water

Article

A GPU-accelerated Shallow-Water Scheme for Surface Runoff Simulations

Francesca Aureli *, Federico Prost, Renato Vacondio, Susanna Dazzi and Alessia Ferrari

Department of Engineering and Architecture, University of Parma, Parco Area delle Scienze 181/A, 43124 Parma, Italy; federico.prost@unipr.it (F.P.); renato.vacondio@unipr.it (R.V.); susanna.dazzi@unipr.it (S.D.); alessia.ferrari@unipr.it (A.F.)

* Correspondence: francesca.aureli@unipr.it

Received: 10 January 2020; Accepted: 23 February 2020; Published: 26 February 2020

Abstract: The capability of a GPU-parallelized numerical scheme to perform accurate and fast simulations of surface runoff in watersheds, exploiting high-resolution digital elevation models (DEMs), was investigated. The numerical computations were carried out by using an explicit finite volume numerical scheme and adopting a recent type of grid called Block-Uniform Quadtree (BUQ), capable of exploiting the computational power of GPUs with negligible overhead. Moreover, stability and zero mass error were ensured, even in the presence of very shallow water depth, by introducing a proper reconstruction of conserved variables at cell interfaces, a specific formulation of the slope source term and an explicit discretization of the friction source term. The 2D shallow water model was tested against two different literature tests and a real event that recently occurred in Italy for which field data is available. The influence of the spatial resolution adopted in different portions of the domain was also investigated for the last test. The achieved low ratio of simulation to physical times, in some cases less than 1:20, opens new perspectives for flood management strategies. Based on the result of such models, emergency plans can be designed in order to achieve a significant reduction in the economic losses generated by flood events.

Keywords: surface runoff; shallow water equations; GPU-parallel scheme; Block-Uniform Quadtree grid (BUQ)

1. Introduction

The numerical distributed simulation of overland and open-channel flows has been studied since the early Seventies ([1–3]), and is being increasingly used to better model and understand the transient flow dynamics of flash floods [4–13]. These flash floods are generally characterized by rapidly varying overland flows as a consequence of fast and complex watershed response to intense rainfall. In a perspective of continuous increase in flash flood frequencies and damage, also owing to climate change, flooding events in Europe rank highly among natural hazards in terms of people involved and casualties, with a huge economic impact of losses characterized by a mean value of about 5.8 billion euros every year in the period 2013–2017 [14]. In addition, a growing number of people are facing high levels of flood risk due to increased urbanization in flood-prone areas. The development and application of tools capable of providing accurate and fast predictions for flood management are hence crucial to obtain a reduction in the damage caused by floods. In the past, lumped hydrological models or simplified hydrodynamic models were often adopted to predict these events at the catchment scale [15–18]. Unfortunately, simplified approaches are generally not capable of describing the rapid watershed responses and complex surface flow processes, and accurately predicting local depth and velocities. Moreover, the reduced representation of the physics of the phenomena can lead to significant dependence on model parameters. Hydrologic models for floods were in the past characterized by increasing levels of complexity starting from lumped models, which treat the whole system as a black box with input and output without accounting for detailed spatial variability of the phenomenon [19], to distributed models, capable of representing the spatial distribution of flows and hydrometeorological inputs over the watershed and better representing the physics of the phenomenon [20–22]. The latter models can nowadays obviously benefit from the new remote-sensing technologies that enable distributed knowledge of Earth surface geometry and forcing inputs [23].

In recent years, several analyses have been carried out to demonstrate that 2D shallow-water distributed numerical models can be proficiently applied to the simulation of flood events at the watershed scale also with the aim of designing measures to prevent flood damage [4–6,8,10,23–25]. As a whole, 2D hydrodynamic models solving the shallow-water equations (SWEs) are capable of accurately simulating the dynamics of overland flows, are much less dependent on parametrization, and can definitely provide reliable predictions of rainfall-runoff processes and of the consequent flood propagation. In particular, the Godunov-type shock capturing finite volume methods prove to be suitable to solve complex unsteady flow problems, even in the presence of rainfall inputs, due to the fact that they guarantee the conservation of flow variables, and they can accurately describe transcritical flows. Different numerical schemes based on simplified versions of the SWEs were developed in the past, with the aim of reducing the computational time while preventing instabilities in the numerical models. The most popular was the kinematic wave approximation (KWA) where the pressure and inertial terms are neglected in the momentum equation. The diffusive wave approximation (DWA) was later introduced; the pressure term in the momentum equation is preserved, while the inertial terms are neglected, thus degrading the starting hyperbolic model to a parabolic one. This simplification is traditionally accepted based on the assumption that flooding over plain areas is characterized by slow velocity propagation. Thanks to the simplification of the governing equations, both the KWA and the DWA could reduce the computational times, although some authors have pointed out that the zero inertia approach could be computationally even more expensive than complete dynamic solvers due to a highly constraining stability requirement [5,12,13,26–35]. Unfortunately, in areas characterized by the presence of roads, railways or channel embankments, or in urban environments, where the flooding dynamics can be strongly influenced by streets and buildings and the main flow variables change rapidly in space and time, the flood inundation simulations performed with simplified models can give rise to unacceptable approximations [13,36,37]. Conversely, a model based on the complete version of the SWEs can be adopted in any situation and with a high degree of detail (provided that the domain geometry is sufficiently well known) [38]. As many authors state, in fact, strongly hyperbolic flows, such as those characterized by moving shock waves, cannot be accurately solved by applying the zero inertia approach [13,29,30,34,39].

Many problems can arise if overland flow phenomena are simulated through a complete 2D hydrodynamic model, due to the presence of solution discontinuities and of very thin shallow flows, especially when the domain slope is steep. The numerical scheme adopted must be robust and capable of preserving the steady state condition of a fluid at rest, and of guaranteeing the positivity of water depth during computations. This is extremely important in situations in which some portions of the computational domain are dry or almost dry. Moreover, mass must obviously be conserved.

Unfortunately, these models are computationally very demanding if applied to simulate long-duration events while using a high-resolution computational grid [40]. However, the choice of a coarse resolution at the catchment scale can seriously compromise the benefit of the adoption of complete 2D hydrodynamic models. Computational efficiency is clearly of the highest importance in the numerical simulation of flash floods at the watershed scale for disaster prediction. If the most relevant geometric features of the terrain (including urban areas) in river basins of several hundreds of square kilometers have to be accurately represented, several millions of computational cells have to be used. Therefore, only the adoption of a high-performance computing technology can allow a drastic reduction in model runtime, thus enabling for instance practical warning services in the case of phenomena characterized by very rapid evolution (in the order of a few hours) [41–47].

Parallel numerical schemes based on message passing interface (MPI) or graphics processing units (GPU) techniques have been recently developed, thereby allowing otherwise prohibitive simulations thanks to the high ratios between physical and computational times [48]. However, most of the GPU-accelerated models are based on uniform Cartesian grids, characterized by two main limitations: (i) a unique resolution must be adopted in the whole domain, and (ii) the domain must be rectangular. The adoption of non-uniform grids in GPU-accelerated models has recently gained attention in the literature [43,49–56], since it allows the use of high resolution only in the portions of the domain where it is needed.

The objective of this paper is to present Parflood Rain, the evolution of a GPU-parallel numerical model for the solution of the 2D SWEs [48,51,57], for the simulation of rainfall-runoff processes in a watershed. The new version of the code required the introduction of a new source term related to rainfall. The model is characterized by a finite volume explicit discretization technique, which is well balanced, second-order accurate, and based on positive depth reconstruction. Thanks to the structure of the algorithm, the scheme is also numerically stable for overland flow. Many issues related to stability had to be addressed due to the thin water layer flowing across the domain. Moreover, the model is based on a recently developed type of grid called Block-Uniform Quadtree (BUQ), designed to exploit the computational power of a GPU and allow the enforcement of the desired resolution in different regions of the computational domain [36]. Since this is the first application of the BUQ grid at the catchment-area scale, we have conducted an in-depth analysis aimed at identifying the appropriate resolution to be adopted at the watershed sides in order to preserve an accurate description of the flow dynamics. A remarkable reduction in computational times and allocated memory was achieved compared to Cartesian high-resolution grids.

The paper is organized as follows: the explicit finite volume numerical scheme for the SWEs is described in Section 2, and the main features of the Computed Unified Data Architecture (CUDA®) implementation and of BUQ grids are illustrated. In Section 3, the accuracy of the numerical model is verified against different challenging synthetic test cases. In Section 4, the capability of the numerical scheme to efficiently simulate real flooding events is assessed by reproducing a real test case. Appendix A contains details of the numerical implementation of source terms and rainfall excess.

2. Numerical Model

2.1. Numerical Scheme

The numerical model, explained in detail in [48,51], solves the 2D-SWE through an explicit finite volume scheme that in integral form [58] can be written as:

$$\frac{d}{dt}\int_A \mathbf{U}\,dA + \int_C \mathbf{H}\cdot\mathbf{n}\,dC = \int_A \left(\mathbf{S_b} + \mathbf{S}_f + \mathbf{S}_r\right)dA\,, \tag{1}$$

where A is the area of integration element, C the element boundary, **n** the outward unit vector normal to C, **U** the vector of the conserved variables, **H** = (**F**, **G**) the tensor of fluxes in the x and y directions, $\mathbf{S}_b$, $\mathbf{S}_f$ and $\mathbf{S}_r$ the source terms representing the bed slope, the frictional effect and the rainfall rate, respectively. A modified form [59] of SWEs is adopted here, and the vector terms are:

$$\mathbf{U} = \begin{bmatrix} \eta \\ uh \\ vh \end{bmatrix} \qquad \mathbf{F} = \begin{bmatrix} uh \\ u^2h + \frac{1}{2}g(\eta^2 - 2\eta z) \\ uvh \end{bmatrix} \qquad \mathbf{G} = \begin{bmatrix} vh \\ uvh \\ v^2h + \frac{1}{2}g(\eta^2 - 2\eta z) \end{bmatrix}$$
$$\mathbf{S_b} = \begin{bmatrix} 0 \\ -g\eta\frac{\partial z}{\partial x} \\ -g\eta\frac{\partial z}{\partial y} \end{bmatrix} \qquad \mathbf{S}_f = \begin{bmatrix} 0 \\ -gh\frac{n_f^2 u\sqrt{u^2+v^2}}{h^{4/3}} \\ -gh\frac{n_f^2 v\sqrt{u^2+v^2}}{h^{4/3}} \end{bmatrix} \qquad \mathbf{S}_r = \begin{bmatrix} r - f \\ 0 \\ 0 \end{bmatrix} \tag{2}$$

in which η is the free surface elevation, h the flow depth, u and v are the velocity components in the x and y directions. The terms uh and vh identify the unit width discharges in the two coordinates directions, respectively (also addressed as qx and qy for the sake of simplicity in notation), g is the gravitational acceleration, z is the bed elevation ($\eta = z + h$), r and f are the rainfall and infiltration rate, respectively.

The conserved variables are updated at each time step by adopting the second order Runge–Kutta method, which ensures the second order of accuracy in time:

$$\mathbf{U}_{i,j}^{n+1} = \mathbf{U}_{i,j}^{n} + \frac{1}{2}\Delta t^{n}\left[\mathbf{D}_i\left(\mathbf{U}_{i,j}^{n}\right) + \mathbf{D}_i\left(\mathbf{U}_{i,j}^{n+1/2}\right)\right]. \quad (3)$$

In Equation (3), n represents the time level, i and j the cell positions in the x and y directions, and Δt^n is the time step calculated accordingly to the Courant–Friedrichs–Lewy condition [58]. The term $\mathbf{U}_{i,j}^{n+1/2}$ is obtained as:

$$\mathbf{U}_{i,j}^{n+1/2} = \mathbf{U}_{i,j}^{n} + \Delta t^{n}\mathbf{D}_i\left(\mathbf{U}_{i,j}^{n}\right) \ , \quad (4)$$

where the operator $\mathbf{D}_i(\mathbf{U}_{i,j}^n)$ is defined as:

$$\mathbf{D}_i\left(\mathbf{U}_{i,j}^{n}\right) = -\frac{\mathbf{F}_{i+\frac{1}{2},j} - \mathbf{F}_{i-\frac{1}{2},j}}{\Delta x} - \frac{\mathbf{G}_{i,j+\frac{1}{2}} - \mathbf{G}_{i,j-\frac{1}{2}}}{\Delta y} + \mathbf{S}_r + \mathbf{S}_b + \mathbf{S}_f \ , \quad (5)$$

in which the numerical fluxes **F** and **G** at the intercells i ± ½ and j ± ½ in the x and y direction respectively, are evaluated by using the states at the cell faces as input in the Harten, Lax and van Leer approximate Riemann solver with the contact wave restored (HLLC). The HLLC scheme, which in many situations behaves like the exact Riemann solver, does not require an iterative scheme, is positive definite and generally computationally more efficient than other well-known solvers, is in fact accurate and robust, and preserves the entropy-satisfaction property of the HLL scheme [58,60]. To obtain the Riemann states, a surface reconstruction method (SRM [10]) with hydrostatic reconstruction [61] is used. The flux and the bed and friction slope source term computations are performed in the present approach by adopting the variables obtained from SRM and local bed modification treatment in order to overcome the numerical instabilities induced by the presence of very small depths related to runoff processes. Considering two adjacent cells in the x direction, i and i + 1, the SRM-reconstructed water surface elevations on the left- and right-hand sides of their common interface are given by:

$$\begin{cases} \eta_{i+\frac{1}{2},j}^{L} = \eta_{i,j} + max\left[0, \min(z_{i+1,j} - z_{i,j} - \delta z, \qquad \eta_{i+1,j} - \eta_{i,j}))\right] \\ \eta_{i+\frac{1}{2},j}^{R} = \eta_{i+1,j} + max\left[0, \min(z_{i,j} - z_{i+1,j} - \delta z, \qquad \eta_{i,j} - \eta_{i+1,j}))\right] \end{cases}, \quad (6)$$

where δz is the difference between the bed elevation at the two sides of the cell interface, obtained through slope-limited interpolation (minmod Ψ) of the related cell center values:

$$\delta z = z_{i+\frac{1}{2},j}^{+} - z_{i+\frac{1}{2},j}^{-} = \left[z_{i+1,j} - \frac{1}{2}\Psi_{i+\frac{1}{2},j}^{+}\left(z_{i+1,j} - z_{i,j}\right)\right] - \left[z_{i,j} + \frac{1}{2}\Psi_{i+\frac{1}{2},j}^{-}\left(z_{i,j} - z_{i+1,j}\right)\right]. \quad (7)$$

