Internal boundary conditions for a GPU-accelerated 2D shallow water model: Implementation and applications

Internal boundary conditions for a GPU-accelerated 2D shallow water model: implementation and applications

SUSANNA DAZZI*, RENATO VACONDIO, PAOLO MIGNOSA

Department of Engineering and Architecture, University of Parma, Parco Area delle Scienze 181/A, 43124 Parma, Italy

susanna.dazzi@unipr.it, renato.vacondio@unipr.it, paolo.mignosa@unipr.it

**corresponding author*

Abstract

Flood propagation in rivers is strongly influenced by the presence of bridges and other hydraulic structures. Among the available approaches for including these elements in numerical models, the adoption of Internal Boundary Conditions (IBC), given its ability to capture backwater, is suitable for field-scale analyses for flood hazard assessment. In this paper, the implementation of internal boundary conditions in the two-dimensional shallow water code named "PARFLOOD" is presented. The application to experimental and real test cases shows that the proposed IBC model can handle both low and high flow conditions for bridges, while being flexible for other types of structures (e.g. flow-through dams). Moreover, the model is computationally efficient (physical/computational time ratio around 20-30 for domains with ~10^6 cells), thanks to the code parallelization on GPU.

Keywords

2D Shallow Water Equations, internal boundary conditions, bridges, hydraulic structures, flood modeling, GPU

1. Introduction

The presence of bridges and other crossing structures can significantly affect the flood propagation in rivers and channels, since the piers, abutments and deck represent an obstruction to the flow. In particular, the lateral and vertical constrictions at bridge sites induce a backwater effect upstream of these structures, possibly increasing the hazard in flood-prone areas. In case of severe flood events, the water level may exceed the elevation of the deck low chord, generating pressure flow under the bridge; the deck may even experience overtopping during extreme events. These "high flow" conditions generate very complex flows; hence, this topic was studied experimentally (e.g. Martin-Vide and Prio, 2005; Picek et al., 2007), with a special focus on the assessment of backwater (Brown, 1985; Seckin, 2004). Besides bridges, other notable types of structures include sluice-gates for flow regulation and irrigation management, weirs, and "flow-through dams" for flood control. All these elements interfere with the flow and their presence should be properly taken into account in numerical simulations performed for flood management applications (Costabile et al., 2015).

Nowadays, two-dimensional (2D) models based on the Shallow Water Equations (SWE) are often adopted for flood risk and hazard assessment, for the design and verification of mitigation strategies, and even for civil protection purposes (Ferrari et al., 2020). However, the markedly three-dimensional characteristics of the flow field around most hydraulic structures, where non-negligible vertical velocities and accelerations can occur, prevent a straightforward integration of these elements in depth-averaged models based on the hypothesis of a hydrostatic pressure distribution. Even if different simple approaches are available for the representation of bridges and other hydraulic structures in one-dimensional (1D) models (i.e. Brunner, 2016; Natale et al., 2004; Catella and Bechi, 2006), fewer analyses can be found in the context of 2D models. According to Ratia et al. (2014), there are three available methods for considering the presence of bridges: mesh discretization (MD), head loss (HL), and internal boundary conditions (IBCs). Maranzoni et al. (2015) propose an additional method, namely the 2D extension of the Preissmann Slot model (2D-PS). The MD approach requires an adequate grid refinement at bridge sites in order to represent the piers as "holes" in the mesh with reflective boundaries. The HL model (Ratia et al., 2014) includes an additional source term in the SWE, which represents the contraction and expansion losses through the bridge. The IBCs modify the numerical scheme only at special cells near the inner structure. Depending on the implementation, IBCs can modify either the flux computation (Zhao et al., 1994) or the value of the conserved variables (Morales-Hernández et al., 2013), usually imposing the unit discharge passing through the hydraulic structure (bridge, weir, dam, etc.) evaluated from a prescribed rating curve or discharge formula. Finally, the 2D-PS model, based on a modified set of SWE, accounts for possible flow pressurization when the water surface reaches the bridge low chord, and must be combined with

the MD approach for low flow conditions. Among these strategies, HL and IBCs predict the "global" effect induced by the bridge presence (i.e. backwater), while MD aims at simulating the flow field near the hydraulic structure, especially as regards the possible transition to supercritical flow downstream. However, the latter (and also 2D-PS) entails longer simulation times due to the higher resolution needed, and to the restriction of the admissible time step. Moreover, MD is unable to reproduce high flow conditions correctly, and 2D-PS can simulate only pressurization (without overtopping), while HL and IBCs are more flexible in this respect, even if empirical formulae are often required. Finally, HL was developed specifically for bridges, while IBCs were successfully applied to different types of structures (Zhao et al., 1994), including gates (Morales-Hernández et al., 2013) and levees (Echeverribar et al., 2019).

