CILAMCE 2017
Florianópolis - Santa Catarina - Brazil

A CONTINUOUS FINITE ELEMENT APPROACH TO THE WELL-BALANCED SHALLOW WATER EQUATIONS

Túlio L. Santos

PetroSoft design desenvolvimento de softwares Ltda.

Department of civil engineering, COPPE/Federal University of Rio de Janeiro

Alvaro L. G. A. Coutinho

Department of civil engineering, COPPE/Federal University of Rio de Janeiro

Abstract. *It is presented a well-balanced two-dimensional single layer shallow water model which considers viscous, wind and Coriolis forces, as well as bed friction. It is outlined how each term can be defined and the whole model is brought out as a generalized convection-diffusion equation. The inherent non-linear system is solved by a continuous finite element approach. Also, two SUPG and three shock capturing operators are applied and compared. Time integration is performed with a predictor-multicorretor algorithm. Additionally, mesh refinement/coarsening and time stepping are employed in an adaptive fashion. Finally, numerical results are presented for some test problems and comparisons with analytical or literature available solutions are made.*

Keywords: *Well-balanced equations, Shallow water model, Finite element, Shock capturing; SUPG*

1 INTRODUCTION

This work is driven by the ongoing demand of the general field of petroleum geology to simulate fluid flows over potentially vast domains with complex shapes and bed topography. In fact, these simulations are usually part of specific purpose systems whose objectives are commonly those of reconstructing or understanding geological features or processes. For instance, their goals might be to recreate channels, deltas and lobes or to reconstruct entire reservoirs and basins, while considering aspects such as sedimentary or diagenetic processes. Nonetheless, it should be noted that the coupling between the fluid dynamics and other physical models or phenomena can be complex and must be analyzed for each individual case.

In consideration of the foregoing, with the intent to faithfully reproduce features observed in nature, one might, at first, want to fully solve the Navier-Stokes complete set of equations. However, this approach can be considered unfeasible for practical applications due to its high computational cost. On this account, reduced models are often employed provided their assumptions are not too restrictive in regard to their applications' objectives. Accordingly, models that take advantage of the shallow water approximation have been substantially studied and applied over the last years. At first, it is good to convey that despite the fact that its label refers to shallow environments, this kind of model can also be employed on deep water regions. In fact, the term 'shallow' indicates that the extent of the simulated regions is significatively greater than its characteristic depth. In being so, some simplifications can be made on standard flow equations.

All in all, this type of model have found its way into the study of geophysical problems such as atmospheric (Giraldo, 2014) and ocean (Sármány et al., 2013) modeling, where vast regions need to be analyzed. Besides, shallow water models have also been derived for compressible flows, such as for gas flows (Karel'skii et al., 2013) and magnetohydrodynamics (Klimachkov et al., 2016).

To numerically solve the system of equations inherent to a shallow water model, often finite volume (Hou et al., 2015) and discontinuous finite element (Ambati et al., 2007) methods are employed. Also, continuous finite element (Castro et al., 2007) and finite difference (Groenenberg et al., 2009) techniques have been applied over the years. Furthermore, significative effort has been made in improving the robustness and accuracy of shallow water models (Hou et al., 2015; Viero et al., 2017) or in creating optimized GPU implementations (Gandham et al., 2015; Vacondio et al., 2017).

Altogether, the objective of this work is to define a ground base on which more complex models can be founded. For instance, a morphodynamic model capable of dealing with the transport, erosion and deposition of sediments as would be incurred, for example, due to turbidity currents. That being said, in Section (2), the adopted physical model is presented while Section (3) explains how it is numerically solved. Then Section (4) gives out a few details about the solver's implementation. Obtained results are discussed in Section (5). At last, concluding remarks and suggestions for future works are displayed at Section (6).

2 PHYSICAL MODEL

An schematic representation of the adopted shallow water model is presented at Fig. (1). On it, $h(x, y, t)$ represents the water column's height (or water depth), while $z_b(x, y)$ and

$\eta(x, y, t)$ are, respectively, the sea bed and water surface elevation from a datum fixed at the bed's lowest point.

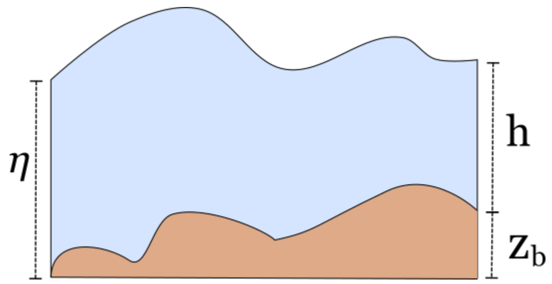

Figure 1: Schematic representation of the adopted shallow water model.

Moreover, the model's unknowns are denoted by:

$$\boldsymbol{U}^T = [hu, hv, \eta] = [q_x, q_y, \eta] \tag{1}$$

Here u and v are depth-averaged velocities and $q_x = hu$ and $q_y = hv$ are specific discharges (or discharges per unit width).

In this context, the set of equations that govern the flow can be given in the form of a generalized convection-diffusion equation:

$$\frac{\partial \boldsymbol{U}}{\partial t} + \boldsymbol{A} \cdot \nabla \boldsymbol{U} - \nabla \cdot [\boldsymbol{K} \nabla \boldsymbol{U}] + \boldsymbol{C}\boldsymbol{U} = \boldsymbol{S} \tag{2}$$

where:

$$\boldsymbol{A} \cdot \nabla \boldsymbol{U} = \boldsymbol{A}^T \nabla \boldsymbol{U} \qquad \boldsymbol{A}^T = [\boldsymbol{A_1} \boldsymbol{A_2}] \tag{3}$$

$$\nabla \boldsymbol{U} = \begin{bmatrix} \boldsymbol{I_3}\, \partial/\partial x \\ \boldsymbol{I_3}\, \partial/\partial y \end{bmatrix} \boldsymbol{U} \qquad (\nabla \cdot) = \begin{bmatrix} \boldsymbol{I_3} \dfrac{\partial}{\partial x} & \boldsymbol{I_3} \dfrac{\partial}{\partial y} \end{bmatrix} \qquad \boldsymbol{I_3} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4}$$

$$\boldsymbol{A_1} = \begin{bmatrix} 2u & 0 & g(\eta - z_b) - u^2 \\ v & u & -uv \\ 1 & 0 & 0 \end{bmatrix} \qquad \boldsymbol{A_2} = \begin{bmatrix} v & u & -uv \\ 0 & 2v & g(\eta - z_b) - v^2 \\ 0 & 1 & 0 \end{bmatrix} \tag{5}$$

$$\boldsymbol{K} = \begin{bmatrix} \boldsymbol{K_{11}} & \boldsymbol{K_{12}} \\ \boldsymbol{K_{21}} & \boldsymbol{K_{22}} \end{bmatrix} \qquad \boldsymbol{K_{11}} = \boldsymbol{K_{22}} = \frac{\mu}{\rho} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \qquad \boldsymbol{K_{12}} = \boldsymbol{K_{21}} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{6}$$

$$\boldsymbol{C} = \begin{bmatrix} \gamma & -f & 0 \\ f & \gamma & 0 \\ 0 & 0 & 0 \end{bmatrix} \qquad \boldsymbol{S} = \begin{bmatrix} -g\eta\dfrac{\partial z_b}{\partial x} + \dfrac{\tau_{\eta x}}{\rho} \\ -g\eta\dfrac{\partial z_b}{\partial y} + \dfrac{\tau_{\eta y}}{\rho} \\ 0 \end{bmatrix} \tag{7}$$

In this case, $\mu = 1\,mPa$ is the fluid's dynamic viscosity and $\rho = 1000\,kg/m^3$ is its density. Likewise, $g = 9.81\,m/s^2$ is the gravity's acceleration and f is the Coriolis parameter.

