Inverse computational algorithms for flood plain dynamic modelling

J Monnier, F Couderc, D Dartus, K Larnier, R Madec, J.-P Vila

Inverse computational algorithms for flood plain dynamic modelling

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

Flood plain dynamic modelling remains a challenge because of the complex multi-scale data, data uncertainties and the uncertain heterogeneous flow measurements. Mathematical models based on the 2d shallow water equations are generally suitable but wetting-drying processes can be driven by small scale data features. The present study aims at deriving an accurate and robust direct solver for dynamic wet-dry fronts and a variational inverse method leading to sensitivity analyses and data assimilation processes. The numerical schemes and algorithms are assessed on academic benchmarks representing well some flood dynamic features and a real test case (Lèze river, southwestern of France). Original sensitivity maps with respect to the (friction, topography) fields are performed and discussed. Furthermore, the identification of inflow discharges (time series) or roughness coefficients defined by land covers (spatially distributed parameters) demonstrate the relevance of the approach and the algorithm efficiency. Inverse computational methods may contribute to breakthrough in flood plain modelling.

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

1. Introduction

Flood plain dynamic modelling is still an active research topic because of the complex multi-scale data (e.g. the topography), data uncertainties (e.g. inflow discharge) and the uncertain sparse heterogeneous flow measurements.

The flow dynamic processes are quite well understood; generally the mathematical models based on the 2d shallow water equations are suitable (in some circumstances the inertial can be neglected). Nevertheless the multi-scale data features, hence the resulting multi-scale flow features, make the modelling of flood plain dynamics a great challenge. Typically the wetting - drying processes (whatever in rural or urban areas) can be greatly correlated to small scale pathways (e.g. dykes, breaks in embankments). Then the global flow dynamic can be driven by small scale data features.

Defining small scale computations require to consider the right physical processes (typically the 2d shallow-water equations, complete or not), extremely high computational costs (High Performance Computing becomes mandatory), more and better terrain data. DTM have greatly improved with LIDAR in the 2000's; their accuracy is higher and higher. Also since recently Shuttle Radar Topography Mission (SRTM) provides rich DTM at large scale (30m horizontal accuracy). In other respect, radar and photo images taken during flood events are more and more common. Remote-sensed data from space are richer and richer; for example the forthcoming data from ESA Sentinel missions or NASA-CNES SWOT mission will make major progresses in river observations and flood plain measurements.
To become reliable in real contexts and operational, distributed flow models will be based on HPC, cocktails of heterogeneous multi-scale data and mathematical methods - algorithms combining all these information.
Data assimilation is a crucial key methodology widely used for the atmosphere and oceans modelling. Sequential methods (Kalman like algorithms) are efficient but since they require huge numbers of model runs, their uses become prohibitive for spatially distributed models. Variational Data Assimilation (VDA, also called 4D-var) based on the adjoint model, becomes the right methodology to combine at best the mathematical model, data and the flow measurements. VDA leads to model uncertainty reductions, parameter identifications and the model calibration. Thus, elaborating adjoint-based methods for 2d distributed models becomes a necessary step even if it may be not sufficient to obtain reliable predictive flow models. Indeed, the terrain data, remote-sensed data, the mathematical model and the computational grid have to be consistent each other; the numerical consistency required between all these information is not clear yet. Furthermore, an accurate descriptive model may not necessarily means reliable predictive model. Typically if the right small scale data features have not been taken into account

correctly in the mathematical model, the calibrated model (describing well the available observations) may not be accurate for prediction.

Following the research directions and the modelling challenges mentioned above, the present paper aims at elaborating an accurate and robust direct solver for dynamic wet-dry fronts (feature potentially crucial in flood plain context), next an adjoint-based variational inverse computational method leading to sensitivity analyses and data assimilation. The forward model is based on the 2d Shallow Water (SW) equations with friction term. (The inertial term is taken into account; obviously it can be straightforwardly skipped if useless). The 2d SW equations is suitable to model many types of shallow geophysical flows and waves propagation e.g. the atmosphere, oceans, coastal flows, rivers and flood plains. As already mentioned, these geophysical flows involve multi-scales data, multi-scale phenomena, badly known topographies (at best well known but at a given scale), uncertain quantities at open boundaries (e.g. inflow discharge), uncertain model parameters (e.g. roughness coefficients). In the flood plain case, the flow dynamic involve wet-dry fronts which are difficult to capture accurately, in particular since topography are rapidly varying. To circumvent this mathematical and numerical difficulty (the water depth tends to 0 which can make blow up the equations and the numerical schemes), the classical technic adopted in the literature is to introduce a numerical cut-off at the front in the momentum equation, see e.g. [1] and references therein. Then the front velocity depends fully on this unphysical regularisation (the numerical cut-off) hence distorting the flood plain dynamics. The numerical solvers derived here do not require any numerical regularisation at the front. To our best knowledge this is new (in particular in presence of rapidly varying topography). In other respect, a higher-order scheme (e.g. second order) can be very interesting in a geophysical context since it makes easier a better consistency between the data (typically the DTM given at a fixed scale) and the computational mesh. In the present study, a first step consist to derive a couple of numerical FV schemes, first order and second order, presenting the important feature mentioned above; also these numerical schemes are assessed into details and compared. The second step of the study consists to elaborate efficient computational inverse methods based on the adjoint method, providing sensitivities maps, robust and affordable variational data assimilation processes.

Deriving FV schemes based on second order methods is quite classical

(even if it remains very technical), but deriving an *actual* second order scheme remains a challenge.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. [2] treating of SW equations with bathymetry and friction. MUSCL technics comes after the pioneering work [3]. First higher-order analysis has been written in [4] on structured meshes and in [5] including for hydraulic applications. Recent reviews can be found for example in [1, 6]. The present contribution aims at elaborating a set of methods leading to an *actual* second order scheme, accurate in presence of wet-dry front situations. The approximate Riemann solver chosen is the HLL solver [7, 8] combined with an ad-hoc estimate of wave velocity [9]. A standard MUSCL technic (mono/multi-slope with corresponding limiters) is combined with a well-balanced property treatment. The well-balanced property (water at rest preserving) comes from the early work [10]. Many technics exist in the literature; in the present study it has been observed that the technics presented in [11, 12] provide a robust and accurate framework. Concerning the time stepping, it has been observed that an implicit-explicit Runge-Kutta (IMEX) scheme leads to an actual second order accuracy, while a more standard Runge-Kutta scheme (order 2) for example does not.
To our best knowledge, the present numerical scheme combination leads to a numerical solver presenting accuracy, stability and robustness features as never published in the literature. The numerical schemes have been assessed by performing classical test cases from the literature *plus* extra new ones presenting crucial features of flood plain dynamics. Detailed comparisons between the first order scheme and two second order versions are made, including on the real test case considered (Lèze river, southwestern of France).

