Robust finite volume schemes for 2D shallow water models. Application to flood plain dynamics

Frédéric Couderc, Jerome Monnier, Jean-Paul Vila, Kévin Larnier, Ronan Madec, Denis Dartus

Robust finite volume schemes and variational inversions for 2D shallow water models. Application to flood plain dynamics.

F. Couderc[a], J. Monnier[b,d], J.-P. Vila[b], K. Larnier[c], R. Madec[b], D. Dartus[c]

[a] *CNRS & Mathematics Institute of Toulouse (IMT), F-31077 Toulouse cedex 4, France*
[b] *INSA & Mathematics Institute of Toulouse (IMT), F-31077 Toulouse cedex 4, France*
[c] *INP & Fluid Mechanics Institute of Toulouse (IMFT), F-31400 Toulouse cedex 4, France*
[d] *Corresponding author: jerome.monnier@insa-toulouse.fr*

Abstract

This study proposes original combinations of higher order Godunov type finite volume schemes and time discretization schemes for the 2d shallow water equations, leading to fully second-order accuracy with well-balanced property. Also accuracy, positiveness and stability properties in presence of dynamic wet/dry fronts is demonstrated. The test cases are the classical ones plus extra new ones representing the geophysical flow features and difficulties. Preliminary numerical experiments give some key features between the topography representation scale and the grid size, and depending on the scheme order. A second aspect aims at performing sensitivities analysis and variational data assimilation, based on the adjoint model. Original sensitivity maps with respect to the basal (roughness, topography) fields are discussed. Inflow discharges and roughness coefficients spatially distributed are efficiently inferred on a real-like flood plain test case.

Keywords: Shallow water, finite volumes, second order, adjoint, data assimilation, sensitivity, wet-dry front, flood plain.

1. Introduction

The 2D Shallow Water (SW) equations, with or without the Coriolis term, can be suitable to model many types of geophysical flows and waves propagation, eg in the atmosphere, the oceans, the rivers. These geophysical flows involves multi-scales, badly known topographies (or well known

but at a fixed scale), uncertain quantities at open boundaries, uncertain values of the model parameter (eg the basal roughness coefficients) and large computational domains. Many of them (eg flood plain in river hydraulics, coastal flows) involves a dynamic wet-dry front. In this context, accurate higher-order numerical schemes can become a key feature to reach acceptable accuracy-computational cost balances, a reliable wet-dry front dynamic computation can be a key feature of the numerical model accuracy too. In other respect, sensitivity analyses become a key ingredient to better understand the complex interactions and set up the model; finally a full data assimilation procedure becomes a crucial step to design a reliable predictive model.
The aim of this paper is to elaborate and assess accurate and robust computations of 2d SW equations with friction and in presence of wet dry front dynamics, typically for real applications in river hydraulics, low-waters or flood plain dynamics.

The first feature of the present study is the direct modeling of this flow type. To do so, we classically use finite volume schemes on unstructured meshes combining Godunov type solvers completed with MUSCL technics and an implicit-explicit Runge-Kutta (IMEX) second order time discretization, together with well balanced features. Original combinations of these already existing methods are elaborated, completed by few necessary modifications; the result leading to demonstrated fully second-order accuracy (both in space and time), well-balanced property (water at rest is preserved), positiveness and stability in presence of dynamic wet/dry fronts. (Note that the second order accuracy can be mathematically reach on smooth solutions only; nevertheless, in presence of dynamic wet/dry fronts, it will be demonstrated that the higher-order schemes are more accurate than the first order ones, even if their convergence rate decreases).

The application of FV schemes to hydraulic problems has started in the early eighties together with the development of these schemes for aerodynamic applications, see e.g. [1] treating of SW equations with bathymetry and friction. MUSCL technics comes after the pioneering work [2]. First analysis higher-order are written in [3] (on structured meshes) and in [4] including hydraulic applications. The success of these MUSCL approaches is mainly due to the efficiency of their extension to unstructured meshes. Complete and recent reviews can be found for example in [5, 6].

As mentioned above, the present contribution aims at elaborating a set of methods leading to a full second order scheme (both in space and time), accuracy and robustness in presence of wet-dry front situations. Concerning the choice of an approximate Riemann solver, it is proposed to use the HLLC solver [7, 8], combined with an ad-hoc estimate of wave velocity [9]. Concerning the MUSCL technics, some tools in relation with mono/multi-slope framework and corresponding limiters combined with the well-balanced property are described. The well-balanced property (to preserve water at rest) comes from the early work [10]; it has been followed by numerous papers, see also e.g. [11] for the theoretical framework. In view of hydraulic applications, it has been found in the present study that the quite recent well-balanced tool due to [12, 13] provides a robust and accurate framework. Concerning the time stepping, it has been found that an appropriate second order time scheme of IMEX type is necessary to achieve an actual second order accuracy, also robustness with respect to a lot of external effects (such as the variability of topographical effects).
To our best knowledge, the resulting numerical scheme combination is new, and it leads to a numerical solver accuracy, stability, and finally robustness, as never published in the literature.
The numerical schemes are assessed by performing classical test cases from the literature plus extra new ones, representing the geophysical shallow flow features and difficulties.

Furthermore, original phenomena are highlighted. Based on the present accurate FV solvers, the SW equations filtering property of the bathymetry variations are numerically analyzed into details, in the case of a 1d open-channel flow. In addition of its mathematical interest, this feature is important in particular in a context of observation of such free-surface shallow flows from above (eg satellite observations of the water surface). Also, this present preliminary study gives some key features to address the issue of the topography scale - grid adequacy. To our best knowledge, this question was not addressed in the literature yet.

Finally, a detailed comparison of the first order scheme and the second order scheme is made in terms of accuracy (and CPU time) on a real-like test case: a (virtual) flood event on the Lèze river, south-western France. The bathymetry and topography are the real ones (the flood plain topography

has been deduced from very fine LIDAR data); the event remains virtual in the sense that no accurate measurements were collected during the event.

A second feature of the present study is the elaboration of variational sensitivities, based on the adjoint of the numerical model described above, and variational data assimilation (4D-var method). Recall that in any geophysical context, usually the boundary conditions at open boundaries, the initial condition, the model parametrization (eg the basal friction), or even the topography are known with large uncertainties or sometimes even unknown. Then, a data assimilation process, based on the measurements available (usually heterogeneous in space and time), becomes crucial to reduce these uncertainties. A classical approach to assimilate data into the model is the variational data assimilation method, based on the adjoint model and an optimal control loop. This method allows to identify input parameters and leads to the model calibration. It is widely employed for few geophysical flow types (see e.g. [14] and references therein).
Concerning 1d river flow models (Saint-Venant equations), calibration and sensitivity analysis, based on the variational approach and on filtering approaches, have been widely studied in the literature. On the contrary, for 2d SW models, few studies only address sensitivity analyses and parameter identification (eg friction parameter, boundary conditions) using the variational approach (adjoint method). One of the reason is the heavy task to elaborate the adjoint model, then to make it run in reasonable CPU time and memory storage. Let us cite [15] which treats of a 2d SW river model coupled with a simplified sediment transport model; [16], [17] treating of the assimilation of a flood plain image (post-treated SAR data) into the 2d flow model; [18], [19] treating simultaneously the assimilation and the coupling 2d local - 1d global flow models in academic configurations; [20] treating of a barotropic tidal model; [21], [22] treating of the assimilation of surface drifting particles (lagrangian data) into the 2d flow model. All these studies are based on the minimization of a cost function, with its gradient computed using the adjoint model, then a descent algorithm (local minimization).
Let us point out that none of these studies are based on higher-order numerical schemes; also the wet-dry front is classically regularized (using a cut-off value at the front). In the context of a flood plain dynamic (observed from above or not), both the high accuracy of the model (direct and adjoint ones) and the wet-dry front dynamics may be crucial features of the global model accuracy and reliability. In other respect, let us cite [23] which proposes an

original analysis of the analytical properties of the solutions of the sensitivity equations for steady-state 2d SW solutions.

In the present article, original variational sensitivities analyses are presented. A cost function measuring the discrepancy of the computed water elevations at two (virtual) gauge stations (synthetic data with realistic noise amplitudes) is classically defined. The cost gradient is computed by performing the adjoint code of the numerical models described previously. The latter is essentially obtained by automatic differentiation using the software Tapenade [24], plus few tricks to optimize the memory required and to make "reverse" the MPI commands. The full model (direct+adjoint) is accurately assessed (using the procedures described here).
The test case is the same portion of Lèze river as previously. The computed sensitivities are with respect to: a) the Manning-Strickler friction coefficient (denoted by n) which is locally defined i.e. without any a-priori on land covers (one value per cell); b) the topography elevation (denoted by z_b).
The bathymetry sensitivity pattern is globally comparable to the friction sensitivity one but the bathymetry one is more point-wise, less diffused. These spatially distributed sensitivity maps greatly help the hydraulic modeler to better understand the mathematical model, the hydraulic model (combining the DTM, the parametrization) and the flow. Also, these sensitivities maps with respect to the bathymetry z_b can give information in view to define a discrete bathymetry field consistent with the friction field.

Next identification - calibration numerical experiments are performed. The identified parameters are: a) The friction Manning-Strickler coefficients defined by land covers (6 in this case); b) the inflow discharge $Q_{in}(t)$ at upstream (open boundary). It is twin experiments since the data are generated by the model (then a realistic random noise is added). Data are the same as previously: elevation time-series measurements at two (virtual) gauge stations. These numerical experiments demonstrate the approach capabilities in a virtual flood plain dynamic context. Based on the know-how exposed here, the computational cost of the data assimilation process becomes affordable (both in terms of CPU time and memory).
As it is well known, the drawback of the adjoint based approach is the large number of cost function evaluations required for parameter identifications (between10 and 35 in the present studies to reach an excellent accuracy on the parameters sought); nevertheless it is much less than any stochastic ap-

proach (which does not require the cost gradient information). The very informative sensitivity maps (spatially distributed information) require the running of the direct and the adjoint model only; hence a sensitivity analysis costs in terms of CPU time, roughly 5 times (1+4) a direct simulation. But, recall that these sensitivity maps are valid at one parameter value only (eg at one roughness field value). An other drawback of the adjoint method is due to the requirement of the adjoint code-model. Nevertheless, in DassFlow software, the adjoint code generation is fully automated. To do so, firstly the direct source code is designed to be differentiated almost entirely by an automatic differentiation tool (Tapenade [24] in the present case); next few tricks for MPI commands need to be introduced, but they are implemented automatically by an extra script. Also the code validation (direct+adjoint), based on classical procedures, is almost fully automatic.

All the numerical schemes and algorithms presented here have been implemented into the computational platform DassFlow (Data Assimilation for Free Surface Flows), [25, 26, 27]. Of course, all the methods elaborated in the present context (the finite volume schemes and the variational data assimilation process based on the adjoint model) and the resulting software, can be extended and applied to any other flow modeled by the 2d SW equations (including those with a non negligible Coriolis force eg tidal flows, by adding it simply).

The paper is organized as follows. In Section 2, the direct mathematical model and the finite volume schemes elaborated (1st and 2nd order) are presented. In Section 3, the inverse model (in the sense the adjoint, the resulting variational sensitivities and the data assimilation process) is presented. Classical and original benchmarks to assess and compare the finite volume schemes are performed in Section 4. The real-like flood plain test case (based on the real topography and a historical-like inflow discharge of the Lèze river) is considered in Section 5, both for the direct numerical comparisons and the inverse features demonstration and analysis (sensitivity analyses and parameter identifications). Some remarks and perspectives are proposed in the conclusion (Section 6). The appendix detail the assessment procedures of the adjoint code, the way to transform the direct MPI commands into the adjoint ones, few tricks to improve the adjoint code memory, and a speed-up curve demonstrating the capabilities of the resulting full model (direct+adjoint).

2. Direct Model

In this section, we describe the direct (or forward) model: the two-dimensional SW equations discretized using a FV method. The schemes are first or second-order, both in space and time, well-balanced (water at rest is preserved), positive, and the friction source term is implicitly discretized insuring stability in presence of dynamic wet/dry fronts. The resulting schemes are obtained from new combinations of existing techniques and/or extension of existing ones. The main features are:

- accurate waves speed estimates considering an HLLC approximate Riemann solver,
- the wet-dry front dynamic independent of any regularization (i.e. no cut-off on the depth value),
- MUSCL reconstruction, well-balanced schemes, "implicitation" of the friction term together with an implicit-explicit Runge-Kutta (IMEX) time stepping method insuring global second order time space accuracy,
- robust implementation of outflow and inflow condition, including some feedback stabilization procedure.

To our best knowledge, such combinations are new; this leads to a numerical solver accurate and stable as never published in the literature. In next section, we present benchmarks, including original ones, demonstrating these properties.

2.1. Mathematical model

The two-dimensional SW equations write in conservative form as follows:

$$\partial_t \mathbf{U} + \nabla.\mathbf{F}(\mathbf{U}) = \mathbf{S}_g(\mathbf{U}) + \mathbf{S}_f(\mathbf{U})$$

$$\mathbf{U} = \begin{bmatrix} h \\ h\mathbf{u} \end{bmatrix}, \ \mathbf{F}(\mathbf{U}) = \begin{bmatrix} h\mathbf{u} \\ h\mathbf{u} \otimes \mathbf{u} + \dfrac{gh^2}{2}\mathbf{I} \end{bmatrix}, \tag{1}$$

$$\mathbf{S}_g(\mathbf{U}) = \begin{bmatrix} 0 \\ -gh\nabla z_b \end{bmatrix}, \ \mathbf{S}_f(\mathbf{U}) = \begin{bmatrix} 0 \\ -g\dfrac{n^2 \|\mathbf{u}\|}{h^{1/3}}\mathbf{u} \end{bmatrix}$$

2.2. Numerical schemes notations

We denote by I_h a discretization of the computational domain Ω with N cells K_i (cells can be triangles or quadrilaterals or mix of both). Let us introduce some notations and conventions (see Fig.1). Considering a given cell K (omitting i index):

- m_K is the area of the cell K, $m_{\partial K}$ its perimeter and x_K its barycenter.
- e is one of the δK boundary edges, m_e its length and x_e its center.
- K_e is the neighboring cell to K across e.
- $\mathbf{n}_{e,K}$ is the unit normal to e oriented from K to K_e.

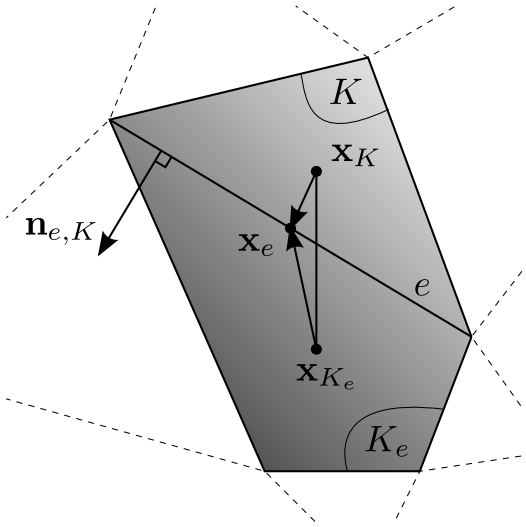


Figure 1: Some notations and conventions

2.3. Numerical fluxes

2.3.1. First order scheme

We integrate over the cells K the hyperbolic system (1) without the source terms $\mathbf{S}_g$ and $\mathbf{S}_f$, and we denote by $\mathbf{U}_K$ the piecewise constant approximation

of $\mathbf{U}$. We consider the Godunov method [28], then the semi-discrete scheme of the homogeneous system writes:

$$m_K \frac{d\mathbf{U}_K}{dt} + \sum_{e \in \partial K} m_e \, \mathbf{F}_e(\mathbf{U}_K, \mathbf{U}_{K_e}, \mathbf{n}_{e,K}) = 0 \tag{2}$$

The rotational invariance property of the SW equations (1) allows to reduce this sum of 2D problem to a sum of 1D local Riemann problems such that $\mathbf{F}_e(\mathbf{U}_K^n, \mathbf{U}_{K_e}^n, \mathbf{n}_{e,K}) = \mathbf{R}_{e,K}^{-1} \hat{\mathbf{F}}_e(\hat{\mathbf{U}}_{e,K}^n, \hat{\mathbf{U}}_{e,K_e}^n)$ with $\hat{\mathbf{U}}_{e,K}^n = \mathbf{R}_{e,K} \mathbf{U}_K^n$ where $\mathbf{R}_{e,K}$ is the rotation matrix, see e.g. [5].

The HLLC approximate Riemann solver is used, see e.g. [5],

$$\left\{ \begin{array}{l} \left[\hat{\mathbf{F}}_e^{HLLC}\right]_{1,2} = \dfrac{s_{K_e}\left[\mathbf{F}(\hat{\mathbf{U}}_K)\right]_{1,2} - s_K\left[\mathbf{F}(\hat{\mathbf{U}}_{K_e})\right]_{1,2} + s_K s_{K_e}(\left[\hat{\mathbf{U}}_{K_e}\right]_{1,2} - \left[\hat{\mathbf{U}}_K\right]_{1,2})}{s_{K_e} - s_K} \\ \left[\hat{\mathbf{F}}_e^{HLLC}\right]_3 = \left[\hat{\mathbf{F}}_e^{HLLC}\right]_1 \hat{v}* \quad \text{with} \quad \hat{v}* = \left\{ \begin{array}{ll} \hat{v}_K & \text{if} \quad s^* \geq 0 \\ \hat{v}_{K_e} & \text{if} \quad s^* < 0 \end{array} \right. \end{array} \right. \tag{3}$$

with the wave speed estimates introduced in [9]:

$$\begin{array}{c} s_K = \min\left(0, \hat{u}_K - \sqrt{g h_K}, \hat{u}_{K_e} - 2\sqrt{g h_{K_e}} + \sqrt{g h_K}\right) \\ s_{K_e} = \max\left(0, \hat{u}_{K_e} + \sqrt{g h_{K_e}}, \hat{u}_K + 2\sqrt{g h_K} - \sqrt{g h_{K_e}}\right) \end{array} \tag{4}$$

The latter insure L^∞ stability, positivity and consistency with entropy condition under a CFL-like condition, see [9]. Finally, this choice is made for the intermediate wave speed estimate expressed as follows, see e.g. [5]:

$$s^* = \frac{s_K h_{K_e} \hat{u}_{K_e} - s_{K_e} h_K \hat{u}_K - s_K s_{K_e}(h_{K_e} - h_K)}{h_{K_e}(\hat{u}_{K_e} - s_{K_e}) - h_K(\hat{u}_K - s_K)} \tag{5}$$

This approximate Riemann solver is the mandatory first step to obtain a stable and positive numerical solver in presence of dynamic wet/dry fronts.

