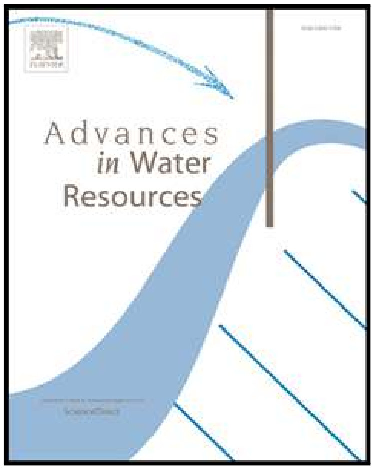
Journal Pre-proof



2D numerical simulation of unsteady flows for large scale floods prediction in real time

I. Echeverribar, M. Morales-Hernández, P. Brufau, P. García-Navarro

PII: S0309-1708(19)30478-6
DOI: https://doi.org/10.1016/j.advwatres.2019.103444
Reference: ADWR 103444

To appear in: *Advances in Water Resources*

Received date: 3 June 2019
Revised date: 7 October 2019
Accepted date: 9 October 2019

Please cite this article as: I. Echeverribar, M. Morales-Hernández, P. Brufau, P. García-Navarro, 2D numerical simulation of unsteady flows for large scale floods prediction in real time, *Advances in Water Resources* (2019), doi: https://doi.org/10.1016/j.advwatres.2019.103444

Highlights

- A sophisticated fully 2D model accelerated with GPU is presented.
- Details of the numerical scheme and the acceleration technique are given.
- The necessity of these numerical fixes in real cases is demosntrated.
- The model is applied to a large stretch of the Ebro River.
- The results are compared with field measurements.

2D numerical simulation of unsteady flows for large scale floods prediction in real time

I. Echeverribar[a,*], M. Morales-Hernández[b], P. Brufau[a], P. García-Navarro[a]

[a] *Fluid Mechanics, University of Zaragoza, Zaragoza (Spain)*
[b] *Computational Science and Engineering Division, Oak Ridge National Laboratory, Tennessee (USA)*

Abstract

The challenge of finding a compromise between computational time and level of accuracy and robustness has traditionally expanded the use simplified models rather than full two-dimensional (2D) models for flood simulation. This work presents a GPU accelerated 2D shallow water model for the simulation of flood events in real time. In particular, an explicit first-order finite volume scheme is detailed to control the numerical instabilities that are likely to appear when used in complex topography. The model is first validated with the benchmark test case of the Toce River (Italy) and numerical fixes are demonstrated to be necessary. The model is next applied to reproduce real events in a reach of the Ebro River (Spain) in order to compare simulation results with field data. The second case deals with a large domain (744 km^2) and long flood duration (up to 20 days) allowing an analysis of the performance and speed-up achieved by different GPU devices. The high values of fit between observed and simulated results as well as the computational times achieved are encouraging to propose the use of the model as forecasting system.

Keywords: computational mesh, shallow water flow, GPU, real time flood prediction

*Corresponding author
Email address: `echeverribar@unizar.es` (I. Echeverribar)

1. Introduction

Flooding events are considered as extreme phenomena, not only due to their severity but also to their high frequency, as a UN survey reveals [1]. River overflows generate floods and the destruction or modification of natural and artificial elements in a basin. The sheet of water flooding urban, rural and industrial areas provokes agricultural, landscape, flora and fauna, as well as economic and social activity alterations. Additionally, the flow volume and the water speed transform the environment, modifying the river banks over long periods of time. They may have destructive effects not only on infrastructures such as bridges, roads and buildings but they can also take human lives [2]. The problem of flooding affects many countries in the world. In Spain, the occurrence of these phenomena has raised the awareness of both public authorities and population [3]. In particular, the flood that took place in the Ebro River in 2015 resulted in a payout of more than 105 million euros for the repair of the consequences. Therefore, the use of resources and better technology for the analysis of floods is justified. While it is impossible to eradicate the occurrence of these events, the development of prediction tools capable of anticipating their damage, allowing improvement of emergency plans, has become one of the main aims of research.

Presently, flood risk evaluations performed by water authorities and decision makers in European countries are based on hydrological models or 1D hydraulic models [4, 5]. Their predictive capacity is limited to the evaluation of water discharges for situations below bankful values. When the flood wave exceeds that value and inundation takes place over the floodplain, the use of 2D hydraulic tools is needed for a correct modelization of the inundation. However, most available 2D hydraulic models are so computationally time consuming that their application to event predictions in real time scenarios is hindered.

Prediction tools can be based on many different procedures and algorithms to forecast the behaviour of a certain flow. They are commonly based on numerical models that are able to provide solutions for the mathematical equations that govern these flow phenomena. A good review of methods is provided in [6]. Several works reveal the suitability of the 2D shallow water model for the proper reproduction of flood events [7, 8]. Some of them, [9, 10] deal with finite volume numerical methods. In this context, some methods have been improved over the years to fix numerical instabilities related with 2D schemes, such as the wet-dry treatment [11, 12, 13, 14]. Nowadays, these improvements allow the creation of robust models that are able to reproduce complex and challenging cases. However, 2D models present their main drawback of requiring large and sometimes unaffordable computational times when applied to realistic scenarios. This situation leads to 2D models not being as widely used as might be expected. It is worth mentioning that the high computational cost represents a drawback not only for the predicting simulations, but also for the calibration process, which involves a great amount of simulations in order to obtain the proper value of parameters, such as roughness coefficient.

Different ways to decrease simulation times in this context can be found in literature. The easiest way to reduce this effort is to go back to 1D models, that are still widely used for flood events [15], since they can reproduce the evolution of certain discharge peaks under specific conditions. However, 1D models usually result in a bad representation of the discharge evolution over time [16] when over-bank events take place. Although this issue can be sometimes successfully overcome [17, 18], the floodplain plays an important role in the peak delay and it is not properly represented through a 1D schematization. As a result, research in coupled models has arisen as an alternative to keep representing the main channel by means of a 1D model, thus avoiding the 2D discretization usually related with the most time consuming part, while a 2D framework is used for the floodplains [19, 20, 21]. The main disadvantage of the 1D-2D models lies in the data preprocessing that can be tedious and complex. The linking task is not trivial as both models must be thoroughly matched to avoid interpolation and to ensure mass conservation. Additionally, their suitability must be assessed on a case-by-case basis, as the time reduction might be negligible in large flood cases in which the wetted flood-prone area plays a more important role than the river bed itself. This could occur in a river basin in which the floodplain requires even more refinement of the mesh than river bed due to the existence of narrow levees, for instance.

Parallel to all those developments, acceleration technologies have been improved and nowadays offer a high number of alternatives for code speed up. This is the High Performance Computing (HPC) research. Within this area, several techniques can be adopted, from massive parallelizing within a CPU network [22, 23] to Graphical Processing Units (GPU) [24] used as a computing device. In any case, all of them are based on the workload division into different threads that can simultaneously compute a part of the numerical solver [25, 26] or any other time consuming function of the application, such as the visualization [27]. Most of them report scalability problems, as the data transfer might dominate the time consumption leading to very inefficient parallelized algorithms [28, 29], thus code optimization becomes crucial. MPI techniques in particular are suitable for great domains due to the large amount of data transfers as they are based on domain partitioning. Recently, some contributions for realistic cases have been reported involving GPU [30] or multi-GPU [31] and comparing with OpenMP [32, 33] to accelerate the computations. The sensitivity of the numerical results to the single or double precision was reported by [31] on cartesian grids. This type of grids were also used in [30] leading to important speed-ups. On the other hand, the simulation of fast dam-break flows on unstructured grids was reported in [33] showing less noticeable computational time acceleration due to the shorter event duration.

With the objective focused on modelling flood events in realistic domains as efficiently that they can be used as real time forecasting tools, this work presents a fully 2D numerical model proposed to reproduce these events with a sufficient accuracy in an affordable computational time by means of GPU devices. The model solves the 2D shallow water equations with a numerical scheme, that was thoroughly enhanced in the recent past to avoid numerical instabilities [34, 35]. These improvements led to a complexity intended to avoid reducing the time step to ensure stable and conservative solutions. The goal of the present work is to focus on the ability of this robust method to model, not only specific and challenging physical phenomena, such as academic test cases with analytical solution [34, 35], but also to reproduce real and large flood events encompassing tens or hundreds of km^2. The model is explained not only in terms of the numerical method, with emphasis on the most relevant details, but also in terms of the acceleration technology. Next, the model is applied to two test cases. The first test case is used as a validation with experimental data measured in a physical model of the Toce River (Italy) to focus on the importance of the numerical fixes included. Different simulations will be performed to show the consequences of not using the numerical fixes. The second case demonstrates the accuracy and suitability of the model for the simulation of large scale floods. A reach of the Ebro River (Spain) is considered and several historical events are simulated in order to compare with field measurements. The set of available observations provides not only maximum extension of flooded area, allowing a comparison of flood shape, but also the time evolution of water surface elevation and discharge at gauging points. These have been used to compare simulated and measured data enabling the calculation of Nash-Sutcliffe errors [36, 37]. Additionally, a study of the speed-up achieved on different GPU cards is carried out not only to confirm the affordable computational times offered, but also to show the potential of the acceleration technologies for the hydraulic research and prediction tool development.

2. Governing equations and numerical scheme

The governing equations and the numerical scheme, proposed in [34, 35], can be found in full detail in the references. It is not the purpose of the present work to repeat them, however, the most important details are explained in the following subsections for the sake of clarity.

2.1. 2D shallow water equations

The mathematical model adopted to represent the surface flow is the hyperbolic shallow water system of equations in its 2D version:

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}(\mathbf{U})}{\partial x} + \frac{\partial \mathbf{G}(\mathbf{U})}{\partial y} = \mathbf{S}(\mathbf{U}, x, y) \tag{1}$$

where the conserved variables:

$$\mathbf{U} = (h, hu, hv)^T \tag{2}$$

are h, water depth (m), and $q_x = hu$ and $q_y = hv$, unit discharges (m^2/s) in x and y direction, respectively. As the model is depth averaged, u and v are the vertical averaged components of the velocity. The fluxes of these conserved variables are:

$$\mathbf{F} = \left(hu, hu^2 + g\frac{h^2}{2}, huv\right)^T; \qquad \mathbf{G} = \left(hv, huv, hv^2 + g\frac{h^2}{2}\right)^T \tag{3}$$

And the source terms are related to bed slope and friction stress as:

$$\mathbf{S} = \left(0, gh\left(S_{0x} - S_{fx}\right), gh\left(S_{0y} - S_{fy}\right)\right)^T \tag{4}$$

where the bed slopes represent the variation of terrain elevation, z_b, in x and y directions:

$$S_{0x} = -\frac{\partial z_b}{\partial x} \qquad S_{0y} = -\frac{\partial z_b}{\partial y} \tag{5}$$

the friction stress is formulated as

$$S_{fx} = \frac{n^2 u\sqrt{u^2+v^2}}{h^{4/3}} \qquad S_{fy} = \frac{n^2 v\sqrt{u^2+v^2}}{h^{4/3}} \tag{6}$$

where n is the Manning roughness coefficient [38, 39, 40, 41].