As already mentioned, the second step of the study aims at deriving sensitivities maps and variational data assimilation processes based on the adjoint model and an optimal control loop. Variational data assimilation (also called 4D-var) is widely employed for few geophysical flows, see e.g. [13] and references therein. Given some measurements of the flow (typically time-series at gauge stations and/or remote-sensed observations), sensitivity analyses and data assimilation algorithms make possible to identify boundary conditions (e.g. inflow discharge) and uncertain parameters (e.g. friction parameters), hence reducing model uncertainties. Concerning 1d river flow models (Saint-Venant's equations), calibration and sensitivity analysis based both on the variational approach and on filtering approaches, have been widely studied in the literature. On the contrary, for 2d SW models very few studies only address sensitivity analyses and parameter identifications (friction parameter, boundary conditions). One of the reason is the heavy task to derive the adjoint model, next to make it run in reasonable CPU time and memory storage. Let us cite [14] which treats of a 2d SW river model coupled with a simplified sediment transport model; [15] analysing some capabilities of the approach given some spatially distributed observations; [16] treating of the assimilation of a flood plain image (post-treated SAR data) into the 2d flow model; [17], [18] treating simultaneously the assimilation and the superimposition of 2d local - 1d global flow models (in academic configurations only); [19], [20] treating of the assimilation of surface drifting particles (lagrangian data) into the 2d flow model. All these studies are based on a first order cost function minimisation algorithm (quasi-newton generally); the cost gradient is computed by performing the adjoint model.

None of these studies are based on higher-order numerical schemes; furthermore if considering wet-dry front dynamics, the computations at the front were classically regularised (by introducing a cut-off hence distorting the front velocity) both in the direct model and in the adjoint model. Also, the past studies considering flood plain flows did not address the sensitivity with respect to the topography. The robustness, stability and high accuracy of the whole model (direct model including wet-dry front dynamics plus the adjoint equations) require the finest numerical analysis knowledges, and the result may be crucial to obtain a reliable descriptive model, hence potentially predictive.

Sensitivity analyses are rich information helping to set up any complex flow model. In the present study, the sensitivity maps, based on the cost gradient values, are performed with respect to the friction parameter and the topography (spatially distributed coefficients). These maps can lead to a better understanding of the complex interactions within the flood plain flow. The VDA process (based on optimal control and minimisation) is performed to identify the inflow discharge or friction coefficients defined by land covers. Thus the inverse variational computational tools derived in the present study, implemented into DassFlow (Data Assimilation for Free Surface Flows) software, [21, 22, 23], may lead to a new generation of 2D shallow flow and flood plain dynamics models.

From a technical point of view, the adjoint code is generated "automatically" by using an automatic differentiation tool (Tapenade, [24]). To do so, the direct source code must be prepared to be differentiated source-to-source;

it is the case in DassFlow software. Next the forward MPI commands are "reversed" by hand (or using an extra shell script). Also, some extra crucial tricks make more affordable the adjoint code in terms of memory storage. Based on the know-hows exposed here, the present computational inverse method becomes affordable both in terms of CPU time and memory (on parallel architectures), including for quite large flood plains.

The capabilities of the algorithms derived here (sensitivity maps, identification parameters, model calibration, data assimilation) are illustrated on a real test case: the Lèze river, southwestern of France. 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. 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 identification parameter 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).

Finally, let us point out that the algorithms elaborated here, and the corresponding software, can be applied to any other flow modelled by the 2d SW equations (e.g. tidal 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 flood plain dynamics. In Section 3, the inverse method (the adjoint-based method, the resulting variational sensitivities and the data assimilation process) is presented. The flood plain test case (Lèze river), based on a fine topography and a historical-like inflow discharge is considered in Section 5, both for direct numerical comparisons and inverse analysis (sensitivity analyses and parameter identifications). Some perspectives are proposed in the conclusion (Section 6). A first appendix present a second benchmark of the forward solver, demonstrating the *actual* second order of the elaborated FV scheme. A second appendix presents the existing link between the cost function differential (the mathematical gradient) and an automatically derived adjoint code. Next a speed-up curve demonstrate the good efficiency of the whole MPI computational code (demonstrating in particular the efficiency of the reverse transforms of the MPI commands introduced by hand).

2. The Direct Model

In this section, the direct (forward) model is briefly described. The equations are the classical 2D shallow-water equations with the Manning-Strickler friction law. Few numerical schemes are developed, all based on finite volume methods. These technical combinations of different numerical methods result in first order and second order schemes. All of the schemes are positive and respect water at rest (well-balanced property). More importantly they remain stable in the presence of a dynamic wet/dry front without any unphysical regularisation (a numerical cut-off at the wet-dry front) like it is classically done in the literature, see e.g. [1] and references therein.
Theses schemes are obtained from original combinations of existing techniques. Their accuracies and capabilities are assessed into details. The basic principles of the schemes are presented below, while all details can be found in [25].
Basically, the required ingredients to reach an *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; b) a MUSCL reconstruction; c) an implicit-explicit Runge-Kutta (IMEX) time scheme.

To our best knowledge, the present scheme accuracy and stability is unequaled in the literature, see e.g. [1, 26, 6] and references therein.
The present finite volume solver capabilities are demonstrated by performing classical and original tests cases (one below and one in Appendix Appendix A); many others are presented in [25].