Also, $\gamma = C_f\sqrt{u^2 + v^2}$. Here, C_f is the bed friction coefficient that can be defined based on other coefficients, such as the ones known as the Chézy's, Strickler's, Manning's or Darcy-Weisbach's coefficients, represented, respectively, by the letters C, S, n e f. They can be related according to the following expression:

$$C_f = \frac{f}{8} = \frac{g}{C^2} = \frac{g\,n^2}{h^{1/3}} = \frac{g}{h^{1/3}S^2} \tag{8}$$

As an aid to define real values, works such as the one from Arcement et al. (1989) can be queried. Alternatively, the friction coefficient can be defined from the bed roughness length z_0. This property depends on the water's viscosity, the speed of the fluid and the scale of the surface's roughness. For flows of sediment mixtures or over a non-flat surface, Soulsby (1997) presents a table with reference values that can be consulted. Hence, the friction coefficient can be computed as:

$$C_f = \alpha\left(\frac{z_0}{h}\right)^{\beta} \tag{9}$$

where different values for the constants α and β have already been proposed by the available literature. For instance, it can be applied the Manning-Strickler law, which defines $\alpha = 0.0474$ and $\beta = 1/3$.

Besides, the shear stress at the water surface due to wind drag forces can be defined as:

$$\boldsymbol{\tau_\eta} = (\tau_{\eta x},\, \tau_{\eta y}) = \rho_{air} C_d \|\boldsymbol{w}_{10}\| \boldsymbol{w}_{10} \tag{10}$$

where $\rho_{air} = 1.29\,kg/m^3$ and $\boldsymbol{w}_{10}$ is the wind speed $10\,m$ above water level. The definition of its value can be aided by the Beaufort scale (Encyclopædia Britannica, 2017). Regarding the drag coefficient C_d, it can be defined using the relation proposed by Wu (1982):

$$C_d = (0.8 + 0.065\|\boldsymbol{w}_{10}\|) * 10^{-3} \tag{11}$$

Therefore, the problem in question can be wrapped up as, given the closed domain $\bar{\Omega} \in \mathbb{R}^2 \times [t_0, t_1]$, of interior region Ω and boundary $\Gamma = \Gamma_e \cup \Gamma_n$, with $\Gamma_e \cap \Gamma_n = \emptyset$, it is desired to solve the Eq. (2) for $\boldsymbol{U}(\boldsymbol{x}, t)$, subject to the initial condition:

$$\boldsymbol{U}(\boldsymbol{x}, t_0) = \boldsymbol{U}_0(\boldsymbol{x}) \tag{12}$$

and to the essential and natural boundary conditions:

$$\boldsymbol{U} = \boldsymbol{G} \text{ in } \Gamma_e \times]t_0, t_1] \tag{13}$$

$$\boldsymbol{K}\frac{\partial \boldsymbol{U}}{\partial x_k}n_k = (\boldsymbol{K}\boldsymbol{\nabla}\boldsymbol{U})\cdot\boldsymbol{n} = 0 \text{ in } \Gamma_n \times]t_0, t_1] \tag{14}$$

where $\boldsymbol{n}$ is the outward-pointing normal at the boundary. In this last case, the diffusive flux across the boundary Γ_n is null. As a result, all the flow across the boundary is advective. This is a common approach in convective-diffusive physical models.

3 NUMERICAL MODEL

3.1 Variational formulation

In order to do define the variational formulation of the problem, the sets of finite-dimensional trial and test functions are respectively defined as $\mathcal{S}^h = \{\boldsymbol{U^h} \in (H^{1h}(\bar{\Omega}))^3 \mid \boldsymbol{U^h} = \boldsymbol{G} \text{ in } \Gamma_e\}$ and $\mathcal{V}^h = \{\boldsymbol{W^h} \in (H^{1h}(\bar{\Omega}))^3 \mid \boldsymbol{W^h} = \boldsymbol{0} \text{ em } \Gamma_e\}$, where $H^{1h}(\bar{\Omega})$ is the finite-dimensional first order Hilbert space, defined in the closed domain $\bar{\Omega}$. Then, employing the continuous Galerkin method, one can define $\boldsymbol{U^h} = \boldsymbol{V}\boldsymbol{N}$ and $\boldsymbol{W^h} = \boldsymbol{C}\boldsymbol{N}$, where $\boldsymbol{V}$ contains the nodal values of the solution, $\boldsymbol{C}$ defines arbitrary constants and $\boldsymbol{N}$ holds the finite element base (or interpolation) functions. Therefrom, the following stabilized integral formulation is adopted:

$$\int_{\Omega}(\boldsymbol{W^h})^T\left(\frac{\partial \boldsymbol{U^h}}{\partial t} + \boldsymbol{A^h}\cdot\boldsymbol{\nabla}\boldsymbol{U^h} + \boldsymbol{C^h}\boldsymbol{U^h} - \boldsymbol{S^h}\right)d\Omega + \int_{\Omega}(\boldsymbol{\nabla}\boldsymbol{W^h})^T\left(\boldsymbol{K}\boldsymbol{\nabla}\boldsymbol{U^h}\right)d\Omega +$$

$$\sum_{e=1}^{n_{el}}\int_{\Omega^e}\left(\boldsymbol{\tau}\boldsymbol{A^h}\cdot\boldsymbol{\nabla}\boldsymbol{W^h}\right)^T\left[\frac{\partial \boldsymbol{U^h}}{\partial t} + \boldsymbol{A^h}\cdot\boldsymbol{\nabla}\boldsymbol{U^h} - \boldsymbol{\nabla}\cdot\left(\boldsymbol{K}\boldsymbol{\nabla}\boldsymbol{U^h}\right) + \boldsymbol{C^h}\boldsymbol{U^h} - \boldsymbol{S^h}\right]d\Omega^e +$$

$$\sum_{e=1}^{n_{el}}\int_{\Omega^e}\delta(\boldsymbol{\nabla}\boldsymbol{W^h})^T\boldsymbol{\nabla}\boldsymbol{U^h}\,d\Omega^e = 0 \tag{15}$$

where Ω^e, $e = 1, ..., n_{el}$ are the n_{el} elements that comprise the domain Ω, with $\Omega^i \cap \Omega^j = \varnothing$, $\forall\,(i, j)$. Also, the forms $\boldsymbol{A^h}$, $\boldsymbol{C^h}$ and $\boldsymbol{S^h}$ are used to indicate that their respective matrices should be computed based on $\boldsymbol{U^h}$. Plus, $\boldsymbol{\tau}$ is the SUPG (Streamline Upwind Petrov-Galerkin) stabilization matrix and δ is the shock capturing operator's coefficient. In sequence, with a suitable choice of the constants $\boldsymbol{C}$, the dynamic system can be defined:

$$\boldsymbol{M}\boldsymbol{A} + \boldsymbol{D}\boldsymbol{V} = \boldsymbol{F} \tag{16}$$

where $\boldsymbol{A} = \partial\boldsymbol{V}/\partial t$. $\boldsymbol{M}$ and $\boldsymbol{D}$ are the generalized mass and stiffness matrices, and $\boldsymbol{F}$ is a source term. In this case:

$$\boldsymbol{M} = \int_{\Omega^e}\boldsymbol{N}^T\boldsymbol{N}\,d\Omega^e + \int_{\Omega^e}\left(\boldsymbol{\tau}\boldsymbol{A^h}\cdot\boldsymbol{\nabla}\boldsymbol{N}\right)^T\boldsymbol{N}\,d\Omega^e \tag{17}$$

$$\boldsymbol{F} = \int_{\Omega^e}\boldsymbol{N}^T\boldsymbol{S^h}\,d\Omega^e + \int_{\Omega^e}\left(\boldsymbol{\tau}\boldsymbol{A^h}\cdot\boldsymbol{\nabla}\boldsymbol{N}\right)^T\boldsymbol{S^h}\,d\Omega^e \tag{18}$$

$$\begin{aligned}
\boldsymbol{D} &= \int_{\Omega^e} \left(\boldsymbol{N}^T \boldsymbol{A^h} \cdot \boldsymbol{\nabla N} + (\boldsymbol{\nabla N})^T \boldsymbol{K} \boldsymbol{\nabla N} + \boldsymbol{N}^T \boldsymbol{C^h} \boldsymbol{N}\right) d\Omega^e \\
&+ \int_{\Omega^e} \left(\boldsymbol{\tau} \boldsymbol{A^h} \cdot \boldsymbol{\nabla N}\right)^T \left(\boldsymbol{A^h} \cdot \boldsymbol{\nabla N} + \boldsymbol{\nabla N})^T \boldsymbol{K} \boldsymbol{\nabla N} + \boldsymbol{C^h} \boldsymbol{N}\right) d\Omega^e \\
&+ \int_{\Omega^e} \delta (\boldsymbol{\nabla N})^T \boldsymbol{\nabla N} \, d\Omega^e
\end{aligned} \tag{19}$$

It should be noted that $\boldsymbol{D}$ and $\boldsymbol{F}$ depend on $\boldsymbol{V}$, which confers non-linearity to the system. Also, it remains to define the matrix $\boldsymbol{\tau}$ and the scalar δ.