The states of bed elevation at the common interface can then be obtained as follows:

$$\begin{cases} z_{i+\frac{1}{2},j}^{L} = \eta_{i+\frac{1}{2},j}^{L} - h_{i,j} \\ z_{i+\frac{1}{2},j}^{R} = \eta_{i+\frac{1}{2},j}^{R} - h_{i+1,j} \end{cases}. \quad (8)$$

To ensure the positivity of the water depths, a unique bed elevation is defined for each cell interface [61] and the water depths are corrected once again to ensure their non-negativity [62]:

$$\begin{cases} h_{i+\frac{1}{2},j}^{L} = \max\left[0, \eta_{i+\frac{1}{2},j}^{L} - \max\left(z_{i+\frac{1}{2},j}^{L}, z_{i+\frac{1}{2},j}^{R}\right)\right] \\ h_{i+\frac{1}{2},j}^{R} = \max\left[0, \eta_{i+\frac{1}{2},j}^{R} - \max\left(z_{i+\frac{1}{2},j}^{L}, z_{i+\frac{1}{2},j}^{R}\right)\right] \end{cases}. \quad (9)$$

The Riemann states of the unit width discharges are then calculated:

$$\begin{cases} qx^L_{i+\frac{1}{2},j} = h^L_{i+\frac{1}{2},j} \cdot u_i \\ qx^R_{i+\frac{1}{2},j} = h^R_{i+\frac{1}{2},j} \cdot u_{i+1} \end{cases} \quad \begin{cases} qy^L_{i+\frac{1}{2},j} = h^L_{i+\frac{1}{2},j} \cdot v_i \\ qy^R_{i+\frac{1}{2},j} = h^R_{i+\frac{1}{2},j} \cdot v_{i+1} \end{cases} \text{ with } u_{i,j} = \frac{qx_{i,j}}{h_{i,j}} \text{ e } v_{i,j} = \frac{qy_{i,j}}{h_{i,j}} \ . \tag{10}$$

In Equation (6)–(10) the variables without superscript (η, h, u, v) are defined at the cell center.

The details of the numerical implementation of bed and friction slope source terms are reported in Appendix A for the sake of space.

2.2. Evaluation of Rainfall Excess

In the numerical model presented here the rainfall source term $\mathbf{S}_r$ represents the inflow due to the net rainfall (Equation (1)). This term, variable in both space and time, can be obtained as the difference between rainfall r and infiltration rate f (Equation (2)).

Several possibilities are available in the code to introduce the input due to the rainfall in order to perform a distributed rainfall simulation over the domain. It is possible to set a unique value of rainfall intensity r (mm·h^{-1}) at each cell or rain time series for predetermined areas (Thiessen polygons). Another way is to exploit the observations at the rain gauges present in the domain by computing the precipitation at each cell (in time) through the inverse distance squared approximation:

$$r_i = \frac{\sum_{g=1}^{N} \frac{r_g}{{d_{g,i,j}}^2}}{\sum_{g=1}^{N} \frac{1}{{d_{g,i,j}}^2}}, \tag{11}$$

where $r_{i,j}$ is the rainfall intensity at cell i, j, r_g is the rainfall intensity measured by the gth rainfall gauge, $d_{g,i,j}$ is the distance from cell i to the gth rainfall gauge and N is the total number of rainfall gauges. Furthermore, a series of maps (e.g., radar maps) containing the amount of precipitation in time (with constant or variable time spacing*)* can be provided as input to be transformed to the rainfall rate. In the present model, the rainfall excess is evaluated through the application of the SCS Method, which allows the evaluation of the depth (mm) of the excess precipitation, P_e, through a simple empirical formulation [63]. More details of the numerical implementation of the SCS approach are reported in Appendix A.

2.3. Graphics Processing Unit (GPU) Implementation of Shallow-Water Equation (SWE)

The numerical scheme described in Section 2.1 is implemented through the CUDA ® in order to overcome the high computational cost of the explicit FV scheme by exploiting NVIDIA™ GPUs. The main aspects of GPU implementation are briefly recalled in this section, for details see [48,57].

In CUDA, the basic work unit is represented by a *thread*, while many threads are grouped into a *block* [64]. In the present model, each thread corresponds to a computational cell used to discretize the physical domain. The CUDA program controls two resources, namely the *host* (CPU) and the *device* (GPU), including the data exchange between them. The former performs pre- and post-processing and controls the simulation's time advancement, while the latter executes the pieces of code (*kernels*) with the instructions for the actual computations, which are performed in parallel over blocks. The update of the conserved variables from time-step n to n + 1 is executed by a sequence of 10 tasks (Figure 1), each corresponding to a distinct CUDA kernel. Compared to the previous model [48], the implementation of Parflood Rain features an additional kernel for processing rainfall maps, and modifies some of the original kernels to include the rainfall source term. Moreover the conserved variables are here reconstructed through the SRM approach.

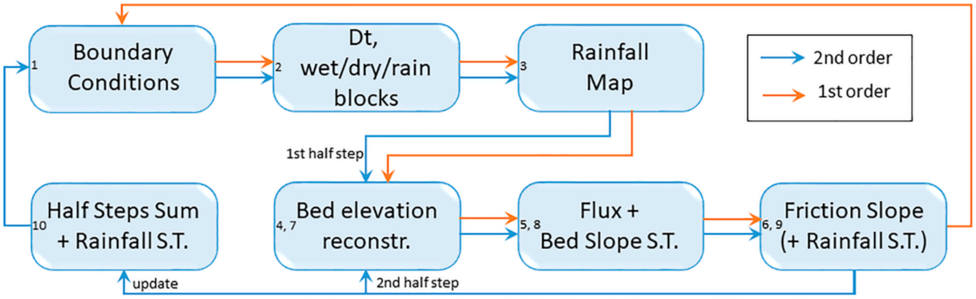


Figure 1. Iteration flux diagram. Each task is handled by a graphics processing unit (GPU) kernel.

The first task is only performed if open boundary conditions are present. In this case, some conserved quantities must be imposed in boundary cells [48]. In the second task, the time-step is calculated according to the CFL condition, and the blocks are reorganized in order to exclude dry blocks from the computations and improve efficiency [48]. However, if at least one cell of a dry block falls in the rainfall area at the current time step, the "rain" block is activated and processed like a wet block. Then, during the third task, a dedicated kernel computes the rainfall intensity for each cell in the "rain" blocks. Tasks 4, 5, and 6 correspond to the first half step of the second order scheme, and include the bed elevation reconstruction, the flux and bed slope source term evaluation, and the computation of the friction slope source term. The updating process continues with the second half step (tasks 7, 8, and 9). Finally, during task 10, the solution is updated according to Equation (3), and the rainfall source term is also added, before a new iteration begins. Conversely, only tasks 4, 5 and 6 are performed for the first-order scheme: the rainfall excess computation and the updating of the conserved variables according to Equation (5) are performed during task 6; the iteration process then starts again immediately from task 1.

2.4. Block-Uniform Quadtree (BUQ) Grids

BUQ grids were first presented in [51] for flood modelling, and this is the first time BUQ grids have been applied at the watershed scale. This circumstance requires an in-depth analysis to determine the minimum resolution (largest cell size) adequate to preserve an accurate description of the flow dynamics. The main characteristics of this type of grid are briefly reported here; for a more detailed description the reader should refer to reference [51].

When the domain is discretized with a Cartesian grid, the cells are organized into regular blocks containing $M \times M$ cells (in this work, $M = 16$), and stored in the memory as a two-dimensional array characterized by a straightforward correspondence between physical localization of a cell and its position in the memory. In this case, retrieving information from cell neighbors is very simple. To overcome the limitation of a Cartesian grid, a new type of structured non-uniform grids (BUQ) was designed to allow a multi-resolution discretization of the domain while maintaining efficiency on GPU. In BUQ grids, each block still contains $M \times M$ cells, but the cell size can differ for different blocks. In particular, cell size can vary between the highest level of resolution (level 1, Δ_1) and the coarsest one (level N, $\Delta_N = 2^{N-1} \Delta_1$), i.e. each block contains cells with size equal to a power-of-two multiple of the minimum size in the domain. The most important criterion for grid generation is the fact that the variation in the resolution level between neighbor blocks cannot exceed one [51]. This idea is similar to quad-tree partitioning [65], the main difference being the fact that single cells are replaced by blocks of cells in the spatial organization of the quad-tree nodes. As an example, Figure 2 shows a detail of a BUQ grid. Starting from a seeding point where the maximum resolution is requested, four different resolutions are originated during pre-processing. The regular blocks (containing $M \times M$ cells each) are shown at the different levels of resolution.

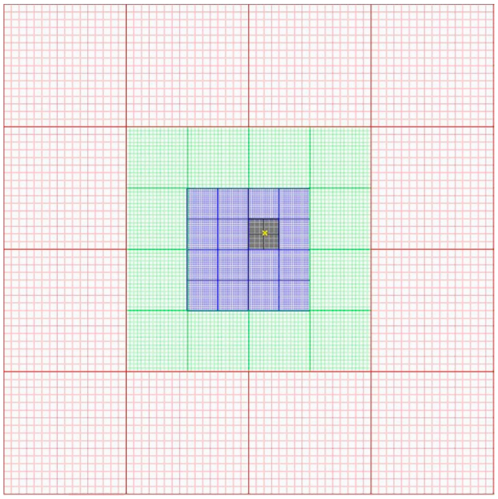


Figure 2. Particular of a Block-Uniform Quadtree (BUQ) grid, with blocks, cells and a maximum resolution seeding point (yellow cross).

The correspondence between physical space and memory, typical of Cartesian grids, cannot be maintained for BUQ grids, since all blocks require the same memory size, but blocks with different resolution have different physical size. For this reason, blocks are stored in memory in a non-preordained way, and additional data structures to efficiently access information on neighbor blocks are introduced. During computations, these arrays will only be used by threads corresponding to cells lying on the border of a block in order to retrieve the values of conserved variables in the neighbor cell that belongs to a different block. The natural neighbor interpolation procedure [66] is adopted to reconstruct the conserved variables in adjacent cells characterized by a different spatial resolution. The C-property in these cells is guaranteed thanks to proper flux computation and bed slope source term treatment, detailed in [51].

This kind of architecture was preferred with respect to other possible procedures for the partitioning of the physical domain (e.g., adaptive mesh refinement (AMR) methods). In the multiresolution approach followed by the writers, the location of the maximum resolution of the computational grid is in fact decided a priori, once and for all, on the basis of the domain's characteristics and of the presence of geometrical elements (such as embankments, built up areas, etc.) capable of sensibly affecting the flow dynamics. On the contrary, the implementation of AMR methods on GPUS poses several challenges mainly related to non-coalesced memory access that leads to considerable overhead.

3. Experimental and Synthetic Rainfall-Runoff Tests

Two synthetic rainfall-runoff experimental cases were selected to demonstrate the applicability, accuracy, and stability of the proposed scheme. In the absence of analytic reference solutions of the complete hydrodynamic model, the computed results were compared with the solutions of the kinematic wave equation or with experimental data if available.

3.1. Test Case 1: One-Dimensional Three Planes Cascade Rainfall-Runoff Case

In this laboratory experiment realized at Kyoto University in 1955 [67] and often adopted for validation purposes [12,25,30,68–71], the flow phenomenon evolved on a three planes cascade of width equal to 19.6 cm having slope $S_{x1} = 0.02$, $S_{x2} = 0.015$ and $S_{x3} = 0.01$ and length $L = 8$ m each (Figure 3). The three planes were subject to different rainfall intensities equal to $r_1 = 3888$ mm·h^{-1}, $r_2 = 2296.8$ mm·h^{-1} and $r_3 = 2880$ mm·h^{-1}. The Manning coefficient $n = 0.009$ m$^{-1/3}$s was suggested by the author [67], even though other researchers in the literature [68] considered a slightly different value of $n = 0.01$ m$^{-1/3}$s to achieve better results in the numerical simulations. This accounts for possible uncertainties in the evaluation of the planes roughness coefficient at the time of the experiments. Both values of the Manning coefficient were considered here, and two different runs of the numerical model were performed with $n = 0.009$ m$^{-1/3}$s (Parflood Rain, test I) and $n = 0.01$ m$^{-1/3}$s (Parflood Rain, test II). In addition to the different values of the rainfall intensities above the three planes, different rainfall durations were also considered, namely $t = 10$ s (case A), $t = 20$ s (case B) and $t = 30$ s (case C). The experimental observations for the three test cases considered are compared in Figure 4 with the resulting hydrographs of unit discharge at the downstream boundary, obtained by the complete hydrodynamic model.