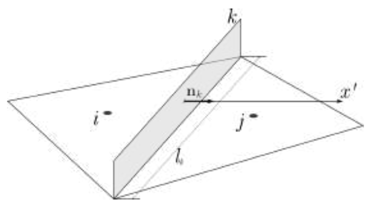


Figure 1: Sketch of a pair of two dimensional cells, i and j, sharing a cell edge, k, of length l_k.

2.2. Finite volume method

The system of equations (1) is time dependent, non-linear, hyperbolic and has source terms. As it has no analytical solution, a Godunov type finite volume scheme is used to discretize the domain into cells, Ω_i, acting like a control volume, leading to a piecewise information with cell-averaged constant values of the variables at time, n, as:

$$\mathbf{U}_i^n = \frac{1}{A_i}\int_{\Omega_i} \mathbf{U}(x, y, t^n) d\Omega; \tag{7}$$

where A_i stands for the cell area. From now on, the (x, y, t) dependence of the variables will be omitted in the notation for the sake of clarity. Thus, the system (1) is integrated at each cell and the Gauss theorem is applied becoming

$$\frac{d}{dt}\int_{\Omega_i} \mathbf{U} d\Omega + \oint_{\partial\Omega_i} \mathbf{E}\mathbf{n} dl = \int_{\Omega_i} \mathbf{S} d\Omega \tag{8}$$

where $\mathbf{n} = (n_x, n_y)$ is the outward normal vector to the volume, Ω_i and $\partial\Omega_i$ is the contour of the volume. The fluxes are included in $\mathbf{E}$, so that $\mathbf{E}\mathbf{n} = (\mathbf{F}n_x + \mathbf{G}n_y)$. Updating the time step size during the simulation, a generic time is defined as t^n and increases adding the time step, Δt, as: $t^{n+1} = t^n + \Delta t$. The used mesh remains constant in time.

Therefore, the fluxes are evaluated at cell edges as $\delta\mathbf{E} = \mathbf{E}_j - \mathbf{E}_i$, where $\mathbf{E}_j$ is the value of fluxes $\mathbf{E}$ at the neighbouring cell j that shares a cell edge, k, of length l_k, with the cell i (as sketched in Figure 1). Source terms are also evaluated in the shared wall, k. And finally, combined with a piecewise constant representation of the variables, $\mathbf{U}_i$, and also assuming a uniform value of $\mathbf{E}_i$ at each cell and the summation of $(\mathbf{n}_k l_k)$ equal to zero, equation (8) can be expressed as

$$A_i \frac{\mathbf{U}_i^{n+1} - \mathbf{U}_i^n}{\Delta t} + \sum_{k=1}^{3} (\delta \mathbf{E})_k \mathbf{n}_k l_k = \sum_{k=1}^{3} \mathbf{S}_k \tag{9}$$

It is worth mentioning that in the 2D framework the solution is obtained by means of a locally linearized 1D Riemann Problem (RP) at each cell edge, k, projected onto the direction $\mathbf{n}$ over an x' axis (as seen in Figure 1), following the Roe approach. The linearized solution must fulfill the Consistency Condition [42].

In system (1) a Jacobian matrix can be defined normal to the direction of the flux, $\mathbf{E}$, given by the unit vector, $\mathbf{n}$, as

$$\mathbf{J}_n = \frac{\partial \mathbf{E}\mathbf{n}}{\partial \mathbf{U}} = \frac{\partial \mathbf{F}}{\partial \mathbf{U}} n_x + \frac{\partial \mathbf{G}}{\partial \mathbf{U}} n_y; \tag{10}$$

This Jacobian matrix can also be locally defined at each wall, k, following also the Roe's linearization, $\tilde{\mathbf{J}}_{n,k}$. Due to the structure of the system, 3 eigenvectors and eigenvalues, $\tilde{\mathbf{e}}^m$ and $\tilde{\lambda}^m$ (with m varying from 1 to 3) can be obtained following:

$$\tilde{\mathbf{J}}_{n,k} = \tilde{\mathbf{P}}_k \tilde{\mathbf{\Lambda}}_k \tilde{\mathbf{P}}_k^{-1} \tag{11}$$

where $\tilde{\mathbf{\Lambda}}_k$ is the diagonal matrix whose elements are the eigenvalues and $\tilde{\mathbf{P}}_k$ is the matrix containing eigenvectors, providing 3 eigenvalues for the 2D model [34, 35].

If the differences in vector $\mathbf{U}$ are expressed as:

$$\delta \mathbf{U}_k = \mathbf{U}_j - \mathbf{U}_i = \sum_{m}^{3} (\tilde{\alpha} \tilde{\mathbf{e}})_k^m \tag{12}$$

the fluxes part of equation (9) can be also expressed as:

$$(\delta \mathbf{En})_k = \tilde{\mathbf{J}}_{n,k} \delta \mathbf{U}_k = \tilde{\mathbf{J}}_{n,k} \sum_{m}^{3} (\tilde{\alpha}\tilde{\mathbf{e}})_k^m \tag{13}$$

Additionally, the source terms are also projected onto the eigenvector basis and, in particular, represented by means of an extra stationary wave leading to a solver defined by $m+1$ states for a problem with m equations. This is the so called Augmented Roe approach (ARoe approach) [35]:

$$\mathbf{S}_k = \sum_{m} (\tilde{\beta}\tilde{\mathbf{e}})_k^m \tag{14}$$

Finally, the updating expression of a single cell i by means of the 3 ingoing contributions (m) at each of the 3 edges (k) shared with the adjacent cells is:

$$\mathbf{U}_i^{n+1} = \mathbf{U}_i^n - \frac{\Delta t}{A_i} \sum_{k=1}^{3} \sum_{m=1}^{3} \left[\left(\tilde{\lambda}^- \left(\tilde{\alpha} - \frac{\tilde{\beta}}{\tilde{\lambda}} \right) \tilde{\mathbf{e}} \right)_k^m l_k \right]^n \tag{15}$$

where superscripts n and n+1 stand for the current and next time step respectively. And the information is propagated according to the upwind philosophy: $\lambda^{\pm} = \dfrac{\lambda \pm |\lambda|}{2}$.

The time step, Δt, is dynamically updated and limited by the CFL condition for stability reasons following:

$$\Delta t = \text{CFL} \min_{k,m} \frac{\delta x_k}{\tilde{\lambda}_k^m} \tag{16}$$

where

$$\delta x_k = \min(\chi_i, \chi_j) \qquad \chi_i = \frac{A_i}{\max\limits_{k=1,NE} l_k}. \tag{17}$$

Coefficient CFL (*Courant-Friedrich-Lewy* number) must be between 0 and 1 to guarantee stability [43] due to the explicitness of the numerical scheme.

The numerical scheme may fail in some situations due to the averaging in Godunov's method. Even without source terms, a common problem is the entropy violation in sonic rarefactions [42, 44]. That must be corrected by means of entropy fixes [45]. Additionally, when taking into account source terms, other problems could arise, such as the appearance of negative values of water depth, as well as non-physical numerical oscillations. These errors have been traditionally avoided by reducing the time step size below that constrained by the CFL condition [9], using new restrictions that consider the influence of source terms. Therefore, cases with high friction values can lead to unaffordably high computational times.

An alternative, as presented in [34, 35], includes new corrections to ensure the robustness of the method and the lack of non physically-based solutions without reducing the time step size. These corrections are mainly based on the detailed analysis of the Riemann Problem formulated as a superposition of waves travelling at the speeds given by the eigenvalues. This concept requires the definition of intermediate states that are next explained for a 1D scheme for the sake of clarity.

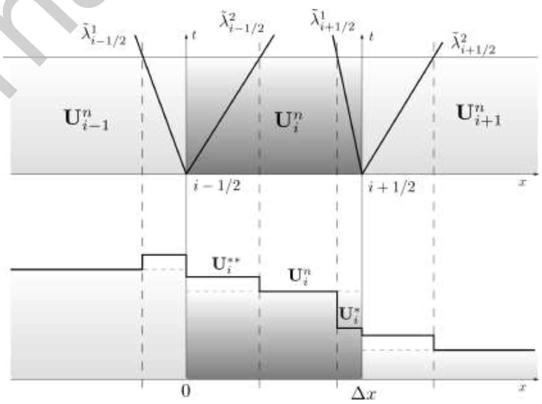


Figure 2: Control volume in Godunov's method and intermediate states for the 1D scheme.

As the resulting solutions obtained with the numerical method for a time step are cell-averaged, they provide a piecewise constant solution at the new time level, t^{n+1}, for each variable (as seen in equation (15)). However, following Godunov's method and the ARoe approach, some intermediate states can be defined before the values in t^{n+1} are averaged [35], as shown in Figure 2. These states are defined for the subcritical case as follows:

$$\begin{aligned}\mathbf{U}_i^{*}(\mathbf{U}_i,\mathbf{U}_{i+1},\mathbf{S}_{i+1/2}) &= \mathbf{U}_i^{n} + (\tilde{\gamma}\tilde{\mathbf{e}})^1_{i+1/2}\\ \mathbf{U}_{i+1}^{**}(\mathbf{U}_i,\mathbf{U}_{i+1},\mathbf{S}_{i+1/2}) &= \mathbf{U}_{i+1}^{n} - (\tilde{\gamma}\tilde{\mathbf{e}})^2_{i+1/2}\end{aligned} \tag{18}$$

where $\tilde{\gamma}_{i+1/2}$ stands for the compact expression of the fluxes and source terms at each edge $i+1/2$: $\tilde{\gamma}^m_{i+1/2} = \left(\tilde{\alpha} - \frac{\tilde{\beta}}{\tilde{\lambda}}\right)^m_{i+1/2}$.

The cell averaged solution for the next level, $\mathbf{U}_i^{n+1}$, is defined by these values, depending on propagation velocities, $\tilde{\lambda}^m_k$. Here, the subcritical case is taken into consideration but more details can be seen in [34]. According to Figure 2:

$$\mathbf{U}_i^{n+1}\Delta x = \mathbf{U}_i^{**}\left(\tilde{\lambda}^2_{i-1/2}\Delta t\right) + \mathbf{U}_i^{*}\left(-\tilde{\lambda}^1_{i+1/2}\Delta t\right) + \mathbf{U}_i^{n}\left(\Delta x - \tilde{\lambda}^2_{i-1/2}\Delta t + \tilde{\lambda}^1_{i+1/2}\Delta t\right) \tag{19}$$

that can be rewritten as

$$\mathbf{U}_i^{n+1}\Delta x = \mathbf{U}_i^{n}(\Delta x) + (\mathbf{U}_i^{**} - \mathbf{U}_i^{n})\left(\tilde{\lambda}^2_{i-1/2}\Delta t\right) + (\mathbf{U}_i^{*} - \mathbf{U}_i^{n})\left(\tilde{\lambda}^1_{i+1/2}\Delta t\right) \tag{20}$$

The intermediate states (*) and (**) are not actually used in the updating scheme. However, they involve the information of the fluxes and source terms and their analysis permit the definition of useful approaches to avoid numerical instabilities and unrealistic solutions, as it will be seen later. Combining (18) and (20), they lead to the 1D updating equation (analogous to the 2D (15) equation):

$$\mathbf{U}_i^{n+1} = \mathbf{U}_i^{n} - (\tilde{\lambda}\tilde{\gamma}\tilde{\mathbf{e}})^2_{i-1/2}\frac{\Delta t}{\Delta x} - (\tilde{\lambda}\tilde{\gamma}\tilde{\mathbf{e}})^1_{i+1/2}\frac{\Delta t}{\Delta x} \tag{21}$$

The extension to a 2D model assumes one dimensional RP at each cell edge, k, projecting the variables onto the normal vector (x' axis in Figure 1). Note that for a 2D model there exist 3 waves with different possibilities and more intermediate states.

