POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Executive Summary of the Thesis

Variational data assimilation for porous shallow water equations

Laurea Magistrale in Mathematical Engineering - Ingegneria Matematica

Author: Marco Spadoni

Advisor: Prof. Anna Scotti

Co-advisors: Antoine Rousseau, Pascal Finaud-Guyot

Academic year: 2024-2025

1. Introduction

Shallow water models with porosity are designed to simulate flood flows orders of magnitude faster than classical shallow-water models thanks to a relatively coarse grid and large time step. Porosity ϕ is introduced in the shallow water model to account for the presence of buildings, vegetation, and other structures in the floodplain, avoiding the need of refining the mesh to the fine scale of such obstacles. In other words, porosity models the obstacles present at the subgrid scale. Physically, porosity represents the fraction of the area available for flow. It ranges from 0 (solid obstruction) to 1 (fully open). The aim of this thesis is to show that variational data assimilation can be applied to improve the knowledge on the spatial distribution of porosity using water depth and unit discharge data and, ultimately, increase accuracy in the prediction of the flood flow.

2. Porous shallow water equations

The two-dimensional single porosity shallow water equations read [1]:

$$\frac{\partial}{\partial t}(\phi \boldsymbol{U}) + \frac{\partial}{\partial x}(\phi \boldsymbol{F}) + \frac{\partial}{\partial y}(\phi \boldsymbol{G}) = \boldsymbol{S}$$

$$\text{with} \quad \boldsymbol{U} = \begin{pmatrix} h \\ hu \\ hv \end{pmatrix}, \ \boldsymbol{F} = \begin{pmatrix} hu \\ hu^2 + gh^2/2 \\ huv \end{pmatrix},$$

$$\boldsymbol{G} = \begin{pmatrix} hv \\ huv \\ hv^2 + gh^2/2 \end{pmatrix}, \ \boldsymbol{S} = \begin{pmatrix} 0 \\ S_{0,x} + S_{f,x} \\ S_{0,y} + S_{f,y} \end{pmatrix},$$

where g is the gravitational acceleration, h is the water depth, u and v are the two velocity components, $S_{0,x}$ and $S_{0,y}$ are the source terms arising from the bottom slopes and porosity variations in the x and y directions, $S_{f,x}$ and $S_{f,y}$ are the source terms arising from friction in the x and y directions, and ϕ is the porosity. Porosity is assumed to depend on the space coordinates only. The topographical source terms are given by

$$S_{0,x} = -\phi g h \frac{\partial z_b}{\partial x} + g \frac{h^2}{2} \frac{\partial \phi}{\partial x} \tag{1}$$

$$S_{0,y} = -\phi g h \frac{\partial z_b}{\partial y} + g \frac{h^2}{2} \frac{\partial \phi}{\partial y}$$

where z_b is the bed elevation with respect to a fixed reference level. The friction terms are given by:

$$S_{f,x} = -\phi g h \frac{(u^2+v^2)^{1/2}}{K^2 h^{4/3}} u - \phi g h s_x (u^2+v^2)^{1/2} u$$

$$S_{f,y} = -\phi g h \frac{(u^2+v^2)^{1/2}}{K^2 h^{4/3}} v - \phi g h s_y (u^2+v^2)^{1/2} v$$

where K is the Strickler coefficient and s_x and s_y are the head loss coefficients accounting for the singular head losses due to the presence of obstacles in the x and y directions, respectively.

3. The adjoint model

Variational data assimilation aims to minimize a cost function that measure the mismatch between observations and model output, through a gradient descent algorithm. Bearing in mind that in real-world applications the number of components of the gradient varies from 10^6 to 10^8, the computationally cheapest way of determining the gradient is through the adjoint model [2], which is derived exploiting the adjoint variables (Lagrange multipliers). We here derive the adjoint equations corresponding to the one-dimensional porous shallow water equations, following a derivation similar to that performed by Sanders et al. for the classical shallow water equations [3]. We consider the one-dimensional case as a starting point to reduce the computational cost of the procedure. We recall that the one dimensional shallow water equations with porosity without friction read:

$$\begin{cases} \dfrac{\partial(\phi h)}{\partial t} + \dfrac{\partial(\phi q)}{\partial x} = 0 \\ \dfrac{\partial(\phi q)}{\partial t} + \dfrac{\partial}{\partial x}\Big(\phi\dfrac{q^2}{h} + \phi g\dfrac{h^2}{2}\Big) = S_{0,x} \end{cases}$$

where the unit discharge q is given by $q = hu$. Let the cost function be:

$$J = \int_0^T \int_0^L r(h, q; x, t)dxdt$$

where r is a residual measuring the distance between observations and the computed solution:

$$r(h,q;x,t) = \frac{1}{2}\Big(h(x,t) - h_{\text{obs}}(x,t)\Big)^2 + \frac{1}{2}\Big(q(x,t) - q_{\text{obs}}(x,t)\Big)^2$$