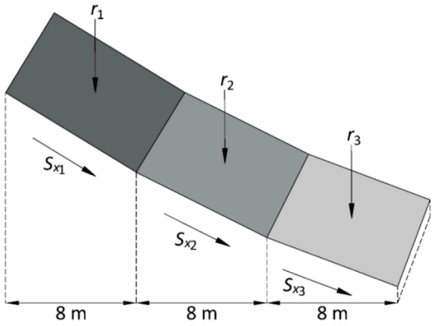
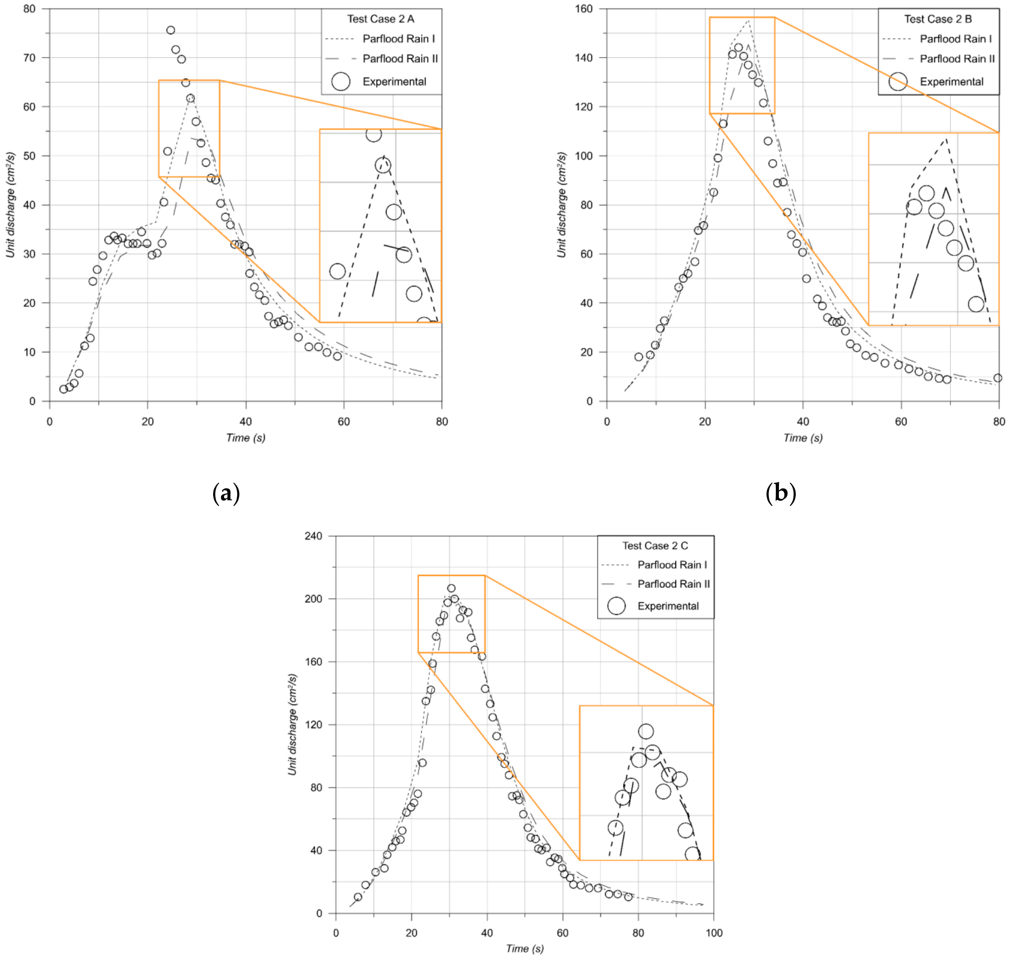


Figure 3. Domain schematization for the three planes cascade test case.



(**a**) (**b**)

(**c**)

Figure 4. Results obtained through the complete hydrodynamic model (tests I and II) compared to the experimental data ([67]) for different rainfall durations: (**a**) t = 10 s, (**b**) t = 20 s and (**c**) t = 30 s.

As a whole, the comparison shows a good agreement between numerical results and observations, especially for test cases B and C. Some high experimental values can be observed in test condition A (Figure 4a), (unit discharge higher than 70 m^{2}s^{-1}). However, the investigator had ascribed this strange behavior to the fact that during the experimental runs the lateral inflow rate q achieved at each reach of the flume was not completely uniform. If the three anomalous values are ignored, the comparison between numerical results and observations for case A is quite good. The numerical results related to the adoption of the two different roughness coefficients proposed in the literature (Parflood Rain, test I and test II) show the sensitivity of the numerical computations to the Manning's parameter adopted. Better results were achieved for case A with n = 0.009 m$^{-1/3}$s and for case B with n = 0.01 m$^{-1/3}$s.

3.2. Test Case 2: Two-Dimensional V-Shaped Rainfall-Runoff Test Case

The V-shaped rainfall-runoff test case has been widely adopted in the literature [3,5,10,12,24,27,38,44,72–74] for evaluation purposes of 2D models.

As shown in Figure 5, the V-shaped catchment is characterized by two hillsides with a 0.05 slope in the x-direction, flowing into a rectangular channel having a 0.02 slope in the y-direction. Each side is 1000 m long and 800 m wide, while the channel width is 20 m. The Manning roughness coefficients for the hillsides and the channel are 0.015 m$^{-1/3}$s and 0.15 m$^{-1/3}$s, respectively. The whole domain was subject to a uniform rainfall intensity equal to r = 10.8 mm·h^{-1} for a duration of 1.5 h. A free outflow condition was imposed at the channel outlet.

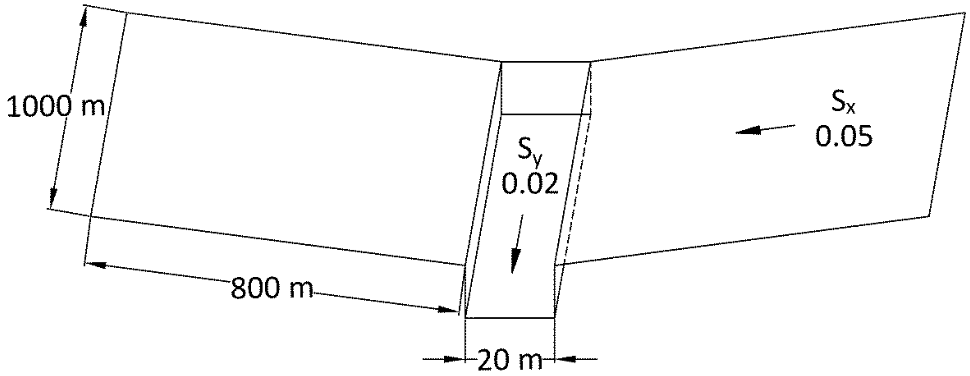


Figure 5. Domain schematization for the V-shaped test case.

The numerical computations were performed with three different grids: the first two were Cartesian, characterized by square cells of size 1 m × 1 m and 10 m × 10 m, while the third one was a BUQ multiresolution grid characterized by square cells of size 1, 2, 4 and 8 m. The maximum resolution was adopted in this last grid for the perimeter of the domain and for the channel, while the planes were described according to the BUQ grid partitioning with increasing cell size in order to reduce the number of computing cells (Figure 6). Indeed, if compared to the 1 m × 1 m Cartesian grid, the adoption of the multiresolution leads to a decrease in the total number of computational elements from over 1.6 × 10^6 to less than 0.3 × 10^6 (with a reduction of about 80%). As a result, the BUQ-based simulation performs 6.2 times faster than the Cartesian-based one (Table 1).

The computational efficiency of the simulations performed with the BUQ grids were verified by means of the compression rate C_R and speed-up S_U defined as:

$$C_R = \frac{N_C}{N}, \quad S_U = \frac{T_C}{T} \tag{12}$$

where N and T are the number of allocated cells and the computational time, respectively, for the simulation with the BUQ grid considered. N_C and T_C are the same quantities referred to an identical simulation run using a Cartesian grid with size equal to the maximum resolution adopted in the BUQ grid considered.

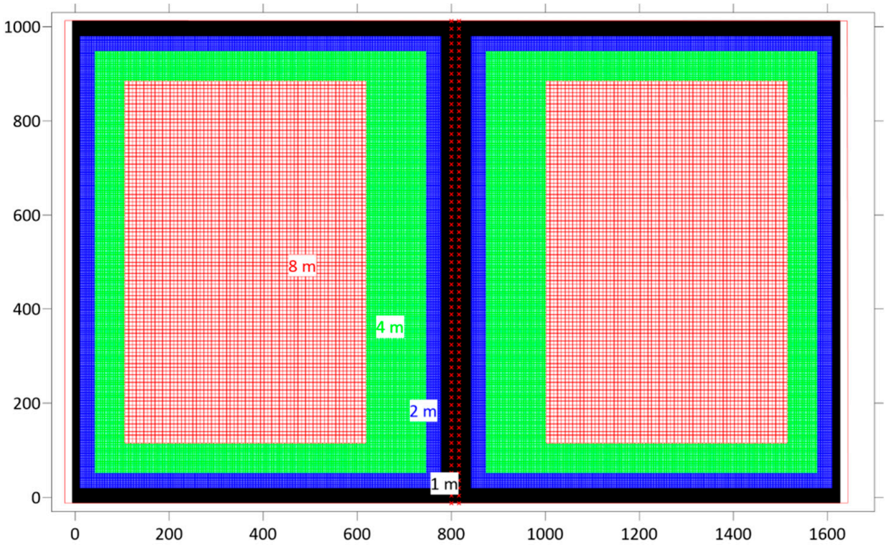


Figure 6. Multiresolution grid for the V-Shaped domain; 1 × 1 m cells in black, 2 × 2 m cells in blue, 4 × 4 m cells in green and 8 × 8 m cells in red.

Table 1. Characteristics of simulations, compression rates C_R and speed-up S_U for the BUQ grid used in the V-shaped test case.

Grid	*N* (Millions)	*T* (Hours)	C_R (–)	S_U (–)
Cartesian 1 m × 1 m	1.62	2.86	–	–
Cartesian 10 m × 10 m	0.016	0.042	–	–
BUQ Grid	0.29	0.46	5.59	6.21

The numerical discharge hydrographs at the end of each hillside and at the channel outlet obtained by using the three different grids are presented in Figure 7 and compared with the analytical solutions of the kinematic wave problem ([75–80]).

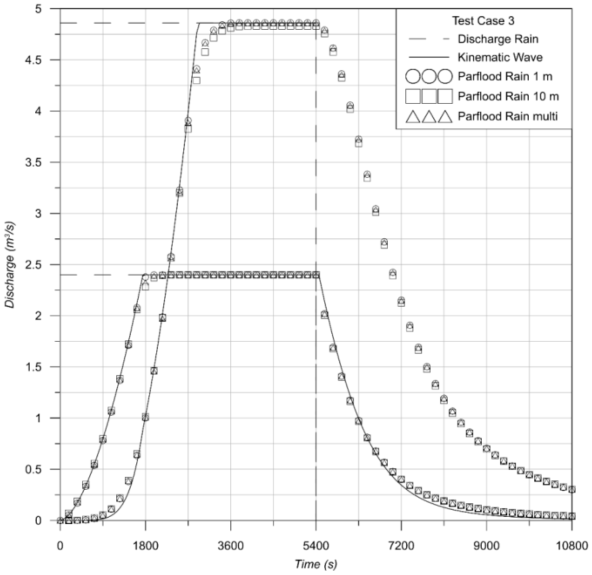


Figure 7. Hydrographs at the end of each hillside and at the channel outlet for different configurations of the Parflood Rain model compared to the kinematic flow wave solution.

The numerical hydrographs are in quite good agreement with the reference solution. In particular the simulation performed by adopting the BUQ grid provides results very close to those obtained by using the fine (1 m × 1 m) Cartesian grid. The simulation performed through the adoption of the 10 m × 10 m Cartesian grid is characterized by a very low number of elements and by a shorter computational time compared to the one achieved through the adoption of the BUQ grid. In this case, the grid size adopted is not adequate to describe the channel geometry in detail, and the resulting hydrograph at the channel outlet does not reproduce the reference solution accurately.

4. Nure Watershed Field Case

Finally the model was applied to simulate a large flood event that occurred in the Western Emilia-Romagna Apennine region, striking the Trebbia and Nure watersheds especially (Figure 8) during the night between 13 and 14 September 2015. Trebbia and Nure are tributaries of the Po river, the longest river in Italy, which flows for over 650 km in the homonymous valley in the North of the country. At the Farini outlet, the Nure watershed has a drainage area of about 208 km^2, with a 23 km long main channel, a 943 m mean altitude above sea level, and a 0.248 mean slope.