Following subsections describe the use of these 2D intermediate states to fix, in particular, unrealistic solutions related with flow direction and negative water depth. When focusing on the flow direction, the variable used to limit the source term is the projected discharge, $q = (hu)n_x + (hv)n_y$, as it will be seen in subsection 2.3.1. On the other hand, intermediate state related with water depth, h, is used to control source terms when trying to avoid negative values of the water column.

2.3. Source term fixes

2.3.1. Friction correction to avoid reverse flow

Although the gravity force can reverse the flow, the friction force should merely reduce it. During a single time step, it is important to ensure that the sign of the flow velocity is not changed by the contribution of the friction term [34]. Otherwise the numerical friction contribution must be redefined and restricted. For that purpose, an approximated water discharge can be defined following (21):

$$(q)_i^* = (q)_i^n + (\tilde{\alpha}\tilde{\lambda})_k^1 - (\tilde{\beta})_k^1 \tag{22}$$

It is useful to split the total source term $\tilde{\beta} = \tilde{\beta}_{\mathbf{S}} + \tilde{\beta}_{\mathbf{F}}$, with the contributions of slope ($\tilde{\beta}_{\mathbf{S}}$) and friction ($\tilde{\beta}_{\mathbf{F}}$). This helps to evaluate the effect of the updated discharge without the friction term:

$$(q)_i^\star = (q)_i^n + (\tilde{\alpha}\tilde{\lambda})_k^1 - (\tilde{\beta})_{\mathbf{S},k}^1 \tag{23}$$

By multiplying (22) and (23) the effect of friction term can be analysed by checking the sign change:

$$(q)_i^\star < 0 \text{ and } (q)_i^* > 0 \rightarrow (q)_i^*(q)_i^\star < 0 \tag{24}$$

In this case, the numerical friction might produce flow reverse, contrary to what is physically possible, and the friction source wave strength can be redefined as:

$$\tilde{\beta}^1_{\mathbf{F}} = \begin{cases} (q)_i^\star & \text{if } (q)_i^*(q)_i^\star \leq 0 \\ \tilde{\beta}^1_{\mathbf{F}} & \text{otherwise} \end{cases} \tag{25}$$

setting $\tilde{\beta}^3_{\mathbf{F}} = -\tilde{\beta}^1_{\mathbf{F}}$ ($\tilde{\beta}^2_{\mathbf{F}} = 0$).

2.3.2. Source term correction to ensure positive water depths

Challenging cases may involve large bed slopes and roughness values that, within the wet domain, locally violate the model hypothesis and lead to numerical issues in the form of negative water depths at a few cells. As a first option to reduce this problem the time step can be restricted in order to ensure positivity in the solution [9]. However, this could lead to extremely small values of the allowable Δt. Thus, a reformulation of the classical wave splitting can be done to preserve the positive values in water depths [34]. Carrying on with the intermediate values, the modification of source strengths, β, enforces positive values of h_i^* and h_j^{***}, leading to positive water depth in the solution [34]. Therefore, for subcritical cases positive values of h_i^* must fulfil that

$$h_i^* = \underbrace{h_i^n + (\tilde{\alpha}\tilde{e}^1)^1_k}_{h^\star} - \left(\frac{\tilde{\beta}}{\tilde{\lambda}}\right)^1_k \geq 0 \tag{26}$$

while positive values of h_j^{***} require that

$$h_j^{***} = \underbrace{h_j^n - (\tilde{\alpha}\tilde{e}^2)^1_k}_{h^\star} + \left(\frac{\tilde{\beta}}{\tilde{\lambda}}\right)^2_k \geq 0 \tag{27}$$

with $h_i^\star > 0$. Then, a redefinition of the source strengths can be written depending on the intermediate states sign. If $h_i^* < 0$, then:

$$\tilde{\beta}^1 = \begin{cases} h^\star \tilde{\lambda}_k^1 & \text{if } h_i^* < 0 \\ \tilde{\beta}^1 & \text{otherwise} \end{cases} \quad ; \quad \tilde{\beta}^3 = -\tilde{\beta}^1 \tag{28}$$

and in case that $h_j^{***} < 0$, the source strength is modify as follows

$$\tilde{\beta}^3 = \begin{cases} -h^\star \tilde{\lambda}_k^3 & \text{if } h_j^{***} < 0 \\ \tilde{\beta}^3 & \text{otherwise} \end{cases} \quad ; \quad \tilde{\beta}^1 = -\tilde{\beta}^3 \tag{29}$$

2.4. Wet-dry treatment

An additional case that could lead to unrealistic solutions of water depth is the wetting/drying process. The interaction between wet and dry cells could lead to negative values when the flow direction is against the slope and the dry cell bottom level is above the water surface level of the wet cell, as depicted in Figure 3. In this case, the basic updating scheme would compute a negative water depth in the dry cell [35].

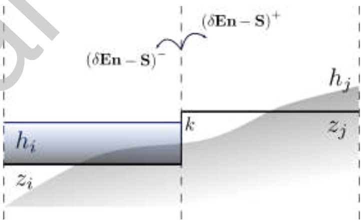


Figure 3: Wet-dry representation.

In order to ensure the positivity of the solution, the flux splitting is performed preventing it from crossing the edge if the result is negative. Following the notation of Figure 3, where **En** are the fluxes through the edge (in **n** direction) and **S** the source terms, the following algorithm is implemented:

- If $h_j^n = 0$ and $h_j^{***} < 0$ then

$$(\delta\mathbf{En} - \mathbf{S})^-_{i,k} = (\delta\mathbf{En} - \mathbf{S})_k, \qquad (\delta\mathbf{En} - \mathbf{S})^-_{j,k} = 0 \tag{30}$$

- If $h_i^n = 0$ and $h_i^* < 0$ then

$$(\delta\mathbf{En} - \mathbf{S})^-_{j,k} = (\delta\mathbf{En} - \mathbf{S})_k, \qquad (\delta\mathbf{En} - \mathbf{S})^-_{i,k} = 0 \tag{31}$$

- Otherwise

$$(\delta\mathbf{En} - \mathbf{S})^-_{i,k} = \sum_{m=1}^{3} \left(\tilde{\lambda}^- \left(\tilde{\alpha} - \frac{\tilde{\beta}}{\tilde{\lambda}} \right) \tilde{\mathbf{e}} \right)^m_k \tag{32}$$

$$(\delta\mathbf{En} - \mathbf{S})^-_{j,k} = (\delta\mathbf{En} - \mathbf{S})^+_{i,k} = \sum_{m=1}^{3} \left(\tilde{\lambda}^+ \left(\tilde{\alpha} - \frac{\tilde{\beta}}{\tilde{\lambda}} \right) \tilde{\mathbf{e}} \right)^m_k \tag{33}$$

that indicates that the flux sent to cell j will be sent back to cell i if it can not ensure positive water depth: $h_j > 0$. Additionally, it is important to impose a zero value on the velocities normal to edge k in cases where the flux is not crossing the cell edge, as depicted in Figure 3.

3. Acceleration technologies

3.1. High Performance Computing (HPC)

The main drawback on these mathematical problems is the high computational cost. The presented two dimensional model must update 3 variables at every cell gathering contributions from 3 cell edges using a global time step dynamically computed. This implies a large amount of operations. Thus, the number of time steps is high and, consequently, the total simulation time might turn unaffordable. This situation leads to the necessity of alternatives when simulating large domains.

There are many different ways of performing accelerated calculations to overcome this problem. Parallel technologies are based on the distribution of workload into units that work simultaneously.*Open Multi-Processing* (OpenMP) applications divide the work load into all the processors available on the computer [22]. However, this technology is extremely limited in scalability to the hardware system. If needed, MPI (*Message Passing Interface*) technologies are available to overcome this problem, using several devices containing several cores to carry out massive calculations [23]. However, this alternative involves not only an increase on budgets that might not be affordable, but also implies a constraint in data-transfer and difficulties in the domain-decomposition strategies.

Another alternative is the use of *Graphical Processing Units* (GPU) to have a high amount of computing threads into a single device, the GPU [24]. These devices were initially developed to deal with graphic operations. Nowadays, NVIDIA has developed a toolkit to run parallel solutions on its devices: the CUDA toolkit. CUDA allows the programmer to implement the code in GPU in its familiar programming environment (C language, in this case) just by incorporating expressions for the parallel parts of the code. This technology is continuously growing and the devices are constantly improving regarding the number of cores, speed on the data transfer and increasing the efficiency of the CUDA toolkit. For instance, the necessity of data transfer (I/O) between CPU and GPU require a computational effort that could entail a bottleneck on a simulation. Recently, NVIDIA unveiled "GPUDirect" storage, a new capability that enables its GPUs to talk directly with NVM-Express storage without needing to involve the host CPU and system memory [46]. Hence, newer cards present better performances than former ones.

3.2. Parallel code and HPC devices details

The numerical scheme afore presented has been implemented to run on GPU cards in order to increase the performance of the simulations. The details regarding the implementation can be found in [26].

This work is devoted to be a study of performance applied to flood simulation. Thus, the cases presented here are carried out with different GPU devices and an analysis of speed-up is done to show the potential performance of the technology and the suitability of those complex models in spite of their compu-

	Type	Processor	CUDA Cores	Memory	Year
GPU1	GPU	GTX 780	2304	3072 MB	2013
GPU2	GPU	GTX Titan Black	2880	6144 MB	2015
GPU3	GPU	Tesla K40	2880	12 GB	2013
GPU4	GPU	Tesla V100	5120 [1]	16 GB	2017

Table 1: Characteristics of the different GPU devices.

tational cost. All the different devices are shown in Table 1.

3.3. Updating flowchart

Figure 4 shows the updating flowchart that is followed in the numerical scheme. On blue rectangles, CPU instructions are represented, while functions executed on GPU are depicted in green rectangles. The preprocess, where the mesh is allocated on the CPU memory and all the input data are read, is done in CPU and the necessary information is later transferred to the GPU. The temporal loop is all allocated in the GPU and is executed as many times as time steps are required to achieve the final time.