3.2 SUPG operators

Here, two forms of the SUPG (Streamline Upwind Petrov-Galerkin) matrix are defined. The first one was initially introduced by Hughes et al. (1986). In this work, it is used the form:

$$\boldsymbol{\tau} = (\boldsymbol{B}^T \boldsymbol{B})^{-1/2} \tag{20}$$

where $\boldsymbol{B}^T = \boldsymbol{A^T} \boldsymbol{J}^{-1}$ is equivalent to the advection matrix $\boldsymbol{A}$ in the canonical referential. Here, $\boldsymbol{J}$ is the Jacobian matrix of the transformation from the real world referential to the canonical referential.

The other SUPG operator employed is the one proposed by Tezduyar (2004) and adapted by Takase et al. (2010) to the shallow water equations:

$$\boldsymbol{\tau} = \left(\frac{1}{(\tau_{SUGN1})^2} + \frac{1}{(\tau_{SUGN2})^2}\right)^{-1/2} \tag{21}$$

where:

$$\tau_{SUGN1} = \left(\sum_{a=1}^{n_{npe}} c|\boldsymbol{j} \cdot \nabla N_a| + |\boldsymbol{u} \cdot \nabla N_a|\right)^{-1} \qquad \text{and} \qquad \tau_{SUGN2} = \frac{\Delta t}{2} \tag{22}$$

Also $n_{npe} = 4$ is the number of nodes per element, $c = \sqrt{gh}$ is the propagation speed of a perturbation at the surface, $\boldsymbol{u}$ is the depth averaged velocity and $\boldsymbol{j} = \nabla\eta/||\nabla\eta||$ is the normalized gradient of the free surface elevation.

3.3 Shock capturing operators

In the case of shock capturing operators, three different types were employed, the first being the one proposed by Rispoli et al. (2006) and later employed with the variable subgrid scale (V-SGS) by Rispoli et al. (2007). Its coefficient is defined as:

$$\delta = \delta_{91-MOD} = max(0, \delta_{91} - \delta_\tau) \tag{23}$$

with:

$$\delta_{91} = \frac{\left\|\boldsymbol{A}_k^h \dfrac{\partial \boldsymbol{U}^h}{\partial x_k}\right\|_{\widetilde{\boldsymbol{A_0}}^{-1}}}{\left(\sum_{j=1}^2 \left\|\dfrac{\partial \xi_j}{\partial x_k} \dfrac{\partial \boldsymbol{U}^h}{\partial x_k}\right\|^2_{\widetilde{\boldsymbol{A_0}}^{-1}}\right)^{1/2}} \qquad \delta_\tau = \frac{\left\|\boldsymbol{A}_k^h \dfrac{\partial \boldsymbol{U}^h}{\partial x_k}\right\|_{\widetilde{\boldsymbol{A_0}}^{-1}\boldsymbol{\tau}}}{\left\|\dfrac{\partial \boldsymbol{U}^h}{\partial x_k}\right\|^2_{\widetilde{\boldsymbol{A_0}}^{-1}}} \tag{24}$$

where ξ_i, $i = 1, 2$ are the canonical reference coordinates. For a vector $\boldsymbol{v}$ and a matrix $\boldsymbol{M}$, it is employed the norm $||\boldsymbol{v}||_{\boldsymbol{M}} = (\boldsymbol{v}^T \boldsymbol{M} \boldsymbol{v})^{1/2}$. Also, $\boldsymbol{\tau}$ is the SUPG matrix and $\widetilde{\boldsymbol{A_0}}^{-1}$ is the inverse Jacobian of the transformation from entropy to conservation variables.

In this regard, for the current physical model, the total energy is given by:

$$E = \frac{gh^2 + u^2 h + v^2 h}{2} \tag{25}$$

from which the entropy variables $\boldsymbol{V}$ and the matrix $\widetilde{\boldsymbol{A_0}}^{-1}$ can defined (Tadmor et al., 2008):

$$\boldsymbol{V} = \frac{\partial E}{\partial \boldsymbol{U}} = \begin{bmatrix} u \\ v \\ gh - \dfrac{u^2 + v^2}{2} \end{bmatrix} \qquad \widetilde{\boldsymbol{A_0}}^{-1} = \frac{\partial \boldsymbol{V}}{\partial \boldsymbol{U}} = \frac{1}{h} \begin{bmatrix} 1 & 0 & -u \\ 0 & 1 & -v \\ -u & -v & gh + u^2 + v^2 \end{bmatrix} \tag{26}$$

where, from a simples analysis, it can be seen that $\widetilde{\boldsymbol{A_0}}^{-1}$ is symmetric positive definite if $h > 0$, which should be true by definition.

Another shock capturing operator adopted is the $YZ\beta$ (Rispoli et al., 2007), which defines its coefficient as:

$$\delta = \frac{\delta_{\beta=1} + \delta_{\beta=2}}{2} \tag{27}$$

with:

$$\delta_\beta = ||\boldsymbol{Y}^{-1}\boldsymbol{Z}|| \left(\sum_{i=1}^{2} \left\| \boldsymbol{Y}^{-1} \frac{\partial \boldsymbol{U}^h}{\partial x_i} \right\|^2 \right)^{\beta/2-1} \|\boldsymbol{Y}^{-1}\boldsymbol{U}^h\|^{1-\beta} \left(\frac{h_{shoc}}{2} \right)^\beta \tag{28}$$

where:

$$\boldsymbol{Y} = \begin{bmatrix} (q_x)_{ref} & 0 & 0 \\ 0 & (q_y)_{ref} & 0 \\ 0 & 0 & (\eta)_{ref} \end{bmatrix} \qquad \boldsymbol{Z} = \boldsymbol{A_i}^h \frac{\partial \boldsymbol{U}^h}{\partial x_i} \tag{29}$$

$$h_{shoc} = 2 \left(\sum_{a=1}^{n_{npe}} |\boldsymbol{j} \cdot N_a| \right)^{-1} \qquad \boldsymbol{j} = \frac{\nabla \eta}{\|\nabla \eta\|} \tag{30}$$

In this case, $(q_x)_{ref}$, $(q_y)_{ref}$, $(\eta)_{ref}$ are reference values for the variables q_x, q_y and η. Alternatively, it could be used $\delta = \delta_{\beta=1}$ or $\delta = \delta_{\beta=2}$, for smoother or sharper shocks, respectively.

The last shock operator used was proposed by Takase et al. (2010):

$$\delta = \delta_{SUGN1} \left(\frac{|\nabla^2 \eta^h|}{max|\nabla^2 \eta^h|} \right) \tag{31}$$

where the maximum value of $|\nabla^2 \eta^h|$ over all the elements acts as an scaling factor.

3.4 Dry/wet interface handling

Another aspect concerning numerical stabilization procedures is the correct handling of the interface between wet and dry regions. Due to the way the domain and source terms are discretized, spurious oscillations often incur at shorelines. These result in unrealistic flow behavior and can even make the fluid height become negative.

On this account, for this paper, a plain solution is employed. When the inherent linear system is being assembled, the fluid height is computed as

$$h = max(\epsilon, |\eta - z_b|) \tag{32}$$

where $\epsilon = 0.01\, m$ is the wet cell classification threshold. This is part of the solution adopted by Heniche et al. (2000). Also, the previously defined SUPG and shock capturing operators, as well as all the source terms are zeroed when evaluating a dry quadrature point's contribution. Even though this is approach works for the cases tested, a more robust method still should be implemented, specially if beds with variable bathymetries are to be used.

3.5 Time evolution

The goal here is to define the way it is performed the evolution of the variables governed by the nonlinear system described by the Eq. (16) over the time range $[t_0, t_1]$, discretized in n steps of size Δt. In this context, a semi-discrete approach is adopted in which the space is treated in a discrete manner and the time continuously. That is, the space is discretized using finite elements and time dependence is accounted through the integration of the nodal values $\boldsymbol{V}$ over time. To this end, the integrals in Eq. (16) are computed with the 4-point Gauss quadrature rule.