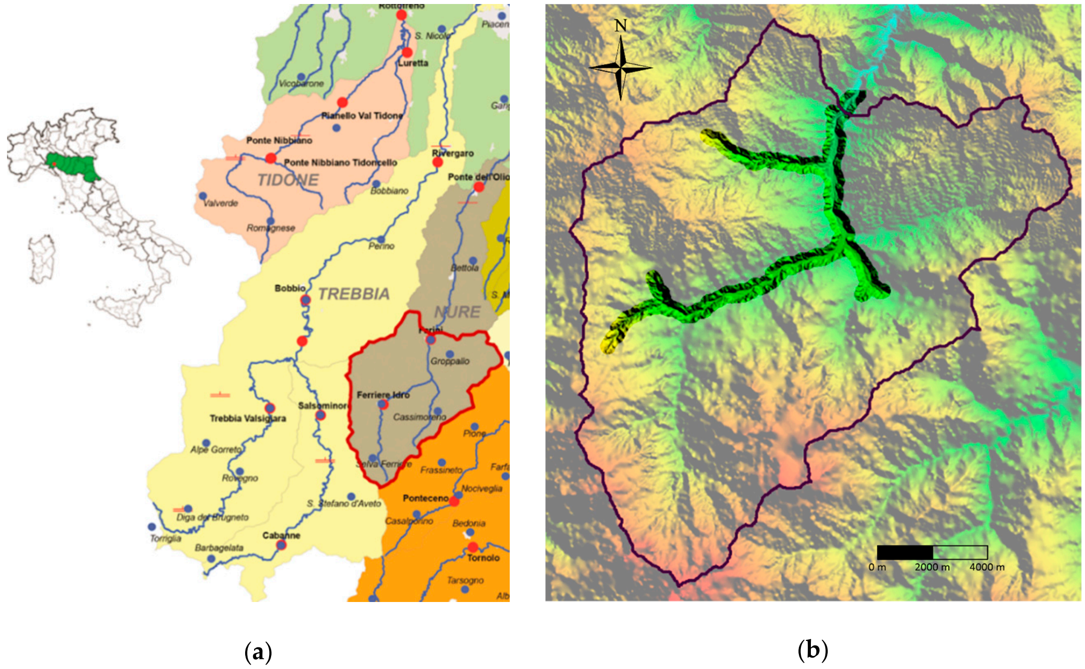


Figure 8. Nure watershed: (**a**) administrative map of Italy with Emilia-Romagna Region highlighted and position of the Nure watershed; (**b**) shaded relief map of the digital elevation model (DEM) of the Nure watershed closed at the Farini outlet.

4.1. Event Description

The rainstorm event was particularly severe with rainfalls that in the most affected areas reached 280 mm in 6 h, with an hourly precipitation peak of 107.6 mm h^{-1}. The precipitation data were acquired every thirty minutes through weather radar and the reflectivity maps obtained were linked to the observations at the rainfall stations present in the drainage area (courtesy of the Regional Agency for Prevention, Environment and Energy of Emilia-Romagna, Italy). The storm center was located South-West of the Nure watershed, yet the total volume of rain on the river basin was over 45 × 10^6 m^3 (Figure 9).

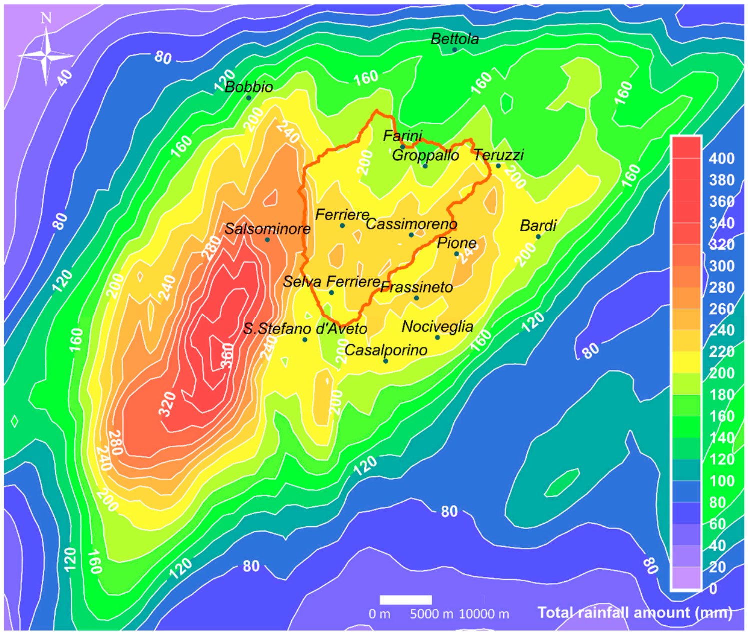
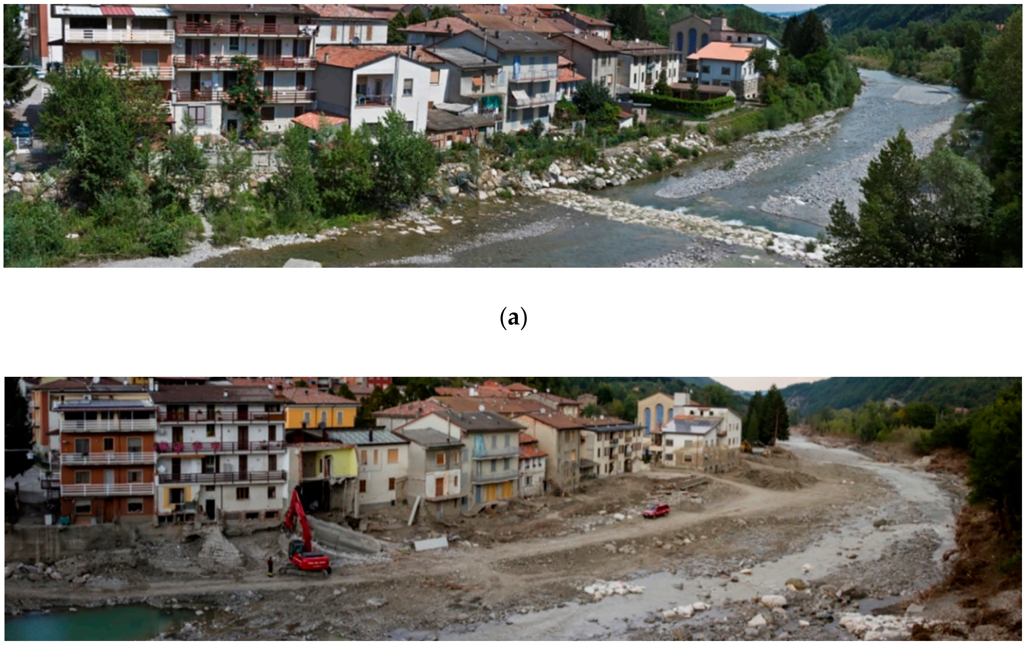


Figure 9. Isohyet map of total rainfall (mm) for the event of interest obtained through the radar reflectivity, linked to rain gauges on the Trebbia and Nure. In red the boundary of the Nure watershed closed at the Farini outlet.

The rainfall event gave rise to such an exceptional flood that it devastated the whole Nure river valley, destroying many buildings located along the river, especially in the village of Farini (Figure 10), and the valley road infrastructure, and damaging the Sassi Neri check dam (located about 500 m upstream of Farini) built to protect the river course from a landslide that tends to obstruct the valley (Figure 11).



(**a**)

(**b**)

Figure 10. Farini village (**a**) before and (**b**) after the flooding event of 13–14 September 2015.

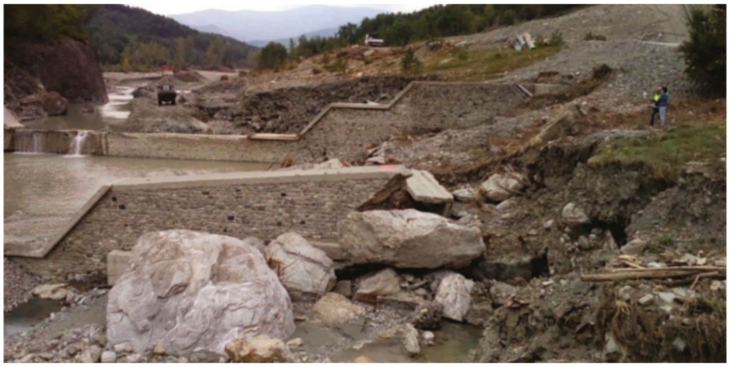


Figure 11. The Sassi Neri check dam after the flood event (from downstream). The flow circumvented and damaged the structure on the left.

4.2. Available Field Data

Several survey campaigns were carried out in the days following the storm. An unmanned aerial vehicle captured ground images at 10 cm per pixel resolution. A light detection and ranging (LiDAR) survey and a 1 m × 1 m DEM of a significant portion of the river valley were also produced, and a traditional ground survey was conducted to identify the maximum levels reached by the water in the village of Farini and upstream. Clear marks left by mud and debris, over 11 meters above the valley floor, were observed upstream of the narrow cross section of the Sassi Neri check dam. These signs allowed identifying the occurrence a posteriori of an ephemeral lake for a few hours only. The water level hydrograph was acquired during the event at the Farini gauging station, which is located downstream the bridge crossing the river Nure in the town center. Unfortunately, the hydrograph was affected by many inaccuracies.

4.3. Bathymetry and Roughness

The 1 m × 1 m DEM of the river valley (visible in the foreground along the main river reach in Figure 8.b) was inlaid through a geographic information system (GIS), in the DEM available of Emilia-Romagna region with a square mesh of size 20 m (in the background of Figure 8.b) to achieve a DEM of the entire watershed under investigation (Figure 12).

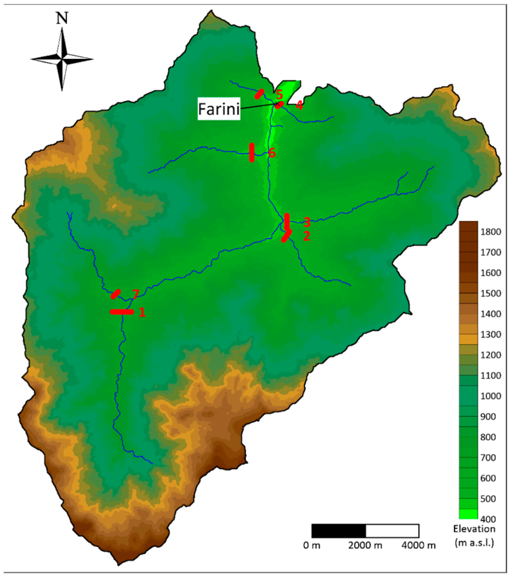


Figure 12. Bathymetry of the Nure watershed and investigated cross sections.

A basic computational mesh of 5 m size was then obtained. The resolution of this basic refined mesh of 15.8 × 10^6 cells (4241 rows × 3725 columns) with over 8.35 × 10^6 active computational cells was chosen to achieve both acceptable accuracy in the geometrical representation (required by the presence of built up areas or infrastructures), and a limited computational burden. Different DEMs characterized by mesh sizes of 10, 20, 40 and 80 m were then obtained through grid decimation.

According to the Emilia-Romagna Region public database, the Nure watershed is characterized by many different soil types and related uses. The original soil polygon map was reworked by appropriately combining some similar subgroups, and seven macro areas were identified with related CN values (Table 2 and Figure 13). The Manning's roughness coefficients for the numerical simulations were then deduced from literature values on the basis of the soil use.

Table 2. Infiltration index and related curve number.

Index	1	2	3	4	5	6	7
CN	57	61	72	75	84	86	90

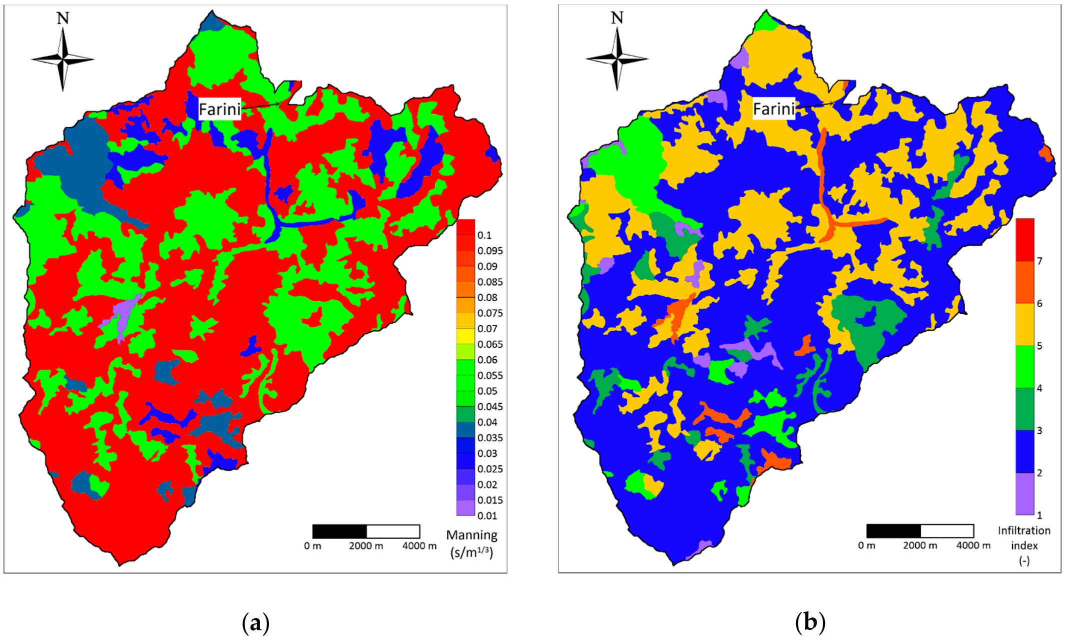


Figure 13. Maps obtained from soil type and land use: (**a**) Manning's roughness map; (**b**) infiltration index map.