Based on this short discussion, IBCs can be identified as a promising strategy for including the presence of hydraulic structures in a 2D model to simulate flood propagation in field-scale analyses, thanks to their flexibility for different structures and flow conditions, and to their ability to capture backwater. Flood simulations in large domains usually imply long computational times, unless optimized and/or parallelized codes are used. In particular, models able to run on Graphics Processing Unit (GPU) devices (e.g. Lastra et al., 2009; Lacasta et al., 2014; Vacondio et al., 2014) are gaining attention due to their computational efficiency, but, to date, the implementation of IBCs in GPU-accelerated codes has never been analyzed in the literature.

The aim of this paper is to present an efficient strategy for including IBCs in the GPU-accelerated 2D SWE code named "PARFLOOD" (Vacondio et al., 2014, 2017). Details about the modifications to the numerical scheme and the GPU implementation are herein provided. The model is validated against experimental tests concerning bridge backwater in steady and unsteady conditions, and then applied to the simulation of the propagation of floods in the Parma River (Italy), with bridges experiencing high flow conditions. Moreover, the case of an IBC to represent the dam which controls a flood detention basin is analyzed.

The paper is organized as follows. Section 2 is dedicated to the model presentation, focusing on the description of the IBC implementation. The validation of the proposed model is described in Section 3, while applications to field-scale tests are shown in Section 4. Discussion and conclusions are provided in the last Section.

2. Model description

2.1. The PARFLOOD model

2.1.1 Mathematical model

The PARFLOOD model (Vacondio et al., 2014, 2017) solves the 2D-SWE through a Finite Volume (FV) scheme. The governing equations are depth-averaged mass and momentum conservation laws, written in integral form as follows (Toro, 2001):

$$\frac{\partial}{\partial t}\int_A \mathbf{U}\,\mathrm{d}A + \int_C \mathbf{H}\cdot\mathbf{n}\,\mathrm{d}C = \int_A \left(\mathbf{S}_0 + \mathbf{S}_f\right)\mathrm{d}A \tag{1}$$

where t is the time, A and C represent the area and boundary of the integration volume, respectively, while $\mathbf{U}$ is the vector of conserved variables, $\mathbf{H} = (\mathbf{F},\mathbf{G})$ is the tensor of fluxes in the x and y directions, and $\mathbf{n}$ is the outward unit vector normal to C; finally, $\mathbf{S}_0$ and $\mathbf{S}_f$ are the bed slope and friction source terms, respectively. Following the well-balanced formulation of Liang and Marche (2009), the terms $\mathbf{U}$, $\mathbf{F}$ and $\mathbf{G}$, $\mathbf{S}_0$ and $\mathbf{S}_f$ are defined as follows:

$$\mathbf{U} = \begin{bmatrix} \eta \\ uh \\ vh \end{bmatrix},\quad \mathbf{F} = \begin{bmatrix} uh \\ u^2h + \frac{1}{2}g(\eta^2 - 2\eta z) \\ uvh \end{bmatrix},\quad \mathbf{G} = \begin{bmatrix} vh \\ uvh \\ v^2h + \frac{1}{2}g(\eta^2 - 2\eta z) \end{bmatrix},$$

$$\mathbf{S}_0 = \begin{bmatrix} 0 \\ -g\eta\frac{\partial z}{\partial x} \\ -g\eta\frac{\partial z}{\partial y} \end{bmatrix},\quad \mathbf{S}_f = \begin{bmatrix} 0 \\ -gh\frac{n_f^2 u\sqrt{u^2+v^2}}{h^{4/3}} \\ -gh\frac{n_f^2 v\sqrt{u^2+v^2}}{h^{4/3}} \end{bmatrix} \tag{2}$$

In Eq. (2), h and z are the flow depth and the bed elevation, respectively, while $\eta = h+z$ represents the water surface elevation; moreover, u and v are the velocity components along the x and y directions, g is the acceleration due to gravity, and n_f is Manning's roughness coefficient.

An explicit FV scheme is employed to discretize the equations. In the first-order accurate approximation, the conserved variables are updated according to the following expression:

$$\mathbf{U}_{i,j}^{n+1} = \mathbf{U}_{i,j}^{n} - \frac{\Delta t}{\Delta x}\left(\mathbf{F}_{i+\frac{1}{2},j} - \mathbf{F}_{i-\frac{1}{2},j}\right) - \frac{\Delta t}{\Delta y}\left(\mathbf{G}_{i,j+\frac{1}{2}} - \mathbf{G}_{i,j-\frac{1}{2}}\right) + \Delta t\left(\mathbf{S}_0 + \mathbf{S}_f\right) \tag{3}$$

where superscript n and subscripts i,j indicate the time level and the cell position, respectively, while Δx (Δy) is the cell size in the x- (y-) direction, and Δt is the time step, which must satisfy the Courant-Friedrichs-Lewy stability condition. Intercell fluxes are computed using the HLLC approximate Riemann solver (Toro, 2001), and a correction to avoid non-physical velocities at wet/dry fronts (Kurganov and Petrova, 2007) is implemented. Moreover, a centered approximation is adopted to discretize the slope source term (Vacondio et al., 2014), while the implicit formulation proposed by

Caleffi et al. (2003) is used for the friction source term. Second-order accuracy in space and time can be achieved by adopting a depth-positive MUSCL extrapolation at cell boundaries (Audusse et al., 2004) and by applying the second-order Runge-Kutta method for advancing the solution in time.