2.1. The Mathematical model

The 2d Shallow Water equations including the Manning-Strickler friction term read in their conservative form as follows:

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

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

The standard 2d SW model described above is numerically solved using finite volume schemes, first and second order. The basic scheme is the standard HLLC approximate Riemann solver, see e.g. [1], but with the intermediate wave speed introduced in [9]. (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 [11, 12] (so-called wb-A in the figures) and those presented in [27, 28] (so-called wb-E in the figures). The second order accuracy is obtained by combining a MUSCL reconstruction and an implicit-explicit Runge-Kutta (IMEX) time scheme, [29].

This combination of these quite technical but standard methods, leads to an *actual* second-order accuracy, well-balanced, positive and stable in presence of dynamic wet/dry fronts , without any unphysical regularisation i.e. a cut-off at the front like it is classically done in the literature, see e.g. [1] and references therein.
To our best knowledge, the present combination is original; it leads to a numerical solver accuracy, stability, and finally robustness as never published in the literature. The numerical schemes have been assessed by performing classical test cases from the literature plus extra new ones, representing geophysical shallow flow features and difficulties. In the present article, we present two selected test cases only: a first one presenting a dynamic wet-dry front and a second one (presented in Appendix Appendix A) considering a

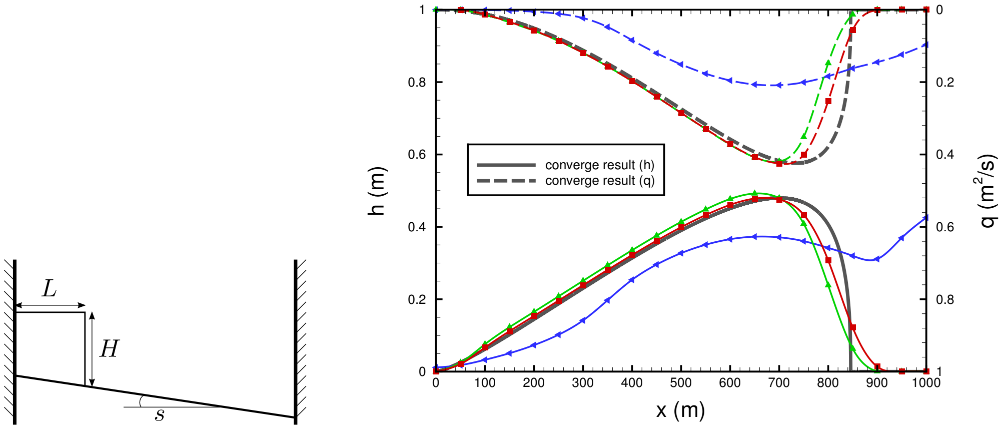


Figure 1: Dam break test case (computed in 2D): (L) Sketch (D) Reference and computed solution (h, q).

"regular" dam-break layout. The goal of this second test case is to demonstrate the *actual* second order feature of the scheme (this is not classical, to our best knowledge). We refer to [25] extra test cases (classical and original ones).
Let us point out that 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 below that the second-order schemes remain more accurate than the first order ones (even if their convergence rate decreases to almost one).

Recall that the aim of the present section is to demonstrate the accuracy and the robustness of the derived numerical schemes in flood plain like configurations. The test case presented below is a dam break on a constant bottom slope, involving a dynamic wet/dry front, see Fig. 1. The Manning-Strickler friction law is taken into account.

The domain length is $l_x = 1000\ m$ 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 $T = 500\ s$. Since no analytical solution is known, the "reference solution" is the numerical solution computed on an extremely fine grid -12 800 cells- Fig. 1; it is considered as almost exact. The "computed solution" is obtained on a rough mesh: 20 cells in length only.

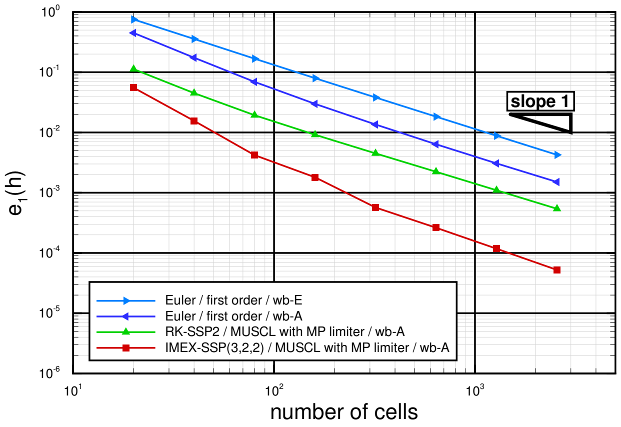


Figure 2: Dam break test case (computed in 2D): convergence curves (relative error norms $e_1(h)$ for the different schemes vs mesh size).

The dynamic wet/dry front is well captured by all finite volume scheme versions (first and second order), without any cut-off water depth h_ϵ (this is an unusual feature). The numerical depths h remain strictly positive.

The computed relative error norm is defined by:

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

where K denotes the mesh cells and m_K their measures.

In the first order scheme case, the convergence rate of $e_1(h)$ equals 1 as expected, Fig. 2.

Two versions of second order scheme are assessed: one based on the RK-SSP2 time scheme (Heun's method) and an other one based on an implicit-explicit Runge-Kutta (IMEX) time scheme. If using the second order RK-SSP2 time scheme (Heune's method), the convergence rate remains equal to 1, but with a better accuracy than the first order scheme one, Fig. 2. If using the IMEX time scheme, the convergence rate is slightly greater than 1 only, but the

accuracy is better than the others.
The loss of the optimal order (order 2) is normal since the solution at wet/dry front is singular. The test case presented in Appendix Appendix A is the regularised version of the present dam break test case; then the solution is regular at the front and the convergence rate demonstrates the actual second order of the IMEX finite volume scheme. (Let us recall that such demonstrated accuracy is unusual in the literature).

The stability condition of the global schemes is the classical CFL-like condition. We refer to [25] for more details and more test cases (classical and original ones).

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 time scheme and to a factor 2 for the MUSCL reconstruction.

Finally, the numerical tests presented above demonstrate that the present numerical schemes are robust and accurate even in presence of wet-front dynamics. A demonstrated numerical scheme accuracy is crucial before studying complex geophysical flows. Indeed it makes the numerical solver "transparent" in the flow analyses, making possible to focus on the modelling errors only (e.g. the data - mathematical model consistency etc).