Accordingly, it is adopted the predictor multi-corrector algorithm introduced by Aliabadi et al. (1995), which can be summarized in the steps depicted in the Fig. (2). Here, $\alpha = 0.5$ is a parameter that controls the stability and precision of the method. For the chosen value, the method is unconditionally stable for linear problems and has a quadratic convergence rate over time. An analysis of the stability and precision of such algorithm can be checked at the work of Hughes et al. (1984).

- Prediction phase:
  - $\boldsymbol{A}_{n+1}^{(0)} = \boldsymbol{0}$
  - $\boldsymbol{V}_{n+1}^{(0)} = \boldsymbol{V}_n + (1-\alpha)\Delta t \boldsymbol{A}_n$
- Correction phase:
  - For $i = 0, 1, 2, ...$ until the convergence criteria is met:
    - $\boldsymbol{R}^{(i)} = \boldsymbol{F}_{n+1}^{(i)} - (\boldsymbol{M}\boldsymbol{A}_{n+1}^{(i)} + \boldsymbol{D}_{n+1}^{(i)}\boldsymbol{V}_{n+1}^{(i)})$
    - $\boldsymbol{M}^* = \boldsymbol{M} + \alpha \Delta t \boldsymbol{D}_{n+1}^{(i)}$
    - Solve $\boldsymbol{M}^*(\Delta \boldsymbol{A}^{(i)}) = \boldsymbol{R}^{(i)}$
    - $\boldsymbol{A}_{n+1}^{(i+1)} = \boldsymbol{A}_{n+1}^{(i)} + \Delta \boldsymbol{A}^{(i)}$
    - $\boldsymbol{V}_{n+1}^{(i+1)} = \boldsymbol{V}_{n+1}^{(i)} + \alpha \Delta t \Delta \boldsymbol{A}^{(i)}$

Figure 2: Pseudocode for the predictor multi-corrector algorithm employed for time integration.

As a convergence criteria, it is used one based on the system residual (Bathe et al., 1980), so that the iterations should stop when:

$$\left\|\boldsymbol{R}^{(i)}\right\| \leq \epsilon_f \left\|\boldsymbol{R}^{(0)}\right\| \tag{33}$$

In this work, it was chosen $\epsilon_f = 0.001$. Additionally, to avoid letting the algorithm run indefinitely, the number of iterations is limited at 100.

The linear system is solved with the iterative GMRes (Generalized Minimal Residual), using a Krylov space of dimension 35. Also, it is applied a SSOR (Symmetric Successive Over-Relaxation) preconditioner with a relaxation parameter of 0.6. The maximum number of iterations for the linear system is fixed at 600 and the convergence threshold is set to 10^{-11} times the initial linear system residual. Another optimization of the linear system is that the degree of freedoms are reordered using the algorithm of Cuthill et al. (1969) to aid the linear solver convergence.

3.6 Adaptive time stepping

For the current model, the CFL number of an element e can be defined as:

$$CFL_e = \left(|\boldsymbol{u_e}| + \sqrt{gh_e}\right)\frac{\Delta t}{l_e} \tag{34}$$

where l_e is the element characteristic length, defined here as the square root of its area. In this case, the speeds $|\boldsymbol{u}_e|$ and $\sqrt{gh_e}$ are evaluated at the element's barycenter.

In order to impose a maximum CFL number on the solution, it is possible to choose a suitable time step:

$$\Delta t = CFL\, min_e \left(\frac{l_e}{|\boldsymbol{u_e}| + \sqrt{gh_e}}\right) \tag{35}$$

Thereby, for the simulations employed in this work with adaptive time steps, it was chosen a CFL number of 0.8.

3.7 Adaptive mesh

Besides adaptive time steps, the mesh itself can be used in an adaptive fashion. This is useful because it makes possible to distribute the computational effort among different areas according to their needs. In effect, stable (or even still) areas can afford to use coarser cells and unstable regions might need to use more refined cells.

In order to refine or coarse the mesh, it necessary to define an error metric per cell to condition the operation. For that matter, it is employed the estimator proposed by Kelly et al. (1983) over the nodal values. Then, the cell with the biggest errors that together compose 80% of the total error are refined. Also, the cells with the smallest errors that combined represent 10% of the total error are coarsened. To avoid that the mesh becomes too refined or coarse at any point, it is limited the minimum refinement level to the initial mesh state and the maximum to the equivalent of 3 subdivisions. Also, 3 mesh refinement-coarsening steps are performed before each time step so that a cell could go from the coarsest to the most refined state (and vice-versa) on the same step. For illustration, a refined mesh can be seen on Fig. (3).

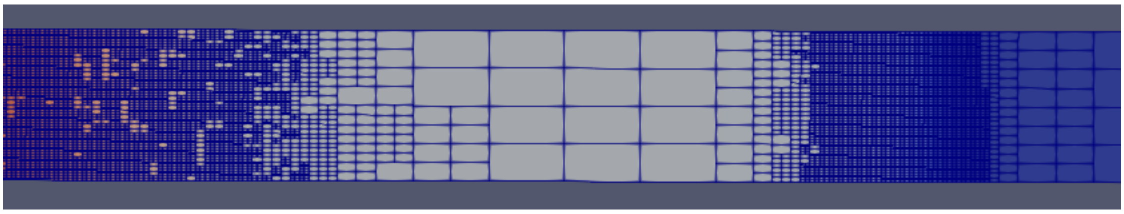

Figure 3: Detail of an adaptively refined mesh.

4 IMPLEMENTATION

All this paper's work was compiled into a c++ library with a well defined API (Application Program Interface) that allows the user to provide input and obtain results in a way that the simulation itself is transparent. One of its main requirements is that it must be multi-platform, being compatible with Windows and Unix-based systems.

That being said, it was employed the CMake (CMake, 2016) tool to configure the compilation project. One of the main dependencies of the library is the deal.II (Bangerth et al., 2007) finite element library, which is the base for all the solver's finite element implementation. One of the deciding factors for its use is it being multiplatform. Others dependencies include Eigen (Guennebaud et al., 2016), Qt (Company, 2016) and Boost (Boost, 2016). Also, Doxygen (Doxygen, 2016) is employed to generate code documentation while unit tests based on Qt's QTest API can be executed with the CMake's CTest tool.

5 RESULTS

Here numerical results of some test cases are presented and comparisons are made with analytical or literature available solutions. For each case, the input mesh's generation was performed with the Gmsh (Geuzaine et al., 2009) tool and results were brought to the ParaView (Ahrens et al., 2005) application for analysis.

5.1 1D dam break

The first problem solved is the 1D dam break configuration which describes two reservoirs that contain the same fluid and are separated by a dam. In one reservoir, the fluid has depth h_1 and, in the other, h_0, with $h_1 > h_0$. At a given time, the barrier is instantly removed without causing any impact to the fluid. Thus, it is sought to calculate the flow of the fluid afterwards.

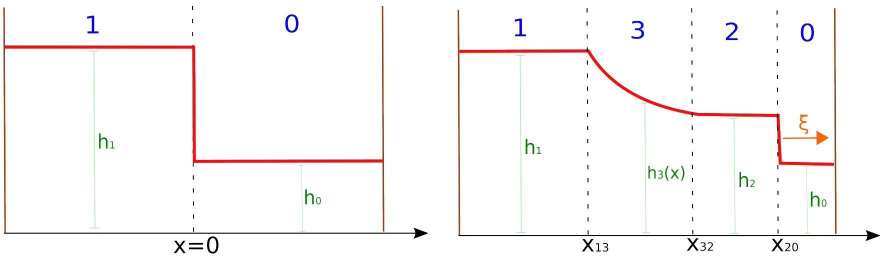

Figure 4: Schematic representation of the problem and its analytical solution.

For this disposition, it is possible to obtain an analytical solution for the state of the fluid after the release. In fact, Stoker (1992) presented the solution adopted in his work. A schematic representation of the solution and the initial state of the problem can be seen in Fig. (4).