2.1.2 GPU implementation

One of the main advantages of the PARFLOOD model is its CUDA/C++ implementation, which allows for the exploitation of the intrinsic parallel computations of GPU devices, thus guaranteeing simulation times up to two orders of magnitude lower than the ones of serial codes (Vacondio et al., 2014). Examples of the good performance of the PARFLOOD model for real applications can be found in Vacondio et al. (2016) and Dazzi et al. (2018, 2019).

In the CUDA framework, the basic work unit is the *thread*, which corresponds to one computational cell of the domain, and many threads are grouped into a *block*, each containing $K \times K$ cells (K=16 in this work). The PARFLOOD model can discretize the domain using either a Cartesian computational grid or a Block Uniform Quadtree (BUQ) grid, which is a non-uniform structured mesh specifically designed for guaranteeing efficiency on the GPU architecture (see Vacondio et al., 2017 for details). The CUDA code takes advantage of two hardware resources: the CPU (the *host*), and the GPU (the *device*). The former controls the execution flow (including the memory exchanges between the device and the host), keeping track of the time advancement during the simulation, while the latter executes the different pieces of code (*kernels*) containing instructions for the actual computations. During each time step (from t^n to t^{n+1}), the code performs a sequence of three main tasks, as summarized in Figure 1, each of which includes CPU instructions for launching one or more kernels. Before describing the implementation of IBCs in detail (Section 2.2), some features of the original model are briefly recalled (see Vacondio et al., 2014 for further details).

2.1.3 External boundary conditions

Boundary conditions (BCs) are handled by adopting the implicit local ghost approach, which does not require the explicit modeling of ghost cells, since ghost values at boundary edges are retrieved at runtime thanks to appropriate instructions included in the kernels. Information concerning the existence and type of BC for each cell must be stored in additional data structures, and later accessed during kernel execution to correct the neighbor values with appropriate expressions (e.g. reflective/transmissive) and compute boundary fluxes. Please notice that different boundary conditions can be associated to different edges of the same cell without any conflicts, and this makes the simulation of complex boundaries in field applications straightforward and efficient.

While wall and far-field BCs are handled directly by the flux computation kernel, in the case of open boundaries (discharge or level hydrograph, stage-discharge relationship, etc.) some conserved variables must be imposed at boundary cells (Task #1 in Figure 1). For an inflow BC, discharge values are interpolated from the discrete time series provided as input data and converted to specific discharges by CPU processing, then a dedicated kernel is launched over the boundary cells to assign the prescribed value of specific discharge (water elevation is also imposed for supercritical inflow). For outflow boundaries, the water level is prescribed for subcritical conditions, while no assignments are performed for supercritical outflow. Finally, a stage-discharge relationship can also be imposed as downstream BC: based on the average water level at boundary cells, the value of discharge is interpolated from the discrete table provided as input data, and the specific discharge is redistributed over boundary cells.

2.1.4 Δt computation and Block Deactivation Optimization (BDO)

In a dedicated kernel, launched during Task #2 in Figure 1, the allowable time step is computed for each cell, and the minimum for each block is determined. Then, the CPU calculates the minimum for the whole domain. At the same time, the kernel also identifies whether a block is wet. Dry blocks are then deactivated (excluded from the computations during the current time step) by CPU processing, in order to avoid unnecessary calculations and thus reduce the computational burden.

2.1.5 Flux computation and time integration

For each cell, a kernel retrieves the left and right states at each of the four edges (north/south/east/west) of a cell, computes the four fluxes using the HLLC solver, and updates the vector of the conserved variables according to Eq. (3). It must be noted that the algorithm actually repeats the calculation of each intercell flux twice (once for each thread sharing that edge), because this strategy was found to be more efficient on the GPU than introducing a separate kernel for flux computation and allocating an extra array for fluxes on the global memory. However, conservation is guaranteed by the fact that the same left/right values are used for both computations. For the first-order approximation, these values coincide with the conserved variables at the center of the current cell and of its four neighbor cells. For the second-order scheme, additional kernels for MUSCL extrapolation and half-step summation are necessary (see Vacondio et al., 2014). For boundary conditions, specific procedures allow the correct definition of the conserved variables in neighbor cells used for flux computation.

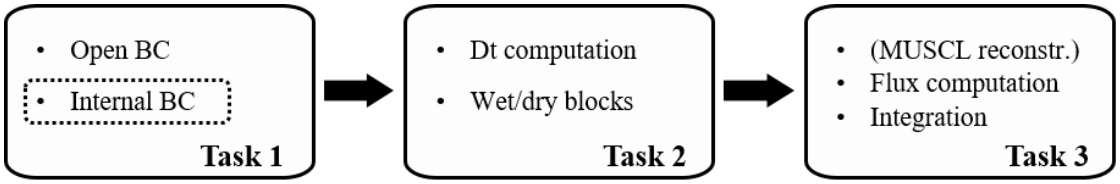

Figure 1. PARFLOOD model: sequence of operations performed during each computational time step.