2.3.2. Second order scheme

The second order scheme is based on the first order scheme above and a monoslope MUSCL reconstruction, see e.g. [29, 30]. It consists to compute local vectorial slopes $[\nabla \mathbf{U}_K]_i$ in each cell K for each variable i. The two reconstructed conservative variables vectors at each side of edge e are denoted by $\mathbf{U}_{e,K}$ and $\mathbf{U}_{e,K_e}$. They satisfy:

$$\begin{array}{c} \mathbf{U}_{e,K} = \mathbf{U}_K + \nabla \mathbf{U}_K . \mathbf{x}_K \mathbf{x}_e \\ \mathbf{U}_{e,K_e} = \mathbf{U}_{K_e} + \nabla \mathbf{U}_{K_e} . \mathbf{x}_{K_e} \mathbf{x}_e \end{array} \tag{6}$$

Then, $\mathbf{U}_{e,K}$ and $\mathbf{U}_{e,K_e}$ replace $\mathbf{U}_K$ and $\mathbf{U}_{K_e}$ in the original first order semi-discrete scheme (2) to evaluate the numerical flux $\mathbf{F}_e$. Thus, we have:

$$m_K \frac{d\mathbf{U}_K}{dt} + \sum_{e \in \partial K} m_e \, \mathbf{F}_e(\mathbf{U}_{e,K}, \mathbf{U}_{e,K_e}, \mathbf{n}_{e,K}) = 0 \tag{7}$$

With such a linear reconstruction, a second-order accuracy in space can be obtained (at least, for solutions regular enough). To this end, a least square method is first employed to predict the vectorial slopes for each primitive variable ($\mathbf{W}_K = [\ h_K\ u_K\ v_K\ ]^T$). These sums of squares,

$$E_i\left(\left[\widetilde{\nabla \mathbf{W}}_K\right]_i\right) = \sum_{e \in \partial K} \left([\mathbf{W}_{K_e}]_i - ([\mathbf{W}_K]_i + [\nabla \mathbf{W}_K]_i \,.\mathbf{x}_K\mathbf{x}_{K_e})\right)^2 \tag{8}$$

are minimized by setting the gradients to zero solution of simple 2 x 2 linear systems. This method represents a good alternative among others to find the hyperplane [29, 31] because of its accuracy and robustness independently to the number of neighbors (this is an important property as it will be shown later with the wet/dry front treatment for a well-balanced scheme). In order to prevent from large numerical dispersive instabilities, hence keep the scheme positive, the predicted vectorial slopes need to be limited. A first and efficient method consists to simply apply the maximum principle to the two edge unlimited reconstructed primitive variable computed from (6) and (8) such that the two limited reconstructed variables $\mathbf{W}_{e,K}$ and $\mathbf{W}_{e,K_e}$ at edge e verify:

$$\min(\mathbf{W}_K, \mathbf{W}_{K_e}) \leq \mathbf{W}_{e,K}, \mathbf{W}_{e,K_e} \leq \max(\mathbf{W}_K, \mathbf{W}_{K_e}) \tag{9}$$

It is what we call the MP limiter (MP for Maximum Principle). We observe in practice that it generates very moderate oscillations at solution singularities whereas diffusion is minimized comparatively to other classic limiters like Minmod or Van Albada; see also e.g. [13, 32, 33, 34]. Nevertheless, the generation of new extrema and the presence of wet/dry fronts can break down the FV mass conservation property in an unacceptable proportion; the scheme is no longer positive. A first solution is to manage "by hand" the mass conservation at the end of each time step. The mass artificially added cutting-off to zero an eventual negative water depth is removed proportionally to neighboring wet cells. A second, and more reliable solution, is to use the Barth limiter [35] for the water depth h:

$$[\nabla \mathbf{W}_K]_i = \min_{e \in \partial K} \left(1, [\phi_{K,e}]_i\right) \left[\widetilde{\nabla \mathbf{W}}_K\right]_i$$

$$[\phi_{K,e}]_i = \begin{cases} 0 & \text{if } ([\mathbf{W}_{K_e}]_i - [\mathbf{W}_K]_i)[\nabla \mathbf{W}_K]_i . \mathbf{x}_K \mathbf{x}_e < 0 \\ \dfrac{[\mathbf{W}_{K_e}]_i - [\mathbf{W}_K]_i}{[\nabla \mathbf{W}_K]_i . \mathbf{x}_K \mathbf{x}_e} & \text{if not} \end{cases} \tag{10}$$

The reconstructed conservative variables vectors $\mathbf{U}_K$ and $\mathbf{U}_{K_e}$ are then obtained by a simple multiplication.

2.4. Well-balanced property with wet/dry front treatment

It is well-known that the numerical schemes must be well balanced in the sense that the pressure term $\nabla\left(gh^2/2\right)$ must equal the gravity source term $(-gh\nabla z_b)$ in a discrete point of view: water at rest must stay at rest. If this property is not satisfied, spurious numerical velocities are generated. Such a phenomena may happen for example at lateral banks of the river. To enforce this balance property, it exists many methods in the literature. In the present study we compare two of them, and observe that they lead to different properties and accuracies. The well-balanced property is very delicate in presence of a wet-dry front, and combining robustness together with formal second-order accuracy revealed to be a challenge. Our experiments lead to recommend the approach of Audusse et al [12, 13] (so-called 'A-well-balanced scheme' below) which is very efficient when combined with suitable correction of HLL flux and wet/dry treatment. For completeness, we also describe, and have assessed, the formulation derived in [36, 37] (so-called 'E-well-balanced scheme' below) which was less robust or accurate in the present tests.

2.4.1. The E-well-balanced scheme

Following [36, 37], to build a well-balanced scheme, the equalities below must be verified when $z_K + h_K = z_{K_e} + h_{K_e}$ and $\mathbf{u}_K = \mathbf{u}_{K_e} = 0$,

$$\begin{aligned} \mathbf{F}_e^h(\mathbf{U}_K, \mathbf{U}_{K_e}, \mathbf{n}_{e,K}) &= 0 \\ \sum_{e \in \partial K} m_e \mathbf{F}_e^{h\mathbf{u}}(\mathbf{U}_K, \mathbf{U}_{K_e}, \mathbf{n}_{e,K}) &= -g \sum_{e \in \partial K} m_e h_e (z_e - z_K) \mathbf{n}_{e,K} \end{aligned} \tag{11}$$

using the property $h\nabla z_b = \nabla(hz_b) - z_b \nabla h$ and the Green's identity to find the piecewise constant approximation of $\mathbf{S}_g(\mathbf{U})$ incorporated into the

semi-discrete scheme (2). Next, the reconstructions at edge e of the water depth and the topography, h_e and z_e respectively, have to be computed in order to derive a well-balanced scheme with a consistent discretization of the gravitational source term $\mathbf{S}_g(\mathbf{U})$. Because of the geometrical property $\sum_{e\in\partial K} m_e \mathbf{n}_{e,K} = 0$, the term $\mathbf{F}_e^{h\mathbf{u}}(\mathbf{U}_K, \mathbf{U}_K, \mathbf{n}_{e,K})$ can be subtracted in the first sum. After some calculations and by identification, one can easily verify, using the HLLC approximate Riemann solver (3) to evaluate $\mathbf{F}_e$, that the two following reconstructions give the second equality in (11):

$$z_e = \frac{1}{2}\left(z_K + z_{K_e}\right) \quad \text{and} \quad h_e = \frac{s_K}{s_{K_e} - s_K}\left(h_K + h_{K_e}\right) \tag{12}$$

To find a zero mass flux, the water depths h_K and h_{K_e} in the diffusive up-winded part of the HLLC approximate Riemann solver are replaced by z_K+h_K and $z_{K_e}+h_{K_e}$ respectively. At wet/dry fronts, the bottom topography is changed in a dry cell K_e such that $z_{K_e} = z_K + h_K$.
This numerical method can be extend to second-order taking into account the reconstructed water depths at edge e but it has not been found stable in all cases. More generally, changing the diffuse up-winded part of the HLLC approximate Riemann solver cannot ensure a stable numerical model.

2.4.2. The A-well-balanced scheme

Following [12, 13], a new vectorial slope for the variable $\eta = h + z$ is first calculated in addition to the primitive variables (6). Then, the so-called hydrostatic reconstructed water depth $h^*_{e,K}$ is defined with the help of a reconstructed topography z_e at edge e as follows:

$$\begin{aligned} h^*_{e,K} &= \max(0, h_{e,K} + z_{e,K} - z_e) \\ \text{with} \quad &\begin{cases} z_{e,K} = \eta_{e,K} - h_{e,K} \\ z_e = \max\left(z_{e,K}, z_{e,K_e}\right) \end{cases} \end{aligned} \tag{13}$$

Next, the standard MUSCL reconstructed conservative variable vector $\mathbf{U}^n_{e,K}$ in semi-discrete scheme (7) is replaced by:

$$\mathbf{U}^*_{e,K} = \begin{bmatrix} h^*_{e,K} \\ h^*_{e,K}\mathbf{u} \end{bmatrix} \tag{14}$$

Including a consistent discretization of the gravitational source term $\mathbf{S}_g$ with the continuous formulation, we finally obtain the following well-balanced scheme, second order:

$$m_K \frac{d\mathbf{U}_K}{dt} + \sum_{e \in \partial K} m_e \left(\mathbf{F}_e(\mathbf{U}^*_{e,K}, \mathbf{U}^*_{e,K_e}, \mathbf{n}_{e,K}) + \right.$$
$$\left. \mathbf{S}_g(\mathbf{U}_{e,K}, \mathbf{U}^*_{e,K}, \mathbf{n}_{e,K}) + \mathbf{S}^c_g(\mathbf{U}_K, \mathbf{U}_{e,K}, z_K, z_{e,K}, \mathbf{n}_{e,K}) \right) = 0$$

$$\mathbf{S}_g(\mathbf{U}_{e,K}, \mathbf{U}^*_{e,K}, \mathbf{n}_{e,K}) = \begin{bmatrix} 0 \\ \frac{g}{2} \left(\left(h^*_{e,K}\right)^2 - \left(h_{e,K}\right)^2 \right) \mathbf{n}_{e,K} \end{bmatrix} \tag{15}$$

$$\mathbf{S}^c_g(\mathbf{U}_K, \mathbf{U}_{e,K}, z_K, z_{e,K}, \mathbf{n}_{e,K}) = \begin{bmatrix} 0 \\ -\frac{g}{2} \left(h_{e,K} + h_K\right) \left(z_{e,K} - z_K\right) \mathbf{n}_{e,K} \end{bmatrix}$$

2.4.3. Robust wet/dry front treatment of A-well-balanced scheme

Our numerical experiments showed that this scheme is not stable at wet/dry fronts perpendicular to the flow streamlines. Let us specify here some additional requirements in order to stabilize in all cases and for all limiters this second order well-balanced scheme. Firstly, all vectorial slopes are enforced to zero in dry cells (what can be viewed as a water mask). Secondly, these same dry cells are not taken into account in the least square method (8) to predict the vectorial slope $\nabla\eta^n$,

$$\begin{gathered} \left[\nabla \mathbf{U}_K\right]_i = \mathbf{0} \quad \text{if} \quad h_K = 0 \\ E(\nabla\eta_K) = \sum_{e \in (\partial K \setminus h_{K_e} = 0)} \left(\eta_{K_e} - \left(\eta_K + \nabla\eta_K . \mathbf{x}_K \mathbf{x}_{K_e}\right)\right)^2 \end{gathered} \tag{16}$$

These conditions are related to the typical situation in Fig.(2) where the well-balanced property is desired with presence of a wet/dry front because the water surface level η is constant around the cell K. The dry cell K_e has a bottom topography z_{K_e} upper than the neibgboring wet cell K water surface level η_K, the fluxes must be enforced to zero at the edge e between the two cells (water cannot "climb" in this situation) and the well-balanced property must be verified in the cell K.

Making the hypothesis that the normal velocity $\hat{u}_{e,K}$ is not necessarily zero, a simple way to enforce $\mathbf{F}_e(\mathbf{U}^*_{e,K}, \mathbf{U}^*_{e,K_e}, \mathbf{n}_{e,K})$ in (15) to be zero

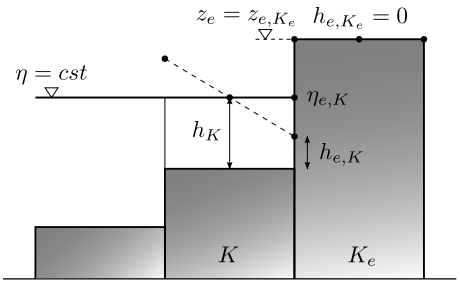


Figure 2: Typical situation of desired well-balanced property in presence of a wet/dry front.

is for the reconstructed hydrostatic water depths $h^*_{e,K}$ and h^*_{e,K_e} to be always zero (we remark that it is effectively always true for the first order version of the scheme since $z_e = \max(z_K, z_{K_e}) = z_{K_e} > h_K + z_K$, $h^*_{e,K} = \max(0, h_K + z_K - z_e) = 0$ and $h^*_{e,K_e} = \max(0, h_{K_e} + z_{K_e} - z_e) = 0$). On the other hand, a key-point in the well-balanced method of Audusse and al. [12, 13], in the version second order, is that the vectorial slope for the variable $\eta = h + z$ must be zero to find the well-balanced property in a cell K. The variable η must be constant in all neighboring cells of a wet cell which is not true in our typical case Fig.(2). So, in order to find a zero vectorial slope for the variable η in the cell K, the dry cell K_e is not taken into account in the least square method (8). Now, a consequence is that $\eta_{e,K} = h_K + z_K < \eta_{e,K_e} = z_{e,K_e}$ implying that $z_e = \max(z_{e,K}, z_{e,K_e}) = z_{e,K_e}$ and concluded that it is always find that $h^*_{e,K} = \max(0, \eta_{e,K} - z_e) = 0$. Since all vectorial slopes are enforced to zero in dry cells, it is obvious that we have also $h^*_{e,K_e} = 0$. The sum of the non negative components of the two sources terms $\mathbf{S}_g(\mathbf{U}_K, \mathbf{U}^*_{e,K}, \mathbf{n}_{e,K})$ and $\mathbf{S}^c_g(\mathbf{U}_K, \mathbf{U}^*_{e,K}, z_K, z_{e,K}, \mathbf{n}_{e,K})$ is reduce to: $-\dfrac{g}{2} h_K^2 \mathbf{n}_{e,K}$ since $z_{e,K} - z_K = h_K - h_{e,K}$. This does not change the demonstration made in [13] to find the well-balanced property in the cell K (that use the geometrical property $\displaystyle\sum_{e \in \partial K} m_e \mathbf{n}_{e,K} = 0$).

Fibnally, we have demonstrated that the present additional treatment (16) to the original scheme (15) are sufficient to find the well-balanced property in presence of wet/dry interfaces without any consideration on limitation such it was done before. Furthermore, this does not change the global scheme

at wet/dry fronts since the predicted vectorial slopes only are changed. At worst, the scheme "falls down" at the first order in cells with a neighboring dry cell.

2.5. Time stepping and friction source term

The friction source term $\mathbf{S}_f$ must be implicitly discretized in presence of dynamic wet/dry fronts because of its stiffness as soon as the water height vanishes. This can be achieved using a splitting method, see e.g. [38, 5], and by resolving the implicit problem:

$$\mathbf{U}_K^{n+1} = \mathbf{U}_K^n + \Delta t^n \mathbf{S}_f(\mathbf{U}_K^{n+1}) \tag{17}$$

where $\mathbf{U}_K^n$ refers to the previous well-balanced scheme (15) using an explicit Euler time step to discretize the time partial derivative. Since the friction source term $\mathbf{S}_f$ is zero in the mass conservation equation (1), and so $h^{n+1} = h^n$, the implicit problem (17) can be then simply to the resolution of:

$$\left\|\mathbf{u}^{n+1}\right\| \mathbf{u}^{n+1} + c\left(\mathbf{u}^{n+1} - \mathbf{u}^n\right) = 0 \quad \text{with} \quad c = \frac{(h^n)^{4/3}}{\Delta t^n \ g \ n^2} \tag{18}$$

Remarking that $\mathbf{u}^{n+1} = \alpha \ \mathbf{u}^n$ with $\alpha \in]0, 1[$ (the vectors are colinear and $\mathbf{q}^n.\mathbf{q}^{n+1} \geq 0$, the flow direction cannot be reversed) reduces this system of non-linear equations to the resolution of a quadratic equation in α. The final analytical solution to the implicit problem (17) is:

$$\mathbf{U}_K^{n+1} = \mathbf{M}(\mathbf{U}_K^n, \Delta t^n) \quad \text{with,}$$

$$\mathbf{M}(\mathbf{U}_K^n, \Delta t^n) = \begin{bmatrix} h^n \\ \dfrac{2\,(h^n)^{5/3}\,\mathbf{u}^n}{(h^n)^{2/3} + \sqrt{(h^n)^{4/3} + 4\Delta t^n \ g \ n^2 \ \|\mathbf{u}^n\|}} \end{bmatrix} \tag{19}$$

The initial full implicit problem (17) is reduced to an explicit step. We note that in [39], the authors have derived a similar expression by linearization of the implicit problem which can be obtained by Taylor expansion at first order of (19), as semi-implicit formulations of (17).

