



HAL
open science

Inverse algorithms for 2D shallow water equations in presence of wet dry fronts. Application to flood plain dynamics

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

► To cite this version:

Jerome Monnier, Frédéric Couderc, Denis Dartus, Kévin Larnier, Ronan Madec, et al.. Inverse algorithms for 2D shallow water equations in presence of wet dry fronts. Application to flood plain dynamics. *Advances in Water Resources*, 2016, 97, pp.11-24. 10.1016/j.advwatres.2016.07.005 . hal-01344008v2

HAL Id: hal-01344008

<https://hal.science/hal-01344008v2>

Submitted on 12 Mar 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Inverse algorithms for 2D shallow water equations in presence of wet dry fronts. Application to flood plain dynamics

J. Monnier^{b,d}, F. Couderc^a, D. Dartus^c, K. Larnier^c, R. Madec^b, J.-P. Vila^b

^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

The 2D shallow water equations adequately model some geophysical flows with wet-dry fronts (e.g. flood plain or tidal flows); nevertheless deriving accurate, robust and conservative numerical schemes for dynamic wet-dry fronts over complex topographies remains a challenge. Furthermore for these flows, data are generally complex, multi-scale and uncertain. Robust variational inverse algorithms, providing sensitivity maps and data assimilation processes may contribute to breakthrough shallow wet-dry front dynamics modelling. The present study aims at deriving an accurate, positive and stable finite volume scheme in presence of dynamic wet-dry fronts, and some corresponding inverse computational algorithms (variational approach). The schemes and algorithms are assessed on classical and original benchmarks plus a real flood plain test case (Lèze river, France). Original sensitivity maps with respect to the (friction, topography) pair are performed and discussed. The identification of inflow discharges (time series) or friction coefficients (spatially distributed parameters) demonstrate the algorithms efficiency.

Keywords: 2D shallow water, wet-dry front, finite volume, variational data assimilation, sensitivity, flood plain.

1. Introduction

The 2D Shallow Water Equations (SWE) adequately model some geophysical flows with dynamic wet-dry fronts e.g. flood plain, tidal, coastal

or estuarine flows. These geophysical flows involve multi-scale and uncertain topography, uncertain quantities at open boundaries, uncertain model parameters (e.g. friction) and large computational domains. An accurate numerical scheme, potentially higher-order, is a key feature to obtain satisfying accuracy-computational cost balance. Also the wet-dry front dynamic has to be accurately approximated since the global flow pattern may depend on local front behaviour. Next, sensitivity analyses become a key tool to better understand the complex interactions and to set up the flow model. Finally a complete data assimilation procedure may be the cornerstone to design a reliable predictive model.

The present article aims at elaborating accurate, stable, conservative Finite Volume (FV) schemes for the 2D SWE in presence of dynamic wet-dry fronts, plus the corresponding adjoint-based variational inverse algorithms providing sensitivity analyses and data assimilation. The typical applications are river hydraulics (flood plain dynamics, low-water), coastal flows or tidal flows. The present schemes are assessed on a classical benchmark widely documented, plus original test cases presenting the key features of wet dry front dynamics; next all the algorithms efficiency (direct and inverse) is demonstrated on a real flood plain case (Lèze river, France).

Flood plain dynamic modelling is still an active research topic because of the complex multi-scale data (the topography), data uncertainties (e.g. inflow discharge, rating curves, friction parameters) and uncertain sparse heterogeneous flow measurements (in-situ and/or remotely sensed), see e.g. some recent discussions in [1] and references therein. Elaborating reliable flood plain dynamic models remains a challenge. The flooded area pattern accuracy mainly depends on the water levels accuracy within the riverbed. In other respect, the wet/dry front position may be a crucial feature of the flow in floodplains, or not, depending on the circumstances. Also, the wetting - drying process, whatever in rural or urban areas, can be greatly correlated to small scale pathways. Then the wet-dry front dynamics may be one of the difficulty to consider when modelling a flood plain dynamic. To become reliable and operational, flood plain dynamic models will be based on High Performance Computing (HPC), on cocktails of multi-scale data and algorithms combining all these heterogeneous information, see e.g. [2].

In the geophysical flows context mentioned above, the wet-dry front dy-

namics are difficult to capture accurately in particular since topographies are rapidly varying. The difficulties are both mathematical and numerical since the water depth vanishes, which can make blow up the equations and the numerical schemes. The most classical technique adopted in the literature is to consider a thin-film over the entire domain (by specifying a minimum threshold depth in the mass equation), see e.g. [3].

A major drawback of this simple approach is the potential front velocity change due to the numerical, unphysical cut-off, see e.g. [4, 3] and references therein. A second classical approach, more complex, is to introduce a sub-grid resolution which may be seen as a porosity description, see e.g. [5] and references therein.

The numerical solvers derived here do not introduce any subgrid resolution, but do not require any numerical cut-off at the front. The numerically observed stability, accuracy and positiveness of the FV schemes naturally derive from a precise combination of existing techniques: the HLL approximate Riemann solver [6, 7] combined with an ad-hoc estimate of wave velocity [8] and the well-balanced techniques presented in [9, 10]. Also, the friction term is discretized semi-implicitly. Next, an *actual* second order scheme is obtained by adding a standard MUSCL reconstruction (combined with the well-balanced property treatment mentioned above) and an IMPLICIT-EXPLICIT Runge-Kutta (IMEX RK) scheme. Note that deriving second order FV schemes is classical, but deriving an *actual* second order scheme remains a challenge. All other combinations of existing methods we have tested (different intermediate wave velocity, different well balanced property treatment, and the order 2 Runge-Kutta scheme) did not provide such an accurate and robust solver. To our best knowledge, the resulting numerical solvers demonstrate an accuracy, stability and robustness as never published in the literature, see e.g. [4, 11] and references therein. The present schemes have been assessed on classical test cases widely documented, plus extra new test cases presenting crucial features of flood plain dynamics. Also, comparisons between the first order scheme and two second order schemes are presented.

As already mentioned, the second step of the study aims at deriving sensitivities maps and a complete variational data assimilation process based on the adjoint equations of the FV schemes described above. Variational Data Assimilation (VDA), also called 4D-var, is widely employed in meteorology and oceanography, see e.g. [12] and references therein. In 1D river hydraulics (Saint-Venant's equations), data assimilation methods and sensitivity anal-

ysis based on sequential, filtering approaches (e.g. Kalman's filter) are classical. On the contrary, for 2D SWE with wet-dry front dynamics, only very few studies exist. For real like 2D hydrodynamic problems, the sequential approach is too much CPU time consuming, while the variational approach (VDA) requires to derive an adjoint model affordable in terms of CPU time and memory amount. This task is non trivial and only few studies managed to design a complete VDA process for real like hydrodynamic problems. In river hydraulics, let us cite [13] which treats of a 2D SWE coupled with a simplified sediment transport model; [14] which analyses some capabilities of the VDA approach given spatially distributed observations; [15] treats of the assimilation of one flood plain image (post-treated SAR data). Also [16, 17] treating simultaneously the assimilation and the superimposition of 2D local - 1D global flow models but in small scale academic configurations only. All these studies are based on the 2D SWE and a first order descent algorithm (quasi-newton generally); the cost gradient is computed by performing the adjoint model. Theses studies are based on the same VDA adjoint-based method as the present one, but none of these studies are based on higher-order numerical schemes, none have been derived without any regularisation at the dynamics front (hence circumventing potential differentiation issues when deriving the adjoint model), none of these studies addressed the sensitivity with respect to the topography, and finally none of these numerical models included the know-how to obtain a fully parallel process (hence preventing to employ them at large scale). In the VDA approach, all the inverse computational algorithms are based on the adjoint equations of the numerical schemes; the adjoint model is generated "automatically" by using an automatic differentiation tool, [18]. To do so, the direct source code must be prepared to be differentiated source-to-source. Next, the forward MPI commands are "reversed" by hand (or using an extra shell script). Also, some extra crucial tricks make affordable the adjoint code in terms of memory storage. Based on all these know-hows exposed in the present article, the inverse computational algorithms becomes affordable for real applications on parallel architectures, both in terms of CPU time and memory. To our best knowledge, such a complete model (robust and accurate schemes for 2D SWE for wet-dry front dynamics, plus the adjoint model, all MPI - HPC) deriving from the numerical analysis and computational science knowledges described here, has never been published in the literature yet. Such complete models are crucial to obtain reliable descriptions of 2D SWE with wet dry front dynamics in a geophysical context. All the methods and algorithms described

here are implemented in the open-source software DassFlow (Data Assimilation for Free Surface Flows) [19, 20, 21]. Hence this experimental software, potentially operational depending on the application, may contribute to make breakthrough in the field.

The capabilities of all present algorithms are illustrated on a real flood plain case: Lèze river, southwestern of France. A cost function measuring the discrepancy between water elevations at two (virtual) gauge stations (synthetic data with realistic noise amplitudes) is classically defined. Then variational sensitivities analyses (sensitivity maps) are performed with respect to: a) the Manning-Strickler friction coefficient locally defined i.e. without any a-priori on land covers (one value per cell); b) the topography elevation. The sensitivity maps required one run of the direct model plus the adjoint model, hence in terms of CPU times, it roughly costs (1+4) times a direct simulation.

Next parameter identification experiments are performed. The identified parameters are: a) the friction Manning-Strickler friction coefficients defined by land covers (6 in this case); b) the inflow discharge $Q_{in}(t)$ at upstream (open boundary).

Of course, all the algorithms elaborated here and the corresponding software, can be applied to any other flow modelled by the 2D SWE with wet-dry fronts e.g. tidal flows, coastal flows.