4. Validation of the model: the Toce River

A physical model of the Toce River, located on the Italian part of the Alps, was built by the Hydraulic Research Laboratory (Milan, Italy), characterizing a 5km stretch of the real valley of the river. The available Digital Terrain Model (DTM) had a resolution of 5cm x 5cm. The physical model was entirely constructed with the same material so that the use of a uniform roughness coefficient was suggested [47]. Over this model, a hydrograph was set as inlet boundary condition, as shown in Figure 5(a), representing a flood event with

[1]Including the new development: 8 x 640 Tensor Cores

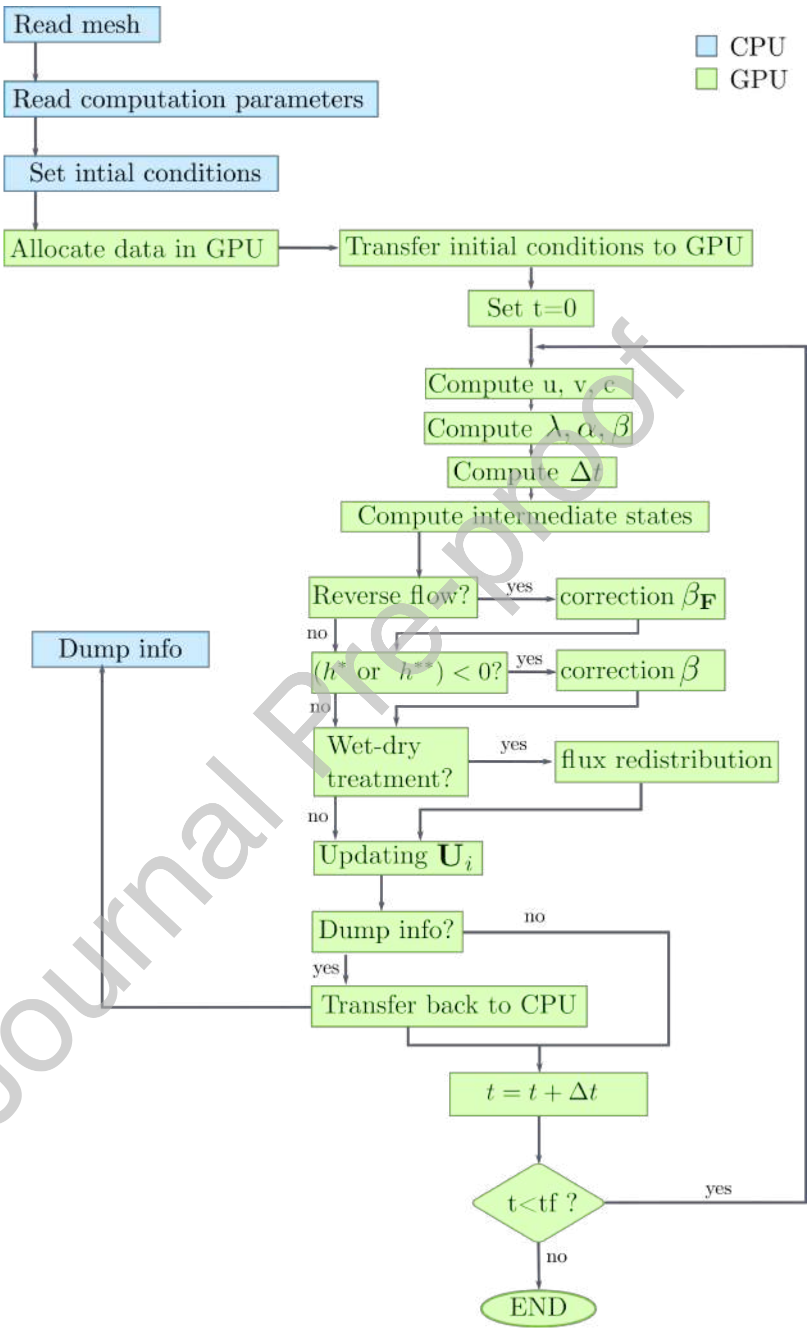


Figure 4: Algorithm flowchart for computation

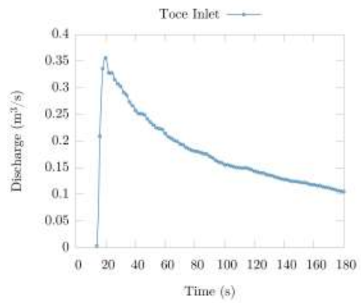


Figure 5: Inlet boundary condition for the Toce River physical and computational model.

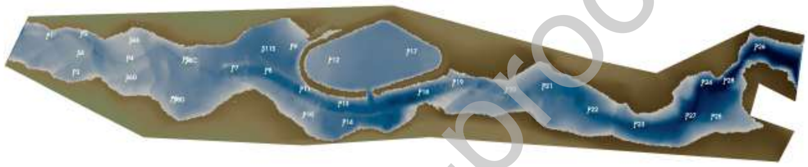


Figure 6: Probes distribution over the physical and computational model of the Toce River.

a sharp discharge peak, whereas free outflow conditions were suggested at the downstream boundary. The evolution of water surface level was registered at several probes spread out over the topography that can be seen at Figure 6, so the data can be compared with computational simulations of the case.

This physical model was created to validate different simulation approaches and now works as a benchmark test case for computational models calibration [47, 48]. In our case, the DTM resolution led to a 98672 triangular cell unstructured mesh as the finest option. As this represents a small size test case, it is more suitable to test the numerical properties of the scheme rather to test the HPC performance. The simulation started from dry bed initial conditions and the numerical stability was controlled by dynamically choosing the time step under the restriction corresponding to CFL = 0.9.

In Figure 7, the comparison between the physical and the computational model is displayed in terms of water surface level evolution at different points, of which only a few are shown to avoid redundancy. The model provides good accuracy in water surface elevation prediction at some of the probes and there is an accurate prediction of the wave front arrival time. This is not only because the terrain is well represented in terms of elevation, but also in terms of friction. This case has been run using a uniform friction coefficient (n=0.0162 s/m$^{1/3}$). However, the Manning coefficient is not provided in the data set, but the case was computed and calibrated by other researchers under different conditions [48], including this one. As reported in [48], the sensitivity of the numerical results to the roughness coefficient, the mesh refinement and other choices is complex in this case and some discrepancies can be observed.

In order to quantify the reliability of the model and the mesh, a Nash-Sutcliffe efficiency coefficient, common to asses the predictive power of a hydrological model [36, 37], has been used. The coefficient for the water depth is defined as:

$$NSE_h = 1 - \frac{\sum_{t_0}^{T_f} (h_s^t - h_o^t)}{\sum_{t_0}^{T_f} (h_o^t - \overline{h_o^t})} \tag{34}$$

where subscript S stands for simulated, O for observed (measurements) and the overbar value is the mean of the whole time series of observed data. The coefficient can take values $-\infty \leq NSE \leq 1$, meaning $NSE = 1$ a perfect fit. The obtained results for the 12 probe locations are displayed in Table 2.

	P2	P3	P4	P5	P8	P9
NSE (h)	0.77	0.63	0.42	0.77	0.82	0.93
	P10	P19	P21	P23	P24	P25
NSE (h)	0.65	0.86	0.97	0.93	0.95	0.97

Table 2: Nash-Sutcliffe coefficient for the water depth at different probes of the Toce river model.

The model must get a Nash-Sutcliffe coefficient over 0.6 to be validated as acceptable. Values over 0.8 indicate a good agreement, while values over 0.9

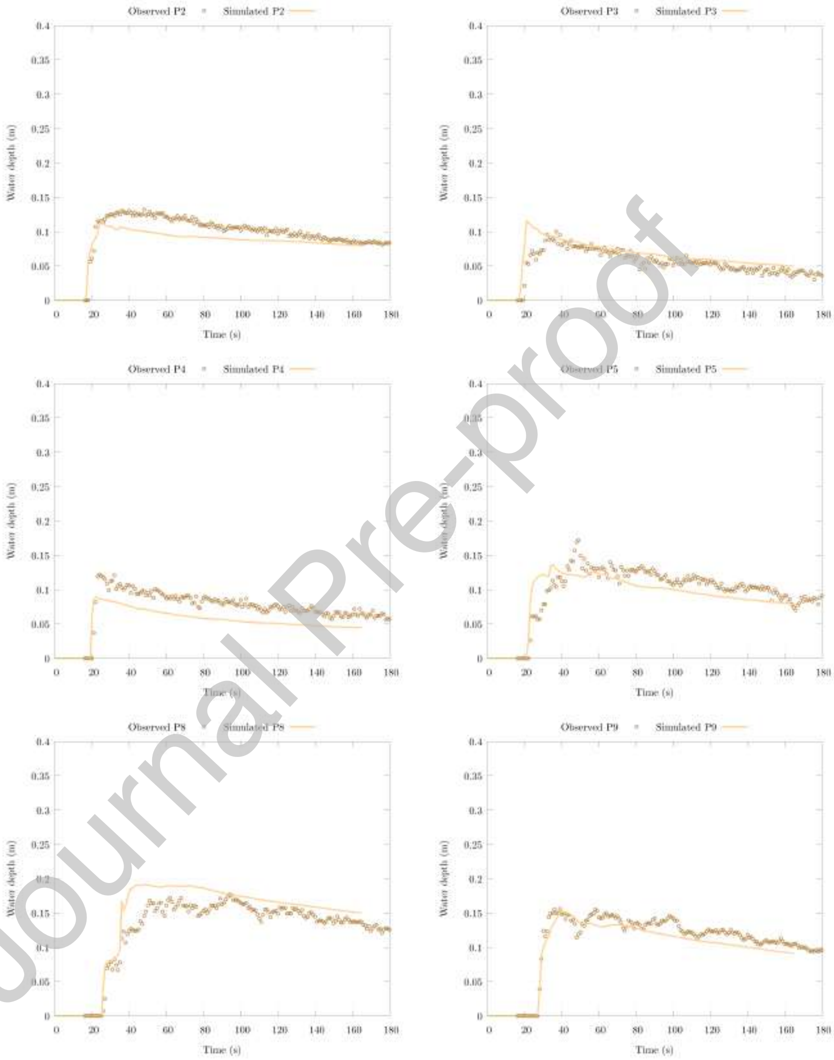


Figure 7: Comparison between experimental and computed data of time evolution water depth (m) at probes P2, P3, P4, P5, P8 and P9.

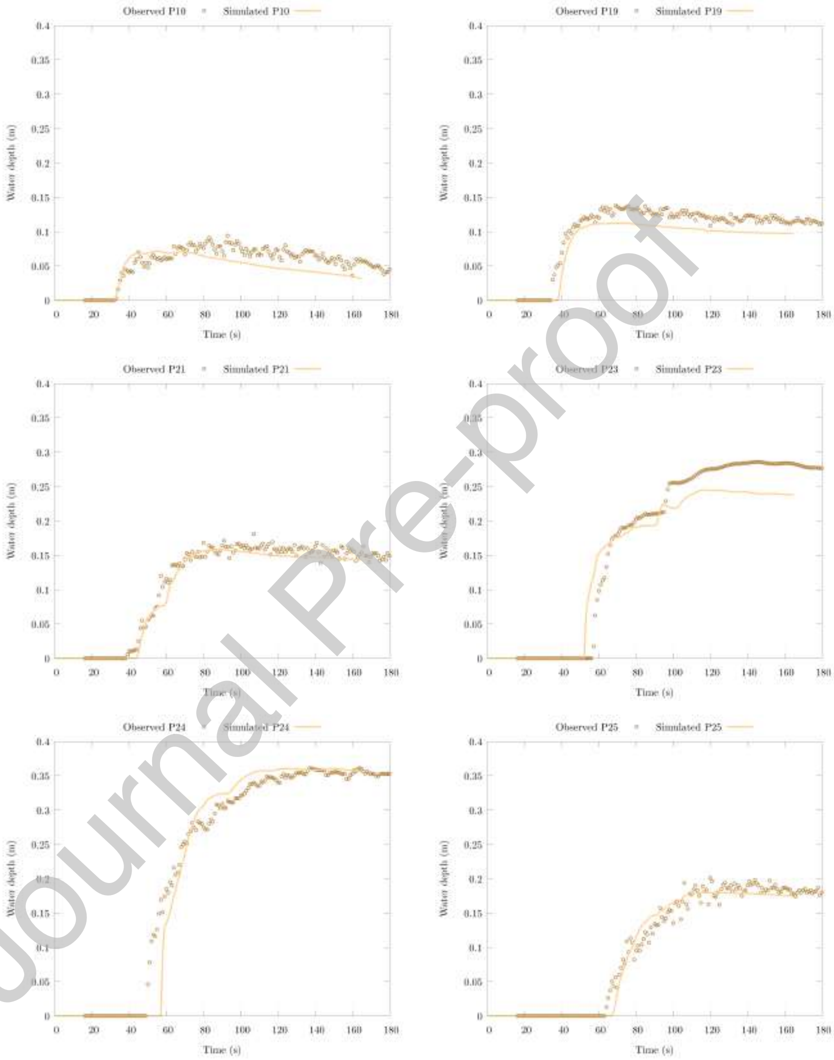


Figure 8: Comparison between experimental and computed data of time evolution water depth (m) at probes P10, P19, P21, P23, P24 and P25.

designate very good results. Therefore, it can be concluded that, apart from P4 whose discrepancies were also reported by other authors [48], the model presents good and very good agreements.