Now, obtaining a second order method using a splitting method is quite complex when the source term is very stiff (The Strang splitting is reduced at

first order for hyperbolic system when the source term is very stiff, see [40]). For example, it is tempting to consider the second order SSP-RK2 method (also known as the Heun method), see e.g. [41], including the splitted implicit step (17) in the two forward Euler steps as follows:

$$\begin{aligned}
\mathbf{U}_K^{(1)} &= \mathbf{U}_K^n + \Delta t^n \mathbf{L}(\mathbf{U}_K^n) \\
\mathbf{U}_K^{(2)} &= \mathbf{M}(\mathbf{U}_K^{(1)}, \Delta t^n) \\
\mathbf{U}_K^{(3)} &= \mathbf{U}_K^{(2)} + \Delta t^n \mathbf{L}(\mathbf{U}_K^{(2)}) \\
\mathbf{U}_K^{(4)} &= \mathbf{M}(\mathbf{U}_K^{(3)}, \Delta t^n) \\
\mathbf{U}_K^{n+1} &= \frac{1}{2}\left(\mathbf{U}_K^n + \mathbf{U}_K^{(4)}\right)
\end{aligned} \tag{20}$$

If this time splitting strategy is stable and preserve positivity without changing the CFL-like condition for the time step Δt^n, the result is only first order accurate (as it will be shown in benchmark problems). In order to resolve this problem, the authors of [42] have recently introduced the implicit-explicit Runge-Kutta schemes (IMEX). The method consists in combining an explicit strong stability preserving Runge-Kutta (RK-SSP) time discretization of the homogeneous system and a diagonally implicit Runge-Kutta (DIRK) time discretization of the stiff source term, taking advantages of both methods. The order conditions have been found using Taylor expansion. The scheme named IMEX-SSP(3,2,2) have been chosen and transformed for numerical implementation purpose as follows:

$$
\begin{aligned}
\mathbf{U}_K^{(1)} &= \mathbf{M}(\mathbf{U}_K^n, \Delta t^n/2) \\
\mathbf{U}_K^{(2)} &= \mathbf{M}(2\mathbf{U}_K^n - \mathbf{U}_K^{(1)}, \Delta t^n/2) \\
\mathbf{U}_K^{(3)} &= \mathbf{U}_K^n + \Delta t^n \mathbf{L}(\mathbf{U}_K^{(2)}) \\
\mathbf{U}_K^{(4)} &= \mathbf{M}(\mathbf{U}_K^{(1)} + \mathbf{U}_K^{(2)} + \mathbf{U}_K^{(3)} - 2\mathbf{U}_K^n, \Delta t^n/2) \\
\mathbf{U}_K^{(5)} &= \mathbf{U}_K^n + \Delta t^n \mathbf{L}(\mathbf{U}_K^{(4)}) \\
\mathbf{U}_K^{n+1} &= \frac{1}{2}\left(\mathbf{U}_K^{(5)} - \mathbf{U}_K^{(3)}\right) + \mathbf{U}_K^{(4)}
\end{aligned}
\tag{21}
$$

The explicit part of this time stepping is the same SSP-RK2 scheme as those presented above, see (20). The three stages implicit part of this time splitting is a DIRK scheme [42] respecting A-stability, L-stability and is stiffly accurate ($R(\infty) = 0$) like it is demonstrated in [43]. Such stability properties allow to deal with dynamic wet/dry front where the friction source term $\mathbf{S}_f$ can be strongly stiff as soon as the water depth is vanishing. In the following numerical experiments, this version of the DIRK scheme has found important to treat non trivial boundaries conditions without any special correction.

The fully second order accuracy presented in the sequel are obtained using all these combinations of methods and corrections.

2.6. Boundary conditions

Classically, we consider ghost cells K_e when the edge $e \in \partial\Omega$ and define the so-called ghost conservative vectors according to the desired boundary condition type. These ghost conservative vectors denoted by $\mathbf{U}_{K_e}^G$ for the first-order schme and by $\mathbf{U}_{e,K_e}^G$ for the second-order scheme are directly used in the well-balanced scheme (15). Let us point that it is quite difficult to impose directly the right numerical flux with the considered well-balanced scheme (15).

2.6.1. Wall

In case of a wall applied to a edge $e \in \partial\Omega$, the following ghost values are specified:

$$\left\{ \begin{array}{rcl} h_{K_e}^G &=& h_K + z_K - z_{K_e} \\ \hat{u}_{K_e}^G &=& - \hat{u_K} \\ \hat{v}_{K_e}^G &=& \hat{v}_K \end{array} \right. \tag{22}$$

The normal velocity is setted to zero and it is consider a slip condition for the tangent velocity.

2.6.2. Inflow discharge when $Fr < 1$

In order to impose a discharge Q_{in} at a subset Γ_{in} of the computational domain boundary $\partial\Omega$, we set the ghost cell values as follows:

$$\left\{ \begin{array}{rcl} h_{K_e}^G &=& h_K \\ \hat{u}_{K_e}^G &=& \dfrac{Q_{in} h_K^{2/3}}{\sum\limits_{e \in \Gamma_{in}} m_e h_K^{5/3}} \\ \hat{v}_{K_e}^G &=& \hat{v}_K \end{array} \right. \tag{23}$$

Such conditions are related to the distribution of the discharge Q_{in} along the edges $e \in \Gamma_{in}$. From a continuous point of view, the discharge is distributed such that $Q_{in} = \displaystyle\int_{\Gamma_{in}} c\, h^{5/3}\, dx$, where c is considered constant all along the subset Γ_{in} in order to close the problem. The power law $q = c\, h^{5/3}$ indicates that it is made the hypothesis that we are always close to a stationnary solution with a balance between the gravitational and the friction source terms. Back to a discrete point of view, the discrete ghost state respects the relations,

$$Q_{in} = \sum_{e \in \Gamma_{in}} m_e \hat{q}_{K_e}^G \quad \text{and} \quad \hat{q}_{K_e}^G = c\, h_K^{5/3}$$

Nevertheless, if the wet surface at inflow Γ_{in} is non trivial, a wrong inflow discharge value Q_{in} (see later) can be obtained since c is no longer a constant (the local slopes can vary in a significant range). In order to overcome this problem, a solution is to apply a feedback process to the ghost bathymetry values as follows:

$$z_b^{new} \;=\; z_b^{old} + c\left(\mathbf{F}_e^h(\mathbf{U}_K^n, \mathbf{U}_{K_e}^n, \mathbf{n}_{e,K}) - \hat{q}_{K_e}^G\right) \tag{24}$$

This process corrects the ghost bathymetry z_b^G along time steps and one obtain very fast a value such that the mass flux $\mathbf{F}_e^h(\mathbf{U}_K^n, \mathbf{U}_{K_e}^n, \mathbf{n}_{e,K})$ is strictly equal to the imposed lineic discharge $\hat{q}_{K_e}^G$.

2.6.3. Outflow control when $Fr < 1$

In order to control at the outflow (on Γ_{out}, a subset of the computational domain boundary $\partial\Omega$), either a water depth or a rating curve $Q_{out}(h)$ are imposed. The ghost cells values write:

$$\left\{ \begin{array}{rcl} h_{K_e}^G & = & h_{user} \text{ or } f_{user}(Q_{out}) \\ \hat{u}_{K_e}^G & = & \hat{u}_K + 2\left(\sqrt{g\,(h_K - h_K^G)}\right) \\ \hat{v}_{K_e}^G & = & \hat{v}_K \end{array} \right. \tag{25}$$

where $f_{user}(Q_{out})$ is a given rating curve. While imposing a water depth in the ghost cell directly (or interpolated from a tabulated rating curve $\eta(Q_{out})$ with $Q_{out} = \sum_{e \in \Gamma_{out}} m_e\ \mathbf{F}_e^h(\mathbf{U}_K^n, \mathbf{U}_{K_e}^n, \mathbf{n}_{e,K})$), it is specified that the Riemann invariant is conserved along each normal edge. The conditions (25) allow to control the flow at Γ_{out} while introducing a minimum of perturbations in the numerical solution.

2.6.4. Second-order scheme case

In the case of the second-order scheme, one needs to reconstruct the conservative variable vector $\mathbf{U}_{e,K_e}^G$ at ghost side of edge $e \in \partial\Omega$. First, $\mathbf{U}_{K_e}^G$ is computed according to the previous conditions (22), (23) and (25) and used in the least square method (8) to compute the vectorial slopes in the interior cell K. Next, the same previous conditions are used a second time replacing $\mathbf{U}_K$ by $\mathbf{U}_{e,K}$ to obtain $\mathbf{U}_{e,K_e}^G$. Finally, the hydrostatic reconstructed water depth $\mathbf{U}_K^{G^*}$ and $\mathbf{U}_{e,K}^{G^*}$ are computed in the same way through equations (13) and (14) with $z_{e,K_e}^G = z_{e,K_e}$.

2.7. Stability criteria

The CFL-like condition for the time step Δt^n is, see e.g. [9],

$$\Delta t^n = c\ \min_{K \in \Omega} \left(\frac{2\ m_K}{m_{\partial K}\left(\|\mathbf{u}_K^n\| + \sqrt{g h_K^n}\right)} \right) \tag{26}$$

with a constant $c \in [0.5, 0.9]$ in practice.

For numerical purpose, the eventual round-off error generating a negative water depth is elimitated at the end of each forward Euler time step $h_K^{n+1} = \max(0, h_K^{n+1})$.

The forthcoming numerical experiments showed that no significant cut-off of the water depth is required while using the present numerical scheme; a cut-off value h_ϵ has been fixed very small (about 10^{-10}), most often to the machine precision 10^{-14}.

3. Inverse Model

In this section, the adjoint-based method used to compute variational sensitivities hence conduct sensitivity analyses is presented. Next, the full optimization process used to calibrate the flow model by identifying its uncertain or unknown input parameters is described into details. As mentioned previously, the optimal control - adjoint based method is classical in the data assimilation community; nevertheless, its fine understanding and implementation always remains a challenge in an application context. To our best knowledge, the present accurate, "automatically" derived and fully assessed approach is still innovative in the numerical river hydraulics research community, in particular if based on the accurate and robust numerical scheme described previously. Variational sensitivity results remain very promising in view to improve surface water flow analyses. The algorithms described below are implemented into the computational software DassFlow, [26, 25, 27].

First, the inverse problem is stated as a optimal control problem. Since it is solved using the adjoint model, since the latter is obtained from the direct source code differentiation (automatic differentiation using Tapenade software, [24]), the link from the exact differential of the cost function and the adjoint code automatically generated is highlighted. Next, the so-called variational sensitivities are defined as being the gradient of the cost function; it is local information in the sense the gradient values are valid at a given control parameter value only (first order Taylor's expansion). Next, the variational data assimilation process (also called 4D-var in the literature) is described. The latter allows to calibrate the model by identifying the input parameters making fit the model with the observations (eg elevation water surface values).

Lest us point out that in the appendix, the process to valid accurately the direct code - adjoint code and the corresponding gradient are presented. In addition, few fundamental tricks to implement the process in a MPI context are presented; also few tricks to make decrease the (huge) memory required intrinsically by the automatic differentiation.

3.1. The variational inverse problem

The so-called "forward code" solves the 2D shallow-water equations (1) and computes the model's response j (a scalar valued function). The model's response j depends on the input parameters of the model. The input parameters are denoted by $\mathbf{k}$, it is the control vector. It can include scalar value or spatially distributed variables; typically $\mathbf{k}$ can contain the inflow discharge, the outflow boundary conditions (eg discharge, rating curve parameter), the Manning-Strickler roughness coefficients (spatially distributed quantity), the bathymetry, the initial condition or a mix of all these "parameters".

In a data assimilation context, $j(k)$ measures the misfit between the numerical solution and the observations; it is the cost function. Otherwise, j can be defined from the state of the system only, or its derivative, for a stability analysis purpose for example. The reader can refer to [44, 45] educational resources for more details on the approach.
A typical cost function in the present data assimilation context writes as follows:

$$j(\mathbf{k}) = \int \|\phi^{obs} - \phi(\mathbf{k})\|_*^2 \, d\mathbf{x} + \text{ regularization terms} \tag{27}$$

In the present context, the quantity ϕ denotes either the water elevation, or the water depth or the discharge, depending on the observations available. Subscript ^{obs} denotes an observed quantity. Usually, the regularization term is a quadratic term in $\mathbf{k}$ or its derivatives, hence regularizing and locally "convexifying" the cost function (Tykhonov's regularization). Also the regularization term is used to introduce a "good" a-priori on the optimal values sough.
The data assimilation problem reads as an optimal control problem as follows :

$$\min_{\mathbf{k}} \; j(\mathbf{k}) \tag{28}$$

where $\phi(\mathbf{k})$ is deduced from the solution of the forward model (1) at $\mathbf{k}$ given.

Calibrating the model or identifying some parameters consists to solve this optimization problem. It is done classically using descent algorithms (typically quasi-newton algorithm). Thus one need to compute the gradient of j with respect to $\mathbf{k}$. The latter is performed by introducing the adjoint model in order to obtain all partial derivatives of j (with respect to all components of $\mathbf{k}$) in one extra model resolution.

3.2. Adjoint method

3.2.1. Source code differentiation and gradient

Let us describe the link between the forward code, the cost function j, the adjoint code generated automatically using an automatic differentiation software source-to-source and the resulting gradient.
For a sake of simplicity, the link is described in the case the input parameters are: 1) the initial condition; 2)the inflow discharge (boundary control, time dependent); 3) the Manning-Strikler roughness coefficient n (time-independent, spatially distributed coefficient). In this case: $\mathbf{k} = (y(0); q_{in}, n)$ with $y_0 = (h_0, \mathbf{q}_0)$.
The total differential $dj(\mathbf{k})$ of the cost function $j(\mathbf{k})$ writes as follows :

$$dj(\mathbf{k}) \;=\; \frac{\partial j}{\partial y_0}(\mathbf{k}) \cdot \delta y_0 + \frac{\partial j}{\partial q_{in}}(\mathbf{k}) \cdot \delta q_{in} + \frac{\partial j}{\partial n}(\mathbf{k}) \cdot \delta n \tag{29}$$

It is showed below how the output of the adjoint code generated by algorithmic differentiation, and using Tapenade software for example [24], corresponds to the partial derivatives of the cost function j with respect to the control variables. The presentation below follows those presented in [46].

Let $\mathcal{K}$ be the space of control variables and $\mathcal{Y}$ the space of the forward code response. The direct code can be represented as the operator: $\mathcal{M} : \mathcal{K} \longrightarrow \mathcal{Y}$ with:

$$Y = \big(y, j\big)^T$$

Let us point out that both the state and the cost function of the system are included into the response of the forward code.

The tangent model writes: $\frac{\partial \mathcal{M}}{\partial \mathbf{k}}(\mathbf{k}) : \mathcal{K} \longrightarrow \mathcal{Y}$. It takes as input variable a perturbation of the control vector $d\mathbf{k} \in \mathcal{K}$, then it gives the variation $dY \in \mathcal{Y}$

as output variable:

$$dY = \frac{\partial \mathcal{M}}{\partial \mathbf{k}}(\mathbf{k}) \cdot d\mathbf{k}$$

The adjoint model is defined as the adjoint operator of the tangent model. This can be represented as follows: $\left(\frac{\partial \mathcal{M}}{\partial \mathbf{k}}(\mathbf{k})\right)^* : \mathcal{Y}' \longrightarrow \mathcal{K}'$. It takes $dY^* \in \mathcal{Y}'$ an input variable and provides the adjoint variable $d\mathbf{k}^* \in \mathcal{K}'$ at output:

$$d\mathbf{k}^* = \left(\frac{\partial \mathcal{M}}{\partial \mathbf{k}}(\mathbf{k})\right)^* \cdot dY^*$$

Next, the link between the adjoint code and the "computational" gradient is as follows. By definition of the adjoint operator, we have:

$$\left\langle \left(\tfrac{\partial \mathcal{M}}{\partial \mathbf{k}}\right)^* \cdot dY^*,\, d\mathbf{k} \right\rangle_{\mathcal{K}' \times \mathcal{K}} = \left\langle dY^*,\, \left(\tfrac{\partial \mathcal{M}}{\partial \mathbf{k}}\right) \cdot d\mathbf{k} \right\rangle_{\mathcal{Y}' \times \mathcal{Y}} \tag{30}$$

or, using the relations presented above:

$$\left\langle d\mathbf{k}^*,\, d\mathbf{k} \right\rangle_{\mathcal{K}' \times \mathcal{K}} = \left\langle dY^*,\, dY \right\rangle_{\mathcal{Y}' \times \mathcal{Y}} . \tag{31}$$

If we set $dY^* = (0, 1)$ and by denoting the perturbation vector $d\mathbf{k} = (\delta y_0, \delta q_{in}, \delta n)$, we obtain:

$$\left\langle \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \begin{pmatrix} dy^* \\ dj^* \end{pmatrix} \right\rangle_{\mathcal{Y}' \times \mathcal{Y}} = \left\langle \begin{pmatrix} \delta y_0^* \\ \delta q_{in}^* \\ \delta n^* \end{pmatrix}, \begin{pmatrix} \delta y_0 \\ \delta q_{in} \\ \delta n \end{pmatrix} \right\rangle_{\mathcal{K}' \times \mathcal{K}}$$

Moreover, by definition:

$$dj = \tfrac{\partial j}{\partial y_0}(\mathbf{k}) \cdot \delta y_0 + \tfrac{\partial j}{\partial q_{in}}(\mathbf{k}) \cdot \delta q_{in} + \tfrac{\partial j}{\partial n}(\mathbf{k}) \cdot \delta n$$

Therefore, the adjoint variable $d\mathbf{k}^*$ (output of the adjoint code with $dY^* = (0, 1)$) corresponds to the partial derivatives of the cost function j:

$$\tfrac{\partial j}{\partial y_0}(\mathbf{k}) = y_0^* \qquad \tfrac{\partial j}{\partial n}(\mathbf{k}) = n^* \qquad \tfrac{\partial j}{\partial q_{in}}(\mathbf{k}) = q_{in}^*$$

In summary, in order to compute the "computational" gradient (partial derivatives of the cost function J using differentiation of the forward code), first, the direct code is runed, second the adjoint code is runed with $dY^* = (0, 1)$ as input.

The adjoint code obtained via automatic differentiation costs approximatively 3-7 times the direct code. A basic calculation shows that a factor 3 is a minimum, while in practice a factor 6 times is not bad.
In the appendix, the process to valid accurately the direct code - adjoint code and the corresponding gradient are presented.
The adjoint code obtained by automatic differentiation is reverse in the memory path, and it requires a very large amount of memory. Furthermore, the variational data assimilation process described below requires a large number of minimization iterates, typically 10-50. Therefore, in practice HPC codes are required combined with a kind-of optimization of the memory management. In the appendix, few fundamental tricks to implement the process in a MPI context are presented; also few tricks to make decrease the (huge) memory required intrinsically by the automatic differentiation. Finally, the speed-up curve obtained in DassFlow case is presented.

3.2.2. Variational sensitivities

Given a perturbation of the control vector $d\mathbf{k} \in \mathcal{K}$, we have:

$$j(\mathbf{k} + d\mathbf{k}) \approx j(\mathbf{k}) + \nabla j(\mathbf{k}).d\mathbf{k}$$

at first order (Taylor's expansion). Thus, the gradient value $\nabla j(\mathbf{k})$ provides a local sensitivity of the cost function (output of the model) with respect to the input parameters. In others words, the ith value $\frac{\partial j}{\partial k_i}(\mathbf{k})$ gives the sensitivity of the response of the model with respect to the ith parameter, eg the Manning-Strickler coefficient at one location point. The sensitivities obtained are local since they are valid at a given point $\mathbf{k}$ only. The resulting sensitivity analysis tool is an important feature which provides a *better understanding of both the physics and the model* by quantifying the roles of the various physical parameters and the influences of parameter variations on the behavior of the system.