2.2. Internal boundary conditions

In the present section, the details of the algorithm adopted to introduce the internal boundary conditions in the PARFLOOD solver are described. The planimetric position of a hydraulic structure is identified by means of a line or polyline, which is discretized through a number of cell edges at the pre-processing stage (see Figure 2). Cells facing at least one of these internal edges are then identified, and marked as "upstream" or "downstream" IBC cells.

The case of a bridge-type IBC will be described in the following, whereas other types of hydraulic structures will be considered at the end of the section. The input data must include the bridge geometry (Figure 3), in particular the low chord elevations: a "ceiling" elevation for pressurization is assigned to each IBC cell, so that complex geometries like single or multiple arched bridges can be described adequately. An overtopping elevation is also provided. Abutments and piers can be either handled as holes in the mesh or included in the bridge geometry as cells with a ceiling coinciding with the bottom elevation ("zero opening", see Figure 2).

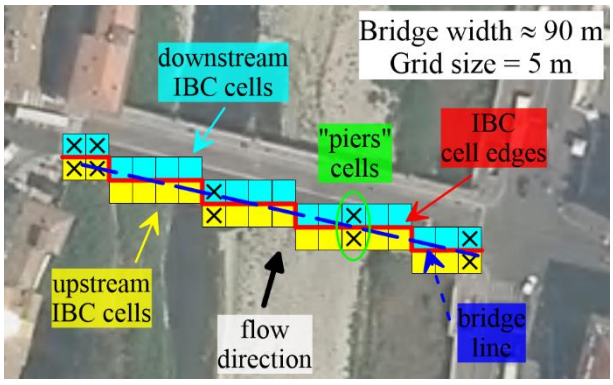

Figure 2. Sketch of IBC elements for a bridge (orthoimage in background). The bridge is identified by the dashed blue line, which the model discretizes as IBC cell edges (red). Upstream and downstream IBC cells are colored in yellow and cyan, respectively. The "zero opening" cells (piers and abutments) are marked with a cross (see for example the ones circled in green).

2.2.1 Selection of flow conditions and discharge estimation

The basic idea is to compute the average water level upstream and downstream of the bridge/structure, and to compare it with the minimum and maximum ceiling elevations, in order to distinguish different flow conditions (low flow, partially pressurized, fully pressurized, overtopped, see Figure 3). These values are then used for estimating the discharge flowing through the bridge by adopting a discharge formula. This approach differs from previous works concerning IBCs in 2D models, where the discharge formula was employed at each cell edge separately (Morales-Hernández et al., 2013; Echeverribar et al., 2019). Although the strategy of using a "global" evaluation is closer to 1D modelling, it is more robust and efficient, and also more flexible to different types of structures (for which "global" rating curves are sometimes available from physical modelling).

For low flow conditions (no interaction with the deck), IBCs are not activated, and all cells behave as ordinary inner cells. Conversely, for high flow conditions the (specific) discharge is imposed in upstream IBC cells. Please note that the water level is not modified in any cell, so that mass is neither added nor subtracted to the domain. The same specific discharge assigned to each upstream IBC cell is replicated in its companion downstream IBC cell. The model assumes that the flow is normal to the IBC edge, hence the specific discharge in the tangential direction is set equal to zero. This hypothesis may induce directionality to flow, which can be particularly noticeable in the case of skewed bridges, but in the meantime it guarantees that mass is conserved through the IBC edge. Moreover, this effect is expected to affect the flow field only locally, while the overall prediction of flood propagation and backwater should not be much influenced.

More in detail, the IBC processing in the PARFLOOD model starts during Task #1 (Figure 1), after open BCs have been assigned. A first kernel is launched over blocks containing IBC cells, with the aim of computing the mean levels upstream (η_U) and downstream (η_D) of the bridge. In particular, the partial summation of water surface elevations in upstream (and downstream) IBC cells for each block is performed during kernel execution, based on a logarithmic reduction (Vacondio et al., 2014); then, these partial values are passed to the CPU, which combines them to compute the result. Some discharge formulae require the additional definition of the total heads (H_U and H_D); these average values are also evaluated if necessary.

Based on the computed mean levels and on the bridge geometry, the flow condition during the current time step is selected by CPU processing. Let us define $Z_{low,min}$ and $Z_{low,max}$ the minimum and maximum elevations of the bridge low chord, Z_{top} the elevation of the top chord, and Z_{bottom} as the minimum bottom elevation along the bridge section (Figure 3). First, a check on the flow direction is performed in order to capture possible flow reversal, which can be observed at bridges immediately upstream of

a retaining structure or a confluence. If $\eta_U > \eta_D$ (most common case), one of the following conditions is verified:

(1) If $\eta_U < Z_{low,min}$, low flow conditions occur and IBCs are ineffective. A simple kernel, which sets to zero the specific discharge normal to upstream and downstream "zero opening" cells (if present), is launched.