4.4. Results for the Nure Watershed Test Case

The BUQ grid is obviously to be preferred to reduce the computational burden and achieve short computational times. For this reason, five preliminary runs adopting Cartesian grids of increasing cell dimensions (5, 10, 20, 40 and 80 m) were performed with the aim of identifying the suitable maximum cell size to be adopted for the watershed sides which gives accurate results. A transmissive boundary condition was assumed in the numerical simulations at the Farini outlet, by imposing a channel slope equal to 0.01, suitable to represent the local river characteristics correctly. Seven cross-sections were identified at the outlet of the main watershed sub-basins (Figure 12), and the corresponding discharge hydrographs were then evaluated and compared.

As Figure 14 shows, no significant differences are present for Cross Sections 1, 2, 5 and 6 (see Figure 12 to identify their locations) in the discharge hydrographs resulting from the simulations performed on Cartesian grids from 5 to 40 m cell size, in terms of neither peak value, nor flood peak timing. The discharge hydrographs show remarkable differences from the ones obtained from the other preliminary runs for the last run adopting a Cartesian grid of 80 m cell size. For this reason, the resolution of 40 m was confirmed to be suitable to describe the watershed sides with no loss of accuracy.

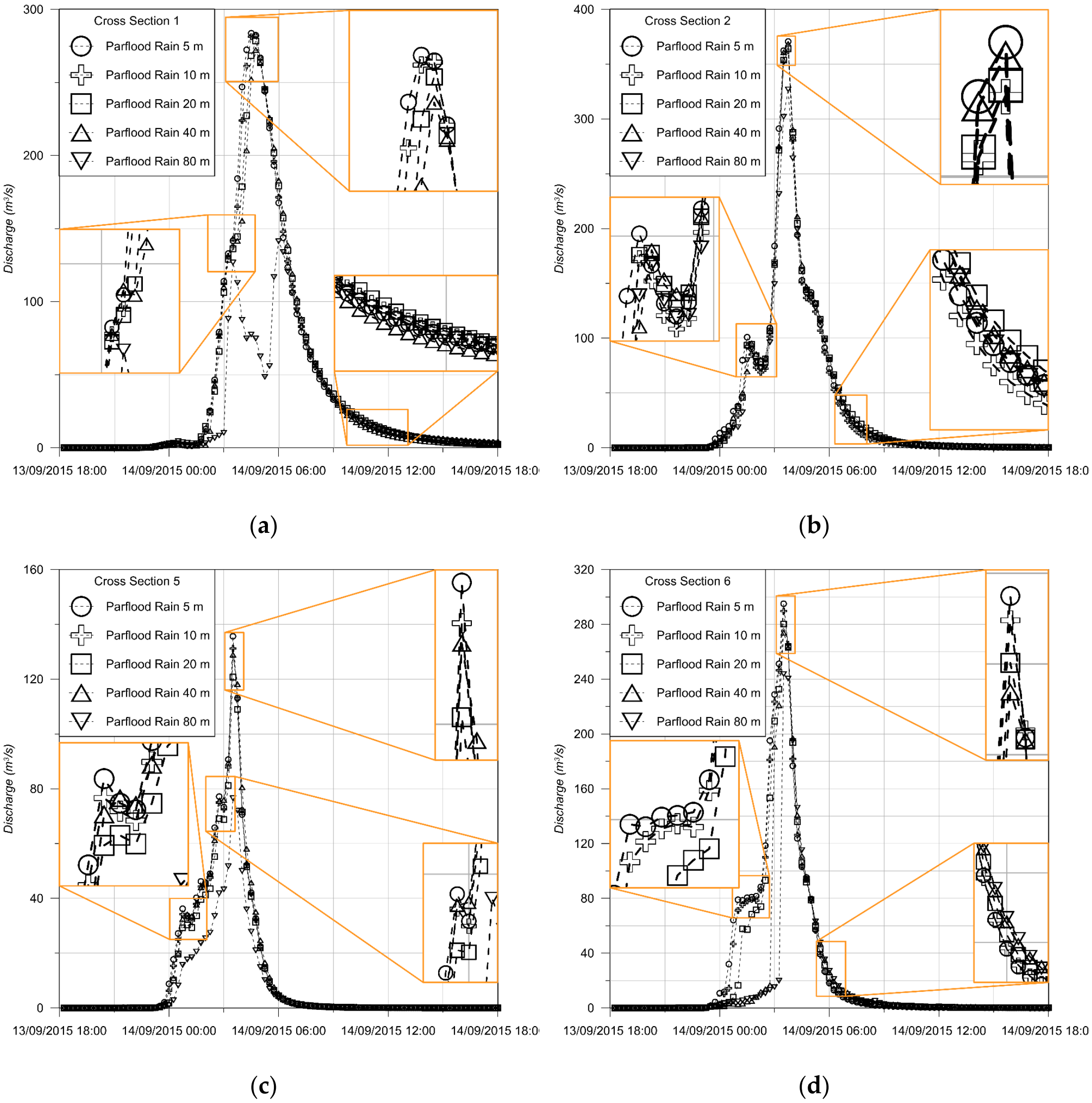


Figure 14. Discharge hydrographs for uniform grid resolutions (5, 10, 20, 40 and 80 m) at: (**a**) Cross Section 1; (**b**) Cross Section 2, (**c**) Cross Section 5 and (**d**) Cross Section 6.

Three BUQ grids, characterized by a decreasing total number of computational cells, were then considered to perform different simulations of the event (Figure 15). In BUQ Grid 1, the minimum grid size of 5 m × 5 m was imposed in correspondence of low-order river branches and at the watershed boundary. Due to both the grid partitioning, and the huge number of cells imposed at the maximum resolution, the maximum cell size of 40 m × 40 m was not present. In BUQ Grid 2, instead, the maximum resolution was required only in higher order river branches and at the watershed boundary, while in BUQ Grid 3 no resolution constraints were imposed at the watershed boundary. As reported in Table 3, the total number of cells of the three BUQ grids decreased from 8.49×10^6 to 0.88×10^6 from BUQ Grid 1 to BUQ Grid 3. As a consequence, a significant speed-up was achieved in the simulations adopting BUQ Grid 3 compared to the reference Cartesian mesh of 5 m × 5 m. The duration of the simulated event was 40 h.

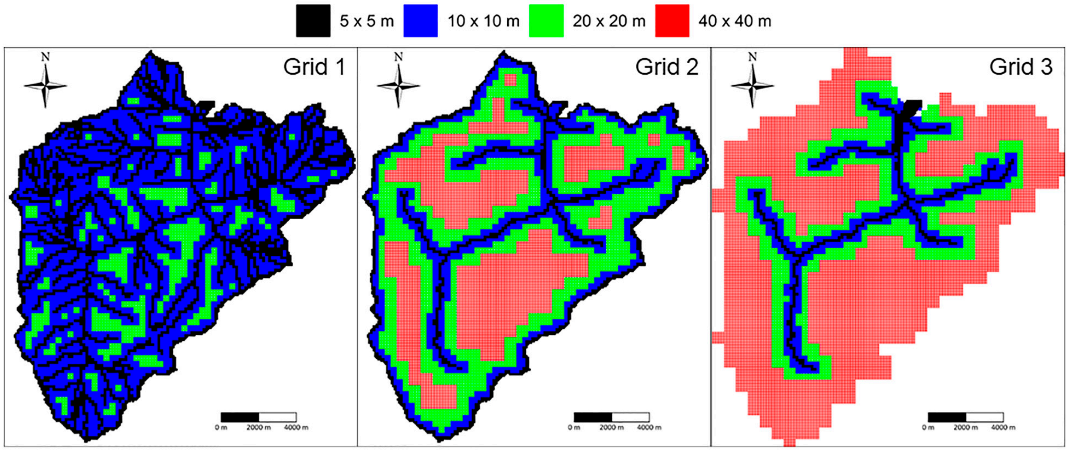


Figure 15. Multiresolution grids: from left to right BUQ Grid 1, 2 and 3.

Table 3. Characteristics of simulations, compression rates C_R and speed-ups S_U for the different BUQ grids adopted.

Grid	N (Millions)	T (Hours)	C_R (–)	S_U (–)
Cartesian 5 m × 5 m	8.49	14.77	–	–
BUQ Grid 1	4.12	7.66	2.06	1.93
BUQ Grid 2	1.55	2.93	5.48	5.04
BUQ Grid 3	0.88	1.88	9.65	7.86

In the simulation adopting the Cartesian mesh of 5 m × 5 m, the number of cells N_C was equal to 8.49×10^6, while T_C was 14.77 h with a simulation to physical time ratio of 0.37. Compression rates and speed-ups achieved with the different BUQ grids adopted are reported in Table 3. The maximum values of C_R and S_U, respectively equal to 9.65 and 7.86, were obtained in the simulation based on BUQ Grid 3, as expected. All simulations were performed using a P100 Tesla® GPU.

Comparable results were achieved in the different simulations in the main river channel at the Farini level gauging station (Figure 16). Malfunctioning of the depth sensor led to many uncertainties affecting the depth hydrograph acquired during the event, so only the maximum water level is reported. This was collected after the event from the marks left by mud and water on the bridge structure. All simulations predicted the maximum water level quite well.

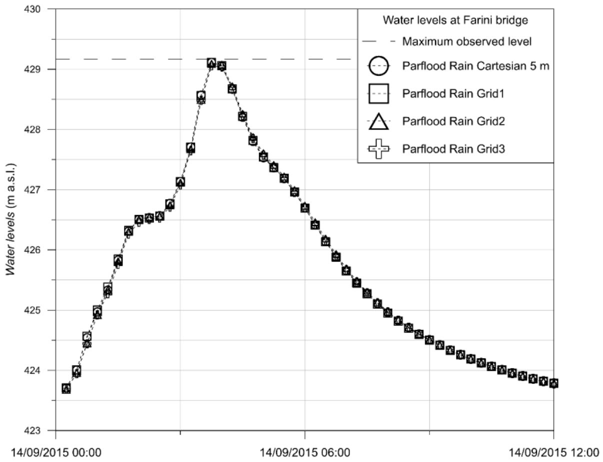


Figure 16. Water level hydrographs at the Farini bridge for the reference simulation, the multiresolution simulations (BUQ Grid 1, 2 and 3) and maximum water observed level.

Figure 17 shows a comparison between the numerical results (adopting BUQ Grid 3) and observations of maximum water levels collected during the post-event surveys in Farini. Satisfactory results were obtained for the majority of the observation points.

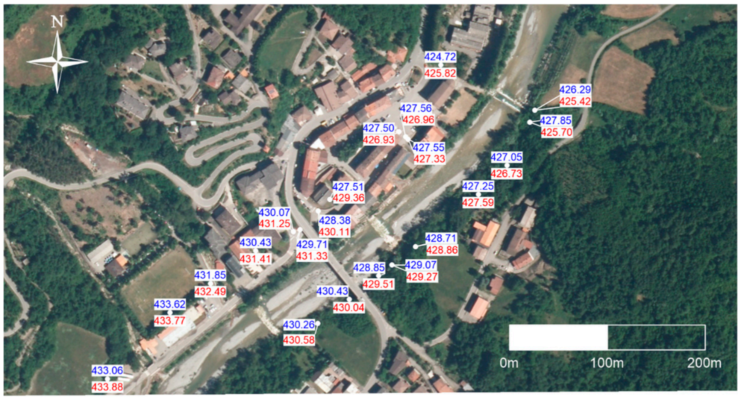


Figure 17. Observed (red) and computed (blue) maximum water elevations at Farini (m a.s.l.).

Figure 18 shows the contour map of the computed water levels at 3:45 a.m. on 14 September in correspondence of the peak flow at Farini. These results can be profitably compared with the data collected through the post-event surveys.