3.3. Data assimilation and twin experiments

As mentioned previously, the (variational) data assimilation problem consists to solve (28). The minimizing procedure operates on the control vector $\mathbf{k}$ to generate a set of parameters which allows to obtain a computed model output closer to the observations. In the computations presented in next section, the classical quasi-newton L-BFGS algorithm is employed. More

precisely, the L-BFGS algorithm implemented in the M1QN3 routine (see [47]) is called ate each minimization iterate. The global optimization process is represented in Figure 3.

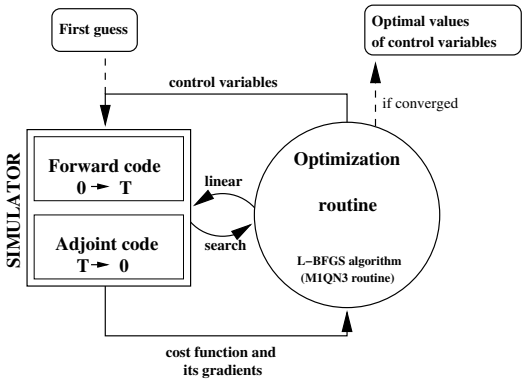


Figure 3: Principle of a 4D-Var type variational data assimilation algorithm.

Let us point out that each variable of the control vector may be actived or not, as an actual control variable. In practice, it is possible to identify only few of them at same time and depending on the observations available of course.

The methodology of twin experiments is used as a first step toward real data processing and calibration model. Twin experiments are designed as follows : the reference parameters of the model k_{ref} are used to generate the observations y_{obs}. Next, the goal is to retrieve the set of parameters k_{ref} starting from an initial guess $k \neq k_{ref}$ and using the minimization process described above. Thus, the algorithm computes at each iteration l a new set of parameters k^l according to the gradient $\frac{\partial j}{\partial k}$ in order to make decrease the cost function value $j(k^l)$.

Twin experiments are an ideal framework but they are a crucial step to better understand and assess the particular inverse problem addressed. While performing twin experiments, random Gaussian noise is added to the synthetic data in view to define real-like test cases.

4. Finite Volume Solver Benchmarks

The present section aims at assessing the accuracy and stability, hence the robustness, of the numerical solvers elaborated in Section 2, in particular in presence of the friction term and wet-dry fronts. We consider the following classical test cases from the literature : the parabolic bowl, and two dam-break test cases: a regularized version and a more classical case (those with the advancing wet-dry front). Both are considered with the realistic Manning-Strickler friction term.
The convergence rate of the methods are carefully investigated. As expected, the actual convergence rates depends on the regularity of the solution sought (it is smooth or not). The present choice of test cases and the computations performed below give detailed information on the actual convergence rates of for each 'global' scheme (i.e. in space and time and including all their ingredients). Let us notice that the same questionings as above have been addressed for example in [48]. Nevertheless the present numerical investigations are wider and are led more into details, in particular in terms of convergence rates.

A fourth test case, relevant in the context of geophysical flows, is explored. It is an open channel flow with a real like topography. The goal is on one hand to compare the performances of first and second order schemes; on the other hand to explore the adequacy required between the topography representation scale and the mesh size, in terms of resulting accuracy, see Fig. 11.

Let us point out that the realistic Manning-Stricker friction formulation is used in all cases, except in the bowl case where the classical analytical solution is used. The results obtained are as follows.
In the dam-break test case, regularized version (the solution is smooth, no front, but there is the Manning-Strickler friction term), the obtained rate of convergence for IMEX-SSP(3,2,2) scheme is about 2 while for RK-SSP2 scheme, it equals to 1 only, see Tab. 2. Actually, the latter scheme is a fake order two scheme.
In the parabolic bowl case, the convergence rate must intrinsically be deteriorated; in the present experiments, while using the fully second order scheme IMEX-SSP(3,2,2), the obtained rate of convergence is about 1.5, while if using the second order scheme RK-SSP2, the convergence rate equals 1 only,

see Tab. 3.
In the classical dam-break test case, like previously and since the presence of the wet-dry front, the convergence rate is intrinsically deteriorated. The numerical experiments demonstrate an order 1 for all schemes, see Tab. 3. Nevertheless, the second order schemes, in particular the actual second order scheme, the accuracy is much higher than those obtained with the first order scheme, even if its convergence rate equals 1, see Fig. 9.
Finally, we point out that the present dam break test case, with the realistic friction formulation (and not a linear one for example) and the wet-dry front, is a good benchmark to assess actual robustness and accuracy of a numerical scheme dedicated to flows in presence of wet-dry fronts (such as in flood plain dynamics or coastal flows).

Notations. In all the numerical tests, the following error norms are considered:

$$e_p(x) = \frac{\|x^{\text{num}} - x^{\text{exact}}\|_p}{\|x^{\text{exact}}\|_p} \quad , \quad e_p^T(x) = \frac{1}{T}\int_0^T e_p(x)\ dt$$
$$\|x\|_p = \left(\sum_{K\in\Omega} m_K\,|x_K|^p\right)^{1/p} \tag{32}$$

where x^{num} is the numerical solution and x^{exact} is either the exact solution if available, or a "converged reference solution" computed on a extremely fine mesh (12800 cells) if not.

The well-balanced property has been verified successfully for two-dimensional test cases of water at rest, for very large simulation times, in presence of very irregular topographies and in presence of wet/dry fronts.

4.1. Parabolic bowl

The analytical solution of the one-dimensional shallow water equations taking into account a linear friction term and a parabolic bottom elevation has been derived by Sampson [49], following the work of Thacker [50]. The solution presents a flat surface dynamics with wet/dry fronts decaying over time because of the friction. It is a famous test case for shallow water numerical solvers, see e.g. [13, 39, 51, 52, 53].

In a parabolic bowl such that $z_b(x) = h_0(x/a)^2$, the solution writes:

$$\begin{aligned}
\eta(x,t) &= h_0 + \frac{a^2 B^2 e^{-\tau t}}{8g^2 h_0}\left(-s\tau \sin 2st + \left(\frac{\tau^2}{4} - s^2\right)\cos 2st\right) \\
&\quad -\frac{B^2 e^{-\tau t}}{4g} - \frac{e^{-\tau t/2}}{g}\left(Bs\cos st + \frac{\tau B}{2}\sin st\right) x \qquad (33) \\
u(x,t) &= Be^{-\tau t/2}\sin st
\end{aligned}$$

with $p = \sqrt{8gh_0}/a$ and $s = \sqrt{p^2 - \tau^2}/2$ (and then valid if $p > \tau$). Considering a computational domain lenght $l_x = 10000\ m$, the classic set of parameters is:

$$\left\{\begin{array}{rcl} h_0 &=& 10\ m \\ a &=& 3000\ m \\ B &=& 5\ m.s^{-1} \\ \tau &=& 0.001\ s^{-1} \end{array}\right. \qquad (34)$$

We plot in Fig.4 and give in Tab.1 the computed error norms $e_1^T(h)$ using this classic set of parameters (34). A cut-off water depth $h_\epsilon \in [10^{-8}, 10^{-10}]$ and the Barth limiter for the MUSCL reconstructions were used to find a proper convergence behavior. Results show that the moving wet-dry fronts are well captured.

	Euler		RK-SSP2		IMEX-SSP(3,2,2)	
3200	$7.89\ 10^{-4}$	-	$2.01\ 10^{-5}$	-	$1.06\ 10^{-5}$	-
6400	$3.92\ 10^{-4}$	1.01	$8.89\ 10^{-6}$	1.18	$3.35\ 10^{-6}$	1.66
12800	$1.95\ 10^{-4}$	1.01	$4.17\ 10^{-6}$	1.09	$1.09\ 10^{-6}$	1.62
25600	$9.73\ 10^{-5}$	1.00	$2.01\ 10^{-6}$	1.05	$3.72\ 10^{-7}$	1.55
51200	$4.86\ 10^{-5}$	1.00	$9.90\ 10^{-7}$	1.02	$1.35\ 10^{-7}$	1.46

Table 1: Parabolic bowl test case: relative error norms $e_1^T(h)$ and associated convergence rates.

While the cells number is lower than 1000, we do not observe any significant difference between the two time stepping methods despite the fact that the RK-SSP2 is first-order and the IMEX-SSP(3,2,2) second-order. That

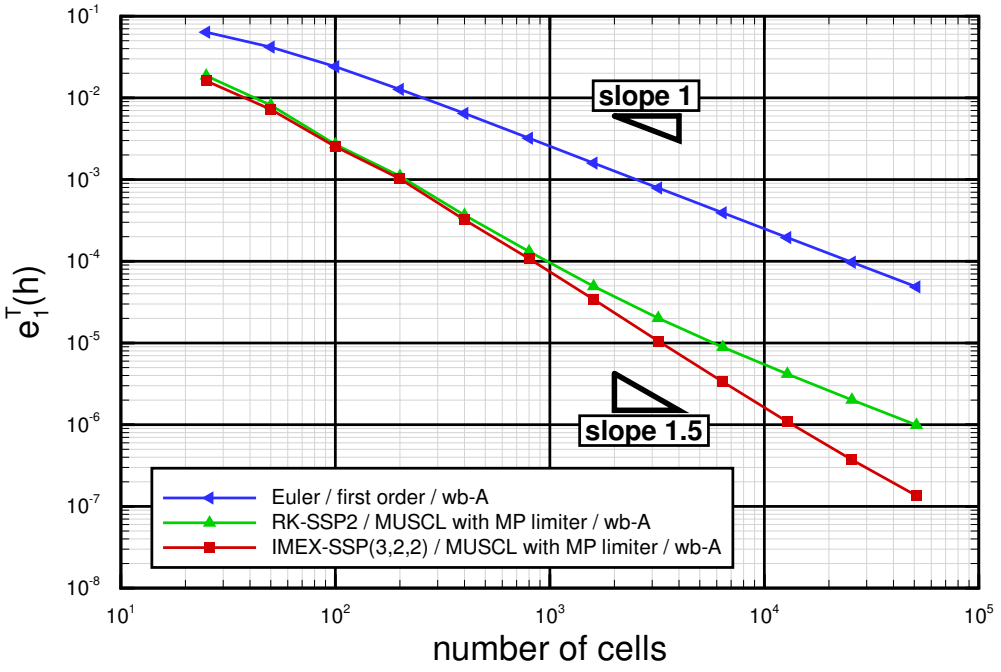


Figure 4: Parabolic bowl test case with the set of parameters (34): relative error norms $e_1^T(h)$ *vs* number of cells.

means that the spatial consistency error is much greater than the time consistency error in this mesh size range. The order of convergence is about 1.5 and not 2 due to the presence of the two moving wet/dry fronts introducing singularities in the solution. Let us note that the relative error norm $e_1^T(h) \simeq 2.5\ 10^{-3}$ for a cells number of 100, which is approximatively the same as the one presented in [39].
Next, if the number of cells is larger than 1000, computations show a significant asymptotic difference between the two time stepping methods. The numerical solver with the RK-SSP2 time stepping method is clearly first-order accurate while a convergence rate upper than one is hold if using the IMEX-SSP(3,2,2) time stepping method.

In conclusion, this test case widely presented in the literature is interesting to assess the numerical solver robustness in presence of dynamic wet/dry fronts, nevertheless it is insufficient to be completely relevant. The correct asymptotic convergence rate is obtained for very large mesh sizes and the singularities at the wet/dry fronts cannot guarantee to compute it precisely. The friction source term is linear and it do not correspond to the original model where the Manning-Strickler friction term is non-linear.

4.2. Dam breaks

The numerical solver behavior is investigated for dam break test cases with a non zero slope and taking into account the non-linear Manning-Strickler friction source term. In a first test case, a regular test case is considered eliminating all eventual singularities in the solution in order to find the right convergence rate. In a second test case, a dynamic wet/dry front is considered in order to mainly check the schemes robustness to deal with.

4.2.1. Regularized dam break

A regularized one-dimensional dam break test case is performed in the sense that all eventual singularities in the solution are eliminated by construction. The following initial condition is considered:

$$\left\{\begin{array}{rcccl} z_b(x) & = & & 0.5 & e^{-(x-l_x/2)^2/2\sigma^2} \\ h(x,t=0) & = & 0.1 + & 0.5 & e^{-(x-l_x/2)^2/2\sigma^2} \end{array}\right. \tag{35}$$

with $\sigma = 100\,m$, a computational domain lenght $l_x = 1000\,m$, a Manning-Strickler roughness coefficient $n = 0.05$, a gravitational acceleration $g =$

$10\ m.s^{-1}$ and a simulation time $t = 100\ s$. A reference simulation (computed on an extremely small grid and considered as exact) is first performed with a mesh of 12800 cells. Initial condition and converged results for the water depth h and the 1d discharge q are plotted in Fig.5. We can indeed see that there is no singularity in the initial condition (two centered gaussians), nor at boundary conditions and nor in the final solution (no hydraulic jump is generated). Let us note that the parameters are suitable in the present context of hydraulic. The topography gradient is chosen not constant in order to avoid an exact topography gradient computation.

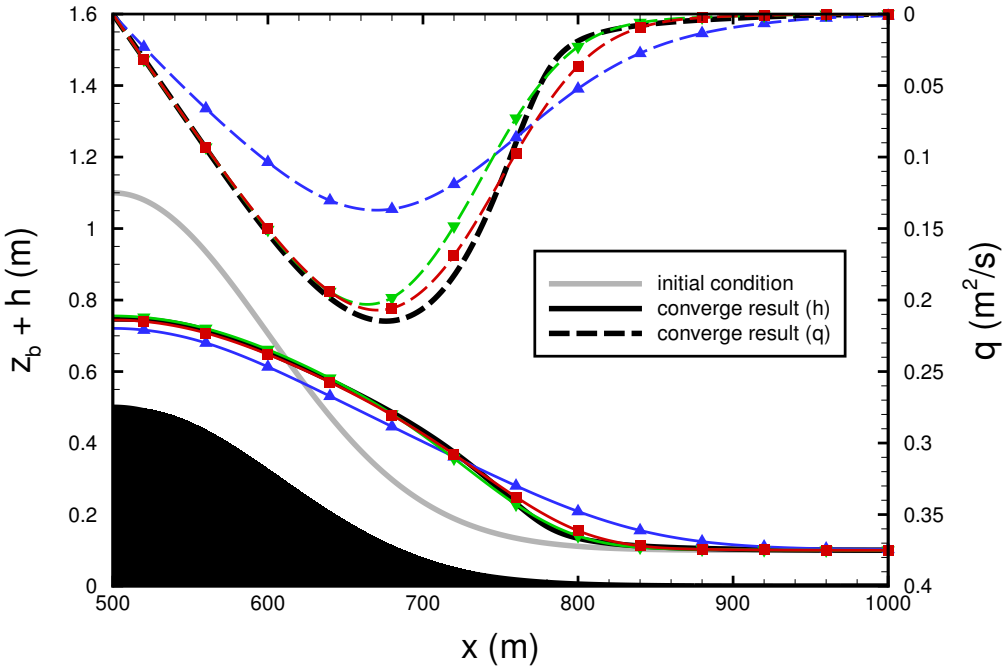


Figure 5: Regularized dam break test case: half-domain initial condition, reference simulations for h and q using a 12800 cells mesh and simulation with a 25 cells mesh.

The computed relative error norms $e_1(h)$ and $e_1(q)$ plotted in the Fig.6 and given in Tab.2 give very clear convergence rates. The RK-SSP2 time stepping method is only first-order accurate with an enhanced accuracy in

comparison with the first-order method in space and time. About the IMEX-SSP(3,2,2) time stepping method, a second-order rate of convergence is found for the water depth h and the 1d discharge q. As a consequence, the relative error norms become very small even for very coarse meshes. This test demonstrates the global second-order accuracy of the presented well-balanced numerical solver (based on MUSCL reconstructions, an appropriate limitation and the IMEX-SSP(3,2,2) time stepping method to implicit the friction source term).

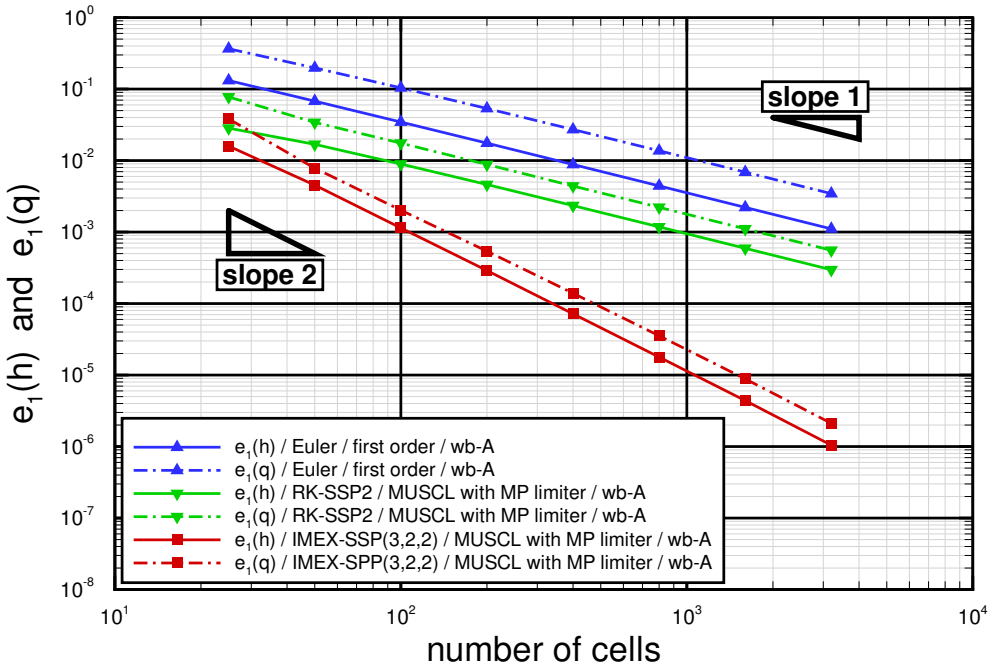


Figure 6: Regularized dam break test case: relative error norms $e_1(h)$ and $e_1(q)$ *vs* the number of cells.