The Lagrangian then is:

$$\mathcal{L} = J + \int_0^T \int_0^L \lambda_h\left(\frac{\partial(\phi h)}{\partial t} + \frac{\partial(\phi q)}{\partial x}\right) + \lambda_q\left(\frac{\partial(\phi q)}{\partial t} + \frac{\partial}{\partial x}\Big(\phi\frac{q^2}{h} + \phi g\frac{h^2}{2}\Big) - S_{0,x}\right)dxdt$$

where λ_h and λ_q are two Lagrange multipliers differentiable in x and t. We substitute $S_{0,x}$ with its expression given by (1) and we integrate by parts to get:

$$\begin{aligned} \mathcal{L} = J + \int_0^T \int_0^L & -\phi h\frac{\partial\lambda_h}{\partial t} - \phi q\frac{\partial\lambda_h}{\partial x} \\ & -\phi q\frac{\partial\lambda_q}{\partial t} - \Big(\phi\frac{q^2}{h} + \phi g\frac{h^2}{2}\Big)\frac{\partial\lambda_q}{\partial x} \\ & -\Big(g\frac{h^2}{2}\frac{\partial\phi}{\partial x} - \phi g h\frac{\partial z_b}{\partial x}\Big)\lambda_q\, dxdt \\ & + \int_0^L \Big[\phi h\lambda_h + \phi q\lambda_q\Big]_0^T dx \\ + \int_0^T & \Big[\phi q\lambda_h + \Big(\phi\frac{q^2}{h} + \phi g\frac{h^2}{2}\Big)\lambda_q\Big]_0^L dt \end{aligned} \tag{2}$$

Taking the first variation with respect to h and q we obtain the adjoint model:

$$\phi\frac{\partial\boldsymbol{\Lambda}}{\partial\tau} + \phi A\frac{\partial\boldsymbol{\Lambda}}{\partial x} + C\boldsymbol{\Lambda} + \boldsymbol{D} = 0 \tag{3}$$

with

$$\boldsymbol{\Lambda} = \begin{pmatrix}\lambda_h \\ \lambda_q\end{pmatrix}$$

$$A = \begin{pmatrix} 0 & \frac{q^2}{h^2} - gh \\ -1 & -\frac{2q}{h}\end{pmatrix}$$

$$C = \begin{pmatrix} 0 & -gh\frac{\partial\phi}{\partial x} + \phi g\frac{\partial z_b}{\partial x} \\ 0 & 0\end{pmatrix}$$

$$\boldsymbol{D} = \begin{pmatrix}\frac{\partial r}{\partial h} \\ \frac{\partial r}{\partial q}\end{pmatrix}$$

where $\tau = T - t$ is measured in the reverse time direction (hence the name backward problem). We finally compute the gradient of the cost function J with respect to the parameter to be assimilated from Eq. (2) by setting the gradient of $\mathcal{L}$ to zero. In particular we obtain that the gradients with respect to the initial conditions $h(x,0) = h_0(x)$ and $q(x,0) = q_0(x)$ are given by $\nabla_{h_0}J(x) = \phi(x)\lambda_h(x,0)$, and $\nabla_{q_0}J(x) = \phi(x)\lambda_q(x,0)$, while the gradient of J with respect to the porosity ϕ is given by $\nabla_{\phi_0}J(x) = h(x,0)\lambda_h(x,0) + q(x,0)\lambda_q(x,0)$. Notice that since ϕ is constant in time (i.e. $\phi(x,t) = \phi(x)\ \forall t$), computing the gradient of J with respect to $\phi_0 = \phi(x,0)$ is enough to perform data assimilation.