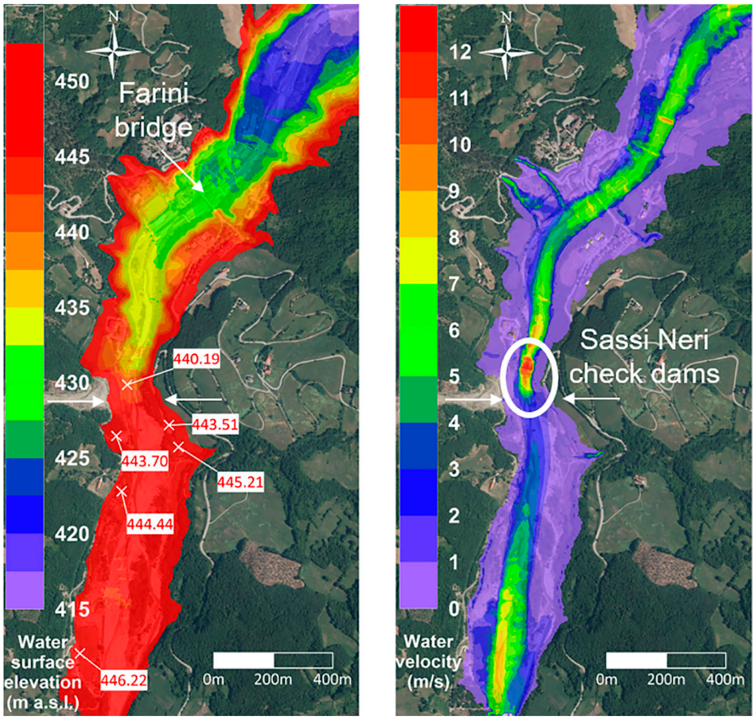
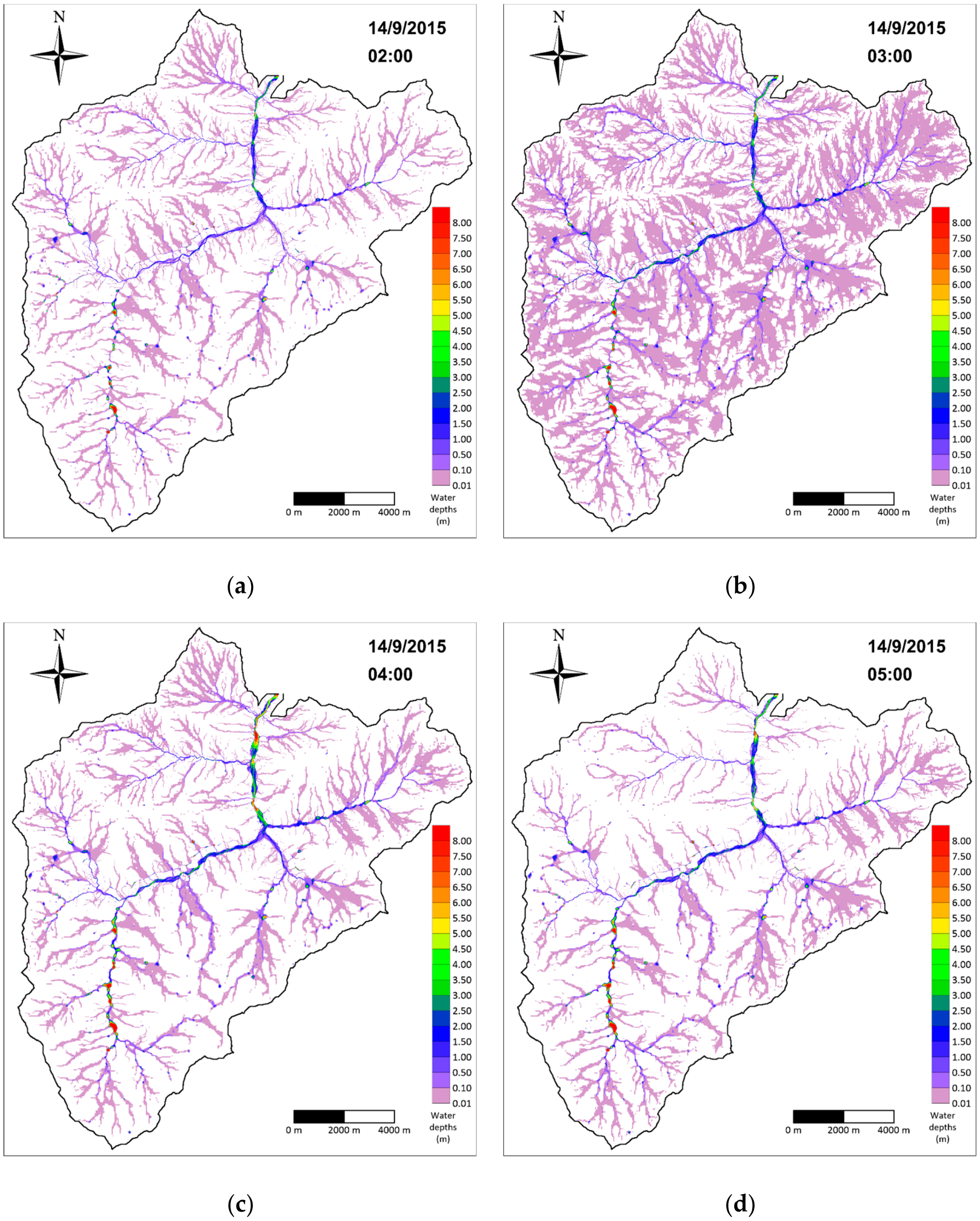


Figure 18. Computed water surface elevation (left frame) and velocities (right frame) in correspondence with the flood peak; the region of supercritical flow is highlighted. In red are reported the observed maximum water levels.

The computed maximum water levels are in good agreement with the observations just upstream of the narrowing of the river valley, where the Sassi Neri check dams are located. Upstream of the check dam the calculated water surface elevation (left frame of Figure 18) is almost horizontal and the velocity (right frame of Figure 18) very low. This clearly shows that an ephemeral lake formed here, storing an estimated water volume of about 0.9·10^6 m^3. The numerical results confirm that the two Sassi Neri check dams were completely submerged and circumvented to the left, as confirmed by the side erosion, and by the damage to the structure and to the wings observed after the event (Figure 11). The flow accelerated and became supercritical with maximum velocities over 10 ms^{-1} downstream of the valley bottleneck originated by the presence of the landslide. Lower velocities of about 5 ms^{-1} were predicted downstream of the bridge in Farini (right frame of Figure 18).

The evolution of the flooding event can be observed at the watershed scale in Figure 19. The numerical model (BUQ Grid 3) provides an accurate reconstruction of the event, characterized by a very high runoff at 3 a.m. on 14 September. Later, the water tended to reach the network of surface streams in which the channel flow occurred with a very quick reaction to rainfall. The maximum flow at Farini occurred at about 3:45 a.m. on 14 September, and the flood event can be definitely referred to as flash flood, as often occurs for the small catchments of the Italian Apennine hydrographic district, characterized by fast hydrological response times.



(**a**) (**b**)

(**c**) (**d**)

Figure 19. Computed water depths for the whole Nure watershed from 02:00 UTC+1 to 05:00 UTC+1 on 14 September 2015: (**a**) 02:00 UTC+1; (**b**) 03:00 UTC+1; (**c**) 04:00 UTC+1 and (**d**) 05:00 UTC+1.

The maximum computed water depths for the whole Nure watershed are shown in Figure 20. The water depths at the catchment sides are generally low (below 0.1 m).

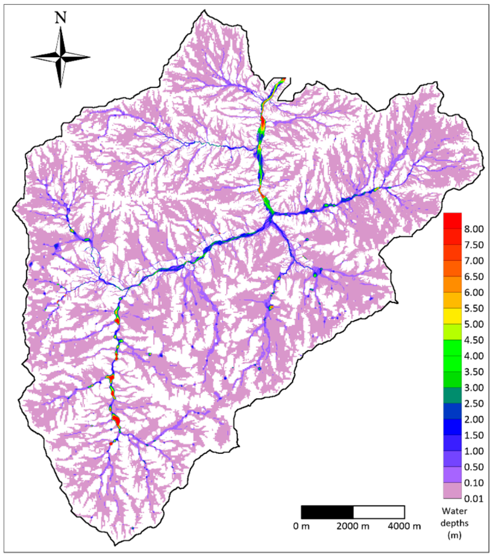


Figure 20. Maximum computed water depths for the Nure watershed for the considered event.

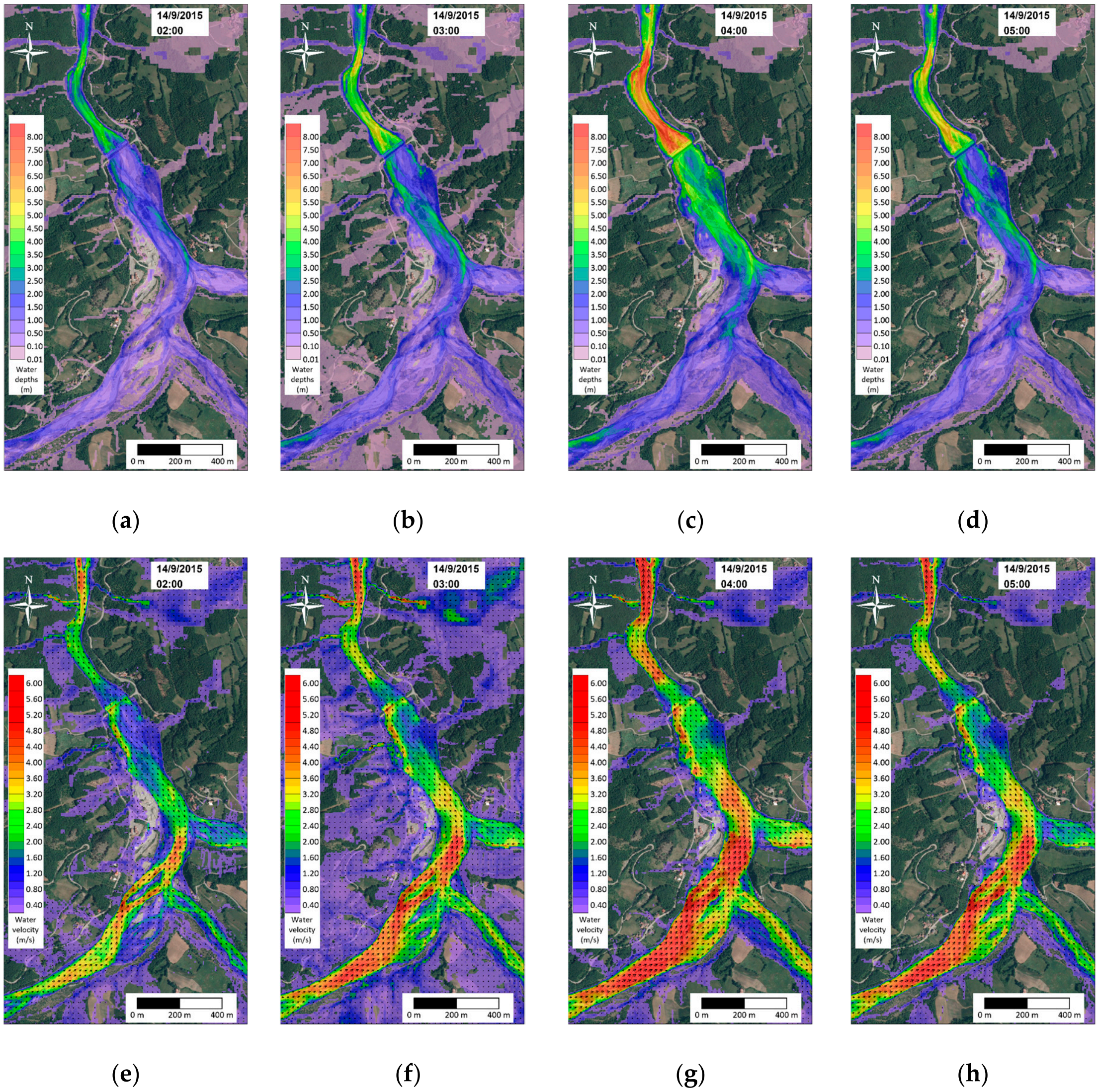
Figure 21 (upper frames) shows a detail of the water depth fields at the confluence of the two right tributaries of the main stream Nure. The model allows a highly detailed description of the flooding dynamics near roads, bridges or other infrastructures located in the studied domain. Figure 21 (lower frames) shows also the velocity field with superimposition of the velocity vectors: as can be seen, the model is capable of providing very detailed information about the velocity field in the different regions of the computational domain.

5. Discussion

The simulation of rapidly varying real flooding scenarios at the watershed scale can nowadays be satisfactorily performed through high spatial resolution 2D models, thanks to the increasingly widespread and affordable LiDAR DEMs. The computational time has always represented a constraint for this kind of simulation. The new multiresolution models, such as that presented here, implemented in a CUDA/C++ code that exploits parallel computation offered by GPUs, allow simulating flooding on several hundreds of square kilometers while maintaining a high spatial resolution in the areas of major interest, with very satisfactory ratios of simulation to physical times.

In particular, the computational model developed and validated here, solves the complete 2D-SWEs, and allows simulating the process of flood propagation starting from the precipitation falling on a watershed. Moreover, due to the adoption of the multiresolution BUQ grids, the modeling tool can also accurately reproduce flood inundations in areas characterized by the presence of roads, railways or channel embankments, or in urban environments, where the flooding dynamics can be strongly influenced by streets and buildings and the geometrical description must be very detailed. In the future, automatic procedures for analyzing terrain models could allow the creation of computational grids based on the identification of some salient features of the basin of interest, completely independent of the hydrodynamic phenomena that can take place in it. These tools, equipped with suitable criteria, could guarantee the generation of a grid consisting of the fewest number of calculation cells compatible with the solution accuracy required.