4.2.2. Idealized dam break with wet/dry front

The idealized dam break test case on a constant bottom slope and involving a dynamic wet/dry front is now investigated, see Fig.7. It allows to

	Euler		RK-SSP2		IMEX-SSP(3,2,2)	
800	$4.420\ 10^{-3}$	-	$1.182\ 10^{-3}$	-	$1.783\ 10^{-5}$	-
1600	$2.213\ 10^{-3}$	1.00	$5.930\ 10^{-4}$	1.00	$4.393\ 10^{-6}$	2.02
3200	$1.107\ 10^{-3}$	1.00	$2.971\ 10^{-4}$	1.00	$1.046\ 10^{-6}$	2.07

Table 2: Regularized dam break test case: relative error norms $e_1(h)$ and associated convergence rates.

verify the numerical solver stability in presence of a dynamic wet/dry front taking into account the non-linear Manning-Strickler friction law.

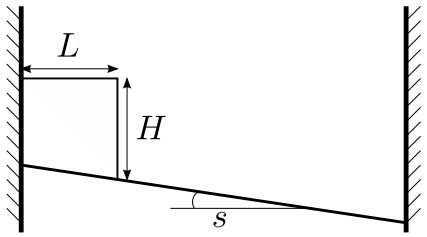


Figure 7: Idealized dam break test case sketch.

The parameters used are: a computational domain length $l_x = 1000\ m$ with wall boundaries on each side, a slope $s = 0.5$ %, an uniform Manning-Strickler roughness coefficient $n = 0.05$, a gravity acceleration $g = 10\ m.s^{-1}$ and a simulation time $t = 500\ s$. The initial water column has a length $L = 50\ m$ and a height $H = 5$ m, see Fig.7. Since no analytical solution is known, a reference solution (computed on an extremely fine grid and considered as exact) is obtained by refining the mesh up to 12800 cells, see the Fig. 8. After the initial simulation time, a rarefaction wave goes upstream and interacts with the left wall boundary while a shock wave goes downstream, both modifying the initial water column shape. After a sufficient simulation time, the water column completely disappears and the new water shape exhibits more mass and stronger gradients downstream.

The dynamic wet/dry front is robustly captured by the well-balanced first and second-order schemes without any cut-off water depth h_ϵ (despite the fact that the well-balanced property is needless for this test case). The numerical solutions remain strictly positive; the numerical schemes do not generate any

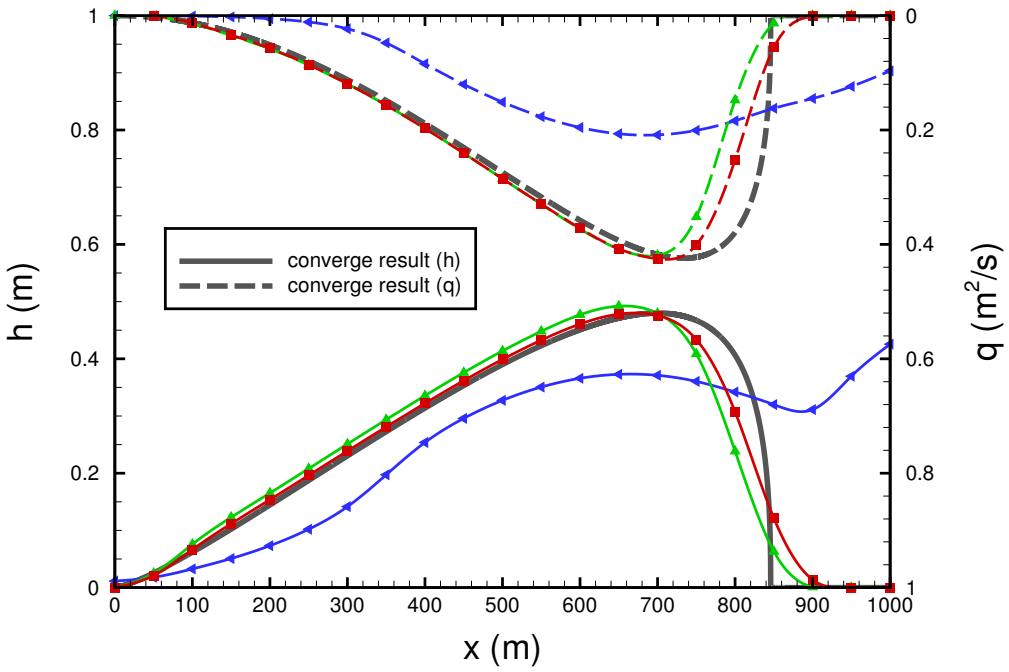


Figure 8: Idealized dam break test case: reference solutions for h and q with 12800 cells mesh and computed ones with a 20 cells mesh (a wall boundary condition has been considered at right and be carefull that the slope is not represented as sketched in the Fig.7).

negative water depth (using the MP limiter for the MUSCL reconstructions was found sufficient in this case). The asymptotic rate of convergence of the norm $e_1(h)$, see Fig. 9 and Tab.3, is as expected, equal to1 (if using the first-order schemes in space and time). This rate of convergence remains equal to 1, but with a better accuracy, if using the RK-SSP2 time stepping method with a second-order spatial accuracy (this is similar to the results obtained for the parabolic bowl 4.1 and the regularized dam break test case 4.2.1).

Using the IMEX-SSP(3,2,2) time stepping method, the accuracy is better, and the asymptotic convergence rate is a bit greater than 1. This loss of the optimal order of 2 is due to the (severe) singularity in the solution at wet/dry front.

The same idealized dam break test case has been performed but changing the initial water column height at $H = 1\ m$. Hence with the water depth closer to the local bottom variation, the schemes behavior described above are enhanced: the second order schemes are more accurate compared to the first order ones (the numerical results are not plotted here).

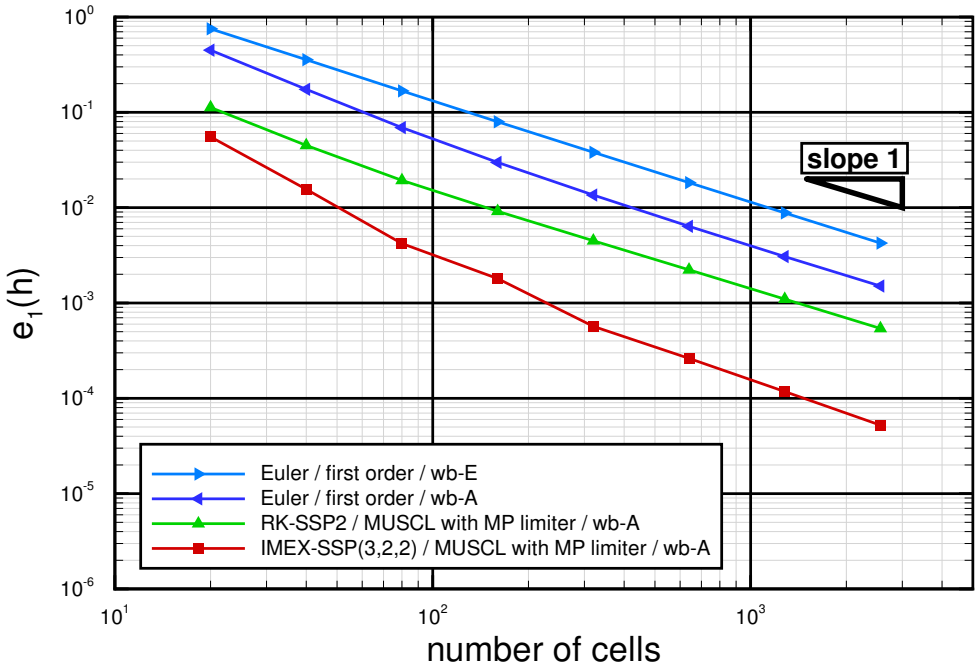


Figure 9: Idealized dam break test case: relative error norms $e_1(h)$ for the different schemes.

	Euler		RK-SSP2		IMEX-SSP(3,2,2)	
640	$6.374\ 10^{-3}$	-	$2.228\ 10^{-3}$	-	$2.623\ 10^{-4}$	-
1280	$3.067\ 10^{-3}$	1.06	$1.098\ 10^{-3}$	1.02	$1.177\ 10^{-4}$	1.16
2560	$1.506\ 10^{-3}$	1.02	$5.405\ 10^{-4}$	1.02	$5.228\ 10^{-5}$	1.17

Table 3: Idealized dam break test case: relative error norms $e_1(h)$ and associated convergence rates.

4.3. Channel flow with real-like bathymetry: topography-grid adequacy

We consider a one-dimensional open-channel flow benchmark with a "geophysical-like" topography. From a channel with a constant slope, topography variations are added with gradually increasing frequencies. The goal is to determine the minimal cell number required for each topography to retrieve a correct convergence behavior (in this simple 1d case). In other words, given the highest frequency in the topography, how many cells are needed to obtain an accurate solution. (Observe that frequencies with an amplitude substantially smaller than the water height are irrelevant in a hydraulic point of view). The numerical solver with the first-order schemes in space and time is compared to the one using the MUSCL reconstructions in space and the IMEX-SSP(3,2,2) time stepping method.

The initial channel length is $l_x = 10000\ m$ and the slope $s = 0,25\ \%$; it is a typical value of main bed river slopes. The local topography perturbations are introduced as follows:

$$z_b = z_b + \sum_{i=1}^{n_p} 0.25 \sin\left(2\pi\left(\frac{p_i\ x}{l_x} + r_i\right)\right) \quad \text{with} \quad r_i \in [0,1] \tag{36}$$

We consider 5 cases: $n_p = 2, 3, 4, 5, 6$ with $p_i = 7, 24, 57, 108, 205, 402$.

The other parameters are a Manning-Strickler roughness coefficient $n = 0.05$, a 1d discharge $q = 1\ m^2/s$ imposed at inflow and a gravitational acceleration $g = 10\ m.s^{-1}$.

Since the exact solution for the constant slope case is $h = 1\ m$ (and $u = 1\ m/s$, giving a Froude number of 0.32, the flow is fluvial), the considered topography perturbations wave amplitude have the same order of magnitude, see Fig.10. Let us note that the flow respects the shallow hypothesis since the smallest wavelength equals approximatively $25\ h$. In order to control the flow

at boundaries, and especially the backwater curve at inflow, the topography gradient is kept constant, see Fig.10. The exact water height is imposed at the outflow according to the boundary condition treatment described in previous section. The steady-state solution is computed until convergence in time is reached. The so-called reference solutions are those based on a 12800 cells mesh. It is plotted in Fig.10, the reference solution corresponds to the $n_p = 6$ case with the associated topography. It can be distinguished regular hydraulic jumps when the local slope becomes sufficiently important to find a Froude number greater than 1. Between these control sections, the water surface η is nearly flat, especially when the Froude number is small.

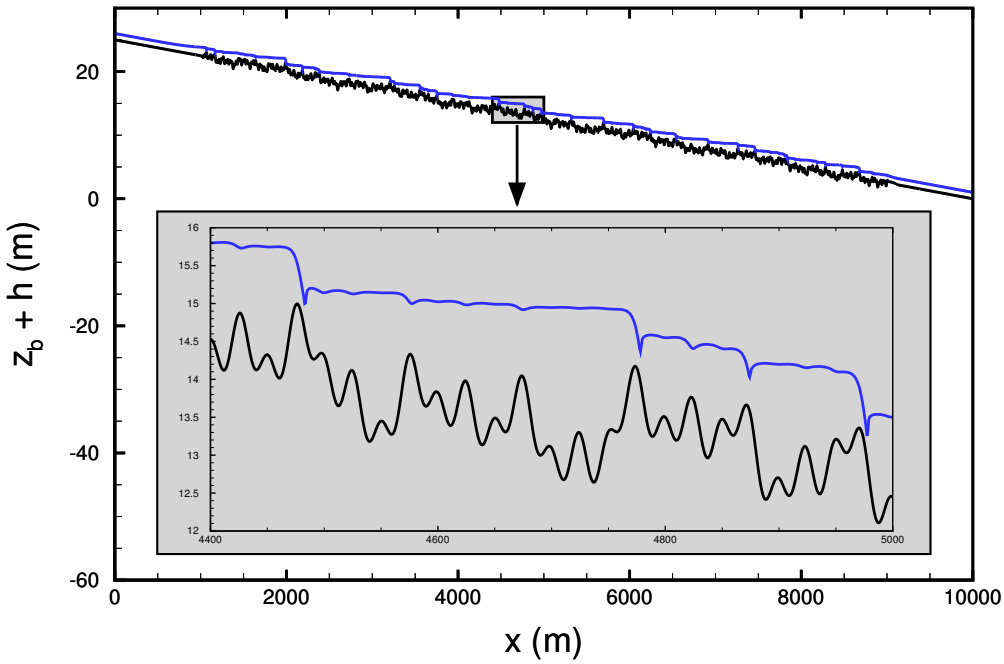


Figure 10: Open-channel flow with "real" topography: the topography variations and the computed reference solution with a 12800 cells mesh for $n_p = 6$, see (36).

We plot in Fig.11 the relative error norms $e_1(h)$ versus the mesh cell number per smallest wave lenght for the 5 cases. In order to find a correct convergence behavior, a minimum of 4 points of discretization for the highest frequency is needed independently to the lowest other frequencies. Otherwise,

it is not easy to make the difference between the solutions, with a relative error around 10 %. If this condition is realized, a second-order convergence rate is obtained if using the MUSCL reconstructions, the MP limiter and the IMEX-SSP(3,2,2) time method. (With the RK-SSP2 time-stepping method, again a first-order convergence rate only is obtained; results are not plotted here). An another remarkable result is that the curves are very close for the 5 cases if using this scaling for the topography discretization.

In order to compare the first- and second-order schemes, one should look at the zoom (not in log scale) in Fig.11 corresponding to the really interesting operating area.

If we want to obtain a relative error of 1 %, a minimum of 25 points of discretization with the first-order schemes must be used, whereas 5 points of discretization are sufficient with the second-order schemes. It gives a ratio of 5 (nearly constant for the 5 cases (36)) between the two numerical solvers.

This ratio is approximatively the same for the time computation cost. However, using more points of discretization reduces the time-step proportionally to the mesh cell size.

In conclusion, the first-order scheme is approximatively five times more expensive in time computation to find the same accuracy to obtain a relative error of 1 %.

This ratio is less important for higher error values expected.

This preliminary test gives a basic rule in the 1d case, to respect relevant ratios between the topography discretization and the mesh size.

5. Leze River : Food Plain Dynamic

In this section, direct flow computations and inverse ones are performed on the real data case: a portion of the Leze river, south-west of France. Few comparisons of the flow computed either with the first order scheme or the second order one are presented. These forward model tests highlight the high robustness of all schemes elaborated in the present context of a real-like flood plain dynamics. Also, these comparisons between the first and second order schemes demonstrate the potential advantages of second order schemes vs the first order ones. Next, the inverse capabilities of the full model are shown: sensitivity maps (gradient values spatially distributed) and data assimilation. Two twin experiments are performed: identification of the friction coefficient in 6 land covers and the inflow discharge $Q_{in}(t)$. These numerical

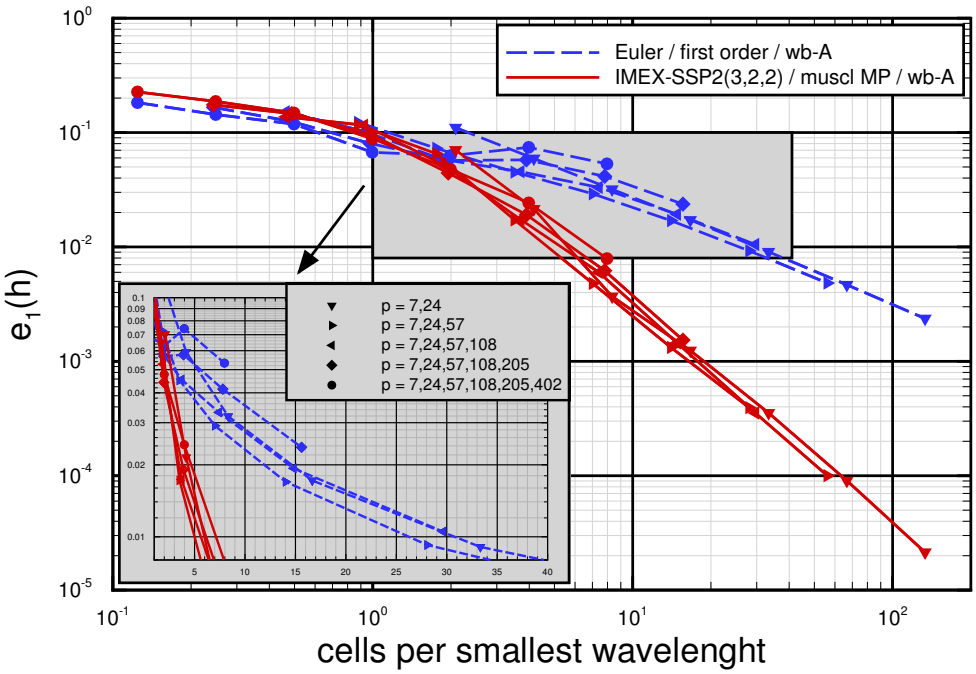


Figure 11: Open-channel flow with "real" topography: relative error norms $e_1(h)$ vs n cells per smallest wavelenght.

experiments demonstrate the capabilities of the approach, and the resulting computational software, in a real-like flood plain dynamic context. It fully assesses the robustness of the whole numerical chain elaborated : accurate, stable direct solver and gradient in presence of wet-dry front dynamics over complex topography, sensitivity analysis and calibration processes.

5.1. The river configuration

The Lèze River is a 70 km long river in southwestern of France near Toulouse. The case presented hereafter is a 2 km long subdomain centered on the hydrological station of Lezat-sur-Leze. This domain was discretized with a relatively coarse mesh of 24 632 cells. Topography for this mesh was computed using data coming from local surveys and Digital Elevation Model. This interesting real test case presents a complex topography, in particular the presence of a bridge - where the hydrological station is located - as well as non flat topography at the upstream an downstream boundaries.