It should be noticed that, in contrast to the basic shallow-water equations, which have well-known

conservation properties, the adjoint shallow-water equations cannot be recast in conservative form. For this reason the adjoint problem shall be solved by different numerical schemes than those of the classical shallow-water equations, which typically are solved in conservative form. In particular we adopt the Roe's one step fluctuation splitting method [4] whose aim is to compute the net effect of all the waves, called fluctuations, arising at each computational cell edge from the Riemann problem. The matrix A can be decomposed as $A = R\Omega R^{-1}$, where

$$\Omega = \begin{pmatrix} -\sqrt{gh} - q/h & 0 \\ 0 & \sqrt{gh} - q/h \end{pmatrix}$$

$$R = \frac{1}{2\sqrt{gh}} \begin{pmatrix} \sqrt{gh} - q/h & \sqrt{gh} + q/h \\ 1 & -1 \end{pmatrix}$$

Ω is the diagonal matrix containing the eigenvalues of A. The columns of R contain the corresponding right eigenvectors. We define $\Omega^{\pm}$ as the the diagonal matrices containing the positive and negative eigenvalues of A respectively. We split the fluctuations $\mathcal{A}\Delta\mathbf{\Lambda}$ into positive and negative moving waves as follows:

$$\mathcal{A}\Delta\mathbf{\Lambda} = \mathcal{A}^{+}\Delta\mathbf{\Lambda} + \mathcal{A}^{-}\Delta\mathbf{\Lambda}$$

where $\mathcal{A}^{\pm} = R\ \Omega^{\pm}R^{-1}$. The fluctuation splitting approach leads to the following discretization of Eq. (3):

$$\mathbf{\Lambda}_i^{n-1} = \mathbf{\Lambda}_i^n - \frac{\Delta t^n}{\phi_i}\left[\frac{(\phi\mathcal{A}^{+}\Delta\mathbf{\Lambda})^n_{i-\frac{1}{2}} + (\phi\mathcal{A}^{-}\Delta\mathbf{\Lambda})^n_{i+\frac{1}{2}}}{\Delta x} + (C\mathbf{\Lambda})_i^n + D_i^n\right]$$

where $\Delta\mathbf{\Lambda}^n_{i+\frac{1}{2}} = \mathbf{\Lambda}^n_{i+1} - \mathbf{\Lambda}^n_i$. Boundary conditions are implemented by linearly extrapolating the adjoint variables to the ghost cells while zero-order extrapolating the gradients. A null initial condition is set for $\mathbf{\Lambda}$, namely $\mathbf{\Lambda}(x, T) = 0$. It should be noticed that initial conditions for the adjoint problem refers to $t = T$ or, in other words, $\tau = 0$.

4. Numerical experiments

Variational data assimilation is performed through an iterative gradient descent algorithm that alternates between the forward model solver (namely a finite volume solver adopting the HLL flux) and the backward problem solver. Nested in each gradient descent iteration, we perform some backtracking line search iterations to adaptively choose a step size that guarantees a sufficient decrease in the objective function: at each iteration we start from a large step and we shrink it by a factor α until the Armijo condition is satisfied. Let γ be the parameter that we aim to assimilate, we report the assimilation algorithm:

Algorithm 1 Variational data assimilation

$k = 0$, $\|\nabla_\gamma J^0\|_2 = 1 + \epsilon$
while $k < N_{\max}$ and $\|\nabla_\gamma J^k\|_2 > \epsilon$ **do**
  Compute $h^k(x,t)$ and $q^k(x,t)$ from the discrete forward model
  Integrate the discrete adjoint model backwards from $t = T$ to $t = 0$ to obtain $\lambda_h^k(x,t)$ and $\lambda_q^k(x,t)$
  Compute $\nabla_\gamma J^k$ and $\|\nabla_\gamma J^k\|_2$
  while $J(\gamma^k - \delta^k\nabla_\gamma J^k) > J(\gamma^k) - c\delta^k\nabla_\gamma J^{k^T}\nabla_\gamma J^k$ **do**
    $\delta^k \to \alpha\delta^k$
  end while
  $\gamma^{k+1} = \gamma^k - \delta^k\nabla_\gamma J^k$
  $k \to k + 1$
end while

We test the assimilation algorithm for the one-dimensional shallow water equations first with the aim of retrieving improved initial conditions, and then with the aim of retrieving the porosity distribution.