The paper is organised as follows. In Section 2, the direct mathematical model and the FV schemes (1st and 2nd order) are presented, then assessed on a benchmark representing well the difficulties encountered when modelling wet-dry front dynamics. In Section 3, the inverse method (the adjoint-based method, the resulting variational sensitivities and the data assimilation process) is recalled. The real flood plain case (Lèze river), based on a fine topography and a historical-like inflow discharge is considered in Section 4 for comparisons between different direct solvers, and in Section 5 for inverse computations: sensitivity analyses and parameter identifications. Some perspectives are proposed in the conclusion (Section 6). A first appendix presents an extra benchmark for the FV solvers, demonstrating the *actual* second order. A second appendix details the link between the cost function differential (the "mathematical gradient") and an automatically derived adjoint code. The last appendix proposes a speed-up curve demonstrating the good efficiency

of the *whole* MPI computational code (hence demonstrating the efficiency of the reverse transforms of the MPI commands introduced by hand too).

2. The Direct Model

In this section, the direct (forward) model is briefly described. The equations are the classical 2D SWE with the non linear Manning-Strickler friction law. Few numerical FV solvers are developed. To our best knowledge, the derived FV solvers are original since they stem from original combinations of existing methods. The resulting FV solvers are either first or second order. The numerical tests demonstrate their accuracy, positiveness, stability and robustness. Deriving second order FV schemes is classical but deriving an *actual* second order scheme remains a challenge. Basically, the employed ingredients to obtain the present *actual* second-order scheme, stable without any regularisation at wet-dry front, are the following: a) an adequate wave velocity in the HLLC approximate Riemann solver [8]; b) a MUSCL reconstruction; c) an IMplicit-EXplicit Runge-Kutta (IMEX RK) time scheme [22]; and of course a semi-implicit scheme for the friction term.

All other combinations which have been tested (different intermediate wave velocity, different well balanced property treatments and the order 2 Runge-Kutta -RK2- scheme) do not lead to such accurate and robust solvers, in particular in the second order case.

All schemes have been assessed and compared into details by performing classical benchmarks widely documented (one is presented below) plus extra original tests cases presenting crucial features of wet dry front dynamics (one is presented below and one in Appendix A). Many extra benchmarks and details can be found in [23].

These numerical tests demonstrate that the present new FV solvers are positive and remain stable in presence of dynamic wet/dry fronts without unphysical regularisation (numerical cut-off). Also, they are well-balanced in the sense they respect water at rest. To our best knowledge, the present FV solvers demonstrate an accuracy, stability and robustness as never published in the literature, see e.g. [4, 24, 5, 11] and references therein.

2.1. The Mathematical model

The 2D SWE including the Manning-Strickler friction term read in their conservative form as follows:

$$\begin{aligned} \partial_t \mathbf{U} + \nabla \cdot \mathbf{F}(\mathbf{U}) &= \mathbf{S}_g(\mathbf{U}) + \mathbf{S}_f(\mathbf{U}) \\ \mathbf{U} &= \begin{bmatrix} h \\ h\mathbf{u} \end{bmatrix}, \quad \mathbf{F}(\mathbf{U}) = \begin{bmatrix} h\mathbf{u} \\ h\mathbf{u} \otimes \mathbf{u} + \frac{gh^2}{2}\mathbf{I} \end{bmatrix}, \\ \mathbf{S}_g(\mathbf{U}) &= \begin{bmatrix} 0 \\ -gh\nabla z_b \end{bmatrix}, \quad \mathbf{S}_f(\mathbf{U}) = \begin{bmatrix} 0 \\ -g\frac{n^2\|\mathbf{u}\|}{h^{1/3}}\mathbf{u} \end{bmatrix} \end{aligned} \tag{1}$$

where h denotes the water depth, \mathbf{u} the depth-averaged velocity, g the gravity, z_b the bottom elevation and n the Manning-Strickler friction coefficient.

2.2. Numerical scheme assessments in presence of wet-dry fronts

The standard 2D SWE described above is numerically solved using FV schemes, first and second order. The basic scheme is the standard HLLC approximate Riemann solver see e.g. [4], but with the intermediate wave speed introduced in [8]. This choice turned out to be essential to reach L^∞ stability without any regularisation. The well-balanced property has been introduced by using two different methods: those presented in [9, 10] (so-called wb-A in the figures) and those presented in [25, 26] (so-called wb-E in the figures). The second order accuracy is obtained by combining a MUSCL reconstruction and an IMEX RK time scheme [22].

This combination of these quite technical but standard methods turned out to be the only combination found to obtain an *actual* second-order accuracy, well-balanced, positive and stable in presence of dynamic wet/dry fronts without any unphysical regularisation (cut-off).

The FV solvers have been assessed by performing few classical test cases from the literature plus extra new ones, representing geophysical shallow flow features and difficulties. Below, two selected test cases only are presented: 1) the classical bowl test case, with a linear friction term which is widely documented in the literature; 2) an original test case presenting a dynamic

wet-dry front.

An extra original test case, considering a "regular" dam-break layout, is presented in Appendix A. The goal of this third test case is to demonstrate the *actual* second order feature of the scheme (recall that to our best knowledge, this is not classical). Indeed, the second order accuracy can be mathematically reached on smooth solutions only. Nevertheless, in presence of a wet/dry front, i.e. in presence of non-smooth solution, it is demonstrated in the second test case that the second-order schemes remain more accurate than the first order ones (even if their convergence rate decreases to almost one).

2.2.1. Parabolic bowl

The exact solution of the 1D SWE with a linear friction term in a parabolic bowl has been derived in [27] following [28]. It is a classical test case see e.g. [10, 29, 30, 31, 32].

The computed relative error norm is defined by:

$$e_1(x) = \frac{\|x^{\text{num}} - x^{\text{exact}}\|_1}{\|x^{\text{exact}}\|_1} \quad \text{with } \|x\|_1 = \sum_{K \in \Omega} m_K |x_K| \quad (2)$$

where K denotes the mesh cells and m_K their measures.

Three FV schemes are compared: a) a first order scheme based on the Euler time scheme; b) an (a-priori) second order scheme based on the RK 2 time scheme; c) an (actual) second-order scheme based on the IMEX RK time scheme. All three schemes are based on the right combination described previously (in particular the intermediate wave velocity in the HLLC solver and the wb-A treatment).

The obtained convergence curves are presented in Fig. 2 (L). Let us point out that to obtain the best results, the MUSCL reconstruction is based on a Maximum Principle (MP) limiter, see [23] and reference therein for more details.

The numerical error obtained for 100 cells is approximatively the same as those presented in [29] (the computed solution is the same). For a standard cell number, typically lower than 1000, no significant difference exists between the RK2 MUSCL scheme and the IMEX RK MUSCL scheme. For extremely

fine meshes (greater than 1000), a discrepancy of convergence rate can be observed: order 1 approx. for RK2 MUSCL scheme and order 1.5 approx. for IMEX RK MUSCL scheme. As expected, the order 2 is not reached since the presence of the two dynamic wet/dry fronts, hence the presence of singularities in the solution.

This first test case assesses the FV schemes since since the results obtained with the three numerical schemes compare fairly well with the exact solution. Nevertheless, the friction source term is linear and not non linear like in the original model (1); also more realistic test cases like those below, highlights accuracy discrepancies between schemes potentially larger.

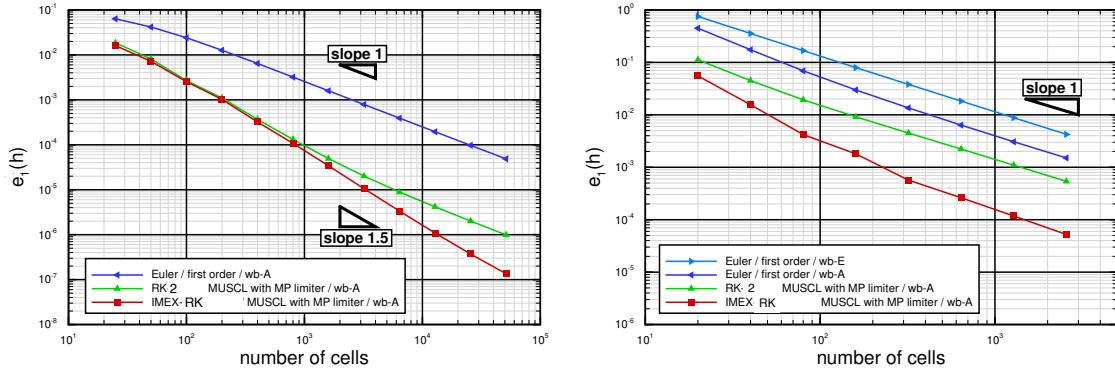


Figure 1: Converge curves: error norm $e_1(h)$ for different schemes and different mesh size. (Left) Parabolic bowl test case. (Right) Dam break test case.

2.2.2. Dam break on a constant bottom slope with a dynamic wet/dry front

The test case presented below is a dam break on a constant bottom slope, involving a dynamic wet/dry front, see Fig. 2. The non linear Manning-Strickler friction law is taken into account.