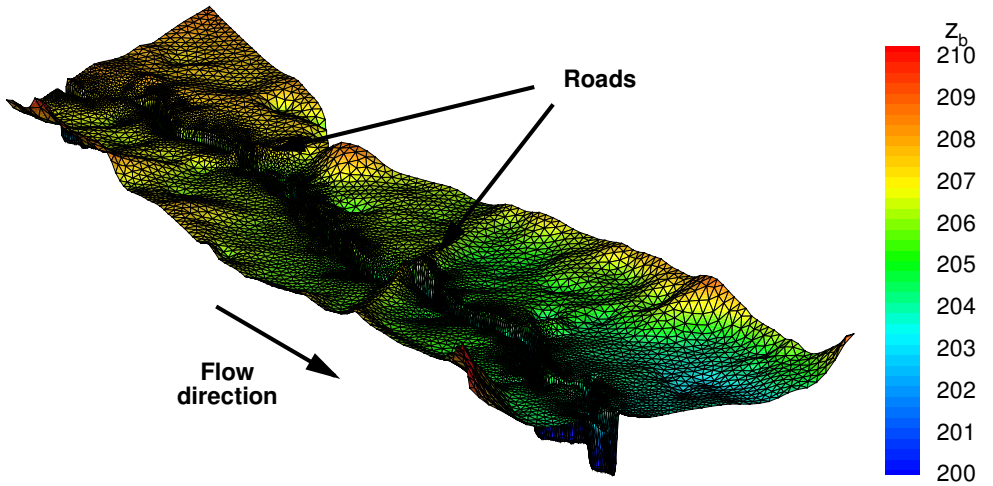


Figure 12: Leze river topography and mesh (24 632 cells)

The domain boundaries consist in two open boundaries - upstream and downstream - and wall boundary is applied elsewhere. At the upstream

boundary an inflow hydrograph (see Fig 13(a)) is imposed; this hydrograph corresponds to the exceptional event of June 2000. A rating curve is imposed at the outflow open boundary (downstream). The Manning-Strickler coefficients are defined as follows. Two scalar values of n are considered: one uniform value in the stream-bed and one uniform value elsewhere (i.e. in the floodplain). The two values of Manning friction parameters are : 0.1 $s^{1/3}.m^{-2}$ in the stream-bed and 0.05 $s^{1/3}.m^{-2}$ in the floodplain. These values have been obtained after a trial-error calibration (i.e. "by hand") leading to a good representation of the flow plain dynamics observed during the June 2000 event (no accurate measurements are available).
The initial condition is set as follows. The whole domain is considered dry, next it is gradually filled in the stream-bed during two hours ("warm-up period") to obtain a realistic initial condition (i.e. the flow state at $t = 0$).

5.2. Direct model: 1st order vs 2nd order

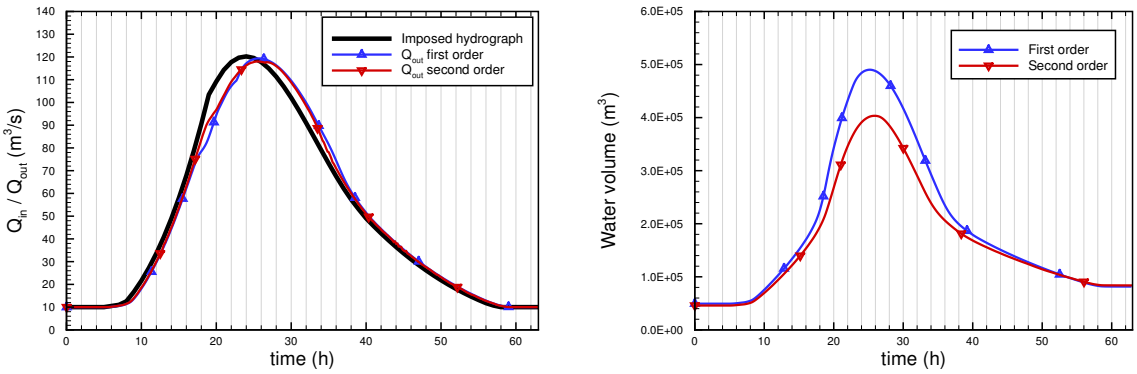
In the sequel, we denote by "1st order scheme" the Euler / A-well-balanced finite volume scheme presented (see sections 2.3.1-2.4.2); and by "2nd order scheme" the RK2 / MUSCL Barth finite volume scheme, A-well-balanced. In this section, we present few comparisons of the computed flow obtained using either the first order scheme or the second order one.

At inflow, a feedback control of the ghost cells is employed in order to compute exactly the prescribed hydrograph (see Section subsubsec:inflow).
The time simulation is T=63 hours; the max CFL number is 0.8 which corresponds to 2 880 282 time steps for the 1st order scheme and 2 872 208 time steps for the 2nd order scheme.

All the results presented here are obtained using a water depth cutoff $h_\epsilon \approx 10^{-15}$. Again, this demonstrates that the numerical schemes elaborated above remain stable even while computing wet-dry front dynamics over complex topography.

The Froude number is at maximum equal to 0.3 at flood peak. The CPU computations times on a : 14 480s (4h, 5ms per dt) for the 1sr order scheme and 61 588s (17h, 20ms per dt) for the 2nd order scheme. Roughly, the corresponding ratio 4.25 is due to a factor 2 for IMEX and a factor 2 for MUSCL and limitations.

As a first test, the hydrograph at outflow Q_{out} obtained by performing the 1st order and 2nd order schemes, are compared in Fig. 13(a). Both scheme give similar outflow discharges (excepted at t =19h, a maximal discrepancy of 8% is observed).



(a) Q_{in} and Q_{out} in time

(b) Total water volume V in time

Figure 13: Left : Prescribed hydrograph at inflow (black curve), resulting outflow hydrographs Q_{out} obtained from the 1st and 2nd order schemes. Right: Water volume in the computational domain in time (in hours): $V = \int_0^T (Q_{\text{in}} - Q_{\text{out}})\, dt$ obtained from the 1st order and 2nd order schemes.

The net mass balance (total water volume) in the computational domain is plotted in time in Fig. 13(b). The first order scheme over-estimates the peak volume by almost a factor of 20% compared to the second order scheme. This over-estimate of the net mass balance by the 1st order scheme, is due to an over estimate of the water level in the stream-bed , see Fig. 14 for the water depth h at the three "stations" and see Fig. 15 for water levels at two cross-sections. Also, the first order scheme gives slightly lower velocities in the stream-bed than those obtained using the second order scheme.

If comparing the spatially distributed water depth h at the flood peak time, both schemes give slightly different flood plain patterns (in particular downstream the crossing road), see Fig 16.

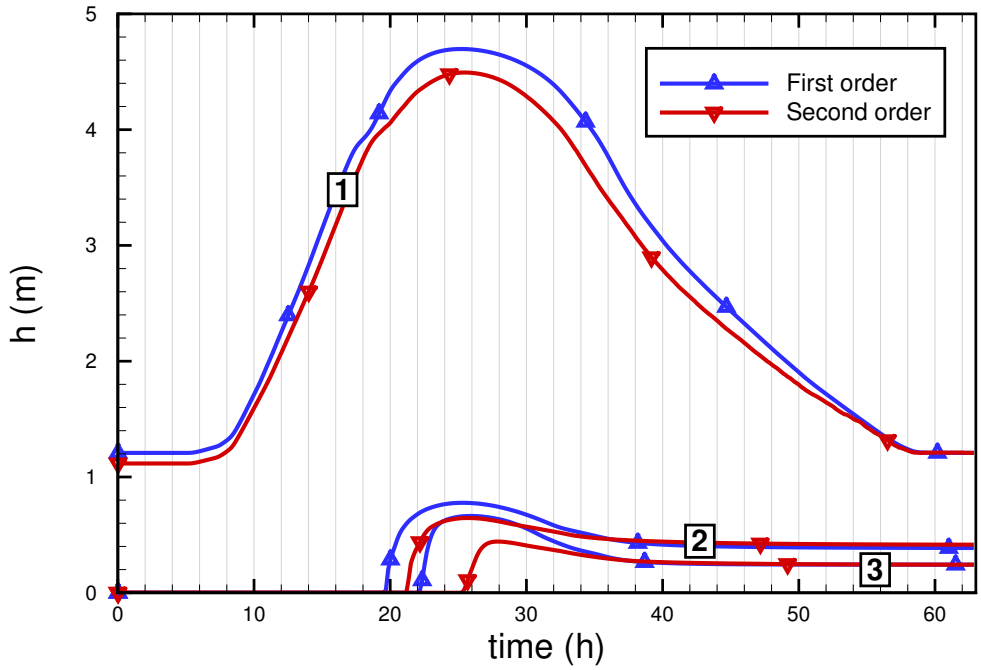
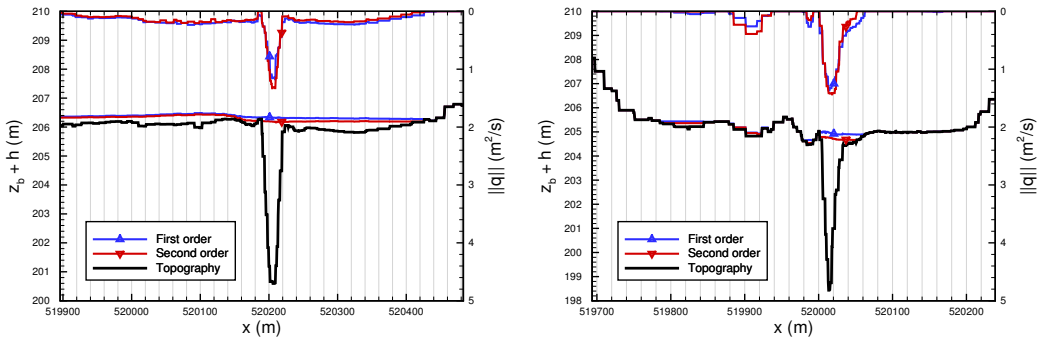


Figure 14: Time evolution of the water depth h at stations 1, 2 and 3 (see their locations in Fig.16). For each stations, the first and second order results are plotted.



(a) Section 1 (b) Section 2

Figure 15: Water surface ($z_b + h$) at time t = 26h at Section 1 (Left) and Section 2 (Right). See the section locations in Fig.16. First and second order results are plotted.

5.3. Sensitivity analysis maps

In this section, variational sensitivities are performed with respect to the Manning-Strickler roughness coefficient n and the bathymetry z_b. The forward model employed is the first order Euler / A-well-balanced scheme, see Section 2.

Boundary conditions are identical as in the previous section: the discharge is imposed at inflow and a rating curve is imposed at outflow.

First the measurements are generated by the forward model (synthetic data), next a realistic Gaussian noise is added. For all experiments, the measurements considered are time-series of water elevation at the (virtual) stations 1 and 2, see Fig. 17. The water elevation recorded are perturbed by a random noise of $+/-10cm$ (representing typical error measurements). The "station 2" measurements are based on the average of a dozen of cells values. Similarly the "station 1" measurements are the average of a dozen of cells values representing a mean cross-section value in the minor bed.

The Manning-Strickler coefficients used to generate the data are defined by areas; their values are given in Table 4.

The cost function is defined as follows:

$$j(n, z_b) = \sum_{time\ \ step\ i} \sum_{station\ \ j} (h_{i,j} - h_{i,j}^{obs})^2$$

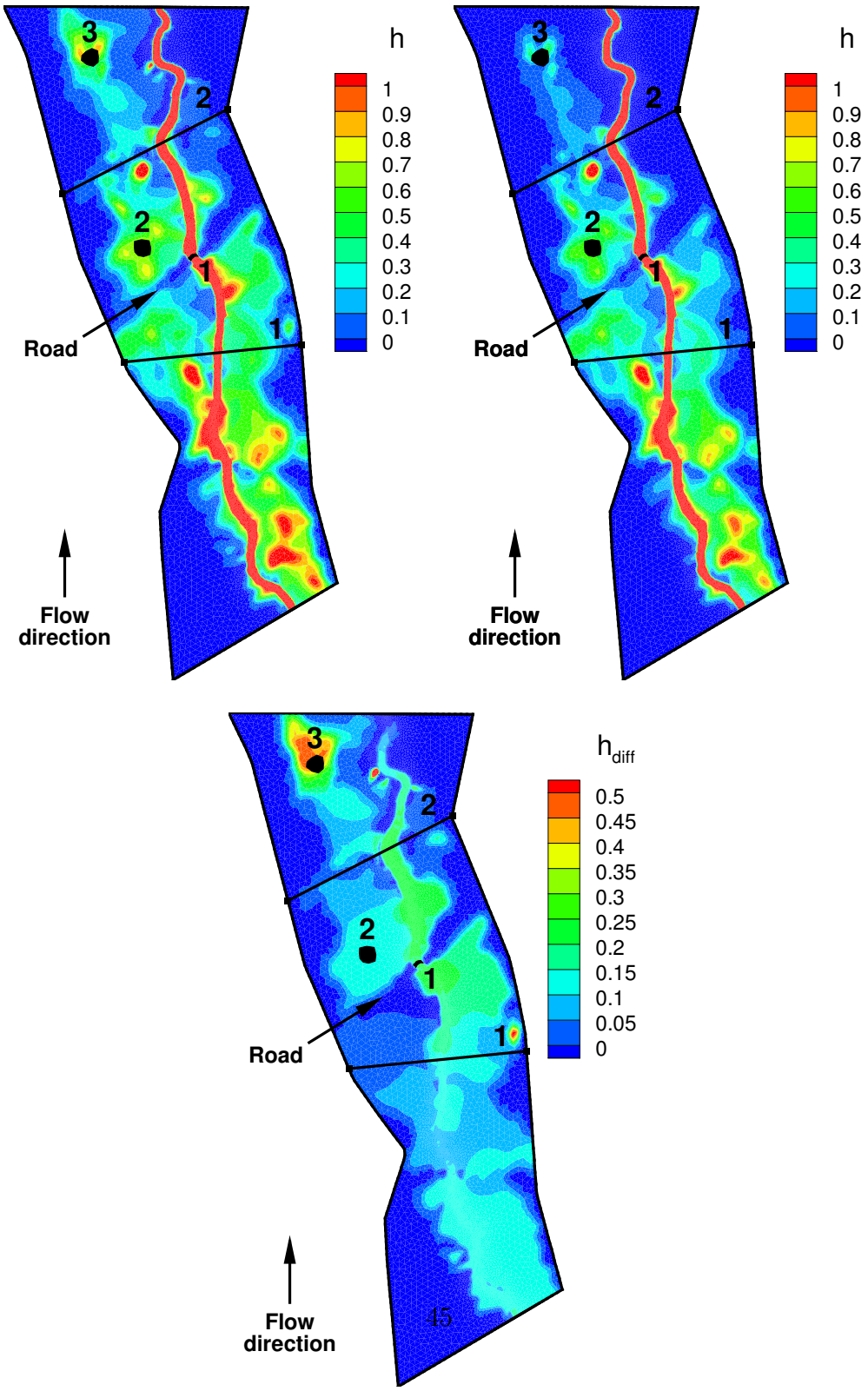


Figure 16: Top views of water depth h at $t = 26$ h (flood peak), locations of the 3 observation points/areas, locations of the 2 cross-sections. (Up left) first-order computed depth h. (Up right) second order computed depth h.

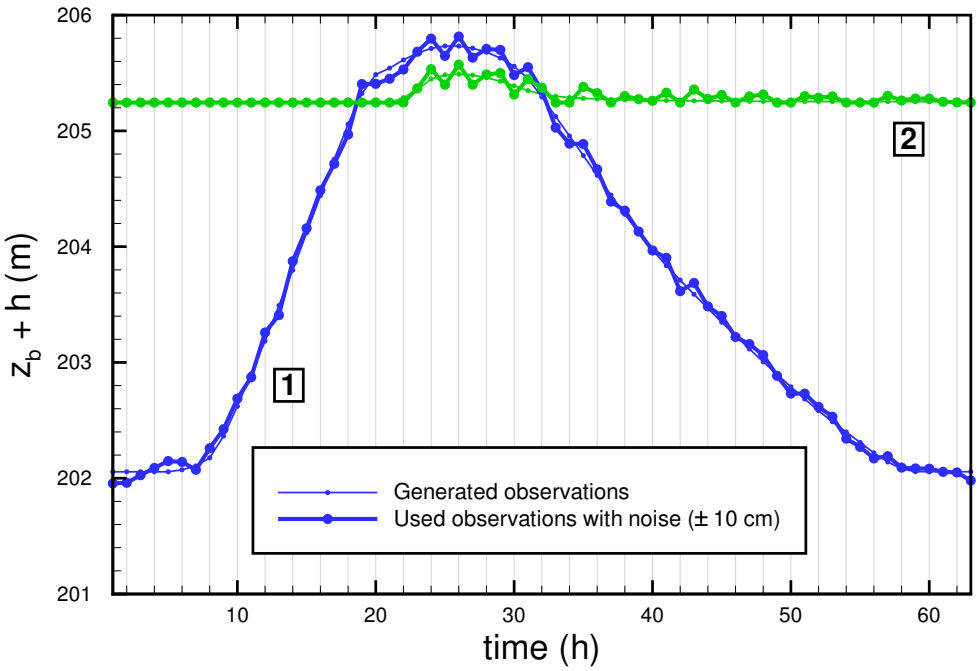
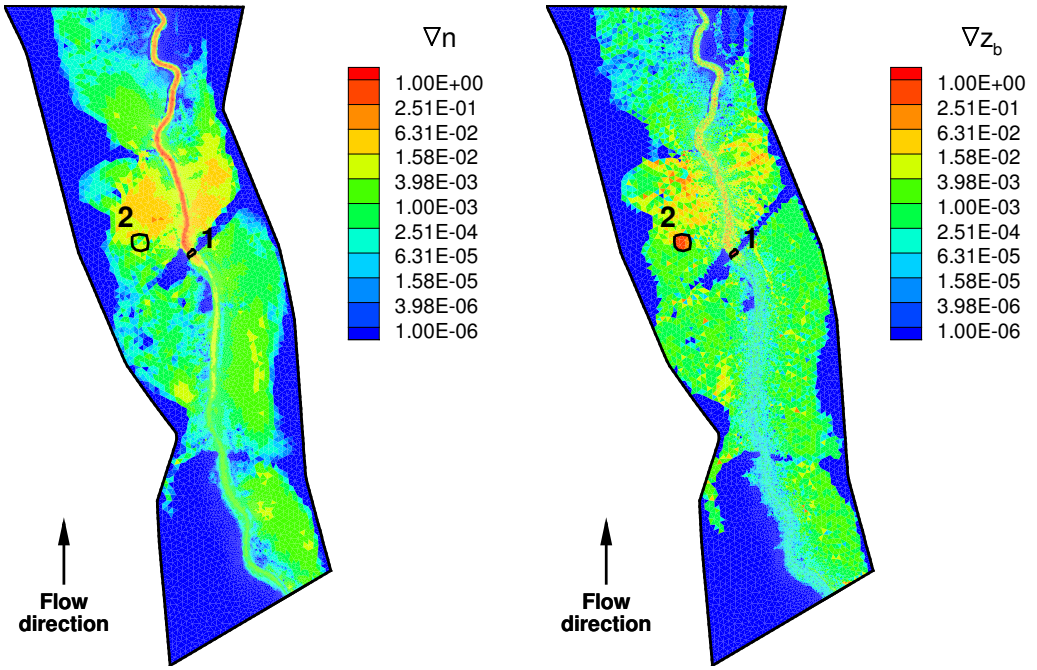


Figure 17: Water surface elevation measurements at Stations 1 (blue) and Station 2 (green). They are synthetic data with realistic random noise added $(+/-10cm)$.