4.1. Assimilation of initial conditions

We run the forward solver with the following initial conditions for h and q to generate synthetic data used as observations:

$$\begin{cases} h_{\text{obs}}(x, t = 0) = 0.05 \cdot e^{-0.1(x-50.0)^2} + 1.0 \\ q_{\text{obs}}(x, t = 0) = 0.0 \end{cases}$$

We then produce a first guess by running the forward solver with the following initial conditions:

$$\begin{cases} h_{\text{start}}(x, t = 0) = 0.02 \cdot e^{-0.1(x-50.0)^2} + 1.0 \\ q_{\text{start}}(x, t = 0) = 10^{-4} \cdot x(x - L) \end{cases}$$

We run the assimilation algorithm until convergence ($\epsilon = 10^{-6}$) with an assimilation window (i.e. the time interval over which observations

are collected and compared to model predictions) of 8 seconds. Figure 1 shows that at the end of the assimilation process, the assimilated initial conditions match the true initial conditions. Indeed the behavior of the cost function J reported in Figure 2 confirms that the assimilated solution is converging towards the observations.

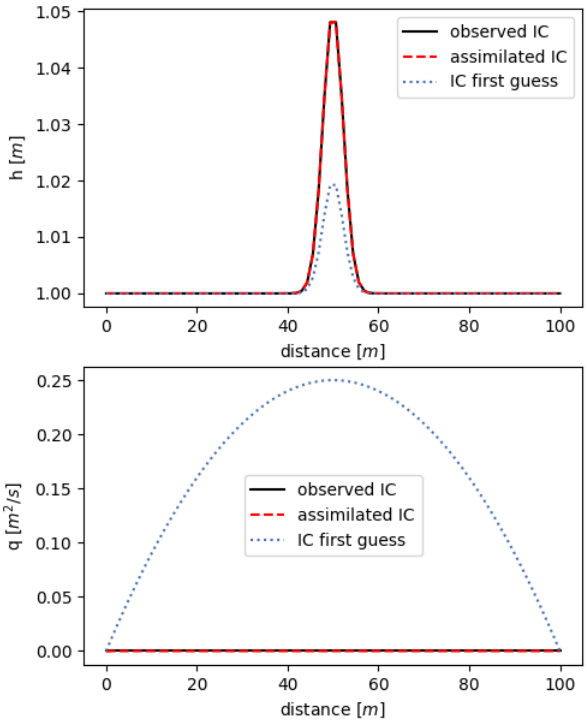

Figure 1: Assimilation of initial conditions.

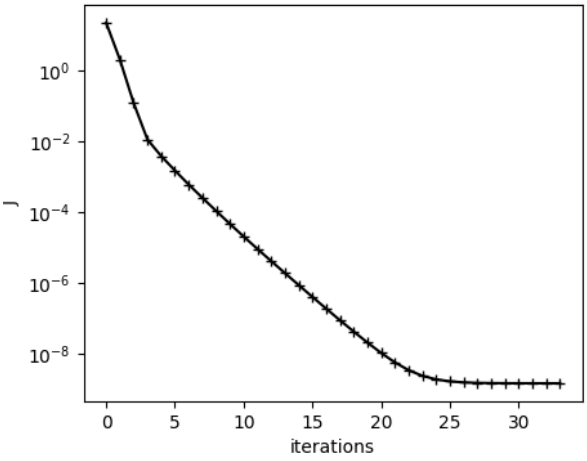

Figure 2: Assimilation of initial conditions: cost function J.

We then test the effect of different time window lengths on the assimilation process. Figure 3 shows how well the initial conditions are assimilated with different assimilation window lengths. In particular, the following assimilation errror is considered at each iteration k:

$$e_A^k = \int_0^L |h^k(x,0) - h_{\text{obs}}(x,0)|dx + \int_0^L |q^k(x,0) - q_{\text{obs}}(x,0)|dx$$

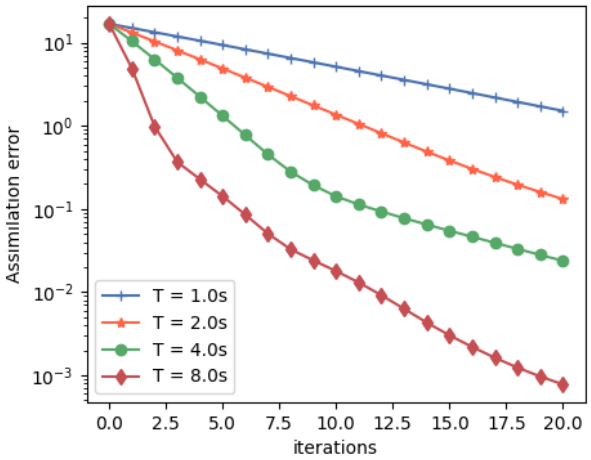

Figure 3: Assimilation of initial conditions using different assimilation windows (fixed $\delta = 0.1$).