The domain length is $l_x = 1000 \text{ m}$ with wall boundaries on each side, the slope $s = 0.5 \%$, the Manning-Strickler coefficient is constant, $n = 0.05$, and the simulation time is $T = 500 \text{ s}$. Since no exact solution is known, the reference solution, indicated as "exact" solution in (2), is the second order

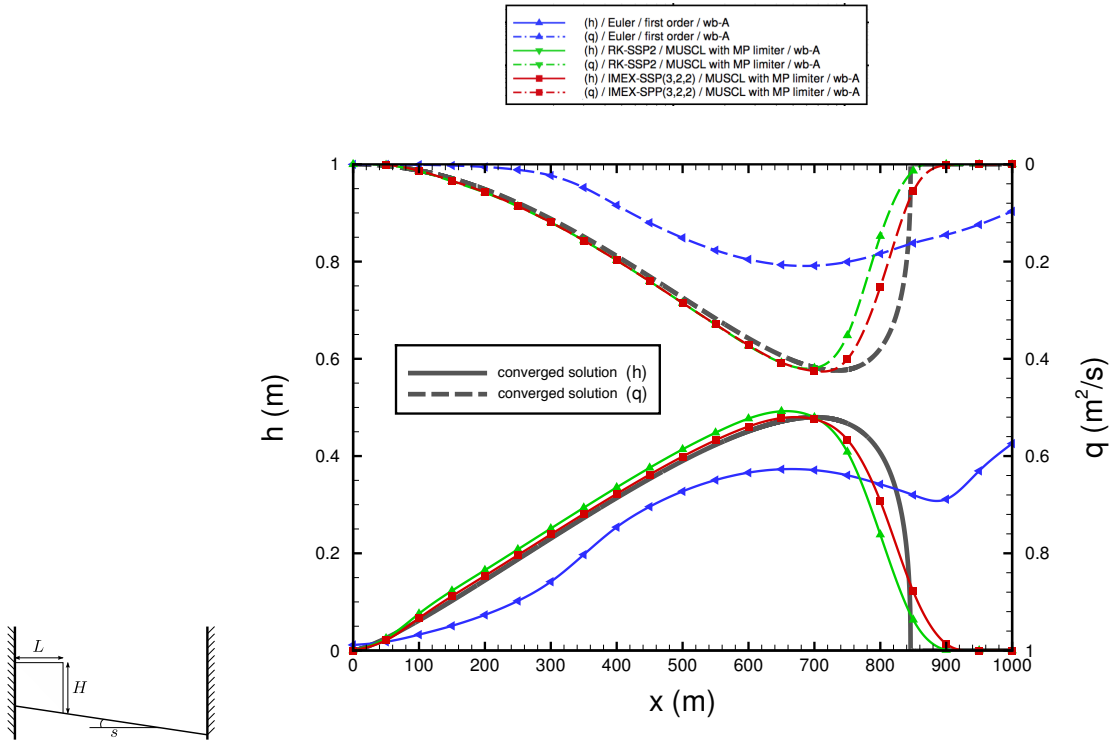


Figure 2: Dam break test case (computed in 2D). (Left) sketch. (Right) reference solution (h, q) and computed ones on 25 cells for the 3 numerical schemes (see the legend above: 1st order in blue, RK2 MUSCL in green, IMEX RK MUSCL in red)

numerical solution computed on an extremely fine grid -12 800 cells- Fig. 2 (Down). It is considered as almost exact. The "computed solution" is obtained on a rough mesh: 20 cells in length only (hence 400 cells).

The dynamic wet/dry front is well captured by all FV scheme versions (first and second order coarse solutions compared to the extremely refined one). The numerical depths h remain strictly positive.

In the first order scheme case, the convergence rate of $e_1(h)$ equals 1 as expected, Fig. 1.

The same two versions of second order schemes as previously are assessed: the RK2 MUSCL scheme (RK2 is the same as Heun's method time step) and the IMEX RK MUSCL scheme. The RK2 MUSCL scheme gives a convergence rate equal to 1 only, but with a better accuracy than the first order scheme one, Fig. 1. The IMEX RK MUSCL scheme gives a convergence rate slightly greater than 1 only, but the accuracy is better than the other

schemes. Again, the optimal order (order 2) cannot be obtained since the solution at wet/dry front is singular. The test case presented in Appendix A is the regularised version of the present dam break test case; then the solution is regular at the front and the computed convergence rate demonstrates the actual second order of the IMEX RK MUSCL scheme.

The stability condition of the global schemes is the classical CFL-like condition. For more details, including details on the discrete boundary conditions, and more test cases (classical and original ones), the reader can consult [23].

Concerning the CPU time, the second order schemes require 4 / 5 times more CPU time than the first order version. This ratio of 4 / 5 is roughly due to a factor 2 for the IMEX RK time scheme and to a factor 2 for the MUSCL reconstruction.

Finally, the numerical tests presented above demonstrate that the present numerical FV schemes are robust and accurate even in presence of wet-dry front dynamics.

3. The Inverse Model

In this section, the adjoint-based inverse method is presented. Recall that the adjoint method makes possible to compute efficiently a cost function gradient. Next it leads to: a) sensitivities analyses (even for spatially distributed coefficients); b) parameter identifications and model calibration, based on an optimal control - optimisation process.

The adjoint-based method remains a key tool to perform full optimisation processes, Variational Data Assimilation (VDA), and to reduce the model parameter uncertainties. More details on the VDA method can be found for example in [33, 34]. As mentioned previously, the optimal control - adjoint based method is classical in the data assimilation community. Nevertheless, the development of robust, efficient algorithms, and their application to real-world problems remain a challenge, see e.g. [2]. Note that all the required knowledge to obtain the present results is described either below or in Appendix B and Appendix C. Variational sensitivity results are promising to improve our surface water flow analyses. The algorithms described below have been implemented into the computational software DassFlow,

[20, 19, 21].

The inverse problem is stated as an optimal control / optimisation problem. Since the uncertain parameter number is generally huge (typically the initial condition is never perfectly known), the optimisation problem is solved using the adjoint method. The adjoint code is obtained from the direct source code by source-to-source differentiation (automatic differentiation using Tapenade software, [18]). The link from the differential of the cost function and the adjoint code automatically generated is highlighted in Appendix B.

The so-called variational sensitivities are defined as being the gradient of the cost function. If the parameter considered is spatially distributed, it leads to sensitivity maps. These sensitivities are local information only since the gradient values are valid at a given control parameter value only (first order Taylor's expansion). Next, the VDA process (also called 4D-var in the literature) is described. The latter makes possible to calibrate the model (parameter identification) by making the model fit the observations (e.g. elevation water surface values).

3.1. Minimisation problem

The so-called "forward code" solves the 2D shallow-water equations as presented previously and computes a model output (or response) j ; it is a scalar valued function. The output j depends on the model input parameters \mathbf{k} ; \mathbf{k} is the control vector. It can include scalar values or spatially distributed variables. Typically \mathbf{k} can contain the (time-dependent) inflow discharge, the outflow boundary conditions (e.g. the rating curve parameters), the Manning-Strickler roughness coefficients (spatially distributed), the bathymetry, the initial condition or any combination of these "parameters".

In a data assimilation context, $j(\mathbf{k})$ 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 educational resources for more details for example in [33, 34].

A typical cost function in the present data assimilation context reads as

follows:

$$j(\mathbf{k}) = \int \|\phi^{obs} - \phi(\mathbf{k})\|_*^2 d\mathbf{x} + \text{regularisation terms} \quad (3)$$

In the present context, the quantity ϕ denotes generally the water elevation. Subscript obs denotes an observed quantity. Usually, the regularisation term is a quadratic term in \mathbf{k} or its derivatives, hence regularising and locally "convexifying" the cost function (Tykhonov's regularisation). Also the regularisation term is used to introduce a "good" guess on the optimal values sought.

The data assimilation problem reads as an optimal control problem as follows:

$$\min_{\mathbf{k}} j(\mathbf{k}) \quad (4)$$

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 optimisation problem. It is done classically using descent algorithms, generally quasi-Newton algorithms. Thus the computation of the gradient of j with respect to \mathbf{k} is required. 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 only.

3.2. Gradient computation

The adjoint code and the cost function gradient are obtained by code/algorithmic differentiation using Tapenade software [18]. The output corresponds to the partial derivatives of the cost function j with respect to all control variables. The non trivial 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, is shown in Appendix B. For a sake of simplicity, this link is described in the case the input parameters are: 1) the initial condition $y_0 = (h_0, \mathbf{q}_0)$; 2) the inflow discharge $Q_{in}(t)$ (boundary control, time dependent); 3) the Manning-Strikler roughness coefficient n (time-independent, spatially distributed coefficient). In this case: $\mathbf{k} = (y(0); q_{in}, n)$.

Then the total differential $dj(\mathbf{k})$ of the cost function $j(\mathbf{k})$ writes as follows :

$$dj(\mathbf{k}) = \frac{\partial j}{\partial y_0}(\mathbf{k}) \cdot \delta y_0 + \frac{\partial j}{\partial q_{in}}(\mathbf{k}) \cdot \delta q_{in} + \frac{\partial j}{\partial n}(\mathbf{k}) \cdot \delta n \quad (5)$$

The adjoint code obtained via automatic differentiation costs approximately 5 – 6 times the direct code. A basic complexity calculation shows that a factor 3 is a minimum, while in practice a factor 5 is good.

The adjoint code obtained by automatic differentiation is reverse in the memory path; this process requires a huge memory amount. Furthermore, the VDA process requires a large number of minimisation iterations, typically 10 – 50. Therefore, in practice HPC codes are required; also it is highly desirable to introduce some memory optimisation tricks within the adjoint code. Developing an affordable (direct + adjoint) computational code requires few crucial tricks. In a MPI context, these tricks are detailed in [23]. They make possible to decrease the (huge) memory required by the automatic differentiation. As a demonstration of the efficiency of DassFlow computational code, a speed-up curve is presented in Appendix B.

3.3. Variational sensitivities

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

$$j(\mathbf{k} + d\mathbf{k}) \approx j(\mathbf{k}) + \nabla j(\mathbf{k}) \cdot d\mathbf{k} \quad (6)$$

at first order (Taylor’s expansion). Thus, the gradient value $\nabla j(\mathbf{k})$ provides a local sensitivity of the cost function (model output) with respect to the input parameters. In other words, the i -th value $\frac{\partial j}{\partial k_i}(\mathbf{k})$ gives the sensitivity of the model output with respect to the i -th parameter, e.g. the Manning-Strickler coefficient at one location point. These sensitivities are local since they are valid at a given point \mathbf{k} only (Taylor’s expansion). Nevertheless, the resulting sensitivity analysis tool is an extremely interesting tool leading to *better understanding of both the physics and the model* by quantifying the roles of the physical parameters and the influences of parameter variations.