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) and make decrease model parameter uncertainties. More details on the VDA method can be found for example in [30, 31]. 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. To our best knowledge, the present accurate, "automatically" derived, HPC version, is new in the surface flow research community, see e.g. [32]. Variational sensitivity results are promising to improve our surface water flow analyses. The algorithms described below have been implemented into the computational software DassFlow, [22, 21, 23].

The inverse problem is stated as a 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, [24]). 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 fit the model with 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 the model response j (a scalar valued function). The model response 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, $j(k)$ 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 [30, 31].
A typical cost function in the present data assimilation context reads as follows:

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

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" a-priori on the optimal values sough.
The data assimilation problem reads as an optimal control problem as follows:

$$\underset{\mathbf{k}}{min} \; j(\mathbf{k}) \tag{3}$$

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 [24]. 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; 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 $y_0 = (h_0, \mathbf{q}_0)$.

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

$$dj(\mathbf{k}) \;=\; \tfrac{\partial j}{\partial y_0}(\mathbf{k}) \cdot \delta y_0 + \tfrac{\partial j}{\partial q_{in}}(\mathbf{k}) \cdot \delta q_{in} + \tfrac{\partial j}{\partial n}(\mathbf{k}) \cdot \delta n \tag{4}$$

The adjoint code obtained via automatic differentiation costs approximatively $5-6$ 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 iterates, 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 [25]. 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:

$$j(\mathbf{k} + d\mathbf{k}) \approx j(\mathbf{k}) + \nabla j(\mathbf{k}).d\mathbf{k} \tag{5}$$

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. Nevertheless, the resulting sensitivity analysis tool is 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 to solve the optimisation problem (3). 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.

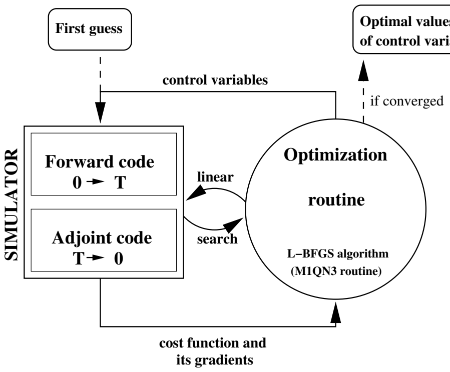


Figure 3: Principle of a 4D-Var type variational data assimilation algorithm.

Here, the classical quasi-Newton L-BFGS algorithm is employed; more precisely, the L-BFGS algorithm implemented in the routine from [33].

Let us point out that each variable of the control vector may be active or not, as an actual control variable. In practice, of copurse it is possible 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 iterate the VDA algorithm computes l a new set of parameters k^l according to the gradient $\frac{\partial j}{\partial k}$, which make 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 (southwestern 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). Furthermore, given a computational grid (the finite volume mesh), given a DTM (derived from a

fine LIDAR), some 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 whole numerical chain implemented into DassFlow software [23], in presence of wet-dry front dynamics over complex topography: accurate, stable direct and adjoint models, sensitivity analysis, parameter identifications and calibration processes.

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, 24632 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 since the presence of a cross road and a bridge (it is the location where the gauge station is), see Fig. 4.

The domain boundaries consist in two open boundaries at upstream and downstream; 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 (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).

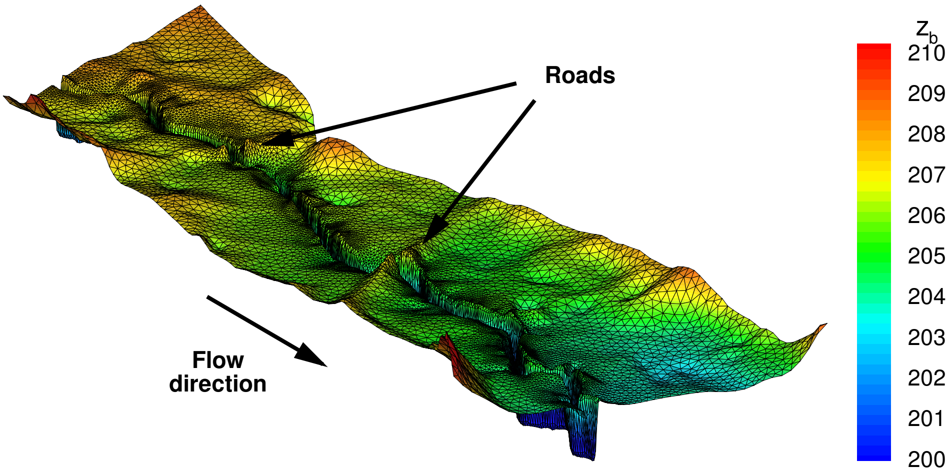


Figure 4: Leze river topography and mesh (24632 cells)

The initial condition is set as follows. The whole domain is considered dry, next it is gradually filled in the stream-bed during a long time (about two hours) to obtain a realistic initial condition; it is the flow state at $t = 0$.

4.2. Direct computations and scheme comparisons

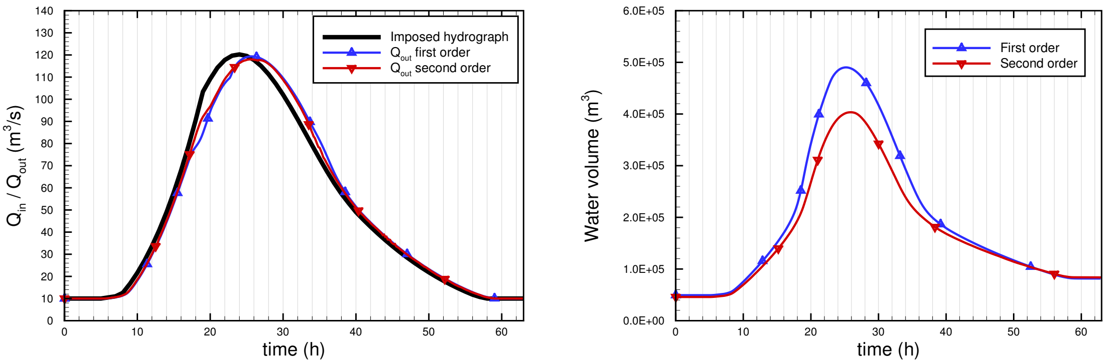
In this section, the direct model only is performed; this shows some flow features. Also, some comparisons between the first order scheme and a second order scheme are presented. In this section, the "1st order scheme" denotes the Euler / A-well-balanced scheme, and the "2nd order scheme" denotes the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

The time simulation 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).

