical Methods in Fluids*, 94(6):655–677, 2022.

- [7] Randall J Leveque. Finite Volume Methods for Hyperbolic Problems.
- [8] Shev MacNamara and Gilbert Strang. Operator splitting. In *Splitting methods in communication, imaging, science, and engineering*, pages 95–114. Springer, 2017.
- [9] Dionyssios Mantzavinos and Dimitrios Mitsotakis. Extended water wave systems of Boussinesq equations on a finite interval: theory and numerical analysis. *Journal de Mathématiques Pures et Appliquées*, 169:109–137, 2023.
- [10] Clawpack Development Team. Clawpack software, 2014. Version 5.1.
- [11] Ali Reza Teymourtash and Mohsen Mokhlesi. Experimental investigation of stationary and rotational structures in non-circular hydraulic jumps. *Journal of Fluid Mechanics*, 762:344–360, 2015.
- [12] Eleuterio F. Toro. *Riemann Solvers and Numerical Methods for Fluid Dynamics: A Practical Introduction*. Springer, Berlin, Heidelberg, 2009.

A Linear stability of diffusive subproblem

To show the linear stability of the hyperdiffusive operator, we consider a plane Fourier mode Ansatz

$$\eta(x, y, t) = \eta_0 \exp [i(k_1 x + k_2 y - \omega t)], \quad (21)$$

for the diffusive subproblem (DIFF) in cartesian coordinates. Then, we get

$$\partial_t \eta = -\omega i, \quad \nabla \eta = i \eta \begin{pmatrix} k_1 \\ k_2 \end{pmatrix}, \quad \Delta \eta = -(k_1^2 + k_2^2) \eta, \quad \Delta^2 \eta = (k_1^2 + k_2^2)^2 \eta, \quad (22)$$

which, after being inserted into the PDE, yields

$$-i\omega \eta = -\mu(k_1^2 + k_2^2)^2 \eta \Rightarrow \omega = -i\mu(k_1^2 + k_2^2)^2. \quad (23)$$

Therefore, the imaginary part of the linear dispersion relation ω will be negative, and thus the PDE will be linearly stable, as long as $\mu > 0$.