3.4. Data assimilation (twin experiments)

As mentioned previously, the VDA approach consists in solving the optimisation problem (4). The minimising procedure operates on the control vector \mathbf{k} to generate a new set of parameters making the model output closer to the observations. The VDA process is sketched in Figure 3.

Here, the classical quasi-Newton L-BFGS algorithm is employed; more precisely, the L-BFGS algorithm implemented in the routine from [35].

Let us point out that each variable of the control vector may be active or not, as an actual control variable. In practice, of course it is possible

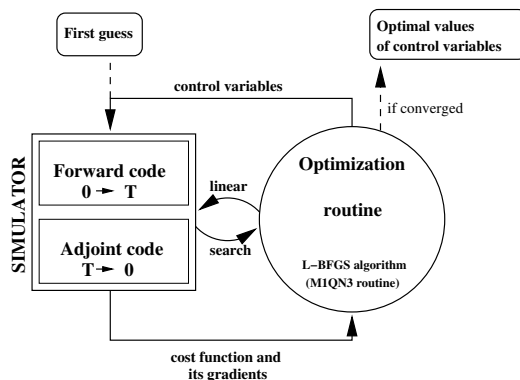


Figure 3: Principle of a 4D-Var type variational data assimilation algorithm.

to identify simultaneously only few unknown parameters (control variables); the identifiability depending on the available observations.

The methodology of twin experiments is used. They are designed as follows : the reference model parameters k_{ref} are used to generate the observations y_{obs} (synthetic data). The latter are perturbed by a realistic Gaussian noise, hence generating real-like observations. Next, the inverse problem consists to retrieve the set of parameters k_{ref} starting from an initial guess $k \neq k_{ref}$ and by performing the minimisation process. At each iteration l , the VDA algorithm computes a new set of parameters k^l (according to the gradient $\frac{\partial j}{\partial k}$) making decrease the cost function value j . The parameter inferences presented below in the case of Lèze river are obtained following this methodology.

4. A Food Plain Dynamic: Leze River (southwest of France)

In this section, the inverse computational method capabilities are demonstrated on a real data case: a portion of the Leze river, southwestern of France. First, the robustness of the developed numerical schemes are highlighted in this context (a real flood plain dynamic). Comparisons between the first and second order schemes are presented and commented. Next, the inverse capabilities of the full model are shown: sensitivity maps (gradient values spatially distributed) and data assimilation. Two twin experiments are performed: 1) the identification of the friction coefficient in 6 land covers; 2) the inflow discharge $Q_{in}(t)$. These numerical experiments

demonstrate the capabilities of the present variational inverse method in a real-like flood plain case.

Also, these numerical tests assess the present original VDA process in presence of wet-dry front dynamics over complex topography: accurate, stable direct and adjoint models, sensitivity analysis, parameter identifications and calibration processes. All the algorithms have been implemented in the open-source computational software DassFlow [21],

4.1. The river configuration

Lèze River is a 70km long river in southwestern of France near Toulouse. The case presented hereafter is a 2 km long subdomain centred on the hydrological station of Lezat-sur-Leze. The domain is discretised with a relatively coarse mesh, 24,632 cells. Topography data come from local surveys (LIDAR data); the resulting DEM is extremely fine; it has been projected onto the computational mesh.

This real test case presents a quite complex topography, in particular because of the presence of a cross road and a bridge (it is the location where the gauge station is), see Fig. 4.

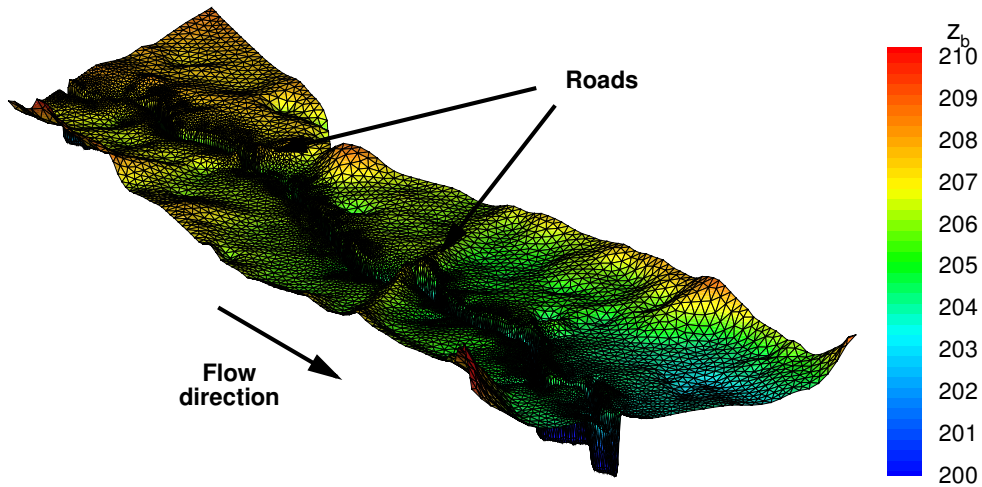


Figure 4: Leze river topography and mesh (24632 cells)

The domain boundaries consist in two open boundaries at upstream and downstream sections; wall boundary is applied elsewhere. At upstream boundary, the inflow hydrograph corresponding to the exceptional flood event of June 2000 is imposed, Fig 5(a). An experimental 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 (flood-plains). The two values of Manning friction parameters are : $0.1 \text{ s.m}^{-1/3}$ for the stream-bed and $0.05 \text{ s.m}^{-1/3}$ for floodplains. 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 flood event (no accurate surveys are available).

4.2. Direct computations and scheme comparisons

In this section, the direct model only is performed; this shows some flow features. Also, some comparisons between the first order scheme and a second order scheme are presented. In this section, the "1st order scheme" denotes the Euler / A-well-balanced scheme, and the "2nd order scheme" denotes the IMEX/MUSCL A-well-balanced scheme, see Section 2.

The simulation time is $T=63$ hours; the max CFL number is 0.8 which corresponds to 2,880,282 time steps for the first order scheme, and 2,872,208 time steps for the second order scheme.

The Froude number equals 0.3 at maximum (during the flood peak). The CPU computation times are: about 4h (5ms per dt) if using the first order scheme and about 17h (21ms per dt) if using the second order scheme.

The resulting hydrographs Q_{out} obtained at outflow are compared in Fig. 5(a). Both schemes give similar outflow discharges; excepted at $t = 19\text{h}$, a maximal discrepancy of 8% is observed.

The net mass balance (total water volume) in the computational domain is plotted vs time in Fig. 5(b). An over-estimate of the net mass balance by the 1st order scheme can be noticed (by a factor of 20% at maximum). This is due to an over-estimate of the water level h in the stream-bed, see Fig. 6 for h at the three "stations", and Fig. 7 for h at two cross-sections. Also, it can be noticed that the first order scheme gives slightly lower velocities in the stream-bed than those obtained using the second order scheme.

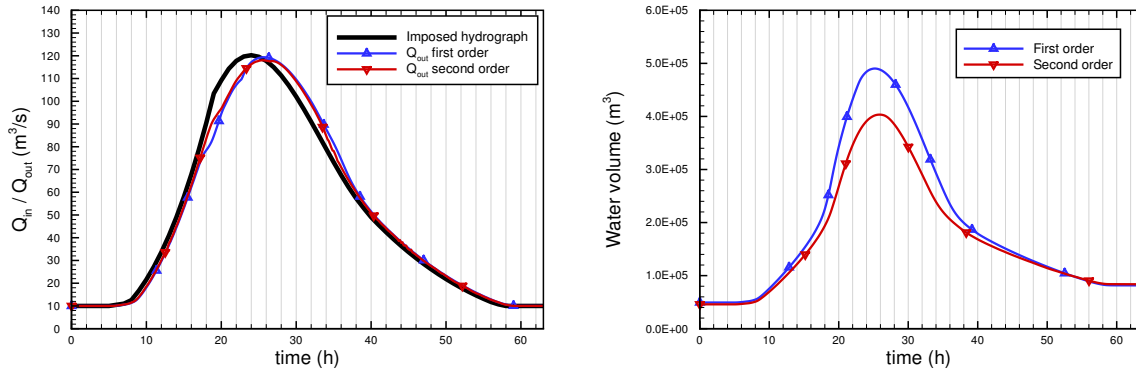


Figure 5: Left : Prescribed hydrograph at inflow (black curve), resulting outflow hydrographs Q_{out} obtained from the 1st and 2nd order schemes. Right: Water volume in the computational domain vs time (in hours): $V = V_0 + \int_0^T (Q_{in} - Q_{out}) dt$ obtained from the 1st order and 2nd order schemes.

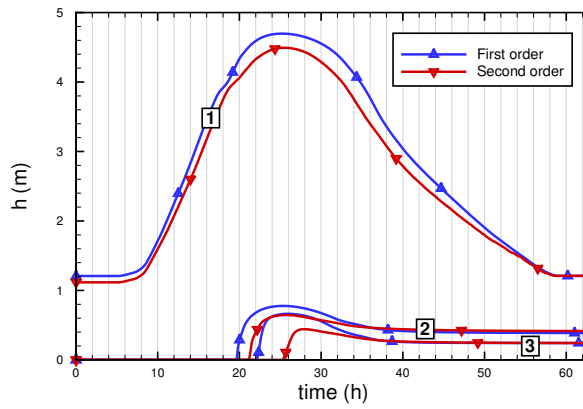


Figure 6: Time evolution of the water depth h at stations 1, 2 and 3 (see their locations in Fig.8). For each stations, the first and second order results are plotted.