As this is a challenging case that could require a correction in the source term strengths, β, and a careful wet-dry front treatment, additional simulations under different numerical conditions have been carried out in the Toce River to highlight the benefit of the numerical fixes presented in the previous sections.

The sensitivity of the numerical results to the numerical fixes presented in section 2.2 is next analyzed. For that purpose, the separate influence of the source term correction and that of the wet-dry treatment have been considered. The numerical scheme with wet-dry treatment and without source term correction will be denoted E1, whereas the numerical scheme with source term correction and without wet-dry treatment will be denoted E2. R represents the original reference simulation with all the numerical fixes. In both cases E1, E2, the CFL condition (16) will not be sufficient to ensure numerical stability. Therefore, as reported in [9], the time step size must be reduced to avoid negative water depths. The actual implementation of the time step reduction is to halve Δt while the water depth is less than the water depth threshold value (TH) used to differentiate a wet cell from a dry cell. It is important to note that the lower this tolerance is chosen, the lower the mass conservation error will be.

As the time step size is dynamically computed during all simulations, an average size is shown for the sake of comparison in Table 3. The reference simulation, R, uses a tolerance TH $= 10^{-12}$ m and performs the simulation with an average time step size $\Delta \mathrm{t} = 4.73 \times 10^{-3}$ s. However, this TH value for E1 and E2 results into extremely small time step sizes (of the order of 10^{-8}), which leads to a practically stop in the simulation. In fact, if contributions are negative for a dry cell, no positive time step could guarantee the water depth positivity preserving. Therefore, only the first 4 seconds of the simulations (until they

TH [m]	Δt_R [s $\times 10^{-4}$]	Δt_{E1} [s $\times 10^{-4}$]	Δt_{E2} [s $\times 10^{-4}$]
10^{-12}	47.3	0.000893 [2]	0.000741 [2]
10^{-6}	-	1.98	0.035
10^{-3}	-	53.46	53.51

Table 3: Average time step size for simulations E1 and E2 when setting different mass tolerances TH and their comparison with the reference simulation, R.

simulation virtually stops) are shown. If TH is relaxed to 10^{-6} m, the simulation is completed although the time step size values are significantly lower than the reference, as seen in Table 3. Only TH=10^{-3} m allows to recover the same order of magnitude of the reference simulation for the time step size. However, increasing the TH has dramatic consequences for the mass conservation.

The absolute mass error is computed every time step as:

$$\epsilon = \sum_{i=1}^{N} A_i h_i^{n+1} - \sum_{i=1}^{N} A_i h_i^{n} - \Delta t(Q_{in}^n - Q_{out}^n) \tag{35}$$

where N is the total number of cells, Q_{in}, Q_{out} stand for the total discharges entering and leaving the computational domain respectively. Table 4 shows the time integrated mass error for the simulations for R, E1 and E2 using the mentioned TH values. The integrated error is in the order of 10^{-11} m^3 for the reference simulation. Although both E1 and E2 reach an even lower mass conservation integrated error, the simulations virtually stop at t = 4 s (as mentioned before) due to the impossibility of satisfying the water depth positivity for this tolerance. If the tolerance is relaxed to 10^{-6} m, E1 and E2 results are different. This fact reveals that if the wet/dry treatment is disabled, the consequences for the mass conservation are critical. Finally, the integrated mass errors for TH = 10^{-3} m are become inadmissible.

[2]Until the simulation virtually stops.
[3]Until the simulation virtually stops.

TH [m]	ϵ_R [m^3]	ϵ_{E1} [m^3]	ϵ_{E2} [m^3]
10^{-12}	1.91×10^{-11}	7.39×10^{-14} [3]	2.059×10^{-12} [3]
10^{-6}	-	-0.0133	-18.38
10^{-3}	-	-23.51	-23.31

Table 4: Cumulative mass error for simulations with E1 and E2 and different mass tolerances and their comparison with the reference simulation, R.

In light of these results it can be said that the aforementioned numerical fixes are important to ensure mass conservation and accurate results without decreasing the time step size, which results crucial when facing real time simulations.

5. Application to a real test case: the Ebro River

The Ebro River basin is managed by the Ebro River Basin Authority (Confederación Hidrográfica del Ebro-CHE) (www.chebro.es). They provided all the topography details as well as hydraulic data from the gauging stations along the basin. The river basin is located in the North-East of Spain (Figure 9) and has an extension of 85362 km^2. Although it is not one of the most populated regions in the country, many urban areas of different size can be found near the river together with areas dedicated to agriculture and farming.

It is a river with an average discharge of Q = 400 m^3/s that increases up to 2500 m^3/s when flooding occurs (return period between 1 and 2 years). The present work is restricted to the physical domain in the middle part of this river (see Figures 9 and 10), which is the most affected by floods. It is a 125 km stretch of river limited by two gauging stations: Castejón de Ebro upstream and Zaragoza downstream. As inlet boundary condition, the hydrographs provided from the measurements at Castejón gauging station are used. The gauging curve (water surface level vs. discharge) provided from measurements in

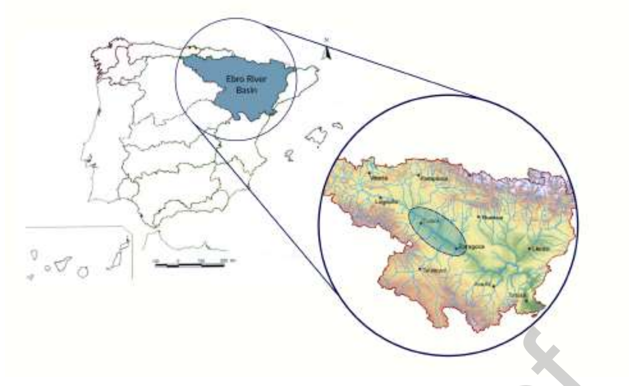


Figure 9: Location of the Ebro river basin in the North East of Spain and location of the analyzed domain (blue ellipse).

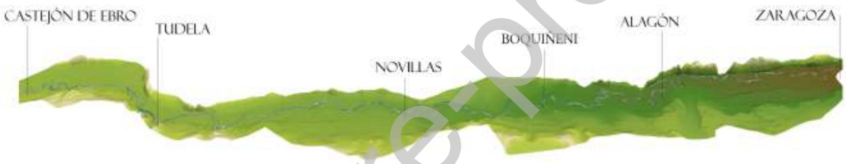


Figure 10: Middle part of the Ebro river object of the study and location of relevant towns.

Zaragoza gauging station is imposed as outlet boundary condition. The rest of the boundaries are chosen far enough not to be reached by the flow. The total computational domain chosen encompasses a total extension of 744 km^2 (see Figure 10).

A calibration process is crucial to set up the model with the most suitable computational mesh and the proper roughness values. When dealing with domains of the size presented here this process could turn unaffordable due to the large amount of simulations that are needed and the high computational times. The numerical model robustness and the GPU implementation play an important role in this process. The data pre-processing, the calibration, and the final numerical results are presented next.

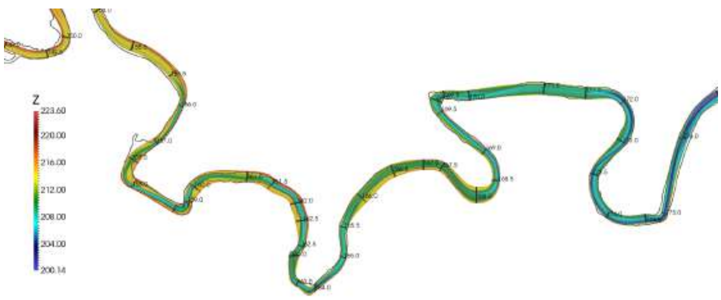
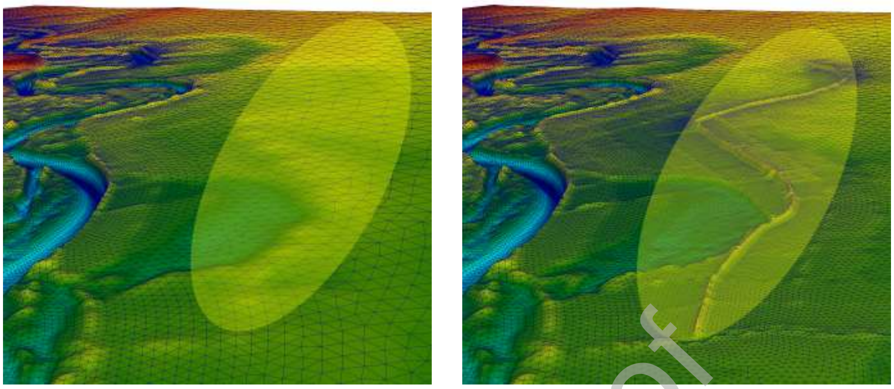


Figure 11: Detail of the bed raster interpolated from cross sections.

5.1. Data pre-processing

For the two-dimensional floodplain, a digital terrain model (DTM) is used to represent the topography. It provides a square-mesh with equal spacing that has a resolution of 5m x 5m. At the river bed itself, the DTM is not valid since LIDAR technology is not able to measure properly under water due to reflection and provides an irregular bed with non-realistic data. Thus, provided river cross sections are used as second data source to represent the main channel. In particular, they are used to interpolate the river bed cross sections and create a new DTM for the river bathymetry [49], as represented in Figure 11, complementary to the floodplain DTM.

A computational mesh is built mapping the terrain elevation from the DTMs. In order to obtain a fast and accurate calculation, the most suitable is a terrain-adapted unstructured triangular mesh [50]. Since it is necessary to reach a compromise between speed of calculation and accuracy of results [50] adaptive meshes in space are used; in such a way that the cells are small where we need much detail of the flow (riverbed, levees, etc.), and big in areas far from the riverbed where practically water does not arrive hardly ever always paying attention to their regularity [51]. In particular, edge sizes ranging from 5 m (near river bed) to 150 m (at boundaries) were first imposed. The mesh was designed starting by the requirement to have enough river bed resolution (at least ten cells at a typical cross section) but less resolution on the floodplain,



(a) (b)

Figure 12: Wrong (a) and correct (b) representation of a levee on the floodplain comparing calibrated (b) and non-calibrated (a) meshes.

leveraging the grid adaptability offered by unstructured triangular meshes. This led to an initial mesh.