The analytical solution can be divided into 4 regions. In the transition between regions 2 and 0, a shock wave propagates to the right with speed ξ. Also, the fluid has zero velocity in regions 1 and 0 ($u_1 = u_0 = 0$). Let the propagation speed of a perturbation on the surface of each region be defined as $c_i = \sqrt{gh_i}$, with $i = 0, 1, 2, 3$. Then, the shock wave speed is obtained by solving the nonlinear equation:

$$\frac{\xi}{2} + \frac{c_0^2}{8\xi}\left[1 + \sqrt{1 + 8\left(\frac{\xi}{c_0}\right)^2}\right] + \left[\frac{c_0^2}{2}\left(\sqrt{1 + 8\left(\frac{\xi}{c_0}\right)^2} - 1\right)\right]^{1/2} = c_1 \tag{36}$$

Since ξ is constant for a given initial configuration, an Octave (Eaton et al., 2015) script was written to solve the above equation, so that ξ can be computed and used as an input to the simulator.

In sequence, it is defined:

$$c_2 = c_0\left[\frac{1}{2}\sqrt{1 + 8\left(\frac{\xi}{c_0}\right)^2} - \frac{1}{2}\right]^{1/2} \qquad c_3 = \frac{1}{3}\left(2c_1 - \frac{x}{t}\right) \tag{37}$$

$$u_2 = \xi - \frac{c_0^2}{4\xi}\left(1 + \sqrt{1 + 8\left(\frac{\xi}{c_0}\right)^2}\right) \qquad u_3(x) = \frac{2}{3}\left(c_1 + \frac{x}{t}\right) \tag{38}$$

where t is the elapsed time since the dam removal. So, the coordinates of each region transition can be defined as:

$$x_{13} = -c_1 t \qquad x_{32} = (u_2 - c_2)t \qquad x_{20} = \xi t \tag{39}$$

For the cases simulated, $t \in [0, 10]s$, $\Delta t = 0.1s$ (when constant), $h_1 = 2\,m$, $h_0 = 1\,m$, $x \in [-50, 50]\,m$. Coriolis forces are ignored and a flat, frictionless horizontal bottom is adopted. The stress at the air-water interface is also not considered. As Fennema et al. (1990) mentions, these assumptions prevent the source term from introducing additional damping to the result. Also, no-penetration boundary conditions are applied at the limits of the numerical domain.

#	SUPG	Shock capturing
1	Hughes	δ_{91-MOD}
2		$YZ\beta$ (*Smooth*)
3		$YZ\beta$ (*Sharp*)
4		$YZ\beta$
5		Takase

#	SUPG	Shock capturing
6	Takase	δ_{91-MOD}
7		$YZ\beta$ (*Smooth*)
8		$YZ\beta$ (*Sharp*)
9		$YZ\beta$
10		Takase

Table 1: Combinations of SUPG and shock capturing operators for each case run.

To evaluate the SUPG and shock capturing operators presented, this problem was solved with the combinations listed in Table (1). A comparison of the obtained results can be seen in Fig. (5). For these cases, it was used a regular mesh of 4×302 elements. Also, the reference values required by the $YZ\beta$'s coefficient were set equal to the exact solution inside the region 2. A similar approach is also applied to the other dam break examples, where it is made a parallel with the current problem using the appropriate fluid depths.

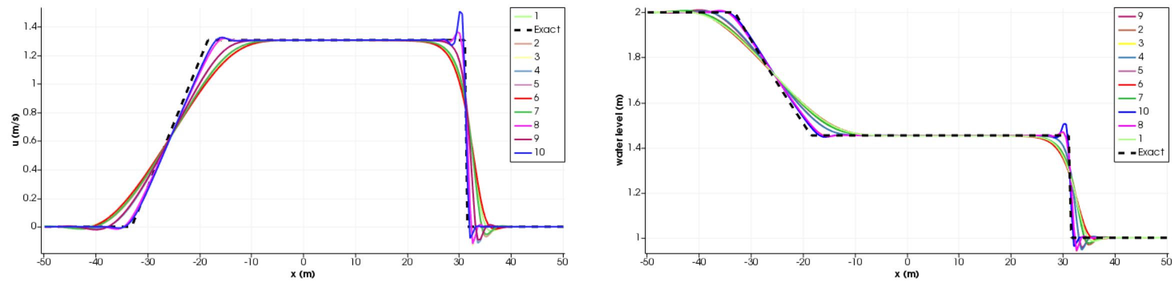

Figure 5: Resulting velocity and water level profiles at $t = 7.5\,s$ of all SUPG and shock operators combinations.

Taking the combination 6, with the smoothest solution profiles, and 10, with the sharpest ones, Figure (6) presents a comparison between them and their solutions with adaptive time steps $6a$ and $10a$. Little improvement can be perceived. However, for the smoothest case, adaptively refining and coarsening the mesh at $6b$ results in a near exact solution. Therein, the initial mesh is the same used in the previous cases.

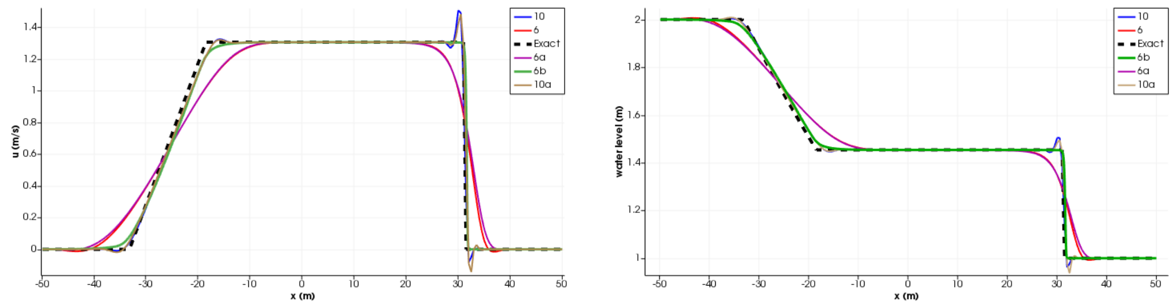

Figure 6: Comparison between solutions 6 and 7, their time adaptive counterparts $6a$ and $10a$, and one with adaptive mesh $6b$.

Further analyzing the combination 6, Figure (7) presents results for different mesh resolutions. It can be seen that as the mesh density increases, the result approaches the exact solution. Due to the quality of the outcomes of the combination 6, the Hughes' SUPG operator and the δ_{91-MOD} shock capturing operator are also employed for the remaining examples.

To be complete, it is also considered the less generic case where $h_0 = 0$. In fact, it contemplates a flooding event (or dry-wet transition), which can be numerically difficult to handle. Thus, Figure (8) presents the profiles of the initial state and the analytical solution. In this case, these relations are valid:

$$u_1 = 0 \qquad c_1 = 0 \qquad x_{10} = -c_1 t \qquad x_0 = 2c_1 t \tag{40}$$

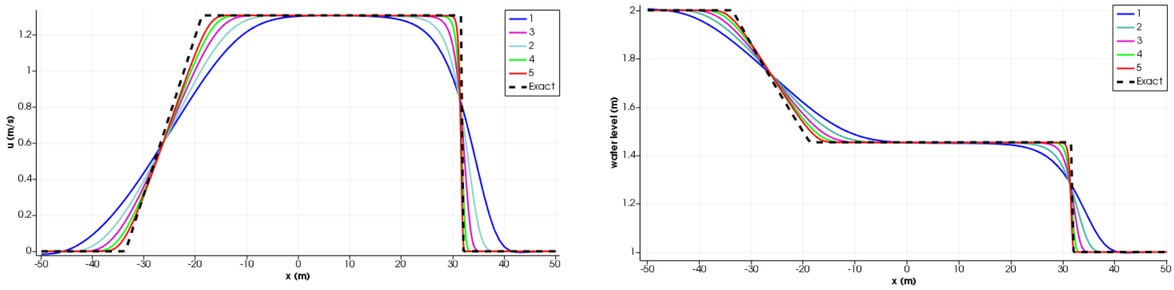

Figure 7: Results for different mesh resolutions for the combination 6**. The number of elements in each case is:** $1-404$**,** $2-1208$**,** $3-4016$**,** $4-14432$**,** $5-54464$**.**

$$c_0 = \frac{1}{3}\left(2c_1 - \frac{x}{t}\right) \qquad u_0 = \frac{2}{3}\left(\frac{x}{t} + c_1\right) \tag{41}$$

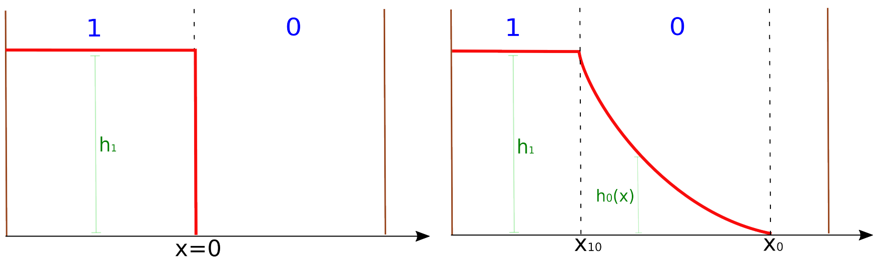

Figure 8: Schematic representation of the problem and its analytical solution.