(2) If $Z_{low,min} < \eta_U < Z_{low,max}$, the bridge is partially pressurized. Only a subset of IBC cells interacts with the bridge deck. This partial obstruction is taken into account by reducing the normal specific discharge $q_\perp$ in the upstream cells whose water surface elevation exceeds the local ceiling ($Z_{low,k}$ for cell *k*, for example), as follows:

$$\overline{(q_\perp)_k} = (q_\perp)_k \frac{Z_{low,k} - z_k}{\eta_k - z_k} \qquad \text{if } \eta_k > Z_{low,k} \tag{4}$$

where $q_\perp$ coincides with *uh* or *vh* depending on the IBC edge inclination (both components are modified if the cell shares two IBC edges). Then, the same specific discharge (overlined) is assigned to the corresponding downstream IBC cell. Note that the specific discharge in all IBC cells with $\eta < Z_{low}$ remains unchanged, while the specific discharge in "zero opening" cells (e.g. with $Z_{low,k} \leq z_k$) is zeroed.

(3) If $Z_{low,max} < \eta_U < Z_{top}$, the bridge is fully pressurized. Two possible flow conditions, namely free-flow and submerged flow, can occur. In the former case, the discharge *Q* passing through the bridge is evaluated as follows:

$$Q = C_d A \sqrt{2g\left(H_U - 0.5\left(Z_{low,max} - Z_{bottom}\right)\right)} \tag{5}$$

For submerged flow, on the other hand, the pressure flow equation is used:

$$Q = C_Q A \sqrt{2g(H_U - \eta_D)} \tag{6}$$

In Eqs. (5) and (6), *A* represents the total area of the bridge opening, while C_d and C_Q are discharge coefficients for the two flow conditions. Typical values used in the practice are 0.5 and 0.8 (Bradley, 1978).

The selection of the appropriate configuration is performed following Cozzolino et al. (2015), namely by evaluating a tentative discharge by means of Eq. (5) and computing the depth at the *vena contracta* and its conjugate depth; if the latter is exceeded by the downstream depth, the flow is submerged, otherwise free-flow conditions occur. Please notice that, in the free-flow case, the discharge is computed by adopting Eq. (5), but the downstream depth is not set equal to the depth at *vena contracta*. Indeed, this operation would violate mass conservation, unless a special procedure is introduced (as done by Echeverribar et al., 2019, as regards the imposition of the critical depth downstream of a weir), which is not straightforward in an

efficient GPU implementation. Moreover, the drawback of neglecting this contraction downstream of bridges is common to other approaches (Ratia et al., 2014) aimed at capturing the global effect of the bridge presence (backwater) rather than the local details of the flow field.

(4) If $\eta_U > Z_{top}$, the bridge is also overtopped. Compared to the previous case, the discharge is increased according to the weir equation:

$$Q = C_W L\sqrt{2g}\left(H_U - Z_{top}\right)^{3/2} \tag{7}$$

where L is the bridge length normal to flow, and C_W is a weir coefficient.

Please notice that the local head losses at the bridge site are represented by the discharge coefficients for case (3). On the other hand, the strategy of imposing a reduced discharge based on the area available to flow in IBC cells for case (2) is preferred over the introduction of an additional head loss in order to guarantee smooth transitions from low flow to partially pressurized flow conditions, and between adjacent pressurized and non-pressurized cells. Moreover, no additional calibration coefficients (which lack experimental grounds) are required for this simple and computationally efficient approach.

For cases (3) (bridge fully pressurized) and (4) (bridge fully pressurized and overtopped), based on the evaluated total discharge, and on the length and orientation of the IBC string (please recall that the flow direction is assumed normal to the bridge), the specific discharge q is computed and redistributed over the upstream IBC cells. The new values of the conserved variables to be imposed for a generic upstream IBC cell k are as follows:

$$\overline{(q_\perp)_k} = q\frac{h_{avail,k}}{\sum_{k=1}^{N} h_{avail,k}}, \tag{8}$$

where, as before, $\overline{(q_\perp)_k}$ coincides with uh or vh depending on the IBC edge inclination (both components are considered for cells with two IBC edges), while N represents the number of IBC edges. Moreover, $h_{avail,k}$ is the weight of cell k, coinciding with the available flow area (divided by the grid size), which can be evaluated as $Z_{low,k} - z_k$ for case (3), and $\left(Z_{low,k} - z_k\right) + \left(\eta_k - Z_{top}\right)$ for case (4). Note that for "piers" cells $h_{avail,k}$ is zero for case (3), and $\left(\eta_k - Z_{top}\right)$ for case (4). If a cell shares two IBC edges, both specific discharge components are actually modified based on $h_{avail,k}$. Then, the same values of the specific discharge are attributed to downstream IBC cells, as in case (2). The water level is never modified in IBC cells during this redistribution. For example, let us consider the simple case of two neighbor cells (i,j) and (i+1,j), separated by an IBC edge (Figure 4a). While $\eta_{i,j}$ and $\eta_{i+1,j}$ are not modified, the following values are assigned to the specific discharges in the x-

and *y*-directions: $(uh)_{i,j} = (uh)_{i+1,j} = \overline{uh}$; $(vh)_{i,j} = (vh)_{i+1,j} = 0$, where $\overline{uh}$ is the result of the discharge redistribution for cell (*i*,*j*) according to Eq. (8).