The mesh calibration process was made with an event occurred in 2015 focusing on flood extension. As not only discharge and elevation data at certain stations, but also maps with the maximum extension of the flooded area were supplied, both a qualitative (using visual exploration) and a quantitative (using a critical success rate index that will be addressed later) comparison between computed and measured flooded area were used to detect the necessity to refine the mesh near levees. This led to the final mesh containing 867672 triangular cells. Figure 12 highlights the differences between final (b) and initial (a) meshes that were generated for the representation of a levee in the floodplain. The figure shows how a local refinement is needed to capture narrow levees. During this process, around 20 simulations were needed to achieve a proper mesh.

The model must be calibrated not only in terms of the mesh, but also in terms of the roughness which is based on a land use map. When dealing with real domains and river representation, a uniform Manning roughness coefficient

Figure 13: Initial rough (left) and final detail (right) roughness map.

is not accurate enough. Thus, land use maps must be used to generate proper roughness distributions. Initially, a coarse Manning roughness distribution map was built by outlying different areas from visual exploration of the land use maps. A trial and error calibration process was carried out to achieve a coincidence between peak values of the measured hydrographs at Tudela and Zaragoza together with the analysis of travelling times. This process showed the necessity of a more complex roughness distribution as the complexity of the two-peak hydrograph, that was associated to different size flooding areas, involved different areas of the floodplain. This led to the improvement of the model by incorporating GIS information to perform the soil distribution effect. The roughness calibration step required another 30 additional simulations. Thus, the final model includes, in addition to a detailed representation of the available topography data, a complex roughness distribution. Figure 13 shows a comparison between the nominal land use map before calibration and the final calibrated roughness values.

The historical events that have been simulated in this work are summarized in Table 5, where the real flood duration is specified in the last column.

A first simulation is carried out for each flood event with initial dry bed un-

Case	Flood peak date	Max. discharge (m^3/s)	Flood duration (days)
1	2/6/2008	1797	8.91
2	13/6/2009	1800	24.27
3	16/01/2010	2000	5.25
4	01/02/2015	2600	21.0

Table 5: Characteristic data of simulated historical events.

til the steady state corresponding to the constant upstream discharge supplied has been reached. Note that the initial conditions must be updated after the calibration process according to the final roughness distribution. The value of the steady state discharge that will be the initial condition for the flooding simulations corresponds to the first discharge value encountered in the upstream boundary condition of the flood hydrograph that is going to be simulated.

5.2. Numerical results

This section shows the comparison between the simulated and the measured data for the different flood events. Due to the existence of gauging stations in the river basin, the temporal evolution of water surface elevation (η=h+z) can be always compared in order to ensure the accuracy of the results. Additionally, some of the stations have also discharge measurements so these data are also used.

In Figure 14 the time evolution of discharge at gauging stations is shown for the four analysed cases. The plot of the inlet hydrograph for each event can be seen in the figure, as well as the conveyance to other gauging points. According to them, the evolution of the hydrograph along the river is in good agreement with the observed data. The correct simulation of the flood wave arrival time at each point is quite important, and the model is able to reproduce it properly. The other relevant effect is the change on the wave shape due to the storage capacity of the floodplain, which can also be seen on the results due to the use of a 2D model guaranteed by the quality in the mesh construction.

Again, the Nash-Sutcliffe coefficient as a measure of agreement between the numerical and the observed results. The coefficient for the discharge is defined as:

$$NSE_Q = 1 - \frac{\sum_{t_0}^{T_f} (Q_s^t - Q_o^t)}{\sum_{t_0}^{T_f} (Q_o^t - \overline{Q_o^t})} \tag{36}$$

The obtained results for the 4 cases at the 2 gauging stations are included in Table 6. The gauging station at Zaragoza for case 4 shows the lowest NSE (0.82) that still indicates a good model. The rest of measurement points show a very good agreement achieving 0.99 in Tudela for case 1.

NSE_Q	Case 1	Case 2	Case 3	Case 4
Tudela	0.99	0.97	0.995	0.93
Zaragoza	0.95	0.95	0.97	0.82

Table 6: Nash-Sutcliffe coefficient for the discharge during different flood events at different gauging stations of the Ebro river.

Since the Ebro River Basin Authority manages interventions when a flood event takes place, their interest on water surface levels at certain locations makes their representation also quite interesting. Figures from 15 to 17 show the time evolution of η at the gauging stations displayed in Figure 10. The results differ mostly at the beginning of the simulation which may be due to the uncertainty in the water surface profile set as initial condition. The initial condition was computed as the steady state corresponding to the discharge measured at the Castejón gauging station (upstream section). This was already an approximation as the discharge was not exactly uniform along the actual river in that moment. Furthermore, the comparison of the water surface level at the gauging sections is full of uncertainty concerning the actual bed level. It is important to stress that the model computes water depth from bed level data. However, the trend followed by the numerical results is quite similar to the measured data,

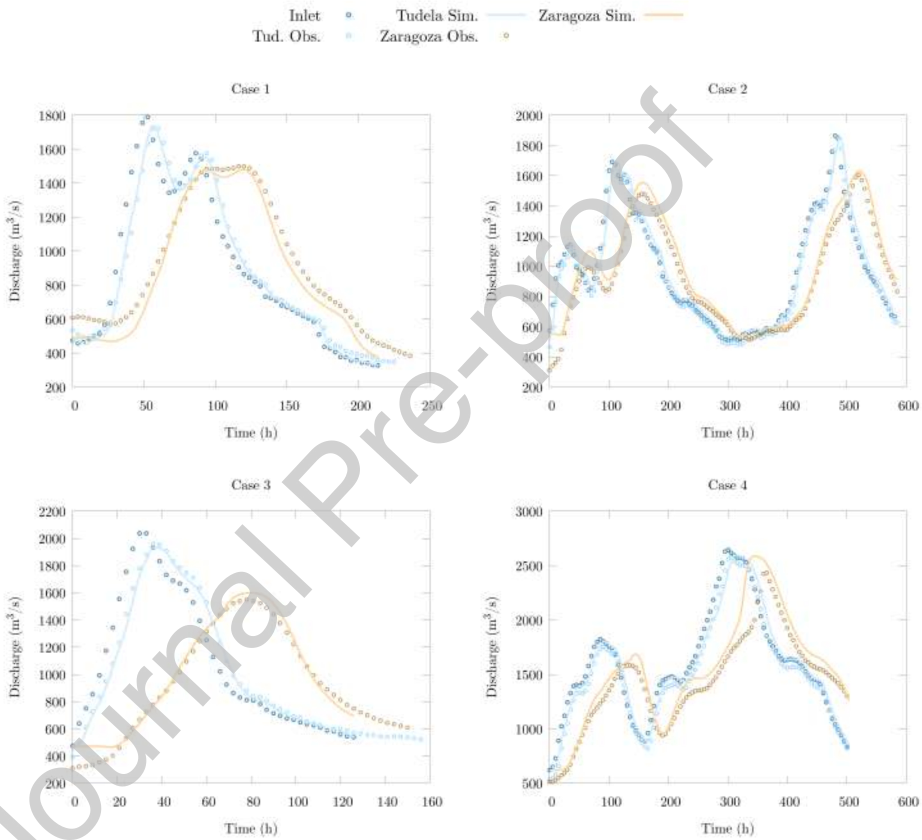


Figure 14: Discharge time evolution observed and computed at Tudela, Novillas and Alagón gauging stations for the 4 different flood events analyzed.

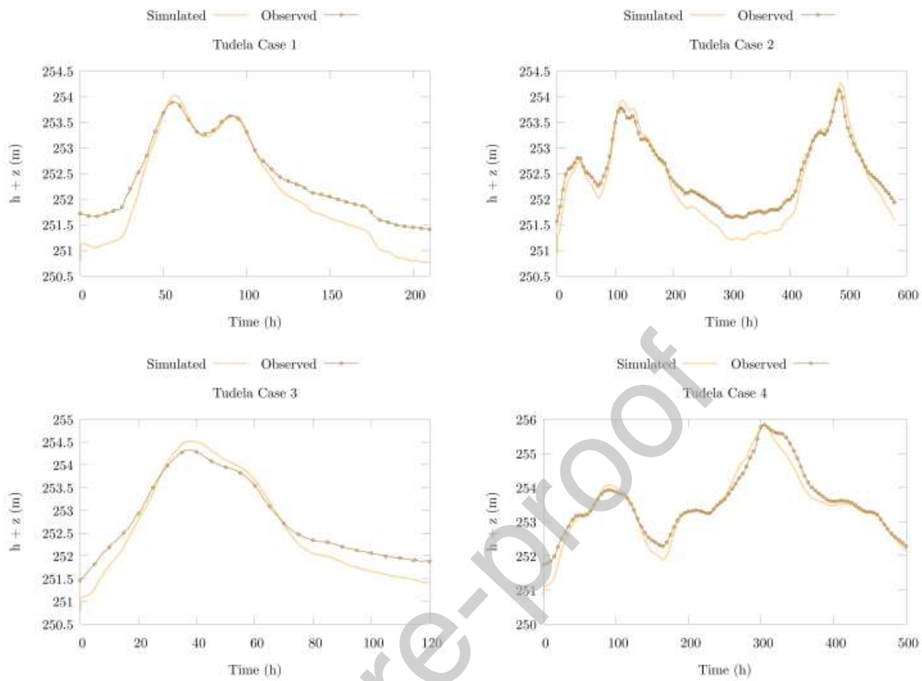


Figure 15: Time evolution of water surface elevation observed and computed at Tudela gauging station for the 4 cases.

and only differences of centimetres are seen in terms of water level.

Finally, in light of the discrepancies, it is worth noting on one hand that the roughness values were adjusted to the overall best fit of the discharge hydrographs involving peak values and arrival times. On the other hand, it is important to mention the lack of available updated data of the terrain in such kind of basins. For instance, river bed cross sections or land uses may have changed along the years since they were measured, or some levees in the floodplain are narrower than the DTM resolution (5m x 5m). When simulating river stretches, an updated set of starting data is essential and decisive and could lead to even better results.