All the results presented here are obtained using a water depth cutoff $h_\epsilon \approx 10^{-15}$. In other words, the numerical schemes remain stable even when computing wet-dry front dynamics over complex topography.

The Froude number equals 0.3 at maximum (it is 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 $t = 19$h, a maximal discrepancy of 8% is observed.



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

(b) Total water volume V in time

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): $V = \int_0^T (Q_{\text{in}} - Q_{\text{out}})\, dt$ obtained from the 1st order and 2nd order schemes.

The net mass balance (total water volume) in the computational domain is plotted 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.

If comparing the spatially distributed water depth h at the flood 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 notable but it may be secondary in a geophysical uncertain context. The second order requires a

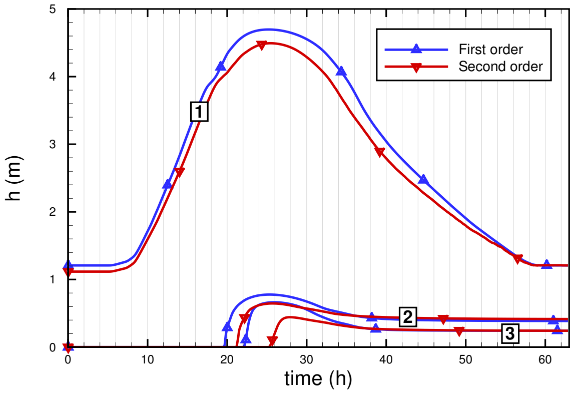
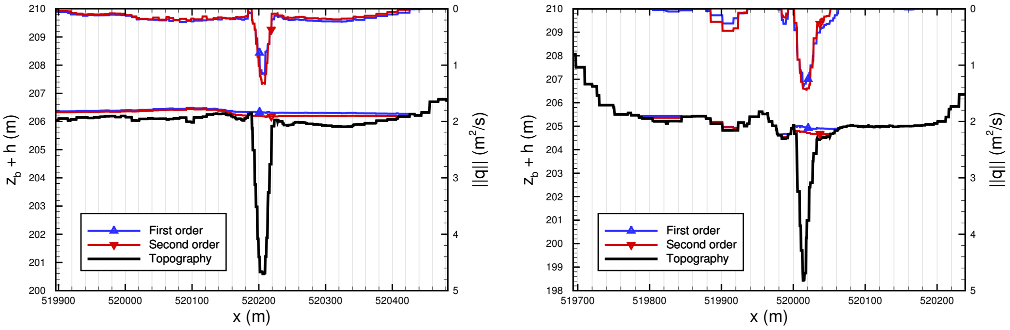


Figure 6: Time evolution of the water depth h at stations 1, 2 and 3 (see their locations in Fig.8). For each stations, the first and second order results are plotted.



(a) Section 1

(b) Section 2

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

CPU time computation about 4 / 5 times higher.

In other respect, let us recall that the front dynamic can be greatly affected by any unphysical regularisation introduced in the numerical solver (areas where h tends to 0) see e.g. [1] and references therein. In a flood plain context, this feature is likely more important than the accuracy discrepancy between the first and second order. Recall that the present numerical solvers (both first and second orders) do not introduce any regularisation at the front, hence do not present this embarrassing numerical modelling bias.