Obtained results can be seen in Fig. (9). Some oscillations are present at the velocity profile but are greatly reduced with the use of an adaptive time step.

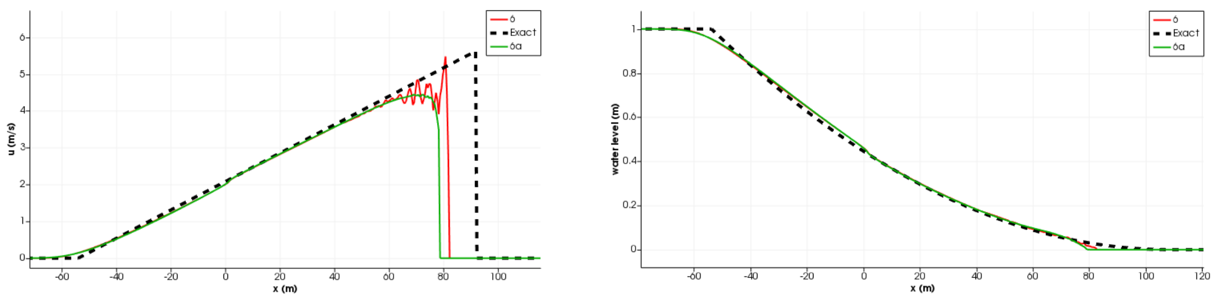

Figure 9: Resulting velocity and water level profiles at $t = 17\,s$**.**

5.2 2D dam break

The natural sequence of the previous problem is its bidimensional case. To simulate it, it was used a square domain with minimum (x, y) position $(0, 0)$ and maximum $(100, 100)$ in which the initial water depth is $h = h_1 = 2$ for $x < 50$ and $y > 50$, and $h = h_0 = 1$ otherwise. Also, it was employed an irregular mesh with 13128 elements. Figures (10) and (11) depict the

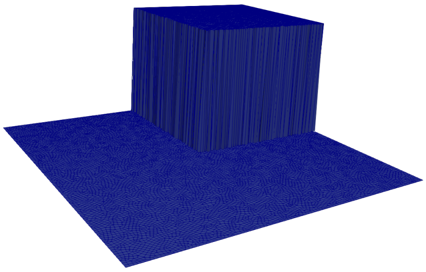

Figure 10: Water surface elevation at $t = 0\,s$.

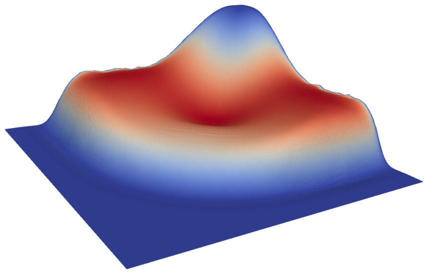

Figure 11: Water surface elevation at $t = 7.5\,s$.

Figure 12: State of the solution presented by Ribeiro et al. (2001) at $t = 7.5\,s$.

initial configuration and the solution at $t = 7.5\ s$, and Fig. (12) shows the solution obtained by Ribeiro et al. (2001) in the same instant.

As a variation of this example, it can be considered the situation where the region with the greatest height is located at the domain's center. So it is defined that the region with depth h_1 is located in a domain centered square with sides of $25\,m$. With this configuration, some simulation steps over a regular mesh of $10k$ elements can be seen on Figs. (13), (14), (15) and (16).

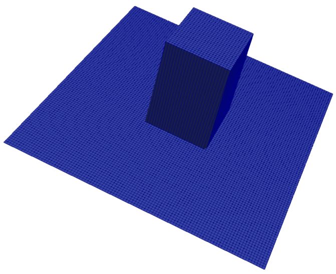

Figure 13: Water surface elevation at $t = 0\,s$.

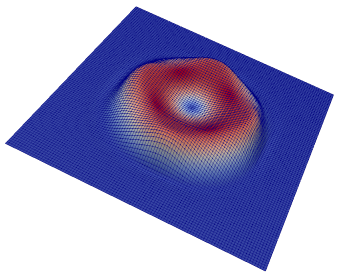

Figure 14: Water surface elevation at $t = 3.5\,s$.

Yet another variation is when the reservoir with the highest water surface level is shaped like a circular section centered at one of the domain‘s corners. In the simulated case, its radius is 40 m. The initial state and the solution at $t = 7.5\,s$ on the $10k$ elements mesh can be seen on Figs. (17) and (18).

5.3 2D dam break through channel

Another problem that can be used to validate the model is the two-dimensional dam break through a channel. The difference from this to the previous one is that the flow between the

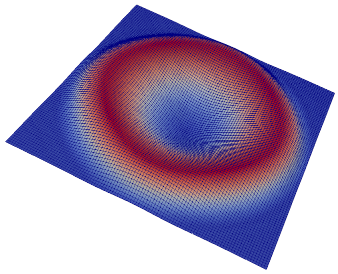

Figure 15: Water surface elevation at $t = 7.5\,s$.

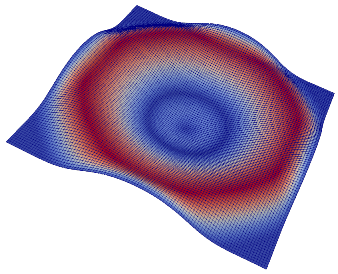

Figure 16: Water surface elevation at $t = 10\,s$.

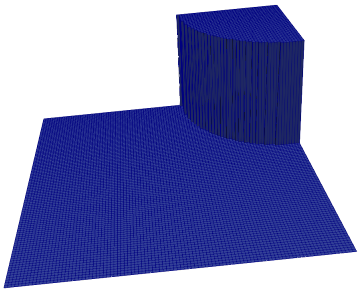

Figure 17: Initial water surface elevation.

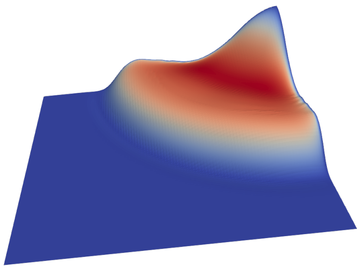

Figure 18: Water surface elevation at $t = 7.5\,s$.

reservoirs is carried out through a channel. This confers different characteristics to the solution, especially when the dimensions and location of the channel are asymmetric.

In this context, a diagram of the domain's geometry and the initial fluid height distribution can be seen in Fig. (19). Similarly, the starting state of the water level over the 1932 elements regular mesh is depicted in the Fig. (20). Two simulation results can be seen in Figs. (21) and (22). For this case, instead of the δ_{91-MOD} operator, it was employed the $YZ\beta$ and its variant with sharper shocks. The latter presented results more similar to the solutions of Fennema et al. (1990) (see Fig. (23)) and Lencina (2007) (see Fig. (24)).

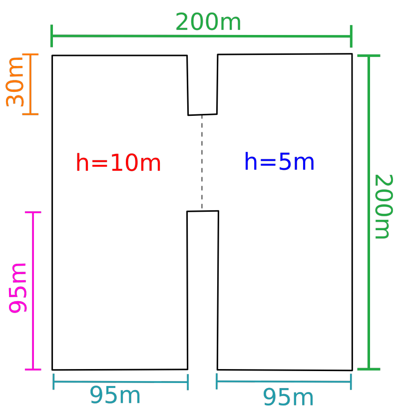

Figure 19: Representation of the domain's geometry and the initial fluid height distribution.