As expected longer assimilation windows result in smaller values of the assimilation error given the same number of iterations of the assimilation algorithm. Of course, the longer the assimilation window, the more computationally expensive the forwards and the backward solvers will be.

4.2. Assimilation of porosity

Synthetic data used as observations are generated by running the forward solver with the following values for porosity and initial conditions:

$$\phi_{\text{target}}(x) = 1.0 - 0.3e^{-0.2(x-75.0)^2}$$
$$h(x,0) = 1.0 + 0.42e^{-0.1(x-50.0)^2}$$
$$q(x,0) = 0.0$$

The initial guess is generated by using the same initial conditions and by setting the porosity to $\phi_{\text{start}}(x) = 1$ in the whole domain. We run Algorithm 1 and we compare the results with those obtained by running a standard gradient descent algorithm with a fine-tuned fixed step

length $\delta = 0.1$: despite the additional cost deriving from the adaptivity of the step size, the first outperforms the latter, meaning that, for the same run time, it returns a smaller value of the cost function J (Figure 4). Figure 5 reports the target and assimilated porosity distributions using Algorithm 1.

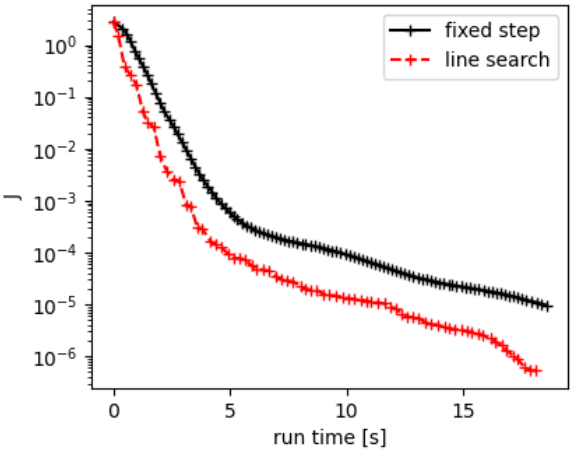

Figure 4: Assimilation of porosity: algorithm comparison.

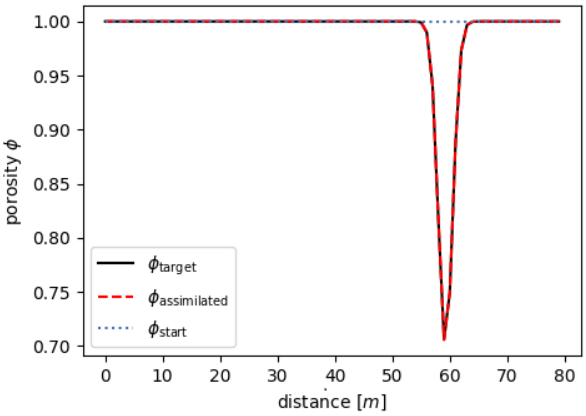

Figure 5: Porosity distributions before and after assimilation

We then repeat the same numerical experiment only changing the target porosity that now reads:

$$\phi_{\text{target}} = 0.8 - 0.3e^{-0.2(x-75.0)^2}$$

We notice that, in this case, we are not able to retrieve the target porosity (Figure 6).
We can solve this issue by adding a regularizing term to the cost function that penalizes the distance from the background state (a prior estimate) ϕ_b. The new cost function reads:

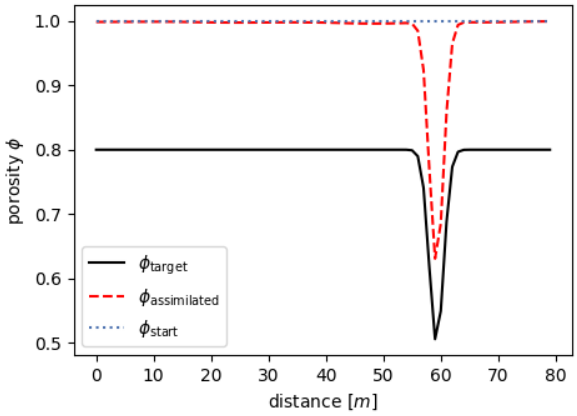

Figure 6: Porosity distributions before and after assimilation

$$\begin{aligned} J = w_1 \int_0^T \int_0^L \Big(h(x,t) - h_{\text{obs}}(x,t)\Big)^2 dxdt \\ + w_2 \int_0^T \int_0^L \Big(q(x,t) - q_{\text{obs}}(x,t)\Big)^2 dxdt \\ + w_3 \int_0^T \int_0^L \Big(\phi(x) - \phi_b(x)\Big)^2 dxdt \end{aligned}$$

We test Algorithm 1 with different values for the weights w_i as reported in Table 1. We set $\phi_b(x) = 0.8$. The resulting assimilated porosity distributions are reported in Figure 7. This experiment highlights the necessity of including all the terms in the cost function in order to retrieve the right porosity distribution.

Test name	w_1	w_2	w_3
A	0.0	0.5	0.5
B	0.5	0.5	0.0
C	0.33	0.33	0.33
D	0.5	0.2	0.3

Table 1: Weight choices

5. Conclusions

In this thesis, we addressed the problem of variational data assimilation applied to the porous shallow water equations, with the aim of improving the spatial estimation of porosity by integrating observational data into the numerical model. The proposed method was success-

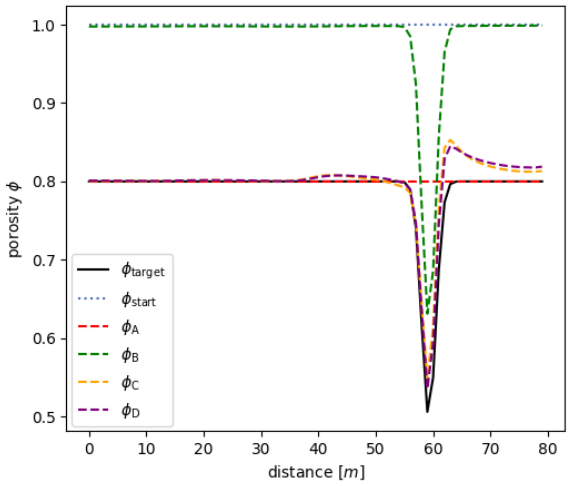

Figure 7: Assimilation of porosity using different weights

fully tested for assimilating both the initial conditions and the spatial distribution of porosity, demonstrating the feasibility and effectiveness of the approach. Notice that, although the numerical experiments were all performed in a one-dimensional setting, all the procedures presented can naturally be extended to the two-dimensional case with just an increased implementation complexity. Nonetheless, this study does have some limitations. In particular, we assumed that observational data are available at every point in space and at each time step, which is rarely the case in real-world applications. While the cost function can easily be modified to accommodate sparse data (for instance by replacing the computation of the integral with a sum over available data points), the quality of the data in terms of their size, location in space and frequency in time plays a major role in the well-posedness of the assimilation problem. Moreover, we assumed observational data to be free of noise and error. Lastly, this approach to variational data assimilation, although computationally efficient, is problem specific: it requires to derive and numerically solve the corresponding adjoint equations, which vary with each different forward model. This manual process could be avoided by exploiting automatic differentiation techniques or by training differentiable surrogate models (e.g. neural networks) to approximate the forward model.

References

[1] V. Guinot and S. Soares-Frazao, "Flux and source term discretization in two-dimensional shallow water models with porosity on unstructured grids," *Int. J. Numer. Meth. Fluids*, pp. 309–345, 2006.

[2] D. Givoli, "A tutorial on the adjoint method for inverse problems," *Computer Methods in Applied Mechanics and Engineering*, vol. 380, p. 113810, 2021.

[3] B. F. Sanders and N. D. Katopodes, "Adjoint sensitivity analysis for shallow-water wave control," *Journal of Engineering Mechanics*, 2000.

[4] B. F. Sanders and S. F. Bradford, "High-resolution, monotone solution of the adjoint shallow-water equations," *Int. J. Numer. Meth. Fluids*, pp. 38:139–161, 2002.