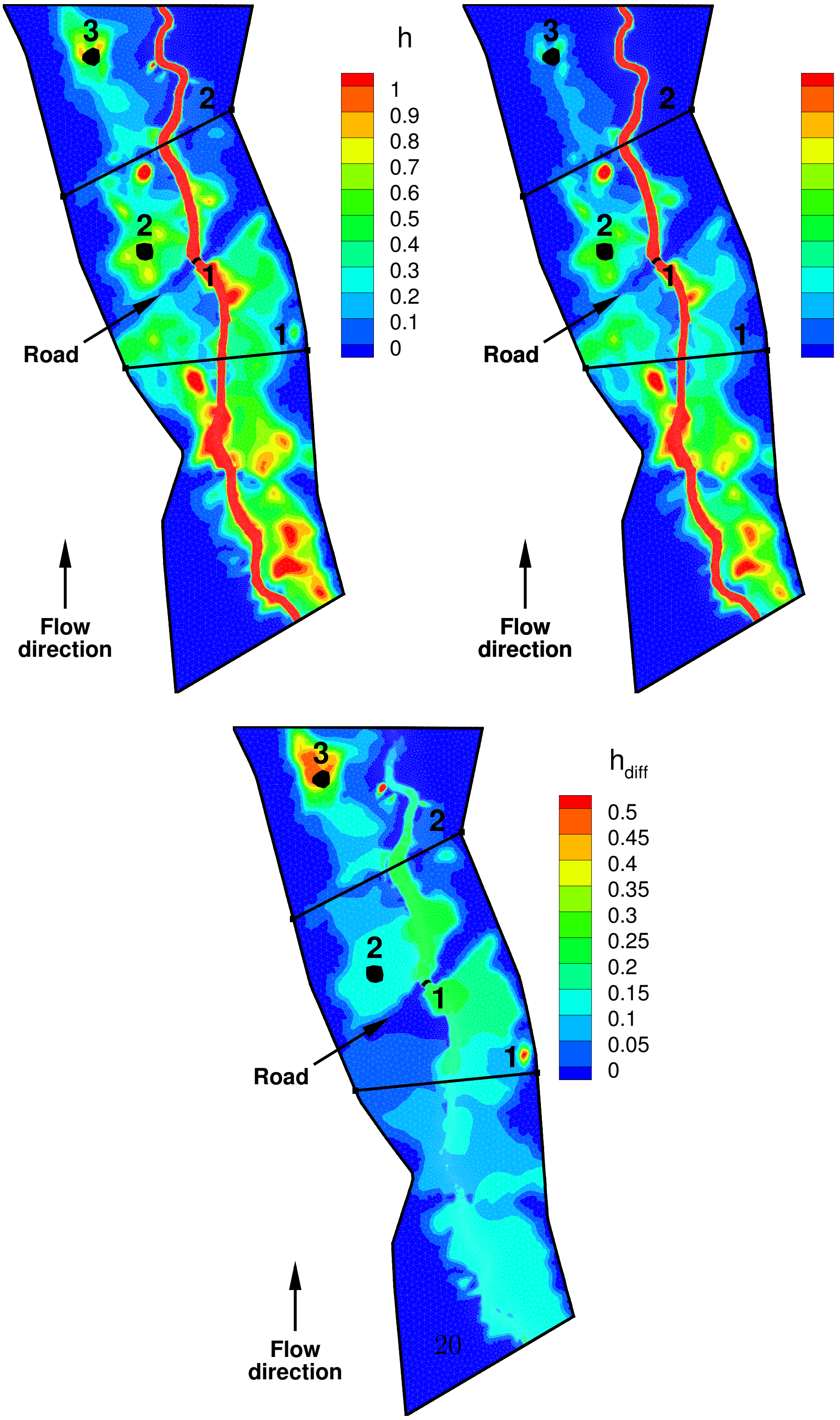


Figure 8: Top views of water depth h at $t = 26\ h$ (flood peak), locations of the 3 observation points/areas, locations of the 2 cross-sections. (Up left) first-order computed depth h. (Up right) second order computed depth h. (Down) Difference between the first and second order depth Δh.

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.

Boundary conditions are identical to 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 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 $+/-10cm$ (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:

$$j(n, z_b) = \sum_{time\ \ step\ i} \sum_{station\ j} (h_{i,j} - h_{i,j}^{obs})^2 \qquad (6)$$

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 (6)). 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 the forward model), see (5).

The gradient value with respect to the Manning friction coefficient is roughly 2 orders of magnitude higher than those with respect to the bathymetry. Then for a sake of clarity, the gradient values plotted in Fig 6 have been normalised.

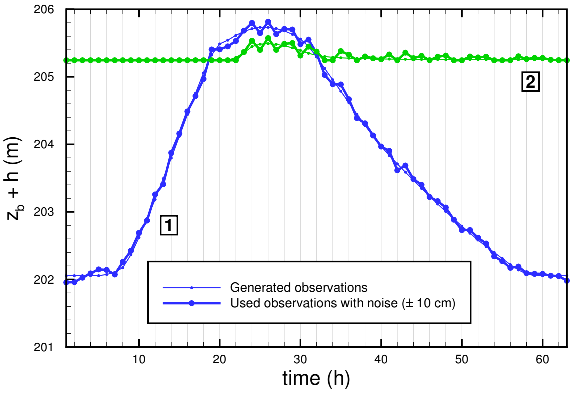
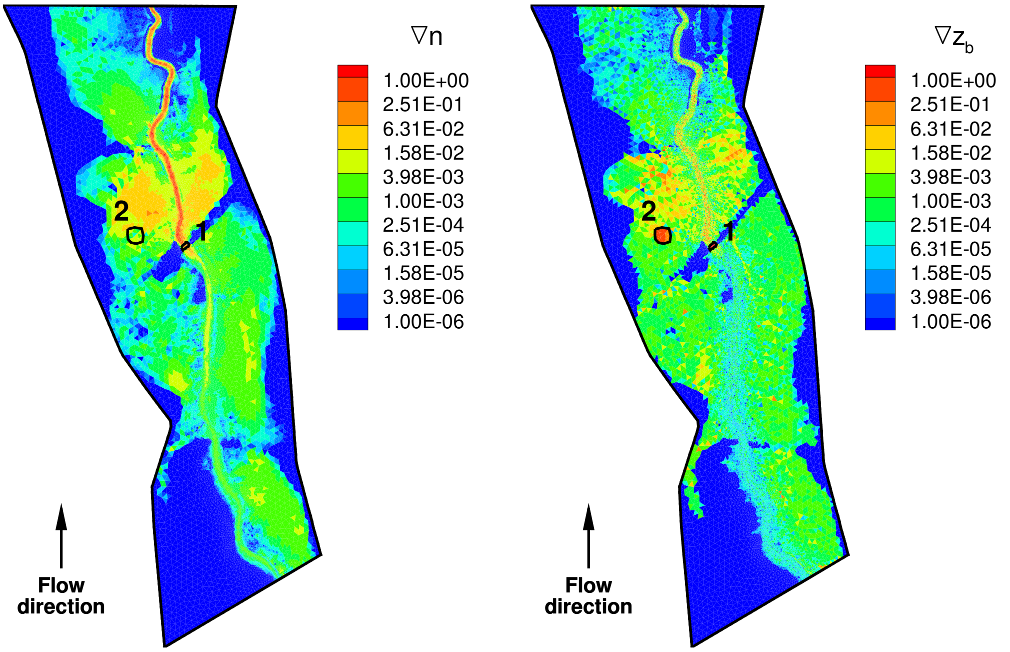


Figure 9: Water surface elevation measurements at Station 1 (blue) and at Station 2 (green). It is synthetic data with realistic random noise added $(+/-10cm)$.



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

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

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.

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 less than 0.3).
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 astandard regime (i.e. without overflowing) is highly desirable before calibrating the same model but applied to a flooding 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 elevation) and the bathymetry elevation.
The bathymetry sensitivity pattern is globally comparable to the friction sensitivity one but more point-wise, less diffused. This remark corroborates the analyses presented in [34] (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 [35] 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.

4.4. Data assimilation and model calibration

In this section, two twin experiments are performed: 1) the identification of the friction coefficient 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. These inverse computational methods should make improve the flood plain modelling.