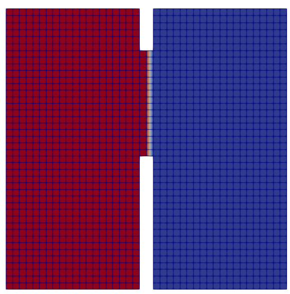

Figure 20: Representation of the domain's mesh colored according to the initial water level.

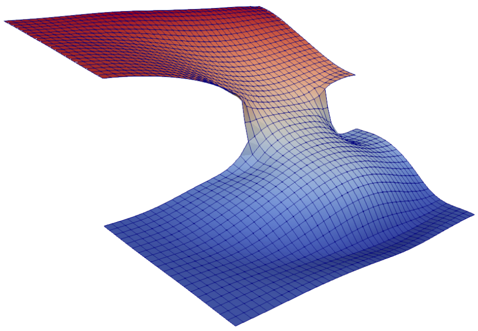

Figure 21: Water surface elevation at $t = 6.0\,s$. Simulated with the $YZ\beta$ operator.

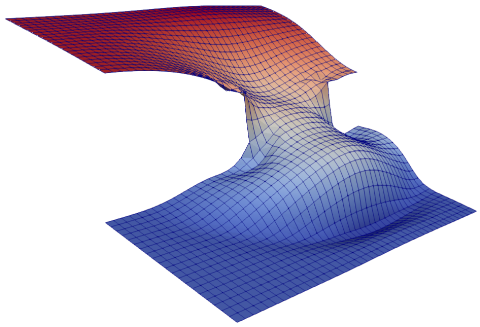

Figure 22: Water surface elevation at $t = 6.0\,s$. Simulated with the $YZ\beta$ operator's sharper variant.

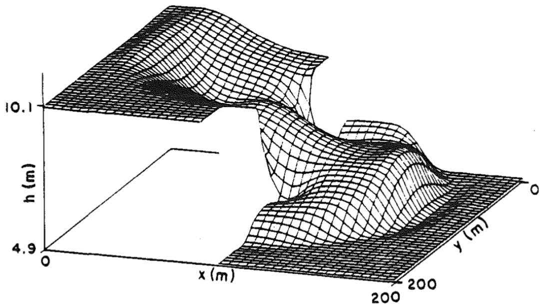

Figure 23: Result presented by Fennema et al. (1990) at $t = 6.0\,s$.

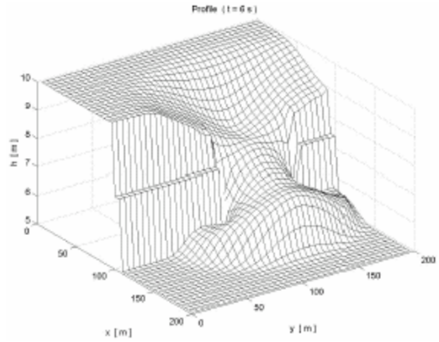

Figure 24: Result presented by Lencina (2007) at $t = 6.0\,s$.

5.4 Steady flow over a bump

To verify the well-balanced property (or the C-property (Bermudez et al., 1994)) of the adopted model, two lake-at-rest configurations were simulated. Both consider a $1 \times 1\,m$ domain composed of 3357 elements and whose bed elevation is given by:

$$z_b(x, y) = max(0.0,\ 0.25 - 5.0 * ((x - 0.5)^2 + (y - 0.5)^2)) \tag{42}$$

which defines a bump in the region's center.

The initial water surface elevation on the first case is $\eta = 0.5$, defining a fully wet region. On the second, $\eta = 0.1$ defines a partially dry bump. In these cases, if the conservation property is not held, spurious velocities would appear and, in the case with a partially dry bump, it would be covered, as demonstrated by Brufau et al. (2003). All in all, after several simulation steps, $||\boldsymbol{u}||$ remains correctly within the order of 10^{-15} and the water surface level remains unchanged, as can be seen in Figs. (25) and (26). In these pictures, the bottom surface represents the ground and the top denotes the water level.

6 CONCLUSION AND FUTURE WORKS

A well-balanced shallow water flow model was presented, together with a continuous finite-element approach to numerically solve its governing equations. Also, numerical stabilization techniques were employed. In this regard, several SUPG and shock capturing operators were

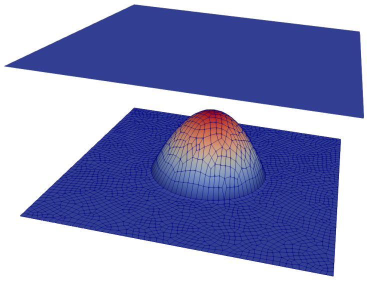

Figure 25: Result of a lake-at-rest simulation within a fully wet region.

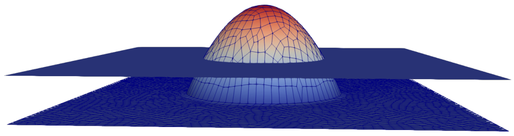

Figure 26: Result of a lake-at-rest simulation with a partially dry bed bump.

evaluated and a plain drying/wetting technique was adopted to prevent unrealistic velocities at shorelines.

Among the combinations of operators tested, the one with Takase's SUPG and δ_{91-MOD} shock operators seemed to provide better results. However, for a more irregular case such as the dam break through a channel, the $YZ\beta$ approach was preferred.

Additionally, adaptive time stepping and mesh refinement/coarsening was adopted. The former presented some improvement in the cases it was used. Nonetheless, when combined with mesh adaptivity, the quality of the result was greatly enhanced. This approach turned out to be more computationally efficient than simply using more refined input meshes. In turn, this case was shown to approximate the exact solution as the number of elements increases.

As for future works, one suggestion is to derive and implement the two-layer shallow water model, where the hydrostatic hypothesis is applied within each layer. This type of model can be employed to simulate flows with different densities(Castro et al., 2007) and tsunamis generated by landslides(Ostapenko, 1999). In the latter case, one bottom layer refers to the landslide, considered as an incompressible liquid, and the other is the sea water. One known issue with this kind of model is that the system is conditionally hyperbolic due to the coupling terms between the layers (Abgrall et al., 2009; Kim et al., 2008; Petcu et al., 2013). In fact, if the velocities of the two layers are too different, it is expected that Kelvin-Helmholtz instabilities arise.

Another related issue is that dry/wet interface instabilities will happen not only on shorelines but also inside the flow region. In this case, the need for a more robust drying/wetting method becomes more critic. Hence, solutions such as the ones presented by Balzano (1998) Vacondio et al. (2017), Defina (2000) or Kärnä et al. (2011) could be taken advantage of.

REFERENCES

Abgrall, Rémi and Smadar Karni. 2009. "Two-Layer Shallow Water System: A Relaxation Approach". In: *SIAM Journal on Scientific Computing* 31.3, pp. 1603–1627.

Ahrens, James, Berk Geveci, and Charles Law. 2005. "36 - ParaView: An End-User Tool for Large-Data Visualization". In: *Visualization Handbook*. Ed. by Charles D. HansenChris R. Johnson. Burlington: Butterworth-Heinemann, pp. 717 –LXXII.

Aliabadi, S. K. and T. E. Tezduyar. 1995. "Parallel fluid dynamics computations in aerospace applications". In: *International Journal for Numerical Methods in Fluids* 21.10, pp. 783–805.

Ambati, V.R. and O. Bokhove. 2007. “Space–time discontinuous Galerkin finite element method for shallow water flows”. In: *Journal of Computational and Applied Mathematics* 204.2. Special Issue: The Seventh International Conference on Mathematical and Numerical Aspects of Waves (WAVES’05), pp. 452–462.

Arcement, George J and Verne R Schneider. 1989. *Guide for selecting Manning’s roughness coefficients for natural channels and flood plains*. US Government Printing Office Washington, DC, USA.

Balzano, Andrea. 1998. “Evaluation of methods for numerical simulation of wetting and drying in shallow water flow models”. In: *Coastal Engineering* 34.1, pp. 83–107.

Bangerth, W., R. Hartmann, and G. Kanschat. 2007. “deal.II – a General Purpose Object Oriented Finite Element Library”. In: *ACM Trans. Math. Softw.* 33.4, pp. 24/1–24/27.