If the flow is reversed (downstream towards upstream), all the previous checks are performed considering η_D instead of η_U, and swapping upstream and downstream IBC cells.

Finally, for other types of internal structures, the flowchart for selecting the appropriate flow condition is modified according to the type of IBC considered. For example, only cases (1) and (3) need to be considered for a sluice gate. For complex structures (e.g. flow-through dams), a stage-discharge relationship obtained from physical modelling is often available, and can be provided in tabular form as input data: checks (1)-(4) are skipped, and the discharge is simply obtained from the rating curve valid for the considered hydraulic structure, then the modified conserved variables at the IBC cell can be computed by means of Equation (8). More complex cases could be handled by supplying multiple rating curves and instructions on how to switch between them (e.g. gate operations).

2.2.2 Flux computation

As regards flux computation and cell updating (Task #3 in Figure 1), the IBC implementation requires few changes to the original scheme, thanks to the implicit local ghost cell approach already implemented for external BCs.

In fact, the most important modification to the usual scheme for IBC cells lies in the definition of left and right states for an IBC edge, while all fluxes are then computed using the HLLC Riemann solver as in standard cells. It is worth recalling that, in the PARFLOOD implementation, each cell computes its own fluxes, hence each flux is actually calculated twice: in the framework of IBCs, this peculiarity is exploited to take into account the bridge discontinuity. In particular, different values of the left/right states are used to compute fluxes at the IBC intercell in the upstream and downstream cell, so that, limited to these particular edges, the fluxes exiting from the upstream cell are not the same as those entering the downstream cell. With reference to the simple case of cells (*i*,*j*) and (*i*+1,*j*) (Figure 4b), let us consider cell (*i*,*j*) first. The fluxes to the east are computed using the conserved variables at the cell center as left state, while the right state is simply determined by mirroring the left state. The result of the HLLC solver with the same left and right states coincides with the physical flux:

$$\mathbf{F}^{U}_{i+\frac{1}{2},j} = \begin{bmatrix} \overline{uh} \\ \overline{uh}h_{i,j} + \frac{1}{2}g(\eta^2 - 2\eta z)_{i,j} \\ 0 \end{bmatrix} \quad (9)$$

Focusing on cell (*i*+1,*j*), the same strategy of mirroring the centered values is applied to define the left state at the western edge, and the physical flux is the following:

$$\mathbf{F}^{D}_{i+\frac{1}{2},j} = \begin{bmatrix} \overline{uh} \\ \overline{uh}h_{i+1,j} + \frac{1}{2}g(\eta^2 - 2\eta z)_{i+1,j} \\ 0 \end{bmatrix} \tag{10}$$

From a comparison between the two fluxes, it can be noticed that the first component is the same for both computations, meaning that mass conservation is always guaranteed. The difference between the momentum fluxes (a part from a possible bottom step discontinuity, which is balanced by a proper discretization of the bed slope source term) equals the reaction exerted by the deck onto the fluid, assuming a hydrostatic pressure distribution (see Appendix A). Please note that other models also introduce violations of the momentum conservation in order to consider the presence of a structure: Ratia et al. (2014) introduced the presence of a bridge as a "momentum sink" (additional source term) using the HL approach. In Maranzoni and Mignosa (2017), a fictitious cell was inserted to mimic the unbalance in the momentum equations caused by a ceiling step discontinuity, while in Cozzolino et al. (2015), different physical fluxes were imposed directly when updating cells separated by a gate.

The advantage of the proposed approach, compared to Zhao et al. (1994), Cozzolino et al. (2015) and Maranzoni and Mignosa (2017), is the fact that code branching is very limited, since basically only the introduction of an additional type of boundary condition, handled by means of the implicit local ghost approach, is required, while the call to the HLLC solver remains unchanged. This is also convenient for enabling both high and low flow conditions in IBC cells: in the latter case, the mirroring of centered values is simply skipped, so that left/right states at the IBC edge are retrieved from the conserved variables in the current cell and in its neighbor as in standard cells.

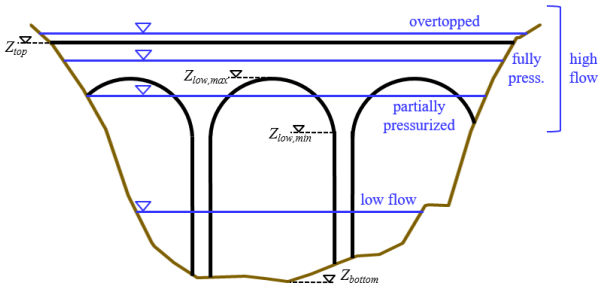

Figure 3. Sketch of a bridge geometry and types of flow condition.