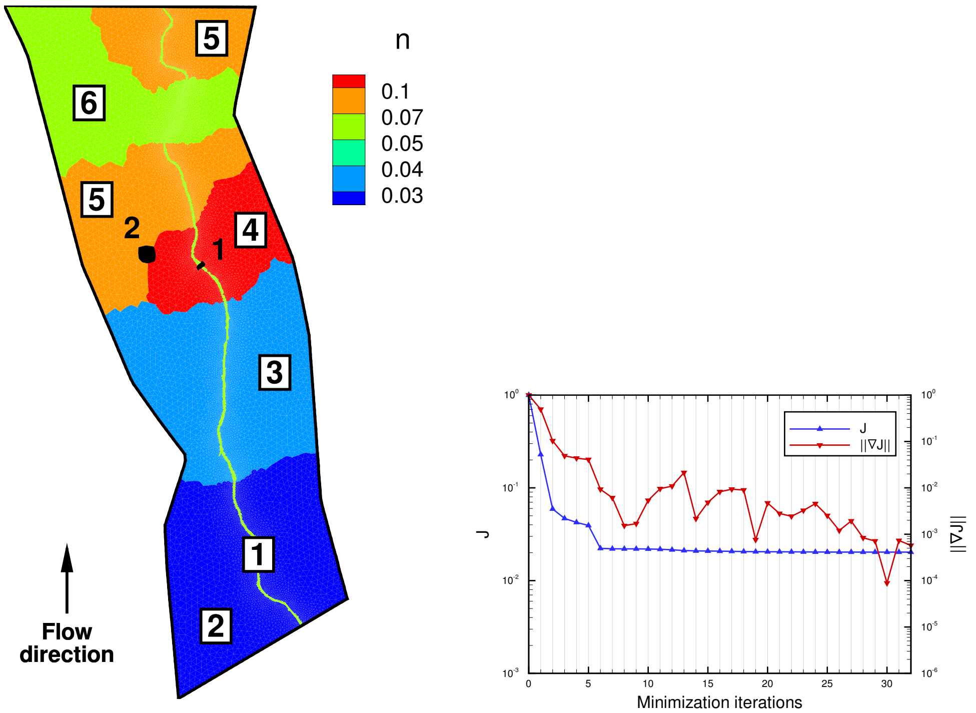
4.4.1. Roughness coefficient identification

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 $+/-10$cm Gaussian noise, the identified friction coefficients are still very good; It is the results presented in Table 1.

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).



(a) Land covers and Manning coefficient values

(b) Cost function and gradient norm vs minimisation iterates (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 iterates. Normalised values.

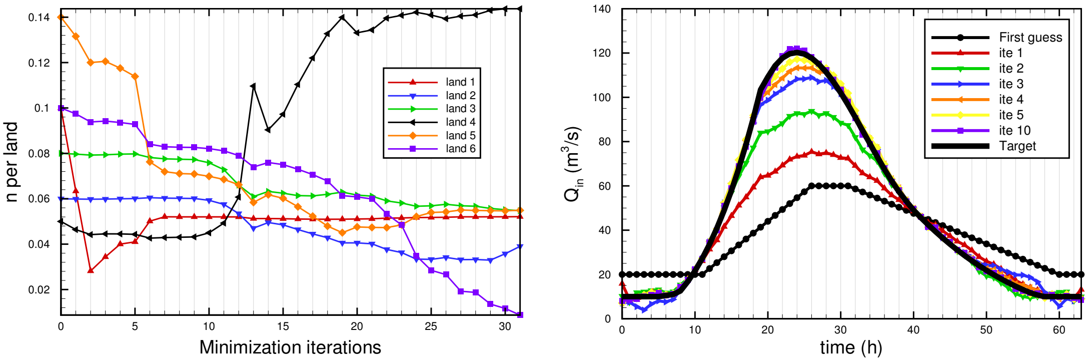


Figure 12: Minimisation process. (L) The 6 Manning coefficient values vs iterates (iterate 0=first guess). (R) Hydrographs (inflow discharge in time) at few iterates.

An accurate convergence is reached into 32 iterations, see Fig 11(b) (the stopping criteria have been set extremely small to make decrease as much as 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 [33].
This optimisation computation enhances the importance of developing accurate and stable numerical schemes, even in the presence of the dynamic wet-dry front (see the discussions led in the previous sections).
This synthetic but real-like data assimilation experiment demonstrates the present inverse method capabilities to enhance flood plain dynamic modelling.

4.4.2. Inflow discharge identification

The present twin experiment aims at identifying the inflow hydrograph. The forward model is strictly the same as the previous one; 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:

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

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 $4dt\frac{(1-0.2)}{0.2}$ (exponential smoothing).

The target hydrograph, the first guess hydrograph and few intermediate hydrographs are plotted in Fig. 12 (R). The first guess hydrograph is defined by dividing by two the difference between the maximum value, plus a phase shift of 4 hours approximatively.

The convergence curves (the decreasing curves of j and $\|\nabla j\|$ vs iterates) are not plotted since they are similar than those in Fig. 11(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 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).

The present identification experiment is very robust and 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, see e.g. [15, 16]. Finally, this numerical experiment demonstrates the inverse method capability to infer inflow hydrographs corresponding to a flood plain event, given time series of water elevation.

5. Conclusion

In this study, a whole computational chain modelling accurately a flood plain dynamics making possible the inference of uncertain parameters (either spatially distributed like the friction coefficient and the bathymetry, or time-dependent like the inflow discharge) has been developed and assessed. As a first step, a particular attention has been paid to the robustness and the accuracy of the forward solver (finite volume method) in presence of a dynamic wet/dry front. The difficulty being to do not introduce any numerical regularisation at the front, regularisation leading to a wrong front velocity. An innovative finite volume 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 scheme has been demonstrated to be *actually* second order, and naturally stable in presence of dynamic wet/dry fronts (no numerical cut-off at the wet-dry front). The numerical schemes have been assessed into details, in particular on benchmarks representing well some flood plain dynamic difficulties.

Next, an inverse variational computational method, adjoint based, has been exposed, including crucial know-hows and tricks in a MPI automatic differentiation context. This inverse method, classical in other geophysical flow modelling contexts (e.g. atmosphere, oceans), provides precious sensitivity information in view to better understand the flow and the model. Given dense in-situ measurements (e.g. water elevation time series), sensitivity maps with respect to the friction parameter (spatially distributed) and the bathymetry have been performed and compared in the Leze river case (past flood event of june 2000). The full Variational Data Assimilation process make possible the identification of uncertain "parameters", time-series like the inflow discharge or spatially distributed like the friction coefficient. These numerical experiments demonstrate the robustness and the great capabilities of the present inverse computational algorithms, making possible to reduce uncertainties hence improving flood plain dynamic models.
As 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 [36].