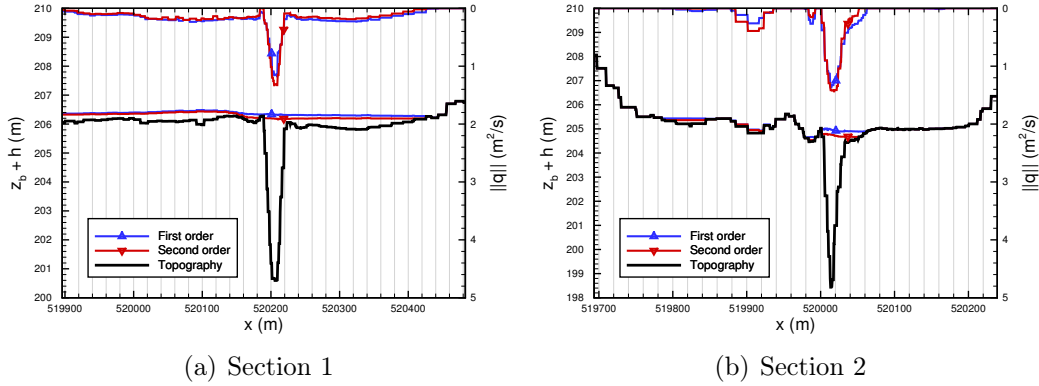


Figure 7: Elevations z_b and $H = (z_b + h)$ (in meters) at time $t = 26h$ at Section 1 (Left) and Section 2 (Right). See the section locations in Fig.8. First and second order results are plotted.

If comparing the spatially distributed water depth h at the inflow peak time, the two schemes give slightly different flood plain patterns (in particular downstream the crossing road); this slight difference is not plotted.

In conclusion, as expected, given a realistic computational mesh and DTM accuracy, the hydraulic model accuracy difference obtained between a first order scheme and an actual second order scheme, is significant. Depending on the modelling context, this difference can be secondary or not (depending on the other uncertainties of the model). The second order requires a CPU time 4 / 5 times higher than those for the first order.

Finally, let us recall that the front dynamics can be affected by numerical, unphysical regularisation (in cells where h tends to 0). The present numerical solvers, both first and second orders, do not introduce such regularisation at the front, hence do not introduce this potentially numerical modelling error.

4.3. Sensitivity maps

In this section, variational sensitivities are performed with respect to the Manning-Strickler roughness coefficient n and the bathymetry z_b . It is spatially distributed quantities hence these sensitivities leads to sensitivity maps. In addition to demonstrate the algorithms efficiency, these inverse computations highlight the similarity and the difference between these intimately linked basal fields: the (bathymetry,friction) pair.

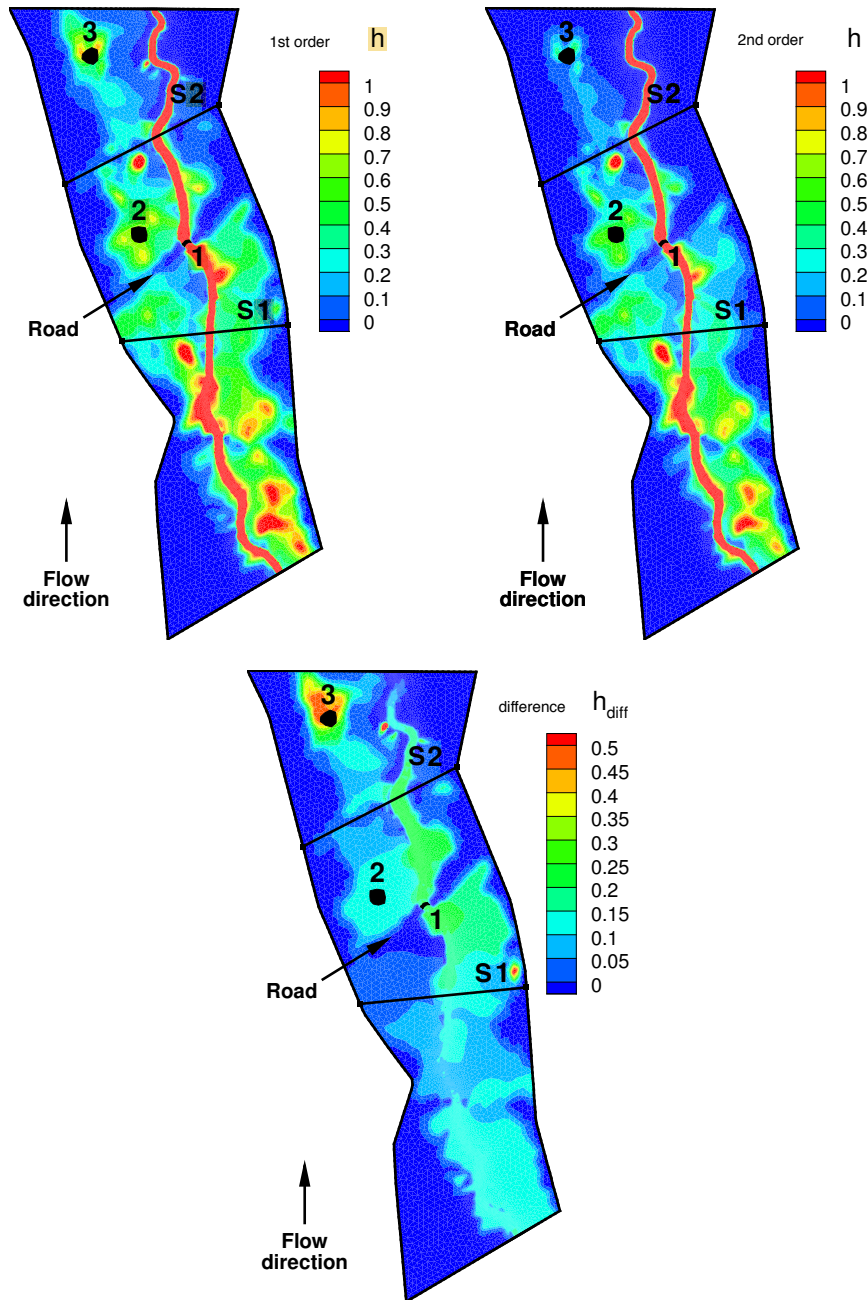


Figure 8: Top views of water depth h at $t = 26$ h (inflow peak instant), locations of the 3 observation points/areas, locations of the 2 cross-sections. (Up left) first-order computed depth h . (Up right) second order computed depth h . (Down) Difference between the first and second order depth Δh .

Let us recall some order of magnitudes in terms of CPU time only (recall that the memory storage may be the biggest issue to address). The adjoint model costs 4-5 times more than the direct model. A complete VDA process requires approximatively 20 minimisation iterations hence it costs approximatively $(1 + 6) \times 20 = 140$ times the direct model.

The second order scheme costs 4-5 times more than the first order one. Hence if using the second order scheme, a complete VDA process costs approximatively $4.5 \times 4.5 \approx 20$ times more than those if using the first order scheme. Since the present goal is to compare sensitivities maps and demonstrate the complete numerical chain capabilities, the forward numerical solver employed here is the first order scheme.

Boundary conditions are identical to the previous section: the discharge is imposed at inflow and a rating curve is imposed at outflow. The measurements are generated by the forward model (synthetic data); next, a realistic Gaussian noise is added to these numerical observations. For all experiments, the measurements are time-series of water elevation at the virtual stations 1 and 2, see Fig. 9. The recorded water elevation values are perturbed by a random noise of +/- 10 cm (representing realistic error measurements). Station 2 measurements are based on the average of a dozen of cell values. Similarly, Station 1 measurements are the average of a dozen of cell values representing a mean cross-section value in the stream-bed.

The Manning-Strickler coefficients used to generate the data are defined by areas; their values are given in Table 1.

The cost function is defined as follows:

$$j(n, z_b) = \sum_{\text{time step}} \sum_{m \text{ station } k} (h_{m,k} - h_{m,k}^{obs})^2 \quad (7)$$

The sensitivity maps with respect to the Manning coefficient n , Fig. 10(a) and with respect to the bathymetry z_b , Fig. 10(b), are the corresponding gradient components for each mesh cell. They are simultaneously obtained by performing one direct - adjoint run. Let us recall that the model outputs are the water elevations at the two stations 1 and 2 (see the cost function j defined by (7)). The gradient values (the sensitivity maps) are relative to the observations (the water elevations at stations 1 and 2) and relative to the "computational point" value (i.e. the value of the fields n and z_b used in

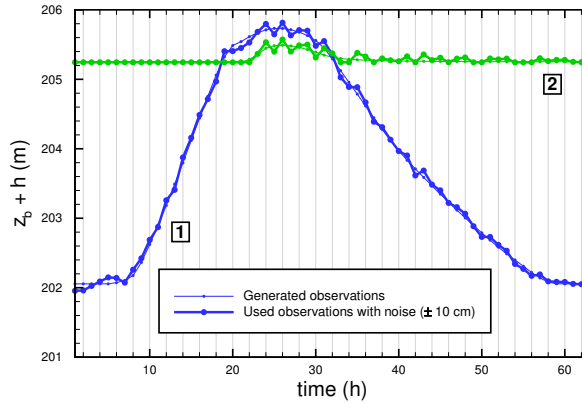


Figure 9: Water surface elevation measurements at Station 1 (blue) and at Station 2 (green). It is synthetic data with realistic random noise added ($\pm 10\text{cm}$).

the forward model), see (6).

The gradient value with respect to the Manning friction coefficient is roughly 2 orders of magnitude higher than those with respect to the bathymetry. For a sake of clarity, the gradient values plotted in Fig. 10 have been normalised. (Also recall that for numerical purposes the cost function normalisation is highly recommendable before calling the minimizer).