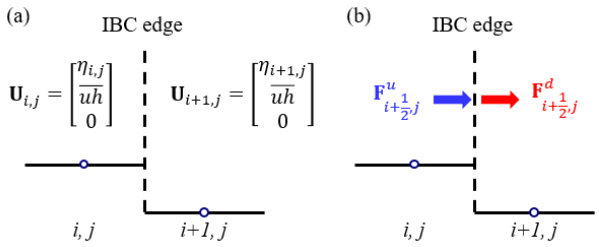

Figure 4. Example of IBC edge, dividing upstream cell (i,j) and downstream cell (i+1,j): (a) assignment of redistributed specific discharge values, normal to the IBC edge; (b) computation of different fluxes at IBC edge (i+1/2,j).

3. Model validation

In this section, the results of numerical tests for model validation are reported. All simulations are performed using a P100 Tesla® GPU.

3.1. Steady flow experiments

3.1.1. Picek et al. (2007)

A subset of the laboratory tests presented by Picek et al. (2007) was simulated for the model validation in high flow conditions. Experiments were carried out in a flume with a 0.75 m-wide rectangular cross-section and a horizontal bottom, where a 0.3 m-long, 7.5 cm-thick bridge was placed 1.1 m from the downstream end of the channel. Different steady-state subcritical conditions, all characterized by pressurized flow through the bridge, were realized by varying the inflow discharge, the deck's vertical position and the water level at the downstream end of the flume. Table 1 reports the 9 test conditions here considered. In particular, tests labelled P1 correspond to pressurized conditions, while P2 and P3 refer to overtopping with unsubmerged or submerged flow, respectively. In the simulations, the domain is discretized by means of cells of size 0.05 m, while Manning's coefficient is set equal to 0.01 $m^{-1/3}s$. A constant inflow discharge and the measured water level are imposed as upstream and downstream boundary conditions, respectively. While for the weir coefficient in Eq. (7) the value 0.32 is assumed, the discharge coefficient in Eq. (6) is set equal to 0.97 after trial-and-error. This value is higher than the usual range (0.7-0.9); however, Picek et al. (2007) showed that, for their deck geometry, the actual range for C_Q is 0.8-0.95 for a partially submerged bridge, and up to 1.0-1.1 for an overtopped bridge; the discrepancy was there justified by the absence of side contractions in the test geometry.

Measured and simulated values of the average water depth upstream of the bridge are compared in Table 1, together with the relative error, which is always below 5%. These results confirm the capability of the model to reproduce the backwater induced by the presence of the deck. A sensitivity analysis to the discharge coefficient was performed for Test P1a, and results show that relative errors of the upstream water level remain below ±10% as long as C_Q is higher than 0.88 (up to 1.1), while overtopping is wrongly predicted for lower values.

Predicted water surface profiles along the channel centerline are analyzed for test cases of type "b" (Figure 5). The comparison between measured and numerical profiles shows that, as expected, the near-field features around the bridge are not reproduced accurately by the model, especially the local water surface upsurge at the upstream face of the deck (probably due to the head loss for the vertical flow contraction) and the flow expansion profile downstream. Actually, the flow through a bridge is characterized by significant accelerations in the vertical direction, which cannot be simulated adopting a depth-averaged model. Also as regards the overtopping tests, the IBC model is unable to reproduce the profile over the deck: the flow field here consists of two independent layers below and above the deck, and, clearly, a shallow water model cannot represent this configuration. However, the backwater profile which develops upstream of the bridge is well caught for all flow conditions; hence, the model is able to predict the overall effect induced by the deck. This can be evaluated quantitatively by computing the root mean square error (RMSE) of the numerical water surface profile along the channel compared to the experimental one (Table 1). Values of RMSE below 1 cm (less than 4% of the mean water level in the channel), except for Test P1a, confirm the quite good performance of the model.

Table 1. Experiments by Picek et al. (2007): test conditions and results.

Test	Q (m^3/s)	Z_{low} (m)	Z_{top} (m)	η_d (m)	η_u exp (m)	η_u sim (m)	η_u error	RMSE h (m)
P1a				0.182	0.202	0.211	4.4%	0.011
P2a	0.111	0.150	0.225	0.248	0.265	0.271	2.4%	0.009
P3a				0.334	0.339	0.342	1.0%	0.007
P1b				0.243	0.254	0.256	0.8%	0.008
P2b	0.155	0.225	0.300	0.325	0.339	0.343	1.2%	0.009
P3b				0.409	0.415	0.416	0.3%	0.008
P1c				0.323	0.328	0.332	1.1%	0.008
P2c	0.184	0.300	0.375	0.405	0.416	0.418	0.5%	0.007
P3c				0.522	0.526	0.526	-0.1%	0.004