Finally in a context of fast growing remote-sensed data volume acquired (e.g. the forthcoming ESA Sentinel missions and NASA-CNES SWOT mission), as mentioned in [32], designing reliable inverse and data assimilation tools should make a breakthrough in the flood plain modelling.

Appendix A. Finite Volume Solver Assessments

The present section aims at assessing the actual accuracy and convergence rate of the numerical solvers. To do so a regular solution is compulsory, hence a flow without wet/dry front (like it has been considered in Section 22). Then the test case is a "regularised" version of the dam-break benchmark. The (non-linear) Manning-Strickler friction term is still taken into account.
If using the IMEX time scheme, the convergence rate obtained equals 2 while for if using the Runge Kutta 2 (RK-SSP2) time-scheme, it equals 1 only, see Tab. A.13(L). In other words, if not considering the right time-space numerical scheme combination, an a-priori order 2 scheme can be actually order 1...

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

$$\left\{ \begin{array}{rcrcl} z_b(x) & = & & & 0.5 \;\; e^{-\left(x - l_x/2\right)^2 / 2\sigma^2} \\ h(x,t=0) & = & 0.1 & + & 0.5 \;\; e^{-\left(x - l_x/2\right)^2 / 2\sigma^2} \end{array} \right. \tag{A.1}$$

with $\sigma = 100\ m$, see Fig. A.13. The computational domain length equals $l_x = 1000\ m$; the Manning-Strickler coefficient $n = 0.05$; the total simulation time is $T = 100$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.

The relative error norms $e_1(h)$ and $e_1(q)$ vs the cell numbers are plotted on Fig.A.13(L). The RK-SSP2 time stepping method leads to a globally first order scheme only. Nevertheless, its accuracy is higher than the first order scheme accuracy. The IMEX time stepping leads to an actual second order convergence rate.

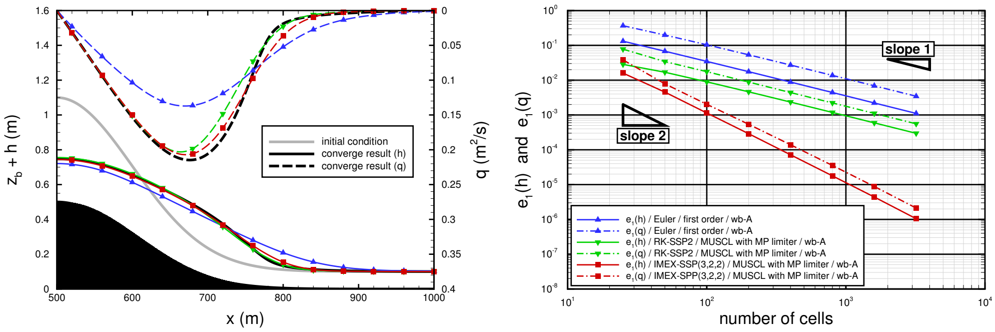


Figure A.13: Regularised dam break test case. (L) (half-domain): initial condition, reference solution (h, q) in black, computed solutions (on 25 cells) for the 3 numerical schemes. (R) Relative error norms $e_1(h)$ and $e_1(q)$ *vs* the number of cells.

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 [9], the well-balanced numerical solver defined in [12], a standard MUSCL reconstruction and the IMEX-SSP(3,2,2) 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; 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 :

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

We show 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 derivation below follows those presented in [37].

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: $\big(\frac{\partial \mathcal{M}}{\partial \mathbf{k}}(\mathbf{k})\big)^* : \mathcal{Y}' \longrightarrow \mathcal{K}'$. 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:

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

or, using the relations presented above:

$$\big\langle d\mathbf{k}^*,\, d\mathbf{k} \big\rangle_{\mathcal{K}' \times \mathcal{K}} = \big\langle dY^*,\, dY \big\rangle_{\mathcal{Y}' \times \mathcal{Y}} \, . \tag{B.3}$$

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:

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

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:

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

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 input.

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 $\frac{j(k+\varepsilon\delta k)-j(k)}{\varepsilon}$, $\varepsilon > 0$, 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 with δk a random vector. 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 [23, 25]; it follows some technics described in [38].
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 [25] to obtain the MPI

version of the adjoint instructions. All these methods and known-hows are implemented in the open source software DassFlow [23].

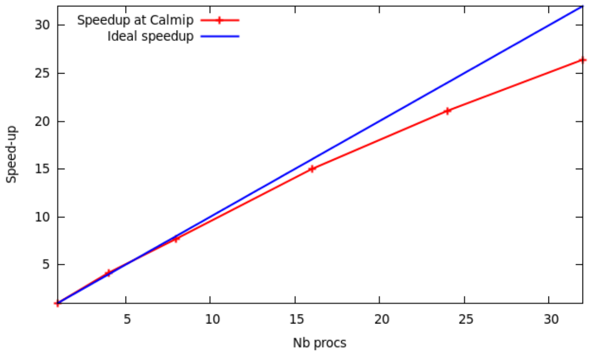


Figure C.14: Speed-up obtained on a sensitivity computation (DassFlow-Shallow software, Lèze river test case, 25 000 cells, Calmip cluster Toulouse).

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 three reviewers for their constructive help and suggestions to improve the manuscript.
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] E. Toro, Shock-capturing methods for free-surface shallow flows, John Wiley, 2001.

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

[3] 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.

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

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

[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.

[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] 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.

[12] 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.

[13] 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.

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

[15] 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.

[16] 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.

[17] 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.

[18] 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.

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

[20] 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.

[21] 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).

[22] D. computational software, Data assimilation for free surface flows, open source.
URL `http://www.math.univ-toulouse.fr/DassFlow/`

[23] 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.

[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] 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`

[26] R. J. LeVeque, Finite volume methods for hyperbolic problems, Vol. 31, Cambridge university press, 2002.

[27] 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.

[28] 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.

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

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

[31] J. Monnier, Variationnal 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

[32] P. D. Bates, Integrating remote sensing data with flood inundation models: how far have we got?, Hydrological Processes 26 (16) (2012) 2515–2521.

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

[34] 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.

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

[36] 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.

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

[38] 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.