The highest sensitivities with respect to the friction coefficient n are downstream of the observation location areas. This sensitivity distribution is logical since the flow is everywhere sub-critical (the Froude number is everywhere lower than 0.3), hence the water depth at one location is controlled by downstream too.

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 a standard regime (i.e. without overflowing) is highly desirable before calibrating the same model but applied to a flood event.

The highest sensitivities with respect to the bathymetry z_b are located at the observation locations. This is logical since the direct correlation between the measurement (water depth) and the bed elevation.

The bathymetry sensitivity pattern is globally comparable to the friction sensitivity one but more point-wise, less diffused. This remark corroborates the

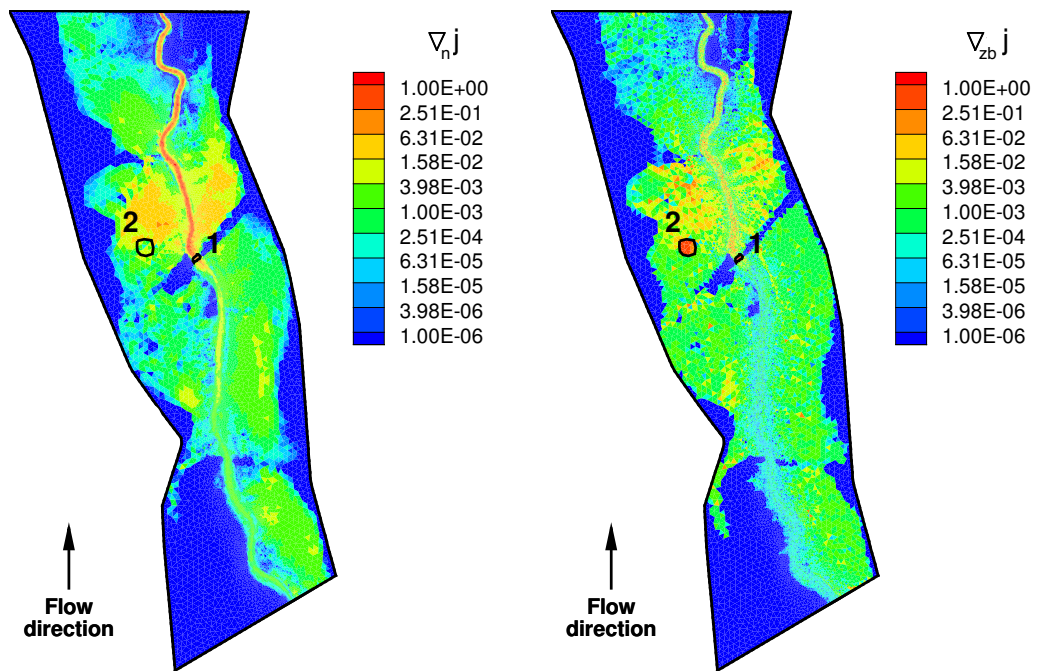


Figure 10: Sensitivity map with respect to: (left) the spatially distributed roughness coefficient n ; (right) the bathymetry z_b . The gradient values plotted are normalised. Data: water elevation at the 2 stations at all time, see Fig 6.

analyses presented in [36] (sensitivities performed on the linearised 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 (potentially making fit accurately the model to data) would not result to an intrinsic model, hence providing a model not necessarily predictive. In other words, these similar patterns illustrate the potential equifinality problem related to the (topography-friction) pair in the 2d shallow-water equations. The (topography-friction) pair equifinality problem and the difficulty to infer this "basal modelling pair" are discussed for example in [37] for 1d flows.

Finally, let us point out that spatially distributed sensitivity maps can greatly help the hydraulic modeller to better understand both the hydraulic model (combining the DTM, the parametrisation) and the flow. Obviously, these sensitivity maps depend on the sensitivity cells considered. For example, changing one cell friction (or topography) value in the river bed does not affect the flow like if changing the value of all cross-section cells.

4.4. Data assimilation and model calibration

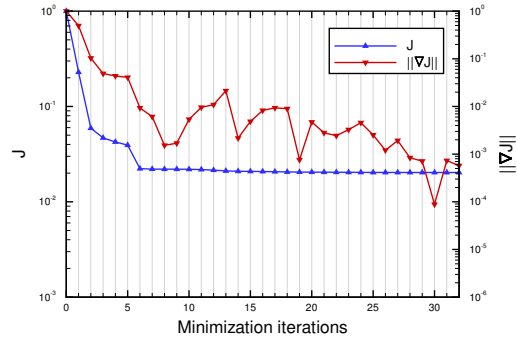
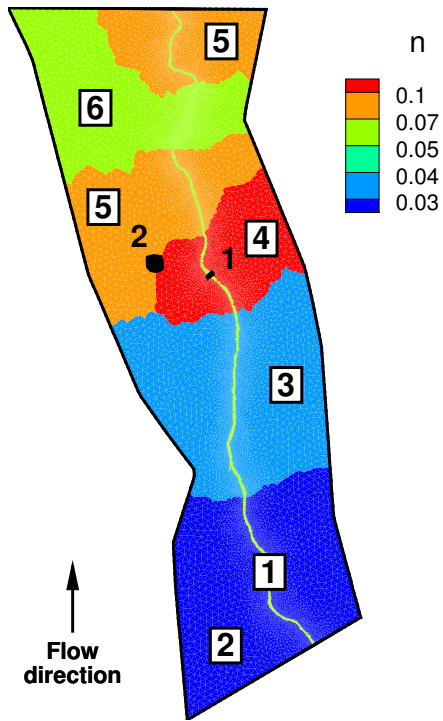
In this section, two twin experiments are performed: 1) the identification of the friction coefficient n in 6 land covers; 2) the identification of the inflow discharge $Q_{in}(t)$. The direct model and the assimilated data are the same as previously (time series of water elevation H at two locations). These two assimilation experiments demonstrate the capabilities of the present inverse computational method in a real like context. Recall that such identification parameters have already been performed in hydrodynamics, including for flood plain flows, see the references mentioned in the introduction. Nevertheless, such inverse computations based on a non regularized FV scheme at the front and in a HPC context (parallel architecture making possible larger computations) are new in a hydrodynamics context. Recall that the water front position is a crucial feature of such flows; furthermore it is the most easily observed quantity (e.g. from satellite). Deriving a reliable numerical model for such complex flows with the present inverse capabilities remains a challenge. Such inverse computational capabilities should make improve the flood plain modelling in particular.

4.4.1. Roughness coefficient identification

The twin experiment performed aims at identifying the Manning coefficient values n for each land cover, see Fig. 11(a).

The entire stream-bed corresponds to one land cover, while the computational flood plain domain is divided into 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. The boundary conditions (inflow discharge, outflow law) and the 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 1. If considering perfect observations (no noise introduced), then the target values are *perfectly* recovered. If the synthetic observations H are perturbed by a +/- 10cm Gaussian noise, the identified friction coefficients are still very good; it is the results presented in Table 1.

An accurate convergence is reached in 32 iterations, see Fig 11(b) (the stopping criteria have been set extremely small to make decrease as much as



(a) Land covers and Manning coefficient values

(b) Cost function and gradient norm vs minimisation iterations (Case: friction coefficients identification)

Figure 11: Left: The land covers, friction coefficient values and the 2 observation stations. Right: The cost function j and the gradient norm $\|\nabla j\|$ vs minimisation iterations. Normalised values.

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 1: The Manning coefficient values n identified per land cover (water elevation measurements include +/- 10 cm Gaussian noise).

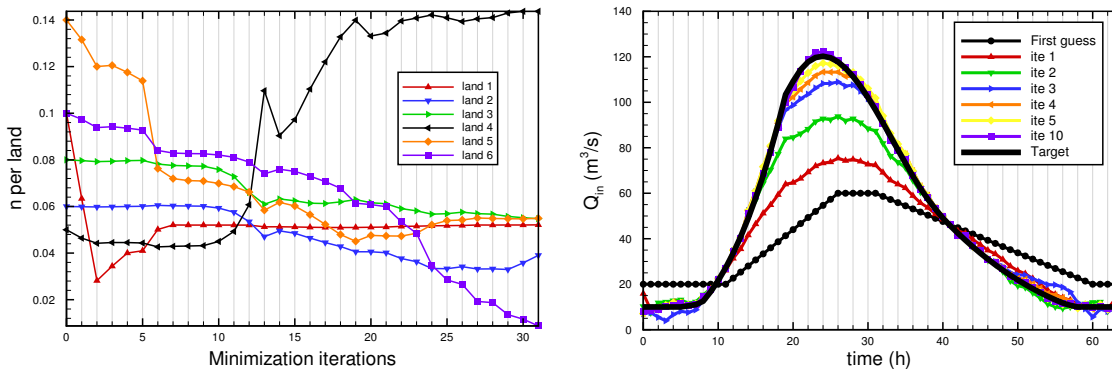


Figure 12: Minimisation process. (Left) The 6 Manning coefficient values vs iterations (iterate 0=first guess). (Right) Hydrographs (inflow discharge in time) at few iterations.

possible the cost function).

It can be noticed that very quickly (after the 6th iterate), the cost function has almost reached its minimal value but the control parameters did not converged yet, see Fig. 12 (L), excepted the dominating one: the stream-bed friction coefficient (land cover #1). The algorithm behaviour is coherent with the presence of a dominating parameter influence. Next, all the remaining friction values can be recovered only if the gradient values are computed accurately, hence an accurate direct and adjoint solvers. Also the minimisation algorithm employed must be efficient and accurate; here it is the L-BFGS version presented in [35].