The present solution algorithm preserves all the terms of the momentum equations and is suitable to deal with the rapidly varying flows that can occur in a two-dimensional flow field during inundation phenomena with low ratios of computational to physical time. Flood management in populated territories crossed by rivers is a crucial issue. Hence, fast and accurate numerical models such as that presented here can be adopted to perform simulations of hypothetical flooding scenarios for flood protection purposes, to evaluate the performance of flood detention structures, or to investigate the behavior of new flood protection hydraulic structures. Finally, the performance of this model could be further improved thanks to the extension of the code to multi-GPUs.



(**a**) (**b**) (**c**) (**d**)

(**e**) (**f**) (**g**) (**h**)

Figure 21. Detail of the water depth (**a, b, c** and **d**) and velocity fields (**e**, **f**, **g** and **h**) at the confluence of two right tributaries of the Nure from 02:00 UTC+1 to 05:00 UTC+1 on 14 September 2015: (**a**, **e**) 02:00 UTC+1; (**b**, **f**) 03:00 UTC+1; (**c**, **g**) 04:00 UTC+1 and (**d**, **h**) 05:00 UTC+1. (Velocity vectors are plotted only one every five for the sake of graphical representation).

Author Contributions: Conceptualization and methodology, F.A., R.V. and F.P.; software, F.P., R.V., A.F. and S.D.; validation, F.P. and F.A.; investigation, F.A. and F.P.; writing—original draft preparation, F.A.; writing—review and editing, F.A., R.V., F.P., A.F. and S.D.; visualization, F.A. and F.P.; supervision, F.A. and R.V. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Acknowledgments: The authors would like to thank Pier Paolo Alberoni (Regional Agency for Prevention, Environment and Energy of Emilia-Romagna, Italy) for providing the radar reflectivity maps linked to the rain gauges related to the Nure watershed event of September 2015. This research benefits from the HPC (High-Performance Computing) facility of the University of Parma.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

A.1. Bed Slope Source Term Treatment

The bed slope source term treatment presented in [62] was adopted to ensure a stable and well-balanced scheme, also in the presence of wet/dry fronts. The bed elevations were obtained from the Local Bed Modification scheme [10] (cell interface i + ½ in the x direction; a similar procedure can be applied to the cell borders i – ½ and j ± ½ in the y direction). The bed slope source term was computed as follows:

$$S_{b_{i,j}} = S_{0_{i,j}} + \bar{S}_{0_{i-\frac{1}{2},j}} + \bar{S}_{0_{i+\frac{1}{2},j}} \,. \tag{A1}$$

The first term represents the central discretization of the bed slope source term and the two other terms were introduced to guarantee the C-property between wet and dry cells:

$$S_{0_{i,j}} = -g\eta\frac{\partial z}{\partial x} = -g\left(\frac{\eta^{R}_{i+\frac{1}{2},j} + \eta^{L}_{i+\frac{1}{2},j}}{2}\right)\left(\frac{\bar{z}_{i+\frac{1}{2},j} - \bar{z}_{i-\frac{1}{2},j}}{\Delta x_{i,j}}\right), \tag{A2}$$

$$\bar{S}_{0_{i-\frac{1}{2},j}} = g\Delta z_{i-\frac{1}{2},j}\left[\frac{\bar{z}_{i+\frac{1}{2},j} - \left(\bar{z}_{i-\frac{1}{2},j} - \Delta z_{i-\frac{1}{2},j}\right)}{2\Delta x_{i,j}}\right], \tag{A3}$$

$$\bar{S}_{0_{i+\frac{1}{2},j}} = g\Delta z_{i+\frac{1}{2},j}\left[\frac{\left(\bar{z}_{i+\frac{1}{2},j} - \Delta z_{i+\frac{1}{2},j}\right) - \bar{z}_{i-\frac{1}{2},j}}{2\Delta x_{i,j}}\right], \tag{A4}$$

with:

$$\Delta z_{i+\frac{1}{2},j} = max\left[0, -\left(\eta^{L}_{i+\frac{1}{2},j} - \bar{z}_{i+\frac{1}{2},j}\right)\right], \tag{A5}$$

$$\Delta z_{i-\frac{1}{2},j} = max\left[0, -\left(\eta^{R}_{i-\frac{1}{2},j} - \bar{z}_{i-\frac{1}{2},j}\right)\right]. \tag{A6}$$

In Equations (A2)–(A6), $\bar{z}$ is obtained through the local bed modification scheme:

$$\bar{z}_{i+\frac{1}{2},j} = z_{i+\frac{1}{2},j} - \Delta z\,, \tag{A7}$$

in which Δz is a function of δz (7):

$$\begin{aligned}
&\text{if } \delta z \geq 0\text{:} \begin{cases} \Delta z = max\left(0,\ z_{i+\frac{1}{2},j} - \eta_{i,j}\right)\ for\ h_{i+1,j} < \varepsilon_h \\ \Delta z = max\left(0,\ min\left(\delta z,\ z_{i+\frac{1}{2},j} - \eta_{i,j}\right)\right)\ for\ h_{i+1,j} \geq \varepsilon_h \end{cases}, \\
&\text{if } \delta z < 0\text{:} \begin{cases} \Delta z = max\left(0,\ z_{i+\frac{1}{2},j} - \eta_{i+1,j}\right)\ for\ h_{i,j} < \varepsilon_h \\ \Delta z = max\left(0,\ min\left(-\delta z,\ z_{i+\frac{1}{2},j} - \eta_{i+1,j}\right)\right)\ for\ h_{i,j} \geq \varepsilon_h \end{cases},
\end{aligned} \tag{A8}$$

where ε_h represents the wet/dry threshold (here 10^{-12} m) and the water surface elevations $\eta_{i,j}$ or $\eta_{i+1,j}$ are the original water surface elevations at cell center rather than the reconstructed value (which is used in Equations (A5) and (A6) instead).

A.2. Friction Source Term Treatment

After the update of the conserved variables at the half time step (4), the friction source term can be discretized in an effective explicit formulation through analytical manipulations. As described in

[74] from the momentum components expanded into a scalar form, the unit width discharge in x the direction, if $DEN = \Delta t^n g n^2 (h^n)^{-\frac{4}{3}} \sqrt{\left(\frac{mx}{h^n}\right)^2 + \left(\frac{my}{h^n}\right)^2}$ takes the form:

$$qx^{n+\frac{1}{2}} = \begin{cases} mx \; for \; \mathrm{DEN} < \varepsilon_h \\ \dfrac{mx - mx\sqrt{1 + 4\Delta t^n g n^2 (h^n)^{-\frac{4}{3}} \sqrt{\left(\frac{mx}{h^n}\right)^2 + \left(\frac{my}{h^n}\right)^2}}}{-2\Delta t^n g n^2 (h^n)^{-\frac{4}{3}} \sqrt{\left(\frac{mx}{h^n}\right)^2 + \left(\frac{my}{h^n}\right)^2}} \; for \; DEN \geq \varepsilon_h \end{cases}, \tag{A9}$$

where:

$$mx = qx^n + \Delta t^n Ax \,, my = qy^n + \Delta t^n Ay \,, \tag{A10}$$

in which Ax and Ay denote the scalar components of Equation (5) neglecting the terms $\mathbf{S}_r$ and $\mathbf{S}_f$. Like Equation (A9), the explicit formulation can be similarly obtained in the y direction.

A.3. Soil Conservation Service SCS Method

The continuity principle establishes that the total precipitation P can be considered as the sum of the excess precipitation, P_e, of the initial abstraction I_a before ponding, for which no runoff occurs, and of the precipitation depth retained in the watershed, F_a:

$$P = P_e + I_a + F_a \,. \tag{A11}$$

Based on this principle, the SCS method hypothesizes the following equivalence between the ratios of the actual to the potential quantities:

$$\frac{F_a}{S_\infty} = \frac{P_e}{P - I_a} \,. \tag{A12}$$

The initial abstraction I_a can be evaluated through the empirical relation:

$$I_a = \alpha S_\infty \,, \tag{A13}$$

in which the parameter α usually assumes the value 0.2. With this assumption, the excess precipitation P_e becomes:

$$P_e = \frac{(P - \alpha S_\infty)^2}{P + S_\infty (1 - \alpha)} \,, \tag{A14}$$

where the potential maximum retention S_∞ (mm) is related to the curve number CN by the expression:

$$S_\infty = 25.4 \cdot \left(\frac{1000}{CN} - 10\right) . \tag{A15}$$

For each computational cell, the cumulative infiltration (mm) can be found by substituting Equation (A14) in Equation (A12):

$$F_a = S_\infty \cdot \frac{P_e}{P - I_a} = S_\infty \cdot \frac{1}{P - I_a} \cdot \frac{(P - I_a)^2}{P + S_\infty - I_a} = S_\infty \cdot \frac{P - I_a}{P + S_\infty - I_a} \,, \tag{A16}$$

with P and F_a calculated and stored at every time step Δt. The infiltration ΔF_a for the current time interval can be obtained through a difference between the cumulative infiltration at times n and $n-1$. The related infiltration rate f is then computed as follows:

$$f^n = \frac{{F_a}^n - {F_a}^{n-1}}{\Delta t} \,. \tag{A17}$$

References

1. Chow, V.T.; Ben-Zvi, A. Hydrodynamic modeling of two-dimensional watershed flow. *J. Hydraul. Div.* **1973**, *99*, 2023–2040.
2. Kawahara, M.; Yokoyama, T. Finite element method for direct runoff flow. *J. Hydraul. Div.* **1980**, *106*, 519–534.
3. Di Giammarco, P.; Todini, E.; Lamberti, P. A conservative finite elements approach to overland flow: The control volume finite element formulation. *J. Hydrol.* **1996**, *175*, 267–291.
4. Singh, J.; Altinakar, M.S.; Ding, Y. Numerical modeling of rainfall-generated overland flow using nonlinear shallow-water equations. *J. Hydrol. Eng.* **2015**, *20*, doi:10.1061/(ASCE)HE.1943-5584.0001124.
5. Yu, C.; Duan, J. Two-dimensional hydrodynamic model for surface-flow routing. *J. Hydraul. Eng.* **2014**, *140*, doi:10.1061/(ASCE)HY.1943-7900.0000913.
6. Yu, C.; Duan, J. Simulation of surface runoff using hydrodynamic model. *J. Hydrol. Eng.* **2017**, *22*, doi:10.1061/(ASCE)HE.1943-5584.0001497.
7. Cea, L.; Garrido, M.; Puertas, J.; Jácome, A.; Del Río, H.; Suárez, J. Overland flow computations in urban and industrial catchments from direct precipitation data using a two-dimensional shallow water model. *Water Sci. Technol.* **2010**, *62*, 1998–2008.
8. Cea, L.; Bladé, E. A simple and efficient unstructured finite volume scheme for solving the shallow water equations in overland flow applications. *Water Resour. Res.* **2015**, *51*, 5464–5486.
9. Liang, D.; Özgen, I.; Hinkelmann, R.; Xiao, Y.; Chen, J.M. Shallow water simulation of overland flows in idealised catchments. *Environ. Earth Sci.* **2015**, *74*, 7307–7318.
10. Xia, X.; Liang, Q.; Ming, X.; Hou, J. An efficient and stable hydrodynamic model with novel source term discretization schemes for overland flow and flood simulations. *Water Resour. Res.* **2017**, *53*, 3730–3759.
11. Zhang, L.; Nan, Z.; Liang, X.; Xu, Y.; Hernández, F.; Li, L. Application of the MacCormack scheme to overland flow routing for high-spatial resolution distributed hydrological model. *J. Hydrol.* **2018**, *558*, 421–431.
12. Costabile, P.; Costanzo, C.; Macchione, F. A storm event watershed model for surface runoff based on 2D fully dynamic wave equations. *Hydrol. Process.* **2013**, *27*, 554–569.
13. Costabile, P.; Costanzo, C.; Macchione, F. Performances and limitations of the diffusive approximation of the 2-d shallow water equations for flood simulation in urban and rural areas. *Appl. Numer. Math.* **2017**, *116*, 141–156.
14. European Environment Agency. *Economic Losses from Climate-Related Extremes in Europe*; European Environment Agency (www.eea.europa.eu), 2019.
15. Anselmo, V.; Galeati, G.; Palmieri, S.; Rossi, U.; Todini, E. Flood risk assessment using an integrated hydrological and hydraulic modelling approach: A case study. *J. Hydrol.* **1996**, *175*, 533–554.
16. Agnese, C.; Baiamonte, G.; Corrao, C. A simple model of hillslope response for overland flow generation. *Hydrol. Process.* **2001**, *15*, 3225–3238.
17. Wang, G.-T.; Chen, S.; Boll, J.; Stockle, C.O.; McCooL, D.K. Modelling overland flow based on Saint-Venant equations for a discretized hillslope system. *Hydrol. Process.* **2002**, *16*, 2409–2421.
18. Morbidelli, R.; Corradini, C.; Govindaraju, R.S. A simplified model for estimating field-scale surface runoff hydrographs. *Hydrol. Process.* **2007**, *21*, 1772–1779.
19. Jasper, K.; Gurtz, J.; Lang, H. Advanced flood forecasting in Alpine watersheds by coupling meteorological observations and forecasts with a distributed hydrological model. *J. Hydrol.* **2002**, *267*, 40–52.
20. Jaber, F.H.; Mohtar, R.H. Stability and accuracy of finite element schemes for the one-dimensional kinematic wave solution. *Adv. Water Resour.* **2002**, *25*, 427–438.
21. Jaber, F.H.; Mohtar, R.H. Stability and accuracy of two-dimensional kinematic wave overland flow modeling. *Adv. Water Resour.* **2003**, *26*, 1189–1198.
22. Lerat, J.; Perrin, C.; Andréassian, V.; Loumagne, C.; Ribstein, P. Towards robust methods to couple lumped rainfall-runoff models and hydraulic models: A sensitivity analysis on the Illinois River. *J. Hydrol.* **2012**, *418–419*, 123–135.
23. Nguyen, P.; Thorstensen, A.; Sorooshian, S.; Hsu, K.; AghaKouchak, A.; Sanders, B.; Koren, V.; Cui, Z.; Smith, M. A high resolution coupled hydrologic–hydraulic model (HiResFlood-UCI) for flash flood modeling. *J. Hydrol.* **2016**, *541*, 401–420.
24. Kim, J.; Warnock, A.; Ivanov, V.Y.; Katopodes, N.D. Coupled modeling of hydrologic and hydrodynamic processes including overland and channel flow. *Adv. Water Resour.* **2012**, *37*, 104–126.
25. Cea, L.; Garrido, M.; Puertas, J. Experimental validation of two-dimensional depth-averaged models for

forecasting rainfall–runoff from precipitation data in urban areas. *J. Hydrol.* **2010**, *382*, 88–102.