In order to quantify the reliability of the model the Nash-Sutcliffe coefficient

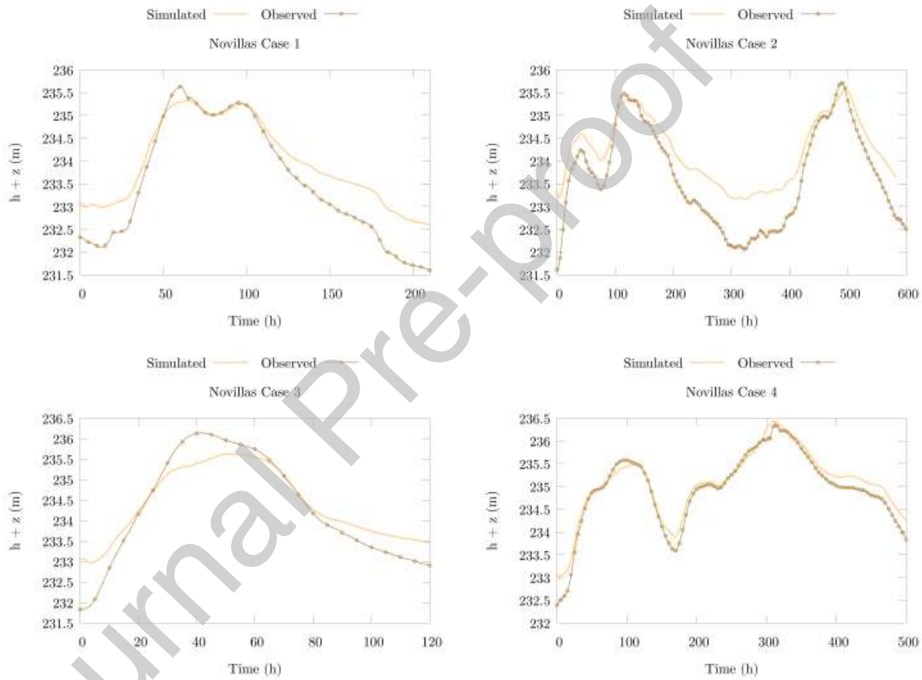


Figure 16: Time evolution of water surface elevation observed and computed at Novillas gauging station for the 4 cases.

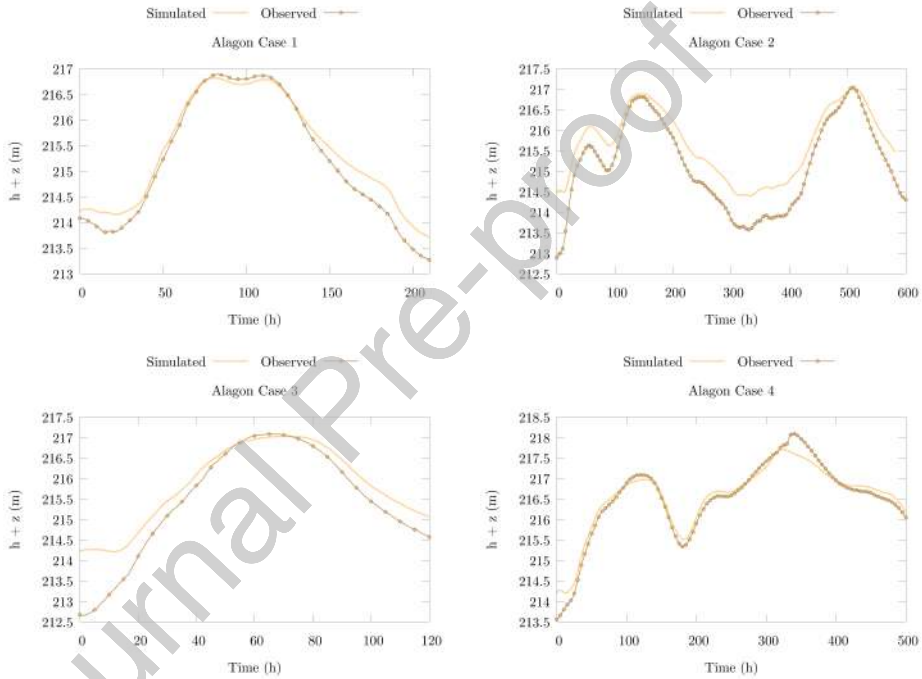


Figure 17: Time evolution of water surface elevation observed and computed at Alagón gauging station for the 4 cases.

for the η computed time series compared with data is defined as:

$$NSE_{\eta} = 1 - \frac{\sum_{t_0}^{T_f} (\eta_s^t - \eta_o^t)}{\sum_{t_0}^{T_f} (\eta_o^t - \overline{\eta_o^t})} \tag{37}$$

The obtained results for the 4 cases in the 3 locations of the stations are included in Table 7. Water elevations are more challenging to reproduce since they do not represent an integrated measurement, as the discharge does, but a specific spatial measurement. Nevertheless, all the points indicate a good or a very good agreement, except for Novillas in case 2, which present an acceptable agreement.

NSE_{η}	Case 1	Case 2	Case 3	Case 4
Tudela	0.728	0.8309	0.8696	0.9344
Novillas	0.8017	0.6387	0.9145	0.9437
Alagón	0.9508	0.7231	0.8294	0.9603

Table 7: Nash-Sutcliffe coefficient for the water surface level during different flood events at different gauging stations of the Ebro river.

Additionally, information concerning the maximum observed flooded area was available for the 2015 flood event. It was used to compare with the maximum flooded area predicted by the model in this event. For that purpose, the flooded area at t=314 h was used. Figure 18 shows the comparison of both maximum inundation areas. The visual agreement between observed and computed data can be quantified by using the critical success index, $C(\%)$, [52, 53]:

$$C(\%) = 100 \frac{A_{Obs} \cap A_{Sim}}{A_{Obs} \cup A_{Sim}} \tag{38}$$

that can vary between 0 and 100%. It is worth mentioning that although this rate penalizes for both under- and over-prediction, a 89.67% of coincidence is reached.

All the flood event simulations were carried out with four different GPU devices. The required computational times are shown in Table 8 together with

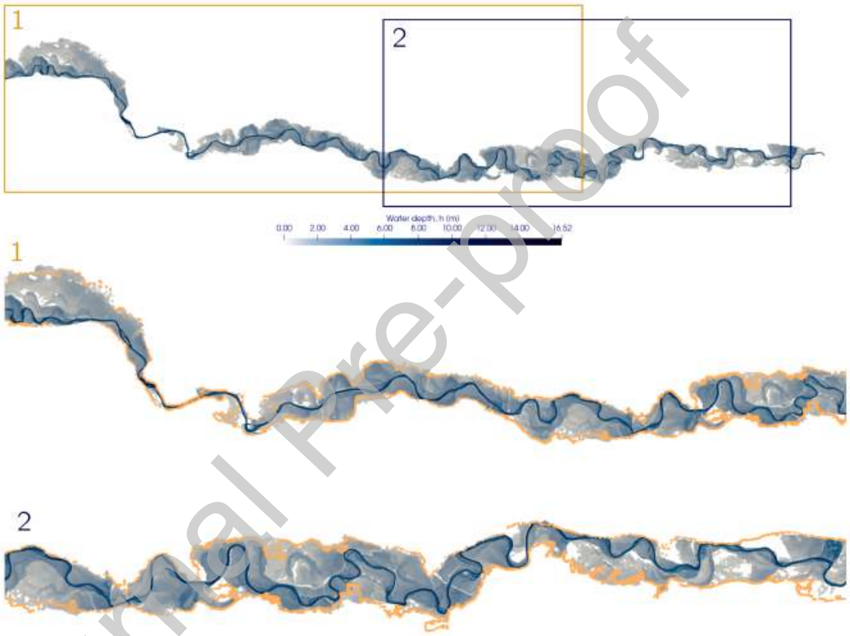


Figure 18: Maximum extension of the flooded area observed (orange points) and computed (in blue scale) for the 2015 flooding event. Zoom in two different zones.

Case	Flood duration	t_{GPU1}	t_{GPU2}	t_{GPU3}	t_{GPU4}
Case 1	8.91 d	7.26 h	5.88 h	3.69 h	1.19 h
Case 2	24.27 d	21.14 h	17.25 h	10.676 h	3.63 h
Case 3	5.25 d	4.75 h	3.83 h	2.38 h	0.495 h
Case 4	21.0 d	21.27 h	16.36 h	10.40 h	3.45 h

Table 8: Different computational times for the 4 events carried out in different processing devices.

Case	Flood duration	r_{GPU1}	r_{GPU2}	r_{GPU3}	r_{GPU4}
Case 1	8.91 d	29.45	36.36	57.95	179.69
Case 2	24.27 d	27.55	33.76	54.55	160.46
Case 3	5.25 d	26.52	32.89	52.94	254.54
Case 4	21.0 d	23.69	30.8	48.46	146.08

Table 9: Different ratios of computational time for the 4 events carried out in different processing devices.

the real flood duration. For the sake of a better comparison, Table 9 shows the ratio, between the real flood duration (d) and the computational time for each simulation at each device (t) as $r = d/t$. Note that this ratio is within a different range for each device, varying depending on the case (i.e. the number of wetted cells).

In order to compare the level of speed up, it is worth mentioning that the 21 day flood event (case 4) is computed in 21 days when a 12 CPU cores (Intel Xeon X5650) is used. Although a 12-cores parallelization is used, the computational time results unaffordable when trying to have a tool with prediction on-line purposes, as seen in Table 8.

6. Conclusions

Since flood consequences result into huge disasters and high amount of human and economic losses, prediction studies become an important tool. In particular, nowadays numerical simulations provide accurate results and their development is increasing with the objective of turning complex models into affordable and useful ones. Recent research has given rise to a new generation of 2D hydraulic models able to run in shorter times thus providing the opportunity to enhance the current flood early warning systems. These hazardous events have been traditionally simulated by means of different simplified or modified models that avoid full 2D frameworks in order to increase the speed performance. The present work proposes an efficient model that is able to reproduce flood events in affordable times by using a 2D model on GPU.

Additionally, not only a 2D model for flood events has been presented, but also a robust 2D model completed with numerical corrections that avoid instabilities and make the model suitable for the simulation of complex phenomena. When dealing with real test cases, numerical fixes result crucial to ensure mass conservation and physically feasible solutions without a decrement of the CFL stability condition or directly the time step size.

Since models and numerical schemes must be first verified with benchmark cases in which the physical terrain, the inlet discharge and recorded variables evolution are properly compared with simulation data, the Toce River physical model has been used. The results demonstrate the necessity to control the numerical source terms as well as the wet-dry front as they have an important influence on the numerical stability. The quality of the results depends on the compromise between minimum water depth tolerance and time step size. The presented approach provides machine accurate conservation errors at the maximum possible time step size.

The application to the simulation of flood events in a reach of the Ebro river highlights that the use of real data introduces some uncertainties related with coarse discretizations of the terrain measurements, non detailed characterization of bed roughness, spurious points on the discharge time series, and other problems that may provoke errors on the results regardless of the numerical scheme. Hence, calibration processes must be carried out. An optimal computational mesh has been generated and calibrated for the Ebro River. The 2015 event has been reproduced reaching an accuracy of 89.67% on area fit. Additionally, other floods have been reproduced in order to ensure mesh accuracy reaching very accurate results in terms of Nash-Sutcliffe Error. As seen in the results, the storage effect of the 2D floodplain is pointed out in the reproduction of hydrographs conveyance, which could not be seen properly with other simplified models. Additionally, although real test cases introduce some errors due to the lack of available data, the model is still able to provide very good results.

Finally, not only the benefits of an accurate and fast numerical method are desirable for flood prediction but also the generation of an appropriate computational mesh and an adequate representation of land use maps are demonstrated to be necessary in order to carry out computations leading to accurate numerical results. The numerical results obtained for different flood events in two river flood cases have been presented and compared with measurements. It should be emphasized that the use of HPC technologies is vitally important when carrying out simulations with large domains and long event durations. In this work, the results for the Ebro River, containing a large domain (744 km^2) and great number of computational cells (867672), are obtained using a GPU-parallelized well-balanced upwind numerical scheme which simulates a hydrograph of 21 days in a bit more than 3 hours, with a Tesla V100, and makes feasible to reproduce events on a real-time basis.