The present data assimilation test case is based on synthetic data but real-like. It demonstrates these inverse algorithms capabilities to improve the

modelling of 2D shallow wet dry front dynamics, and in particular flood plain dynamics models.

4.4.2. Inflow discharge identification

The present twin experiment aims at identifying the inflow hydrograph. The forward model is strictly the same as previously; the friction coefficients are set to the target values indicated in Table 1. The observations generated at the two virtual stations are the same as previously (synthetic water elevation values plus a Gaussian noise); the boundary conditions (inflow discharge, outflow law) are the same too. The cost function is defined as follows:

$$j(Q_{in}) = \sum_{\text{time step } m} \sum_{\text{station } k} (h_{m,k} - h_{m,k}^{obs})^2 + \alpha (Q_{in} - Q_{in}^{filtered})^2$$

The second term in the cost expression is a regularisation term (Tykonov-type term) aiming at smoothing the identified parameter $Q_{in}(t)$. The coefficient α equals 10^{-3} ; the smoothed function $Q_{in}^{filtered}(t)$ is defined as a low-pass filter based on a window size of 16 dt.

The target hydrograph, the first guess hydrograph and few intermediate hydrographs are plotted in Fig. 12 (R). Hence the first guess hydrograph maximum value is twice lower than the target hydrograph maximum value; its minimum value is twice greater; furthermore, it presents a phase shift of 4 hours approximatively.

The convergence curves (the decreasing curves of j and $\|\nabla j\|$ vs iterations) are not plotted since they are similar than those in Fig. 11(b): a fast decrease of j the first 7 iterations then a much slower decrease of j , while the gradient norm $\|\nabla j\|$ keep decreasing until an accurate convergence is reached at iterate 20. In other words, the target hydrograph is perfectly retrieved by the minimisation process at iterate 20 while it is already very close at iterate 7, see Fig. 12 (R).

This numerical experiment demonstrates the inverse algorithms capability to infer inflow hydrographs corresponding to a flood plain event, given time series of water elevation. The present identifications are very accurate because the assimilated data are very constraining, in particular the water elevation time series in the stream-bed. Less dense data would lead to less accurate identified inflow discharges. The present inverse algorithms capabilities, reliability and efficiency would make possible to resume hydrodynamics

studies like those initiated in [14, 15] for example, or those in progress for example in the context of the future SWOT space mission [38], the first mission partly dedicated to terrestrial surface waters.

5. Conclusion

In this study, a complete inverse computational chain dedicated to the 2D hydrodynamics shallow flows with wet dry fronts has been developed. As a first step, a particular attention has been paid to the derivation of conservative FV schemes, accurate and stable for the 2D SWE in presence of dynamic wet/dry fronts. The challenge has been to do not introduce any numerical regularisation at the front since such regularisation (numerical cut-off on the water depth somehow representing a thin film water at rest) may modify the front velocity hence the water extend. An innovative FV scheme has been built up from a combination of classical methods (HLLC solver, well-balanced treatment, MUSCL reconstruction and IMplicit-EXplicit Runge-Kutta time scheme) completed with a less classical modification (the intermediate wave speed in the solver). The resulting global schemes has been demonstrated to be naturally stable in presence of dynamic wet/dry fronts, accurate and *actually* second order in the case of the second order version. These FV schemes have been assessed and compared into details on classical benchmarks and also on more original benchmarks presenting the wet dry front dynamics difficulties.

Next variational inverse algorithms have been derived. These inverse algorithms are based on the adjoint equations of the numerical schemes (the adjoint code is generated by using a source-to-source automatic differentiation tool); hence circumventing potentially complex derivations of the wet dry front treatments. This variational approach provides precious sensitivity information making possible a better understanding of the flow and the model. Next, the full Variational Data Assimilation (VDA) processes make possible the identification of uncertain input "parameters", time-series like the inflow discharge or spatially distributed like the friction coefficient.

The variational approach is widely used in other geophysical flow modelling (atmosphere, oceans) but much less in hydrodynamics. In hydrodynamics, the challenges are to derive reliable direct and inverse solvers for the wet dry front dynamics in particular since the water extent may be a key feature of the observed flow, and affordable computations codes both in terms of CPU time and memory amount. The present algorithms and resulting computa-

tional chain undertakes this challenge.

The typical applications of the present model and algorithms are river hydraulics (flood plain dynamics, low-water), some coastal, estuarine and tidal flows. The present algorithms capabilities and efficiency (direct and inverse) have been demonstrated on a real flood plain case (Lèze river, France, past flood event of june 2000). Given dense in-situ measurements (e.g. water elevation time series), original sensitivity maps with respect to the friction parameter and the bathymetry have been performed and compared. Next, the identification of inflow discharge (time series), and the spatially distributed friction parameter have been performed correctly. These numerical experiments on a real test case, demonstrate the robustness and the capabilities of the present inverse computational algorithms, making possible great improvements of flood plain modelling, in particular in the context of fast growing remote-sensed data volume acquired (e.g. the forthcoming ESA Sentinel missions and Nasa/Cnes SWOT mission).

As it is well-known and already mentioned, the adjoint method has few drawbacks: the potentially complex generation of the adjoint code, its CPU-time and memory use. Nevertheless, it has been demonstrated that if the direct computational code is designed for VDA, using automatic differentiation it is possible to derive quite quickly a reliable assimilation chain. Next, to circumvent the memory and CPU time drawback of the VDA method, an incomplete adjoint model, tuneable in terms of accuracy and CPU time, leading to a large gain of computational efficiency with a minimal accuracy loss, could be considered like it has been done in [39]. Finally, let us recall that all the algorithms elaborated here (and the corresponding computational software) can be applied to other flows satisfactorily modelled by the 2D SWE with wet-dry fronts, like tidal flows or some coastal flows.

Appendix A. Finite Volume Solvers Assessment

The present section aims at assessing the *actual* accuracy and convergence rate of the numerical FV solvers. To do so a regular solution is compulsory, hence a flow without wet/dry front. Then the test case is a "regularised" version of the dam-break benchmark presented in Section 2. The non-linear Manning-Strickler friction term is still taken into account.

If using the IMEX RK time scheme, the convergence rate obtained equals 2 while for if using the RK2 time-scheme, it equals 1 only. In other words, if not considering the right time-space combination, an a-priori order 2 FV scheme can be actually order 1...

The test case is as follows. The initial condition is defined by:

$$\begin{cases} z_b(x) = & 0.5 e^{-(x - l_x/2)^2 / 2\sigma^2} \\ h(x, t = 0) = & 0.1 + 0.5 e^{-(x - l_x/2)^2 / 2\sigma^2} \end{cases} \quad (\text{A.1})$$

with $\sigma = 100 \text{ m}$, see Fig. A.13. The computational domain length equals $l_x = 1000 \text{ m}$; the Manning-Strickler coefficient $n = 0.05$; the total simulation time is $T = 100\text{s}$. A reference simulation is computed on an extremely small grid (12,800 cells); it is considered as exact. The initial condition and the "exact" solution (h, q) are plotted on Fig.A.13. It can be noticed that the solution is regular.

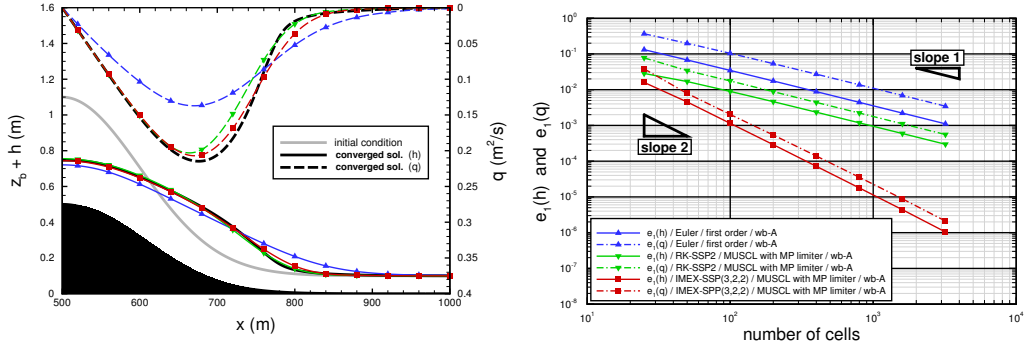


Figure A.13: Regularised dam break test case. (Left) (half-domain): initial condition, reference solution (h, q) in black, computed solutions on 25 cells for the 3 numerical schemes: 1st order in blue, RK-SSP2 in green, IMEX-SPP(3,2,2) in red. (Right) Relative error norms $e_1(h)$ and $e_1(q)$ vs the number of cells.

The relative error norms $e_1(h)$ and $e_1(q)$ vs the cell numbers are plotted in Fig.A.13(L). The RK2 MUSCL scheme (denoted RK-SSP2 in Fig.A.13(R)) leads to a globally first order scheme only. Nevertheless, its accuracy is higher than the first order scheme accuracy. The IMEX RK MUSCL scheme

(denoted IMEX-SPP(3,2,2) in Fig.A.13(R)) leads to an actual second order convergence rate.

This test case demonstrates the *actual* second-order accuracy of the global scheme resulting from the right combination of: the HLLC solver with the intermediate wave velocity presented in [8], the well-balanced numerical solver defined in [10], a standard MUSCL reconstruction and the IMEX RK time stepping method. To our best knowledge, such a demonstrated second-order accuracy is not classical in the literature.

Appendix B. On the gradient derivation

Let us describe the existing 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 $y_0 = (h_0, \mathbf{q}_0)$; 2) the inflow discharge q_{in} (boundary control, time dependent); 3) the Manning-Strikler roughness coefficient n (time-independent, spatially distributed coefficient). In this case: $\mathbf{k} = (y_0; q_{in}, n)$.