26. Jaber, F.H.; Mohtar, R.H. Dynamic time step for one-dimensional overland flow kinematic wave solution. *J. Hydrol. Eng.* **2002**, *7*, 3–11.
27. Warnock, A.; Kim, J.; Ivanov, V.; Katopodes, N.D. Self-adaptive kinematic-dynamic model for overland flow. *J. Hydraul. Eng.* **2014**, *140*, 169–181.
28. Costabile, P.; Costanzo, C.; Macchione, F. Comparative analysis of overland flow models using finite volume schemes. *J. Hydroinform.* **2012**, *14*, 122–135.
29. Caviedes-Voullième, D.; Fernández-Pato, J.; Hinz, C. Cellular automata and finite volume solvers converge for 2D shallow flow modelling for hydrological modelling. *J. Hydrol.* **2018**, *563*, 411–417.
30. Aricò, C.; Nasello, C. Comparative analyses between the zero-inertia and fully dynamic models of the shallow water equations for unsteady overland flow propagation. *Water* **2018**, *10*, 44.
31. Aricò, C.; Sinagra, M.; Begnudelli, L.; Tucciarelli, T. MAST-2D diffusive model for flood prediction on domains with triangular Delaunay unstructured meshes. *Adv. Water Resour.* **2011**, *34*, 1427–1449.
32. Bates, P.D. Remote sensing and flood inundation modelling. *Hydrol. Process.* **2004**, *18*, 2593–2597.
33. Jahanbazi, M.; Özgen, I.; Aleixo, R.; Hinkelmann, R. Development of a diffusive wave shallow water model with a novel stability condition and other new features. *J. Hydroinform.* **2017**, *19*, 405–425.
34. Savant, G.; Trahan, C.J.; Pettey, L.; McAlpin, T.O.; Bell, G.L.; McKnight, C.J. Urban and overland flow modeling with dynamic adaptive mesh and implicit diffusive wave equation solver. *J. Hydrol.* **2019**, *573*, 13–30.
35. Su, B.; Huang, H.; Zhu, W. An urban pluvial flood simulation model based on diffusive wave approximation of shallow water equations. *Hydrol. Res.* **2017**, *50*, 138–154.
36. Mignosa, P.; Vacondio, R.; Aureli, F.; Dazzi, S.; Ferrari, A.; Prost, F. High resolution 2D modelling of rapidly varying flows: Some case studies. *Ital. J. Eng. Geol. Environ.* **2018**, *2018*, 143–160.
37. Dottori, F.; Todini, E. Testing a simple 2D hydraulic model in an urban flood experiment. *Hydrol. Process.* **2013**, *27*, 1301–1320.
38. Hou, J.; Wang, T.; Li, P.; Li, Z.; Zhang, X.; Zhao, J.; Hinkelmann, R. An implicit friction source term treatment for overland flow simulation using shallow water flow model. *J. Hydrol.* **2018**, *564*, 357–366.
39. Caviedes-Voullième, D.; Fernández-Pato, J.; Hinz, C. Performance assessment of 2D Zero-Inertia and shallow water models for simulating rainfall-runoff processes. *J. Hydrol.* **2020**, doi:10.1016/j.jhydrol.2020.124663.
40. Aureli, F.; Maranzoni, A.; Mignosa, P.; Ziveri, C. A weighted surface-depth gradient method for the numerical integration of the 2D shallow water equations with topography. *Adv. Water Resour.* **2008**, *31*, 962–974.
41. Brodtkorb, A.R.; Sætra, M.L.; Altinakar, M. Efficient shallow water simulations on GPUs: Implementation, visualization, verification, and validation. *Comput. Fluids* **2012**, *55*, 1–12.
42. Smith, L.S.; Liang, Q. Towards a generalised GPU/CPU shallow-flow modelling tool. *Comput. Fluids* **2013**, *88*, 334–343.
43. Lacasta, A.; Morales-Hernández, M.; Murillo, J.; García-Navarro, P. An optimized GPU implementation of a 2D free surface simulation model on unstructured meshes. *Adv. Eng. Softw.* **2014**, *78*, 1–15.
44. Liang, Q.; Xia, X.; Hou, J. Catchment-scale high-resolution flash flood simulation using the GPU-based technology. *Procedia Eng.* **2016**, *154*, 975–981.
45. Le, P.V.V.; Kumar, P.; Valocchi, A.J.; Dang, H.V. GPU-based high-performance computing for integrated surface-sub-surface flow modeling. *Environ. Model. Softw.* **2015**, *73*, 1–13.
46. Hu, X.; Song, L. Hydrodynamic modeling of flash flood in mountain watersheds based on high-performance GPU computing. *Nat. Hazards* **2018**, *91*, 567–586.
47. Xing, Y.; Liang, Q.; Wang, G.; Ming, X.; Xia, X. City-scale hydrodynamic modelling of urban flash floods: The issues of scale and resolution. *Nat. Hazards* **2019**, *96*, 473–496.
48. Vacondio, R.; Dal Palù, A.; Mignosa, P. GPU-enhanced finite volume shallow water solver for fast flood simulations. *Environ. Model. Softw.* **2014**, *57*, 60–75.
49. Lacasta, A.; Morales-Hernández, M.; Murillo, J.; García-Navarro, P. GPU implementation of the 2D shallow water equations for the simulation of rainfall/runoff events. *Environ. Earth Sci.* **2015**, *74*, 7295–7305.
50. Juez, C.; Lacasta, A.; Murillo, J.; García-Navarro, P. An efficient GPU implementation for a faster simulation of unsteady bed-load transport. *J. Hydraul. Res.* **2016**, *54*, 275–288.
51. Vacondio, R.; Dal Palù, A.; Ferrari, A.; Mignosa, P.; Aureli, F.; Dazzi, S. A non-uniform efficient grid type

for GPU-parallel Shallow Water Equations models. *Environ. Model. Softw.* **2017**, *88*, 119–137.

52. García-Navarro, P.; Murillo, J.; Fernández-Pato, J.; Echeverribar, I.; Morales-Hernández, M. The shallow water equations and their application to realistic cases. *Environ. Fluid Mech.* **2019**, *19*, 1235–1252.
53. Dazzi, S.; Vacondio, R.; Mignosa, P. Internal boundary conditions for a GPU-accelerated 2D shallow water model: Implementation and applications. *Adv. Water Resour.* **2020**, *137*, 103525.
54. Ferrari, A.; Dazzi, S.; Vacondio, R.; Mignosa, P. Enhancing the resilience to flooding induced by levee breaches in lowland areas: A methodology based on numerical modelling. *Nat. Hazards Earth Syst. Sci.* **2020**, *20*, 59–72.
55. Dazzi, S.; Vacondio, R.; Dal Palù, A.; Mignosa, P. A local time stepping algorithm for GPU-accelerated 2D shallow water models. *Adv. Water Resour.* **2018**, *111*, 274–288.
56. Dazzi, S.; Vacondio, R.; Mignosa, P. Integration of a levee breach erosion model in a GPU-accelerated 2D shallow water equations code. *Water Resour. Res.* **2019**, *55*, 682–702.
57. Turchetto, M.; Dal Palu, A.; Vacondio, R. A general design for a scalable MPI-GPU multi-resolution 2D numerical solver. *IEEE Trans. Parallel Distrib. Syst.* **2019**, doi:10.1109/TPDS.2019.2961909.
58. Toro, E.F. *Riemann Solvers and Numerical Methods for Fluid Dynamics*; Springer: Berlin/Heidelberg, Germany, 1999.
59. Liang, Q.; Borthwick, A.G.L. Adaptive quadtree simulation of shallow flows with wet–dry fronts over complex topography. *Comput. Fluids* **2009**, *38*, 221–234.
60. Toro, E.F. *Shock-Capturing Methods for Free-Surface Shallow Flows*; Wiley: New York, NY, USA, 2001.
61. Audusse, E.; Bouchut, F.; Bristeau, M.-O.; Klein, R.; Perthame, B. A fast and stable well-balanced scheme with hydrostatic reconstruction for shallow water flows. *SIAM J. Sci. Comput.* **2004**, *25*, 2050–2065.
62. Liang, Q.; Marche, F. Numerical resolution of well-balanced shallow water equations with complex source terms. *Adv. Water Resour.* **2009**, *32*, 873–884.
63. Chow, V.T.; Maidment, D.R.; Mays, L.W. *Applied Hydrology*; McGraw-Hill: New York, NY, USA, 1988.
64. NVIDIA CUDA. *Compute Unified Device Architecture Programming Guide*. Available Online: www.nvidia.com (accessed on 1 November 2019).
65. Greaves, D.M.; Borthwick, A.G.L. Hierarchical tree-based finite element mesh generation. *Int. J. Numer. Methods Eng.* **1999**, *45*, 447–471.
66. Liang, Q. A structured but non-uniform Cartesian grid-based model for the shallow water equations. *Int. J. Numer. Methods Fluids* **2011**, *66*, 537–554.
67. Iwagaki, Y. Fundamental studies on the runoff by characteristics. **1955**, *10*, 1–25.
68. Feng, K.; Molz, F.J. A 2-D, diffusion-based, wetland flow model. **1997**, *196*, 230–250.
69. Busaman, A.; Mekchay, K.; Siripant, S. Dynamically adaptive tree grid modeling for simulation and visualization of rainwater overland flow. *Int. J. Numer. Methods Fluids* **2015**, *79*, 559–579.
70. West, D.W.; Kubatko, E.J.; Conroy, C.J.; Yaufman, M.; Wood, D. A multidimensional discontinuous Galerkin modeling framework for overland flow and channel routing. *Adv. Water Resour.* **2017**, *102*, 142–160.
71. Fiedler, F.R.; Ramirez, J.A. A numerical method for simulating discontinuous shallow flow over an infiltrating surface. *Int. J. Numer. Methods Fluids* **2000**, *32*, 219–240.
72. Simons, F.; Busse, T.; Hou, J.; Özgen, I.; Hinkelmann, R. A model for overland flow and associated processes within the Hydroinformatics Modelling System. *J. Hydroinform.* **2014**, *16*, 375–391.
73. Hou, J.; Guo, K.; Liu, F.; Han, H.; Liang, Q.; Tong, Y.; Li, P. Assessing slope forest effect on flood process caused by a short-duration storm in a small catchment. *Water* **2018**, *10*, 1256.
74. Xia, X.; Liang, Q. A new efficient implicit scheme for discretising the stiff friction terms in the shallow water equations. *Adv. Water Resour.* **2018**, *117*, 87–97.
75. Wooding, R.A. A hydraulic model for the catchment-stream problem. I. Kinematic-wave theory. *J. Hydrol.* **1965**, *3*, 254–267.
76. Wooding, R.A. A hydraulic model for the catchment-stream problem. II. Numerical solutions. *J. Hydrol.* **1965**, *3*, 268–282.
77. Overton, D.E.; Brakensiek, D.L. A kinematic model of surface runoff response. *Hydrologie* **1970**, *1*, 100–112.
78. Overton, D.E. Estimation of surface water lag time from the kinematic wave equations. *J. Am. Water Resour. Assoc.* **1971**, *7*, 428–440.
79. Liggett, J.A.; Woolhiser, D.A. The use of the shallow water equations in runoff computation. In Proceedings of the Third annual American Water Resources Conference, San Francisco, CA, USA, 8–10 November 1967; pp. 117–126.

80. Woolhiser, D.A.; Liggett, J.A. Unsteady, one-dimensional flow over a plane-The rising hydrograph. *Water Resour. Res.* **1967**, *3*, 753–771.