Bathe, Klaus Jürgen and Arthur P. Cimento. 1980. “Some practical procedures for the solution of nonlinear finite element equations”. In: *Computer Methods in Applied Mechanics and Engineering* 22.1, pp. 59–85.

Bermudez, Alfredo and Ma Elena Vazquez. 1994. “Upwind methods for hyperbolic conservation laws with source terms”. In: *Computers & Fluids* 23.8, pp. 1049–1071.

Boost. 2016. *Boost C++ Libraries*. `http://www.boost.org/`.

Brufau, P. and P. García-Navarro. 2003. “Unsteady free surface flow simulation over complex topography with a multidimensional upwind technique”. In: *Journal of Computational Physics* 186.2, pp. 503 –526.

Castro, Manuel J. et al. 2007. “Improved FVM for two-layer shallow-water models: Application to the Strait of Gibraltar”. In: *Advances in Engineering Software* 38.6. Advances in Numerical Methods for Environmental Engineering, pp. 386–398.

CMake. 2016. *CMake*. `https://cmake.org/`.

Company, Qt. 2016. *Qt || Cross-platform software development for embedded & desktop*. `http://www.qt.io/`.

Cuthill, E. and J. McKee. 1969. “reducing the bandwidth of sparse symmetric matrices”. In: *Proceedings of the 1969 24th National Conference*. ACM ’69. New York, NY, USA: ACM, pp. 157–172.

Defina, Andrea. 2000. “Two-dimensional shallow flow equations for partially dry areas”. In: *Water Resources Research* 36.11, pp. 3251–3264.

Doxygen. 2016. *Doxygen : Main Page*. `http://www.doxygen.org`.

Eaton, John W. et al. 2015. *GNU Octave version 4.0.0 manual: a high-level interactive language for numerical computations*.

Encyclopædia Britannica, inc. 2017. *Beaufort scale*.

Fennema, Robert J. and M. Hanif Chaudhry. 1990. “Explicit methods for 2-D transient free surface flows”. In: *Journal of Hydraulic Engineering* 116.8, pp. 1013–1034.

Gandham, Rajesh, David Medina, and Timothy Warburton. 2015. "GPU Accelerated Discontinuous Galerkin Methods for Shallow Water Equations". In: *Communications in Computational Physics* 18.1, 37–64.

Geuzaine, Christophe and Jean-François Remacle. 2009. "Gmsh: A 3-D finite element mesh generator with built-in pre- and post-processing facilities". In: *International Journal for Numerical Methods in Engineering* 79.11, pp. 1309–1331.

Giraldo, F.X. 2014. "Continuous and discontinuous Galerkin methods for atmospheric modeling". In: *Seminar on Recent Developments in Numerical Methods for Atmosphere and Ocean Modelling, 2-5 September 2013*. ECMWF. Shinfield Park, Reading: ECMWF, pp. 167–181.

Groenenberg, Remco M., Kees Sloff, and Gert Jan Weltje. 2009. "A high-resolution 2-DH numerical scheme for process-based modeling of 3-D turbidite fan stratigraphy". In: *Computers & Geosciences* 35.8, pp. 1686–1700.

Guennebaud, Gaël, Benoît Jacob, et al. 2016. *Eigen v3*. `http://eigen.tuxfamily.org`.

Heniche, Mourad et al. 2000. "A two-dimensional finite element drying-wetting shallow water model for rivers and estuaries". In: *Advances in Water Resources* 23.4, pp. 359 –372.

Hou, Jingming et al. 2015. "An efficient unstructured MUSCL scheme for solving the 2D shallow water equations". In: *Environ. Model. Softw.* 66.C, pp. 131–152.

Hughes, Thomas J.R. and Michel Mallet. 1986. "A new finite element formulation for computational fluid dynamics: III. The generalized streamline operator for multidimensional advective-diffusive systems". In: *Computer Methods in Applied Mechanics and Engineering* 58.3, pp. 305–328.

Hughes, T.J.R. and T.E. Tezduyar. 1984. "Finite element methods for first-order hyperbolic systems with particular emphasis on the compressible euler equations". In: *Computer Methods in Applied Mechanics and Engineering* 45.1–3, pp. 217–284.

Karel'skii, K. V., A. S. Petrosyan, and A. V. Chernyak. 2013. "Nonlinear theory of the compressible gas flows over a nonuniform boundary in the gravitational field in the shallow-water approximation". In: *Journal of Experimental and Theoretical Physics* 116.4, pp. 680–697.

Kelly, D. W. et al. 1983. "A posteriori error analysis and adaptive processes in the finite element method: Part I–Error Analysis". In: *Int. J. Num. Meth. Engrg.* 19, pp. 1593–1619.

Kim, Jihwan and Randall J LeVeque. 2008. "Two-layer shallow water system and its applications". In: *Proceedings of the Twelth International Conference on Hyperbolic Problems, Maryland*.

Klimachkov, D. A. and A. S. Petrosyan. 2016. "Nonlinear theory of magnetohydrodynamic flows of a compressible fluid in the shallow water approximation". In: *Journal of experimental and theoretical physics* 123.3, pp. 520–539.

Kärnä, Tuomas et al. 2011. "A fully implicit wetting–drying method for DG-FEM shallow water models, with an application to the Scheldt Estuary". In: *Computer Methods in Applied Mechanics and Engineering* 200.5, pp. 509 –524.

Lencina, Ignacio Valcárcel. 2007. "Comparison between 1D and 2D models to analyze the dam break wave using the FEM method and the shallow water equations". MA thesis. Royal Institute of Technology (KTH).

Ostapenko, V. V. 1999. "Numerical simulation of wave flows caused by a shoreside landslide". In: *Journal of Applied Mechanics and Technical Physics* 40.4, pp. 647–654.

Petcu, Madalina and Roger Temam. 2013. "An interface problem: The two-layer shallow water equations". In: *Discrete and Continuous Dynamical Systems* 33.11/12, pp. 5327–5345.

Ribeiro, F., A. Galeão, and L. Landau. 2001. "Edge-based finite element method for shallow water equations". In: *International Journal for Numerical Methods in Fluids* 36.6, pp. 659–685.

Rispoli, Franco and G.Z. Rafael Saavedra. 2006. "A stabilized finite element method based on SGS models for compressible flows". In: *Computer Methods in Applied Mechanics and Engineering* 196.1, pp. 652 –664.

Rispoli, Franco et al. 2007. "Computation of inviscid compressible flows with the V-SGS stabilization and YZβ shock-capturing". In: *International Journal for Numerical Methods in Fluids* 54.6-8, pp. 695–706.

Sármány, D. and M.E. Hubbard. 2013. "Upwind residual distribution for shallow-water ocean modelling". In: *Ocean Modelling* 64, pp. 1–11.

Soulsby, R. 1997. *Dynamics of marine sands: a manual for practical applications*. Telford.

Stoker, James Johnston. 1992. *Water waves: The mathematical theory with applications*. Vol. 36. John Wiley & Sons, Inc.

Tadmor, Eitan and Weigang Zhong. 2008. "Energy-Preserving and Stable Approximations for the Two-Dimensional Shallow Water Equations". In: *Mathematics and Computation, a Contemporary View: The Abel Symposium 2006 Proceedings of the Third Abel Symposium, Alesund, Norway, May 25–27, 2006*. Ed. by Hans Munthe-Kaas and Brynjulf Owren. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 67–94.

Takase, Shinsuke et al. 2010. "Space-time SUPG formulation of the shallow-water equations". In: *International Journal for Numerical Methods in Fluids* 64.10-12, pp. 1379–1394.

Tezduyar, Tayfun E. 2004. "Finite element methods for fluid dynamics with moving boundaries and interfaces". In: *Encyclopedia of Computational Mechanics*. John Wiley & Sons, Ltd. Chap. 17.

Vacondio, Renato et al. 2017. "A non-uniform efficient grid type for GPU-parallel Shallow Water Equations models". In: *Environmental Modelling & Software* 88, pp. 119 –137.

Viero, Daniele Pietro and Mohammad Valipour. 2017. "Modeling anisotropy in free-surface overland and shallow inundation flows". In: *Advances in Water Resources* 104, pp. 1–14.

Wu, Jin. 1982. "Wind-stress coefficients over sea surface from breeze to hurricane". In: *Journal of Geophysical Research: Oceans* 87.C12, pp. 9704–9706.