The total differential $dj(\mathbf{k})$ of the cost function $j(\mathbf{k})$ writes as follows :

$$dj(\mathbf{k}) = \frac{\partial j}{\partial y_0}(\mathbf{k}) \cdot \delta y_0 + \frac{\partial j}{\partial q_{in}}(\mathbf{k}) \cdot \delta q_{in} + \frac{\partial j}{\partial n}(\mathbf{k}) \cdot \delta n \quad (\text{B.1})$$

We show how the output of the adjoint code generated by algorithmic differentiation, and using Tapenade software for example [18], corresponds to the partial derivatives of the cost function j with respect to the control variables. The derivation below follows those presented in [40].

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: $\mathcal{M} : \mathcal{K} \rightarrow \mathcal{Y}$ with:

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

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

The tangent model writes: $\frac{\partial \mathcal{M}}{\partial \mathbf{k}}(\mathbf{k}) : \mathcal{K} \rightarrow \mathcal{Y}$. 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: $(\frac{\partial \mathcal{M}}{\partial \mathbf{k}}(\mathbf{k}))^* : \mathcal{Y}' \rightarrow \mathcal{K}'$. 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:

$$\left\langle \left(\frac{\partial \mathcal{M}}{\partial \mathbf{k}} \right)^* \cdot dY^*, d\mathbf{k} \right\rangle_{\mathcal{K}' \times \mathcal{K}} = \left\langle dY^*, \left(\frac{\partial \mathcal{M}}{\partial \mathbf{k}} \right) \cdot d\mathbf{k} \right\rangle_{\mathcal{Y}' \times \mathcal{Y}} \quad (\text{B.2})$$

where $\langle \cdot, \cdot \rangle$ denotes the duality product in the indicated spaces.

Using the relations presented above, (B.2) reads:

$$\langle d\mathbf{k}^*, d\mathbf{k} \rangle_{\mathcal{K}' \times \mathcal{K}} = \langle dY^*, dY \rangle_{\mathcal{Y}' \times \mathcal{Y}}. \quad (\text{B.3})$$

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:

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

Moreover, by definition:

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

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 :

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

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 executed, second the adjoint code is executed with $dY^* = (0, 1)$ as an input variable.

Gradient validation. Finally let us clarify that how the adjoint code (MPI or not) is classically validated. As usual, a gradient test is performed by computing the quantity $(\frac{j(k+\varepsilon\delta k)-j(k)}{\varepsilon})$ with ε strictly positive, small. This ratio has to converge to the partial derivative of the cost function $\frac{\partial j(k)}{\partial k} \cdot \delta k$ when ε tends to 0. The finite difference approximation is evaluated by performing twice the direct code with $\delta k = \mathbf{1}$ or a random vector of the right order of magnitude. Next a convergence curve in function of ε is plotted; the difference between the computed gradient using the adjoint code and the finite difference approximation (using the direct code only) converge at order 1 in ε . The same convergence curve is obtained while using an adaptive time step for the forward code.

Appendix C. Speed-up of the full MPI computational code

The direct code is written in Fortran and uses the MPI library. The automatic differentiation softwares, source-to-source, do not handle MPI instructions yet; in particular Tapenade software. Then, to derive a MPI version of the adjoint code, it is required to re-write "by hand" the adjoint instructions of the MPI standards calls such as MPI_SEND, MPI_RECV or MPI_ALLREDUCE. This known-how is described into details in [21, 23]; it follows some techniques described in [41].

Then, it becomes interesting to measure the efficiency of the resulting whole MPI inverse code (i.e. the direct code plus the adjoint code). In Fig. C.14 is presented the speed-up curve obtained if performing sensitivity analysis (Lèze river test case, 25 000 cells approx.). For example, at 32 processors, 82.5% of the idealistic speed-up is reached. A speed curve performed on a larger mesh (containing typically 1 million of cells) should give a good performance up to a much higher processor number. This curve illustrates the good operating of the technics and known-hows described in [23] to obtain the MPI version of the adjoint instructions. All these methods and known-hows are implemented in the open source software DassFlow [21].

Acknowledgments

The authors warmly thank J. Chorda from CNRS & IMFT for the mesh generation and his help to set up the hydraulic model based on the real topography data. The authors warmly thank the four reviewers for their constructive help and suggestions to improve the manuscript.

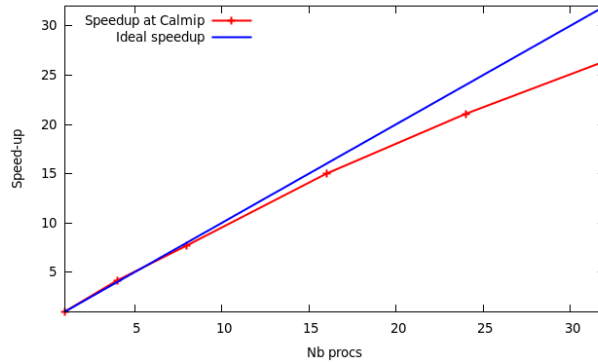


Figure C.14: Speed-up obtained on a sensitivity computations (DassFlow software, Lèze river case, 25,000 cells, Calmip cluster Toulouse).

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. The authors warmly thank P.-A. Garambois, INSA Strasbourg & ICUBE, for his final lecture of the manuscript.

- [1] K. Yan, G. Di Baldassarre, D. P. Solomatine, G. J.-P. Schumann, A review of low-cost space-borne data for flood modelling: topography, flood extent and water level, *Hydrological Processes* 29 (15) (2015) 3368–3387.
- [2] P. D. Bates, Integrating remote sensing data with flood inundation models: how far have we got?, *Hydrological Processes* 26 (16) (2012) 2515–2521.
- [3] S. C. Medeiros, S. C. Hagen, Review of wetting and drying algorithms for numerical tidal flow models, *International Journal for Numerical Methods in Fluids* 71 (4) (2013) 473–487.
- [4] E. Toro, *Shock-capturing methods for free-surface shallow flows*, John Wiley, 2001.
- [5] V. Casulli, A high-resolution wetting and drying algorithm for free-surface hydrodynamics, *International Journal for Numerical Methods in Fluids* 60 (4) (2009) 391–408.

- [6] B. Einfeldt, On godunov-type methods for gas dynamics, *SIAM Journal on Numerical Analysis* 25 (2) (1988) 294–318.
- [7] E. F. Toro, M. Spruce, W. Speares, Restoration of the contact surface in the hll-riemann solver, *Shock waves* 4 (1) (1994) 25–34.
- [8] J.-P. Vila, Simplified godunov schemes for 2 x 2 systems of conservation laws, *SIAM J. Numer. Anal.* 23 (6) (1986) 1173–1192.
- [9] 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.
- [10] 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.
- [11] V. Guinot, *Wave propagation in fluids: models and numerical techniques*, John Wiley & Sons, 2012.
- [12] 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.
- [13] E. Belanger, A. Vincent, Data assimilation (4D-VAR) to forecast flood in shallow-waters with sediment erosion, *Journal of Hydrology* 300 (2005) 114–125.
- [14] 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.
- [15] 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.
- [16] 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.

- [17] 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.
- [18] 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>
- [19] 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).
- [20] Dassflow computational software: Data assimilation for free surface flows, open source.
URL <http://www.math.univ-toulouse.fr/DassFlow/>
- [21] C. Couderc, R. Madec, J. Monnier, J.-P. Vila, Dassfow-shallow: Numerical schemes, user and developer guides, Tech. rep., Mathematics Institute of Toulouse (IMT), CNRS - INSA.
- [22] L. Pareschi, G. Russo, Implicit-explicit runge-kutta schemes and applications to hyperbolic systems with relaxation, *Journal of Scientific Computing* 25 (2005) 129–155.
- [23] C. Couderc, J. Monnier, J.-P. Vila, K. Larnier, R. Madec, D. Dartus, Robust finite volume schemes and variational inversions for 2d shallow water models. application to flood plain dynamics., Research report, Math. Inst. of Toulouse (IMT) and Fluid Mech. Inst. of Toulouse (IMFT) (2014).
URL <http://hal.archives-ouvertes.fr/hal-01133594>
- [24] R. J. LeVeque, Finite volume methods for hyperbolic problems, Vol. 31, Cambridge university press, 2002.
- [25] 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 Mathématique* 338 (1) (2004) 85 – 90.

- [26] 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 hyperbolic systems. application to the dam break of aznacóllar, *Comput. Methods Appl. Mech. Engrg.* 197 (2008) 3932–3950.
- [27] J. Sampson, A. Easton, M. Singh, Moving boundary shallow water flow above parabolic bottom topography., *ANZIAM Journal* 47.
- [28] W. Thacker, Some exact solutions to the nonlinear shallow-water wave equations, *J. Fluid Mech.* 107 (1981) 499–508.
- [29] 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.
- [30] 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.
- [31] 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.
- [32] 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.
- [33] L. Bouttier, P. Courtier, Data assimilation concepts and methods, ECMWF Training course. [www.ecmwf.int.](http://www.ecmwf.int/), 1999.
- [34] J. Monnier, Variational data assimilation: from optimal control to large scale data assimilation, Open Online Course, INSA - University of Toulouse.
URL <http://www.math.univ-toulouse.fr/~jmonnie/Enseignement/VDA.html>
- [35] J. Gilbert, C. Lemaréchal, Some numerical experiments with variable-storage quasi-newton algorithms.
- [36] V. Guinot, B. Cappelare, 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.

- [37] P.-A. Garambois, J. Monnier, Inference of effective river properties from remotely sensed observations of water surface, *Adv. Water Res.* 79 (2015) 103–120.
- [38] 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.
- [39] 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.
- [40] M. Honnorat, Assimilation de données lagrangiennes pour la simulation numérique en hydraulique fluviale., Ph.D. thesis, Institut National Polytechnique de Grenoble - INPG (2007).
- [41] 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.