Declaration of Competing Interest

All authors have participated in (a) conception and design, or analysis and interpretation of the data; (b) drafting the article or revising it critically for important intellectual content; and (c) approval of the final version.

This manuscript has not been submitted to, nor is under review at, another journal or other publishing venue.

The authors have no affiliation with any organization with a direct or indirect financial interest in the subject matter discussed in the manuscript

Acknowledgements

This work is part of the PGC2018-094341-B-I00 research project funded by the Ministry of Science and Innovation/FEDER. The authors would like to thank also the Confederación Hidrográfica del Ebro staff for their availability and for the supply of the data. Additionally, Mario Morales-Hernández was partially supported by the U.S. Air Force Numerical Weather Modeling Program.

References

[1] The human cost of weather related disasters (1995-2015), Centre for Research on the Epidemiology of Disasters (CRED) Brussels (2015) 1–30.

[2] D. P. Hoyois, D. Guha-Shapir, Three decades of floods in Europe: a preliminary analysis of EMDAT data, Working paper. Brussels, CRED.

[3] J. Olcina, D. Sauri, M. Hernández, A. Ribas, Flood policy in Spain: a review for the period 1983-2013, Disaster, prevention and management 25 (1) (2016) 41–58.

[4] J. Thielen, J. Bartholmes, M.-H. Ramos, A. de Roo, The european flood alert system. Part 1: Concept and development, Hydrology and Earth System Sciences 13 (2) (2009) 125–140.

[5] J. M. V. D. Knijff, J. Younis, A. P. J. D. Roo, LISFLOOD: A GISbased distributed model for river basin scale water balance and flood simulation, International Journal of Geographical Information Science 24 (2) (2010) 189–212.

[6] J. Teng, A. Jakeman, J. Vaze, B. Croke, D. Dutta, S. Kim, Flood inundation modelling: A review of methods, recent advances and uncertainty analysis, Environmental Modelling & Software 90 (2017) 201 – 216.

[7] R. Hu, F. Fang, P. Salinas, C. Pain, N. Sto.Domingo, O. Mark, Numerical simulation of floods from multiple sources using an adaptive anisotropic unstructured mesh method, Advances in Water Resources 123 (2019) 173 – 188.

[8] S. J. Noh, J.-H. Lee, S. Lee, K. Kawaike, D.-J. Seo, Hyper-resolution 1D-2D urban flood modelling using LIDAR data and hybrid parallelization, Environmental Modelling & Software 103 (2018) 131 – 145.

[9] J. Murillo, P. Garcia-Navarro, J. Burguete, P. Brufau, The influence of source terms on stability, accuracy and conservation in two-dimensional shallow flow simulation using triangular finite volumes, International Journal for Numerical Methods in Fluids 54 (5) (2007) 543–590.

[10] S. F. Bradford, B. F. Sanders, Finite-volume model for shallow-water flooding of arbitrary topography, Journal of Hydraulic Engineering 128 (3) (2002) 289–298.

[11] P. Brufau, P. García-Navarro, M. Vázquez-Cendón, Zero mass error using unsteady wetting-drying conditions in shallow flows over dry irregular topography, International journal for numerical methods in fluids 45 (10) (2004) 1047–1082.

[12] R. Briganti, N. Dodd, Shoreline motion in nonlinear shallow water coastal models, Coastal Engineering 56 (5) (2009) 495 – 505.

[13] M. E. Hubbard, N. Dodd, A 2D numerical model of wave run-up and overtopping, Coastal Engineering 47 (1) (2002) 1 – 26.

[14] M. Morales-Hernández, M. Hubbard, P. Garcia-Navarro, A 2D extension of a large time step explicit scheme (CFL > 1) for unsteady problems with wet/dry boundaries, Journal of Computational Physics 263 (2014) 303–327.

[15] M. S. Horritt, P. D. Bates, Evaluation of 1D and 2D numerical models for predicting river flood inundation, Journal of Hydrology 268 (2002) 89–99.

[16] P. Costabile, F. Macchione, L. Matale, G. Petaccia, Flood mapping using LIDAR DEM. limitations of the 1-D modeling highlighted by the 2-D approach, Natural Hazards 77 (2) (2015) 181–204.

[17] G. Petaccia, L. Natale, F. Savi, M. Velickovic, Y. Zech, S. Soares-Frazo, Flood wave propagation in steep mountain rivers, Journal of Hydroinformatics 15 (1) (2012) 120–137.

[18] J. Murillo, P. Garcia-Navarro, Accurate numerical modeling of 1d flow in channels with arbitrary shape. application of the energy balanced property, Journal of Computational Physics 260 (2014) 222–248.

[19] E. Bladé, J. Gómez-Valentín, J. Dolz, J. L. Aragón-Hernández, G. Corestein, M. Sánchez-Juny, Integration of 1D and 2D finite volume schemes for computations of water flow in natural channels, Advances in Water Resources 42 (2012) 17–29.

[20] M. Morales-Hernández, P. García-Navarro, J. Burguete, P. Brufau, A conservative strategy to couple 1D and 2D models for shallow water flow simulation, Computers & Fluids 81 (2013) 26 – 44.

[21] M. Morales-Hernández, A. Lacasta, J. Murillo, P. Brufau, P. García-Navarro, A Riemann coupled edge (RCE) 1D2D finite volume inundation and solute transport model, Environmental Earth Sciences 74 (11) (2015) 7319–7335.

[22] O. A. R. Board, OpenMP Application Programming Interface, High-Performance Computing Center Stuttgart, 2015.

[23] Message-Passing Interface Standard Forum, MPI: A Message-Passing Interface Standard, Version 3.1, 2015.

[24] NVIDIA, NVIDIA CUDA Programming Guide Version 3.0, http://developer.nvidia.com/object/cuda_3_0_downloads.html, 2010.

[25] M. J. Castro, S. Ortega, M. de la Asunción, J. M. Mantas, J. M. Gallardo, GPU computing for shallow water flow simulation based on finite volume schemes, Comptes Rendus Mécanique 339 (2-3) (2011) 165–184.

[26] A. Lacasta, M. Morales-Hernández, J. Murillo, P. García-Navarro, An optimized GPU implementation of a 2D free surface simulation model on unstructured meshes, Advances in Engineering Software 78 (2014) 1–15.

[27] A. Brodtkorb, M. Saetra, M. Altikanar, Efficient shallow water simulations on GPUs: implementation, visualization, verification and validation, Computers and Fluids 55 (2012) 1–12.

[28] Z. Shang, High performance computing for flood simulation using Telemac based on hybrid MPI/OpenMP parallel programming, International Journal of Modeling, Simulation, and Scientific Computing 05 (04) (2014) 1472001.

[29] S. Zhang, Z. Xia, R. Yuan, X. Jiang, Parallel computation of a dam-break flow model using OpenMP on a multi-core computer, Journal of Hydrology 512 (2014) 126 – 133.

[30] R. Vacondio, F. Aureli, P. Mignosa, A. Dal Pal, Simulation of the January 2014 flood on the Secchia River using a fast and high-resolution 2D parallel shallow-water numerical scheme, Natural Hazards 80 (1) (2016) 103–125.

[31] L. Smith, Q. Liang, Towards a generalized GPU/CPU shallow flow modelling tool, Computers and Fluids 88 (2013) 334–343.

[32] O. Orlando García-Feal, J. González-Cao, M. Gómez-Gesteira, L. Cea, J. M. Domínguez, A. Formella, An accelerated tool for flood modelling based on iber, Water 10 (10).

[33] G. Petaccia, F. Leporati, E. Torti, Openmp and cuda simulations of Sella Zerbino dam break on unstructured grids, Computational Geosciences 20 (5) (2016) 1123–1132.

[34] J. Murillo, P. García-Navarro, Wave riemann description of friction terms in unsteady shallow flows: Application to water and mud/debris floods, Journal of Computational Physics 231 (2012) 1963–2001.

[35] J. Murillo, P. García-Navarro, Weak solutions for partial differential equations with source terms: Application to the shallow water equations, Journal of Computational Physics 229 (0) (2010) 4237–4368.

[36] J. E. Nash, J. V. Sutcliffe, River flow forecasting through conceptual models. part I - A discussion of principles, Journal of Hydrology 10 (1970) 282–290.

[37] A. Ritter, R. Munoz-Carpena, Performance evaluation of hydrological models: Statistical significance for reducing subjectivity in goodness-of-fit assessments, Journal of Hydrology 480 (2013) 33–45.

[38] G. Arcement, V. Schneider, Guide for Selecting Manning's Roughness Coefficients for Natural Channels and Flood Plains, no. 2339 in U.S. Geological Survey. Water-supply paper, 1984.

[39] V. T. Chow, Open-channel hydraulics, McGraw-Hill Civil Engineering Series, McGraw-Hill, 1959.

[40] F. Palmeri, F. Silván, I. Prieto, M. Balboni, I. García-Mijangos, Manual de técnicas de ingeniería naturalística en ámbito fluvial, Departamento de Ordenación del territorio y Medio Ambiente, Gobierno del País Vasco, España, 2002.

[41] B. W. P. Staff, G. Witheridge, B. Q. C. Council, Natural Channel Design Guidelines. Appendix C, Technical Document, Brisbane City Council, 2000.

[42] F. J. Leveque, Finite Volume Methods for Hyperbolic Problems, Cambridge University Press, New York, 2002.

[43] R. LeVeque, Numerical Methods for Conservation Laws, Lectures in Mathematics. ETH Zrich, Birkhuser Basel, 1992.

[44] E. F. Toro, Riemann Solvers and Numerical Methods for Fluid Dynamics, Springer, Berlin, 1997.

[45] A. Harten, J. M. Hyman, Self adjusting grid methods for one-dimensional hyperbolic conservation laws, Journal of Computational Physics 50 (1983) 235–269.

[46] M. Feldman, NVIDIA GPU accelerators get a direct pipe to big data, The Next Platform.

[47] Concerted action on dam-break modelling, Office for Official Publications of the European Comunities. Proceedings of the CADAM meeting Wallingford, UK, 1998.

[48] S. Soares-Frazao, The Toce River test case: numerical results analysis, Proceeding of the 3rd CADAM workshop. Milan, Italy.

[49] D. Caviedes-Voullième, M. Morales-Hernández, I. López-Marijuan, P. García-Navarro, Reconstruction of 2D river beds by appropriate interpolation of 1D cross-sectional information for flood simulation, Environmental Modelling & Software 61 (0) (2014) 206 – 228.

[50] D. Caviedes-Voullième, P. García-Navarro, J. Murillo, Influence of mesh structure on 2D full shallow water equations and SCS curve number simulation of rainfall/runoff events, Journal of hydrology 448 (2012) 39–59.

[51] A. Bomers, R. M. J. Schielen, S. J. M. H. Hulscher, The influence of grid shape and grid size on hydraulic river modelling performance, Environmental Fluid Mechanics.

[52] J. M. Hoch, R. van Beek, H. C. Winsemius, M. F. Bierkens, Benchmarking flexible meshes and regular grids for large-scale fluvial inundation modelling, Advances in Water Resources 121 (2018) 350 – 360.

[53] M. Gonzalez-Sanchís, J. Murillo, B. Latorre, F. Comín, P. García-Navarro, Transient two-dimensional simulation of real flood events in a mediterranean floodplain, Journal of Hydraulic Engineering-ASCE 138 (7) (2012) 629–641.