The sensitivities maps with respect to the Manning coefficient n, Fig. 18(a), and the bathymetry z_b, Fig. 18(b), are the corresponding gradient components for each cell. Recall that these gradient values are spatially distributed information, fully relative to the observations considered (here water elevations at stations 1 and 2) and relative to the value of the "computational point" (i.e. the value of the fields n and z_b used in the forward model).
Let us recall that the model output considered are the water elevation at the two stations 1 and 2 (the cost function j defined above).
The gradient value with respect to the Manning friction coefficient is roughly 2 orders of magnitudes higher than those with respect to the bathymetry.
For a sake of clarity, the sensitivity maps (gradient values) plotted in Fig 14 have been normalized.

The highest sensitivities with respect to the friction coefficient n are downstream of the observation location areas. This sensitivity repartition is logical



(a) Sensitivity map w.r.t. friction coefficient n

(b) Sensitivity map w.r.t. bathymetry z_b

Figure 18: Sensitivity map with respect to: (left) the spatially distributed roughness coefficient n; (right) the bathymetry z_b. The gradient values plotted are normalized. Data: water elevation at the 2 stations at all time, see Fig 14 (1st order scheme curves).

since the flow is sub-critical everywhere (recall the Froude number is less than 0.3 everywhere).
The dominant sensitivity with respect to the friction coefficient n is in the stream-bed. One of the consequence is the following. Calibrating the mean stream-bed roughness coefficient during standard regime (i.e. without over-flowing) is desirable before calibrating the same flow model but applied to a flood plain dynamic event.

The highest sensitivities with respect to the bathymetry z_b are at the observation location areas. This is logical. The bathymetry sensitivity pattern is globally comparable to the friction sensitivity one but the bathymetry sensitivity pattern is more point-wise, less diffused. This remark corroborates the analyses presented in [23] (sensitivities performed on the linearized steady-state system around an uniform flow): the bathymetry sensitivity is local (it does not depend on the perturbation surface area) while the friction sensitivity depends on the perturbation surface area (non-local sensitivity).
In other respect, the similarity between the bathymetry and the friction coefficient sensitivity global patterns suggest that a simultaneous "blind" calibration of both quantities (making fit well the model with data) would not result to an intrinsic model, hence not necessarily predictive. In others words, these similar patterns illustrate the potential equifinality problem related to the (topography-friction) pair in the shallow-water equations. The (topography-friction) pair equifinality problem and the difficulty to infer this "basal modeling pair" are discussed for example in [54] in the 1d SW context.
Such spatially distributed sensitivity maps can greatly help the hydraulic modeler to better understand both the hydraulic model (combining the DTM, the parametrization) and the flow.

5.4. Data assimilation, calibration

In this section, two twin experiments are performed: identification of the friction coefficient in 6 land covers and the inflow discharge $Q_{in}(t)$. The direct model and the data assimilated (time series of water elevation at two locations) are the same as previously. These two assimilation experiments demonstrate the capabilities of the approach and the computational software in a real-like flood plain dynamic context, and the robustness of the whole numerical chain elaborated (robust direct solver and gradient, in presence of wet-dry front dynamics over complex topography).

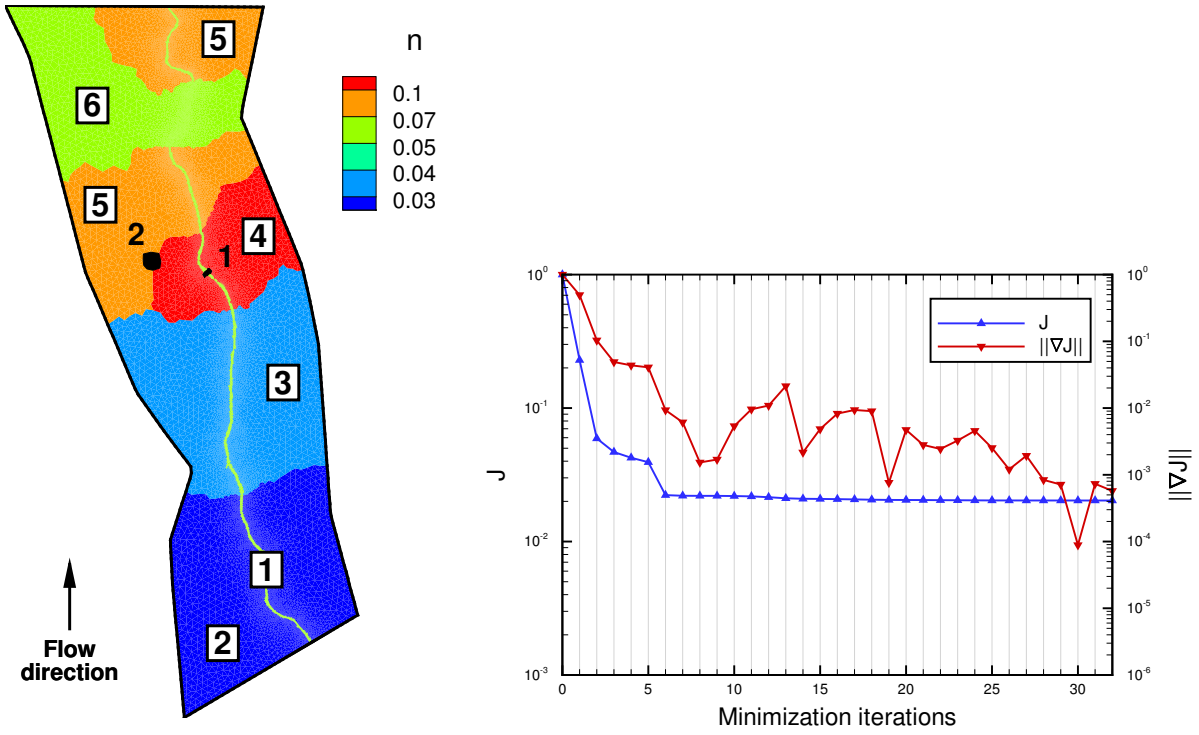
5.4.1. Roughness coefficient identification

The twin experiment performed aims at identifying the Manning-Strickler coefficient values n for each land cover, see Fig. 19(a).

The entire stream-bed corresponds to one land cover, while the computational flood plain domain is divided into the five land covers. The forward model is strictly the same as the previous one (those used for the sensitivity analyses): the observations are water elevations measured at the two virtual stations. All the boundary conditions (inflow discharge, outflow law) and input parameters (friction coefficients) are the same as previously. The first guess values were arbitrarily defined as twice or half the target values (depending on the target value), see Table 4. If considering perfect observations (no noise introduced), then the target values are perfectly recovered. If the observations are perturbed by a $+/-10$cm Gaussian noise, the friction coefficients identified are still good, they are presented in Table 4.

Land cover #	first guess	identified	target
1 (stream-bed)	0.10	0.0521	0.05
2 (flood plain)	0.06	0.0378	0.03
3 (flood plain)	0.08	0.0550	0.04
4 (flood plain)	0.05	0.1436	0.10
5 (flood plain)	0.14	0.0548	0.07
6 (flood plain)	0.10	0.0096	0.05

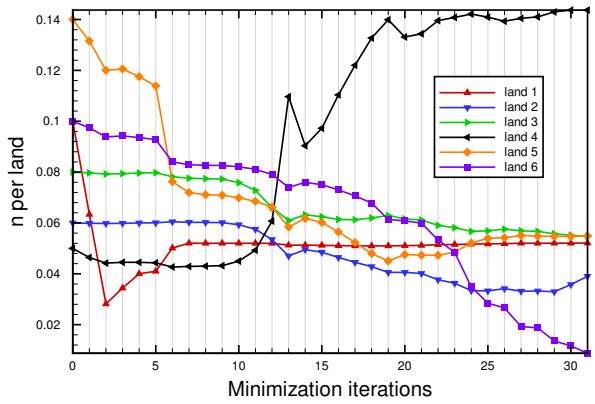
Table 4: The Manning coefficient values n identified per land cover (elevation measurements include $+/-10$cm Gaussion noise).



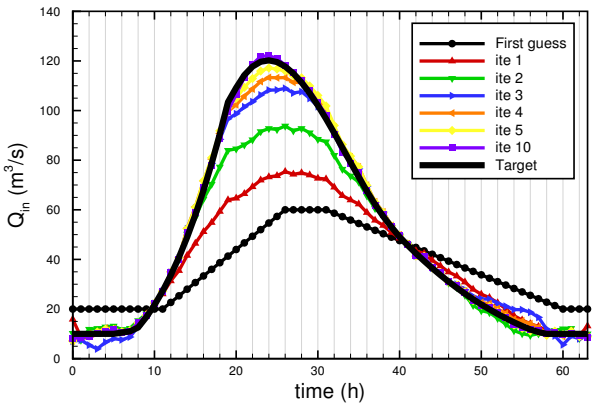
(a) Land covers and Manning coefficient values

(b) Cost function and gradient norm vs minimization iterates (Case: friction coefficients identification)

Figure 19: Left: The land covers, friction coefficient values and the 2 observation stations. Right: The cost function j and the gradient norm $\|\nabla j\|$ vs minimization iterates. Normalized values.



(a) The 6 friction coefficient values n vs minimization iterates



(b) Hydrographs identified

Figure 20: Minimization process. Left: The 6 Manning coefficient values vs iterates (iterate 0: first guesses). Right: Hydrographs (inflow discharge in time) at few minimization iterates.

An accurate convergence is reached into 32 iterations, see Fig 19(b) (the stopping criteria have been set extremely small to make decrease as much as possible the cost function).
Observe that after the 6th iterate, the cost function has almost reached its minimal value but the control parameters did not converged yet, see Fig. 20(a), excepted the dominating one: the stream-bed friction coefficient (land cover #1). Next, the gradient computed (accurate and stable even in the presence of the dynamic wet-dry front, see discussions in the previous sections), and the robustness of the minimization algorithm employed [47], allow to retrieved all the target friction values (iterates $\approx 7 - 32$).

This numerical experiment demonstrates the method capabilities within a real-like flood plain dynamic context, and the robustness of the whole numerical chain implemented within DassFlow, [27].

5.4.2. Inflow discharge identification

The twin experiment performed here aims at identifying the inflow hydrograph. The forward model is strictly the same as the previous one, with the friction coefficient set at the values "targets" indicated inTable 4. The observations generated at the two virtual stations are the same (synthetic water elevation values plus a gaussian noise with a same amplitude as previously), and the boundary conditions (inflow discharge, outflow law) are the same too.
The cost function is defined as follows:

$$j(Q_{in}) = \sum_{time\ step} \sum_{i\ station\ j} (h_{i,j} - h_{i,j}^{obs})^2 + 10^{-3} * (Q_{in} - Q_{in}^{filtered})^2$$

The second term in the cost expression is a regularization term (Tykonov type) aiming at smoothing the control variable Q_{in} computed. The function $Q_{in}^{filtered}(t)$ is defined as an exponential moving average low-pass filter based on a window size of $4dt\frac{(1-0.2)}{0.2}$.

The target hydrograph, the first guess hydrograph and few intermediate ones are plotted in Fig. 20(b). The first guess hydrograph was defined by dividing by two the difference between the peak target value and the baseflow ($10\ m^3.s^{-1}$), plus a phase shift of 4 hours approximatively.

The convergence curves (the decreasing curves of j and $\|\nabla j\|$) are not plotted since they are similar than those in Fig. 19(b) : a fast decrease of j the first 7 iterates then a much slower decrease of j, while the gradient norm $\|\nabla j\|$ keeps decrease until an accurate convergence is reached at iterate 20. In others words, the target hydrograph is perfectly retrieved by the minimization process at iterate 20 while it is very close already at iterate 7, see Fig. 20(b).

This accurate inflow hydrograph identification is due to the very constraining data assimilated; the most crucial data set being the time series of water elevation in the stream-bed. Again, this second numerical experiment demonstrates the method capability to infer the inflow hydrograph within a real-like flood plain dynamic, given time series of water elevation at two (virtual) stations (one inside the stream-bed, and one in the plain).

6. Conclusion

First order and second order finite volume schemes built up from original combinations of existing methods (Godunov type solvers with MUSCL technics and an implicit-explicit Runge-Kutta time discretization), completed with original modifications (in particular at the wet-dry front) have been elaborated. The resulting numerical schemes have been assessed on classical benchmarks plus extra new ones representing better the geophysical flow features and difficulties. A fully second-order accuracy with well-balanced property is obtained; positiveness and stability are demonstrated in presence of dynamic wet/dry fronts. An original numerical study analyzing the topography scale - mesh size in terms of accuracy has been conducted (in the case of a 1d flow channel).
Next, variational sensitivities based on the adjoint of the numerical model, and the corresponding variational data assimilation process based on a BFGS type algorithm, has been elaborated in a MPI and automatic differentiation context. All the know-how and the assessment procedures are exposed. In a real-like test case (real topography of a flood plain with the inflow discharge corresponding to a past flood event), some sensitivity maps with respect to the basal pair (friction,topography) are analyzed. Next, the identification of the inflow discharge and the friction coefficients (defined by land covers) are accurately inferred. The observations (water elevation time series at two virtual stations were synthetic; nevertheless all these numerical tests demon-

strate the robustness and capabilities of the whole modeling chain (direct and inverse) in a real-like flood plain context.

The first part of this work have lead to an effective higher-order finite volume scheme for the 2d shallow-water model with arbitrary topography and friction term, hence gives a reliable answer to the direct numerical modeling of this type of flows. Note that on unstructured grids, Discontinuous Galerkin (DG) methods appear as an alternative tool. DG methods open interesting issues; nevertheless, at the present time, it seems that they do not reach the same degree of robustness as FV technology yet, especially for flood plain dynamic applications (which include wet/dry front dynamics).
As regards the prospects, numerical studies aiming at analyzing the topography scale - grid adequacy in 2d, would deserve to be pursued; in particular in the forthcoming context of rich satellite data of surface waters (e.g. the forthcoming SWOT mission [55]). Also, the innovative sensitivity maps with respect to the basal (friction,topography) would deserve to be further analyzed, and employed in operational analyses, in view to better understand the flow and its complex interactions, the model and its calibration. The goal being not to be descriptive only, but predictive too. Finally as already mentioned, the adjoint method has two main drawbacks: the potentially complex generation of the adjoint model (in particular in the context of an already existing operational direct model) and its computational cost (both in terms of CPU time and memory storage). Nevertheless, we have demonstrated that if the direct model is designed for variational data assimilation (potentially based on automatic differentiation), it is possible to derive fully automatically an accurate, reliable and affordable full assimilation chain based on the adjoint. Furthermore, it has been demonstrated in [56] that in this context, an incomplete adjoint model, tunable in terms of accuracy and cpu time, can be easily employed, and leading to a large gain of computational efficiency with a minimal accuracy loss.

Adjoint code: some fundamental techniques

We present below how the adjoint code is validated, next how to transform the MPI commands of the direct code to the corresponding adjoint commands; finally a speed-up curve, based on real topography data, demonstrate the efficiency of the procedure. Let us recall that all the numerical schemes and algorithms presented in this manuscript have been developed into DassFlow computational software, [26, 27] (open-source available online).

Appendix .1. Validation of the adjoint code

We present briefly how the adjoint code (MPI or not) is classically validated. A gradient test is performed by computing the quantity $\frac{j(k+\varepsilon\delta k)-j(k)}{\varepsilon}$, $\varepsilon > 0$, small, which should converge to the partial derivative of the cost function $\frac{\partial j(k)}{\partial k} \cdot \delta k$ when $\varepsilon \to 0$. The finite difference approximation is evaluated by performing twice the direct code. We set δk to a random vector value, then we compute the quantity:

$$I(\varepsilon) = \frac{j(k+\varepsilon\delta k) - j(k)}{\varepsilon \frac{\partial j(k)}{\partial k} \cdot \delta k}$$

The computed values of $|I(\varepsilon) - 1|$ versus ε are plotted in Fig. .21. Up to the simple precision, the difference between the gradient computed by the adjoint code and its first order finite difference approximation (using the direct code only) converge at order 1 in ε.

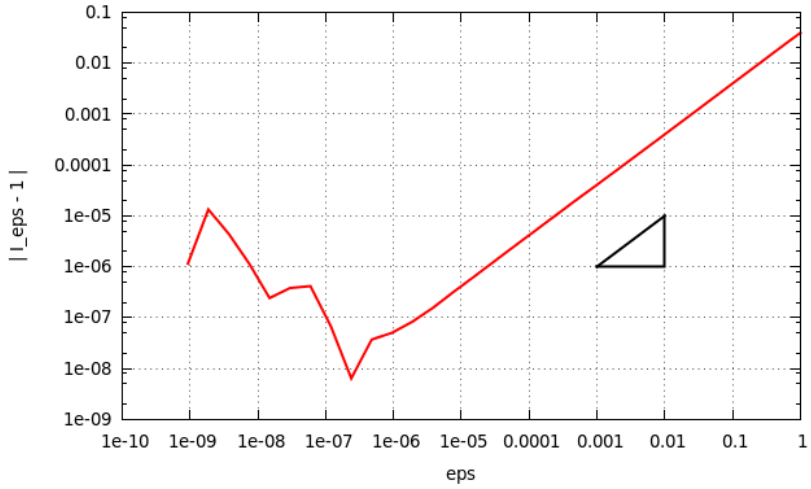


Figure .21: Adjoint code validation: the gradient test.

Let us notice that the same convergence curve is obtained while using an adaptive time step for the forward code (obviously in that case, the same time grid must be used for for all computations: the two direct wes and the adjoint we).

Appendix .2. Writing a MPI adjoint code

The direct code, written in Fortran, is implemented using parallel MPI library. An overlapping strategy is used, see Fig. .22. Since the current

versions of the automatic differentiation tools, and in particular Tapenade, does not handle MPI instructions, in order to derive a MPI adjoint code, we need to re-write "by hand" the equivalent adjoint of the basic MPI subroutine calls such as MPI_SEND, MPI_RECV or MPI_ALLREDUCE. We follow the technics described in [57].

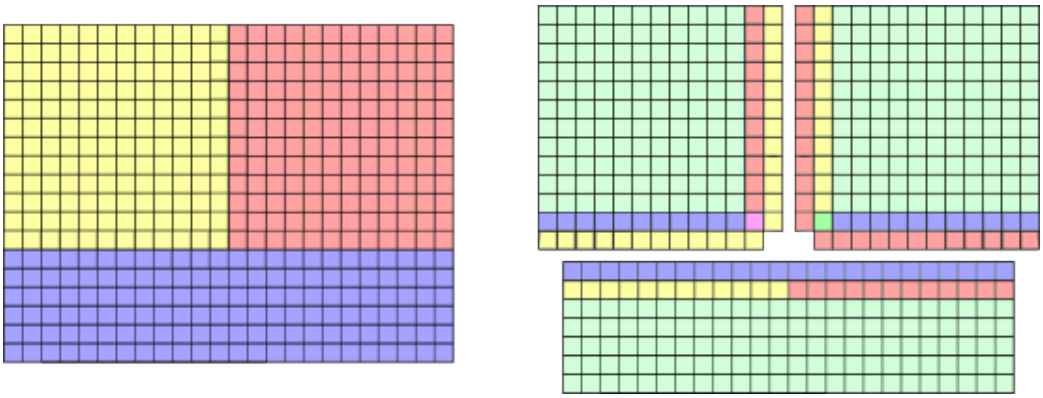


Figure .22: Representation of the partitionning (left) of the domain in three MPI subdomains and its overlapping (right). Each subdomain is extended to the first line present in neighboring subdomain.

i) Send/Recv.

The Send/Recv MPI subroutines consist in exchanging values between subdomains through the overlapping zwe, Fig .22, Let us give an example. Sending an variable a of processor 1 to variable b on processor 2 is equivalent for processor 2 to do the assignment $b = a$. The adjoint of this simple operation is not $ab = bb$ but $ab = ab + bb$, $bb = 0$. Then, as described in [57], the MPI adjoint of a Send/Recv is shown on Tab. .5.

In the case of MPI_ISEND and MPI_IRECV (non blocking communications), and if using overlapping, we must be careful if we value is sent to two different neighbors, see Fig. .23. When performing the backward operation, we have to recover the two adjoint variables in different temporary variables that must be added afterwards. Alternatively, in [57] the author uses the MPI_WAIT subroutine to update the adjoint variable received.

ii) Allreduce (sum).

Direct MPI code	Adjoint MPI code

`if (proc == 1) then`
`  call MPI_SEND(a, 2)`
`else if (proc == 2) then`
`  call MPI_RECV(b, 1)`
`end if`
	`if (proc == 1) then`
	`  call MPI_RECV(temp,2)`
	`  ab = ab + temp`
	`else if (proc == 2) then`
	`  call MPI_SEND(bb,  1)`
	`  bb = 0.`
	`end if`

Table .5: Adjoint of a Send/Recv call.

The Allreduce MPI sum subroutine consists in summing all values of a variable on every processors. In a sequential context, the adjoint consists to compute each corresponding adjoint instructions, see Tab. .6. Using the Allreduce MPI sum subroutine, first we must make an Allreduce MPI sum to unify the global variable value ("`a_globb`" variable in Tab. .6). Next, the adjoint of the sum is obtained by adding the global backward variable to the local we.

iii) Allreduce (min/max).

If trying to differentiate the *min* (or *max*) function using Tapenade software, first it is transformed as an *if* instruction. In order to compute the adjoint of the Allreduce MPI max subroutine, we have to identify the processor where the maximum value is reached (during the direct computation), then update the adjoint value on this processor only. To do so, we use the MPI_ALLGATHER subroutine that regroup values from every processor in we vector.

Appendix .3. Speed-up of the full computational process

In order to measure the efficiency of the resulting MPI inverse code (DassFlow-Shallow software, including the direct code and the adjoint code), we plot the speed-up curve for sensitivity analysis performed on real data set: the Lèze river. The curve obtained up to 32 processors is presented in

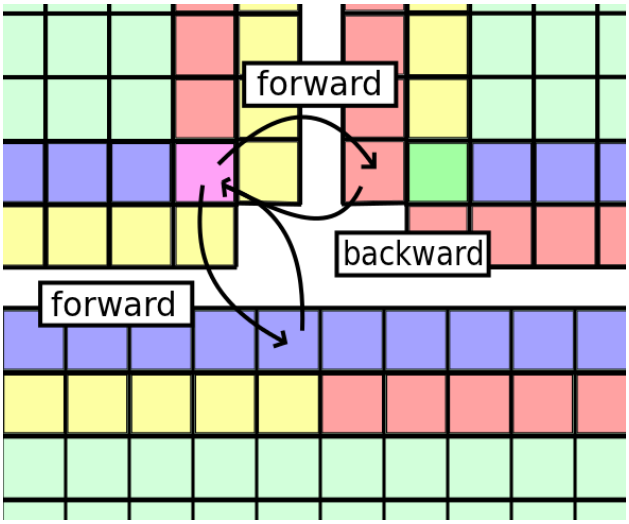


Figure .23: Zoom of Fig. .22 with detailed communications for we cell in the forward context (direct code) and backward context (adjoint code).

Fig. .24. At 32 processors, we obtain 82.5% of the idealistic speed-up. It illustrates the good operating of the technics described above to obtain the MPI version of the adjoint instructions.

Acknowledgments

The authors thank warmly J. Chorda from CNRS & IMFT for the mesh generation and his help to set up the hydraulic model based on the real topography data.
This work was supported by a grant overseen by the French National Research Agency (ANR) AMAC 2009-2013. Also it was granted access to the HPC resources of CALMIP supercomputing center, Toulouse.

[1] A. Le Roux, Numerical stability for some equations of gas dynamics, Mathematics of Computation 37 (156) (1981) 307–320.

[2] B. Van Leer, Towards the ultimate conservative difference scheme. v. a second-order sequel to godunov's method, Journal of computational Physics 32 (1) (1979) 101–136.

Direct sequential code	Adjoint sequential code

`a_glob = a1 + a2 + a3`	`a1b = a1b + a_globb`
	`a2b = a2b + a_globb`
	`a3b = a3b + a_globb`
	`a_globb = 0`

Direct MPI code	Adjoint MPI code

`call MPI_ALLREDUCE(a, &`
`&  a_glob, sum)`
	`call MPI_ALLREDUCE(a_globb, &`
	`&  a_globb_sum, sum)`
	
	`a_globb = 0.`
	`ab = ab + a_globb_sum`
	`a_globb_sum = 0.`

Table .6: Adjoint of a sum, in a sequentiel code and using the Allreduce MPI sum subroutine.

[3] S. Osher, Convergence of generalized muscl schemes, SIAM Journal on Numerical Analysis 22 (5) (1985) 947–961.

[4] J.-P. Vila, An analysis of a class of second-order accurate godunov-type schemes, SIAM J. Numer. Anal. 26 (4) (1989) 830–853.

[5] E. Toro, Shock-capturing methods for free-surface shallow flows, John Wiley, 2001.

[6] V. Guinot, Wave propagation in fluids: models and numerical techniques, John Wiley & Sons, 2012.

[7] B. Einfeldt, On godunov-type methods for gas dynamics, SIAM Journal on Numerical Analysis 25 (2) (1988) 294–318.

[8] E. F. Toro, M. Spruce, W. Speares, Restoration of the contact surface in the hll-riemann solver, Shock waves 4 (1) (1994) 25–34.

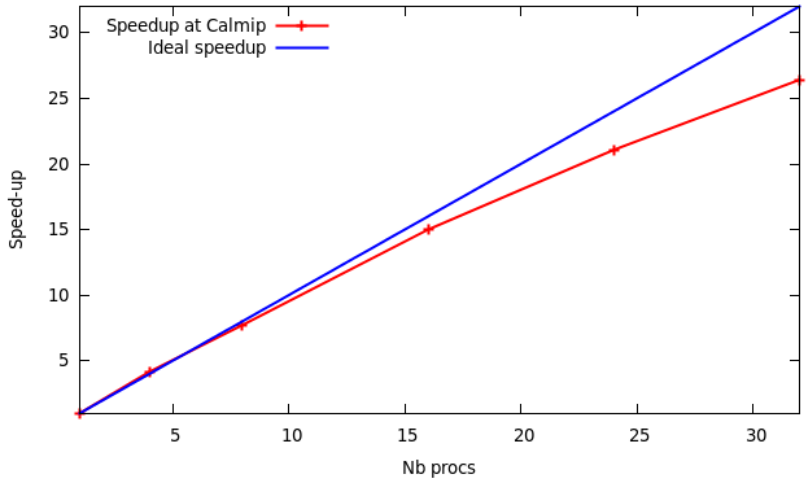


Figure .24: Speed-up obtained on a sensitivity computation (DassFlow-Shallow software, Lèze river test case, Calmip cluster Toulouse).

[9] J.-P. Vila, Simplified godunov schemes for 2 x 2 systems of conservation laws, SIAM J. Numer. Anal. 23 (6) (1986) 1173–1192.

[10] J. M. Greenberg, A.-Y. Leroux, A well-balanced scheme for the numerical processing of source terms in hyperbolic equations, SIAM Journal on Numerical Analysis 33 (1) (1996) 1–16.

[11] L. Gosse, Computing qualitatively correct approximations of balance laws, SIMAI Springer Series 2.

[12] E. Audusse, F. Bouchut, M.-O. Bristeau, R. Klein, B. Perthame, A fast and stable well-balanced scheme with hydrostatic reconstruction for shallow water flows, SIAM J. Sci. Comput. 25 (6) (2004) 2050–2065.

[13] E. Audusse, M.-O. Bristeau, A well-balanced positivity preserving "second-order" scheme for shallow water flows on unstructured meshes, J. Comput. Phys. 206 (1) (2005) 311–333.

[14] J. Blum, F.-X. Le Dimet, N. I.M., Data assimilation for geophysical fluids, in: Temam, Tribbia (Eds.), Handbook of Numerical Analysis, Vol. 14, North-Holland, 2009.

[15] E. Belanger, A. Vincent, Data assimilation (4D-VAR) to forecast flood in shallow-waters with sediment erosion, Journal of Hydrology 300 (2005) 114–125.

[16] X. Lai, J. Monnier, Assimilation of spatial distributed water levels into a shallow-water flood model. part i: mathematical method and test case, J. Hydrology 377 (1-2) (2009) 1–11.

[17] R. Hostache, X. Lai, J. Monnier, C. Puech, Assimilation of spatial distributed water levels into a shallow-water flood model. part ii: using a remote sensing image of mosel river, J. Hydrology 390 (3-4) (2010) 257–268.

[18] I. Gejadze, J. Monnier, On a 2d zoom for 1d shallow-water model: coupling and data assimilation, Comp. Meth. Appl. Mech. Eng. 196 (45-48) (2007) 4628–4643.

[19] J. Marin, J. Monnier, Superposition of local zoom model and simultaneous calibration for 1d-2d shallow-water flows, Math. Comput. Simul. 80 (2009) 547–560.

[20] J. Zhang, X. Lu, Parameter estimation for a three-dimensional numerical barotropic tidal model with adjoint method, International journal for numerical methods in fluids 57 (1) (2008) 47–92.

[21] M. Honnorat, J. Monnier, F. Le Dimet, Lagrangian data assimilation for river hydraulics simulations, Comput. Visual. Sc. 12 (3) (2009) 235.

[22] M. Honnorat, J. Monnier, N. Rivière, E. Huot, F. Le Dimet, Identification of equivalent topography in an open channel flow using lagrangian data assimilation, Comput. Visual. Sc. 13 (3) (2010) 111.

[23] V. Guinot, B. Cappelaere, Sensitivity analysis of 2d steady-state shallow water flow. application to free surface flow model calibration, Advances in Water Resources 32 (4) (2009) 540–560.

[24] L. Hascoët, V. Pascual, The Tapenade Automatic Differentiation tool: Principles, Model, and Specification, ACM Transactions On Mathematical Software 39 (3).
URL http://dx.doi.org/10.1145/2450153.2450158

[25] M. Honnorat, J. Marin, J. Monnier, X. Lai, Dassflow v1.0: a variational data assimilation software for 2d river flows, Research Report RR-6150, INRIA (2007).

[26] DassFlow, Data assimilation for free surface flows - computational software, open source software (2013).
URL http://www.math.univ-toulouse.fr/DassFlow/

[27] C. Couderc, R. Madec, J. Monnier, J.-P. Vila, Dassfow-shallow, variational data assimilation for shallow-water models: Numerical schemes, user and developer guides., Research report, CNRS, Mathematics Institute of Toulouse (IMT), INSA (2013).
URL http://hal.archives-ouvertes.fr/hal-01120285v1

[28] S. K. Godunov, A difference method for numerical calculation of discontinuous solutions of the equations of hydrodynamics, Math. Sbornik 47 (1959) 271–306.

[29] P. Chévrier, H. Galley, A van leer finite volume scheme for the euler equations on unstructured meshes, ESAIM: Mathematical Modelling and Numerical Analysis 27 (2) (1993) 183–201.

[30] T. Buffard, S. Clain, Monoslope and multislope muscl methods for unstructured meshes, Journal of Comptutational Physics 229 (2010) 3745–3776.

[31] F. Haider, J.-P. Croisille, B. Courbet, Stability analysis of the cell centered finite-volume muscl method on unstructured grids, Numer. Math. 113 (4) (2009) 555–600.

[32] I. Nikolos, A. Delis, An unstructured node-centered finite volume scheme for shallow water flows with wet/dry fronts over complex topography, Computer Methods in Applied Mechanics and Engineering 198 (47-48) (2009) 3723–3750.

[33] A. Delis, I. Nikolos, M. Kazolea, Performance and comparison of cell-centered and node-centered unstructured finite volume discretizations for shallow water free surface flows, Archives of Computational Methods in Engineering 18 (2011) 57–118.

[34] A. I. Delis, I. K. Nikolos, A novel multidimensional solution reconstruction and edge-based limiting procedure for unstructured cell-centered finite volumes with application to shallow water dynamics, International Journal for Numerical Methods in Fluids 71 (5) (2013) 584–633.

[35] T. Barth, Numerical methods for convservative laws on structured and unstructured meshes, Tech. rep., VKI Lecture Series (2003).

[36] T. Chacón Rebollo, A. Domínguez Delgado, E. D. Fernández Nieto, Asymptotically balanced schemes for non-homogeneous hyperbolic systems - application to the shallow water equations, Comptes Rendus Mathematique 338 (1) (2004) 85 – 90.

[37] M. Castro Díaz, T. Chacón Rebollo, E. D. Fernández-Nieto, J. M. González-Vida, C. Parés, Well-balanced finite volume schemes for 2d non-homogeneous hyperbolicsystems. application to the dam break of aznalcóllar, Comput. Methods Appl. Mech. Engrg. 197 (2008) 3932–3950.

[38] A. Quarteroni, A. Valli, Numerical Approximation of Partial Differential Equations, Springer Series in Computational Mathematics, Springer, 2008.

[39] Q. Liang, F. Marche, Numerical resolution of well-balanced shallow water equations with complex source terms, Advances in Water Resources 32 (6) (2009) 873–884.

[40] T. Jahnke, C. Lubich, Error bounds for exponential operator splittings, BIT Numerical Mathematics 40 (4) (2000) 735–744.

[41] S. Gottlieb, C.-W. Shu, E. Tadmor, Strong stability-preserving high-order time discretization methods, SIAM Rev 43 (2001) 89–112.

[42] L. Pareschi, G. Russo, Implicit-explicit runge-kutta schemes and applications to hyperbolic systems with relaxation, Journal of Scientific Computing 25 (2005) 129–155.

[43] E. Hairer, G. Wanner, Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems, Springer, 2004.

[44] L. Bouttier, P. Courtier, Data assimilation concepts and methods, ECMWF Training course. www.ecmwf.int., 1999.

[45] J. Monnier, Variationnal data assimilation, from optimal control to large scale data assimilation, Open Learn. Res. Ed. INPT.
URL http://pedagotech.inp-toulouse.fr/130908/co/Mon130908-variational\textunderscoreweb.html

[46] M. Honnorat, Assimilation de données lagrangiennes pour la simulation numérique en hydraulique fluviale., Ph.D. thesis, Institut National Polytechnique de Grenoble - INPG (2007).

[47] J. Gilbert, C. Lemar´ echal, Some numerical experiments with variable-storage quasi-newton algorithms. mathematical programming 45, 407-435 45 (1989) 407–435.

[48] L. Begnudelli, B. F. Sanders, Unstructured grid finite-volume algorithm for shallow-water flow and scalar transport with wetting and drying, Journal of hydraulic engineering 132 (4) (2006) 371–384.

[49] J. Sampson, A. Easton, M. Singh, Moving boundary shallow water flow above parabolic bottom topography., ANZIAM Journal 47.

[50] W. Thacker, Some exact solutions to the nonlinear shallow-water wave equations, J. Fluid Mech. 107 (1981) 499–508.

[51] L. Song, J. Zhou, J. Guo, Q. Zou, Y. Liu, A robust well-balanced finite volume model for shallow water flows with wetting and drying over irregular terrain, Advances in Water Resources 34 (7) (2011) 915–932.

[52] Q. Kesserwani, G.and Liang, Locally limited and fully conserved rkdg2 shallow water solutions with wetting and drying, Journal of Scientific Computing 50 (2012) 120–144.

[53] J. Hou, Q. Liang, F. Simons, R. Hinkelmann, A 2d well-balanced shallow flow model for unstructured grids with novel slope source term treatment, Advances in Water Resources 52 (0) (2013) 107 – 131.

[54] P.-A. Garambois, J. Monnier, Inference of effective river properties from remotely sensed observations of water surface, Adv. Water Ress. Accepted, to appear.

[55] L.-L. Fu, D. Alsdorf, E. Rodriguez, R. Morrow, N. Mognard, J. Lambin, P. Vaze, T. Lafon, The swot (surface water and ocean topography) mission: spaceborne radar interferometry for oceanographic and hydrological applications, in: OCEANOBS 09 Conference, 2009.

[56] N. Martin, J. Monnier, Adjoint accuracy for the full-stokes ice flow model: limits to the transmission of basal friction variability to the surface, The Cryosphere 8 (2014) 721–741.

[57] J. Utke, L. Hascoët, P. Heimbach, C. Hill, P. Hovland, U. Naumann, Toward adjoinable mpi, in: Proceedings of the 10th IEEE International Workshop on Parallel and Distributed Scientific and Engineering, PDSEC'09, 2009.