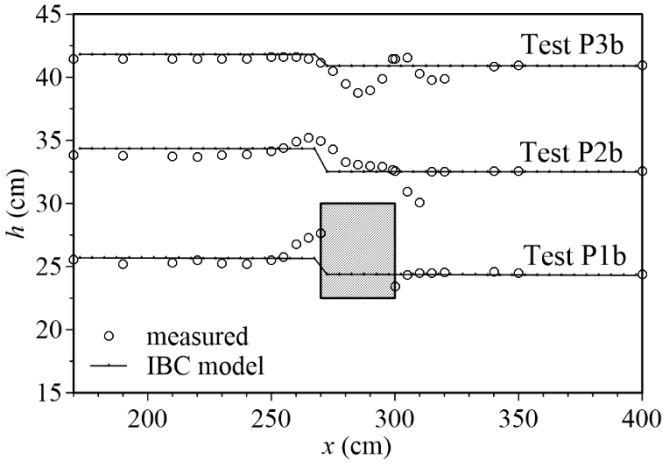

Figure 5. Experimental and numerical water surface profiles for test cases of type "b" by Picek et al. (2007).

3.1.2. Brown (1985)

The laboratory investigation performed at the HR Wallingford laboratory and reported by Brown (1985) was selected for validating the model in the case of arched bridges. A large number of experimental runs in steady flow conditions were performed in a 15 m-long channel with variable width where a bridge was inserted 7.6 m from the downstream end. Different bridge configurations, inflow discharges and tailwater levels were considered. The three types of bridge (simple arch unit "HR1", widened arch unit "HR2", and multiple arches "HR3") selected for this work are represented in Figure 6.

A large subset of these tests, all characterized by fully pressurized conditions with no overtopping, were simulated with the proposed model, assuming $\Delta x=\Delta y=0.02$ m, and Manning's coefficient equal to 0.01 $m^{-1/3}s$. Test conditions are summarized in Table 2. Bridge piers are modelled as cells with "zero opening" in the IBC model.

Measured and simulated upstream levels, compared in Table 2, are in good agreement; in fact, the relative error is always below 3%. These results were obtained using a calibrated value for C_Q, set equal to 1.0, 0.86, and 0.92 for bridge types HR1, HR2, and HR3, respectively. These values were selected after performing a sensitivity analysis of the upstream water depth to C_Q for one test for each bridge geometry (namely, tests 4D, 19C and 29B), which allowed selecting the most suitable value. It was also observed that relative errors of the upstream water level were always below 10% considering discharge coefficients in the range 0.8-1.10 for tests 4D and 29B, and 0.7-1.0 for test 19C. This suggests that, even though the predicted backwater somehow depends on the discharge coefficient when the bridge is fully pressurized, moderate variations of this parameter still guarantee acceptable results.

As described in Section 2.2, the model redistributes the specific discharge based on the available area. For arch bridges in pressurized conditions, this treatment gives origin to a markedly 2D flow field upstream of the bridge, as can be observed in Figure 7 as regards test 29B. At bridge piers, the water surface level slightly increases and velocity falls to zero, while the maximum velocity is noticeable at the center of arches.

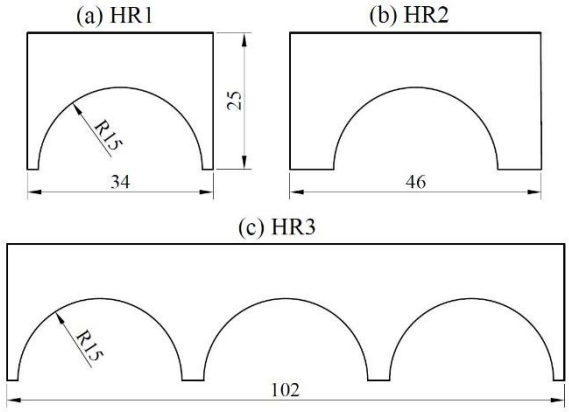

Figure 6. Bridge geometry for the test cases by Brown (1985). All dimensions are in cm. The bridge length is 4 cm.

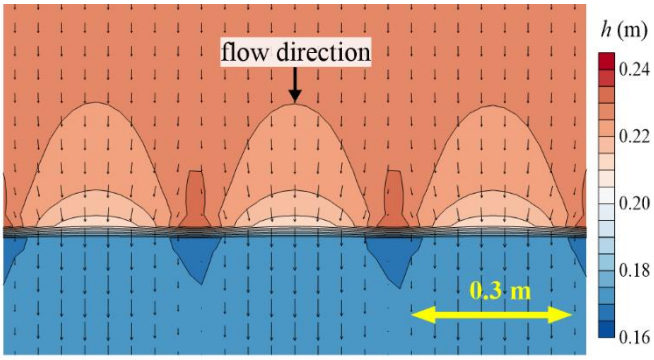

Figure 7. Local flow field around the bridge for test 29B by Brown (1985): contour map of water depth and velocity vector map (only one vector out of four is represented for clarity).

Table 2. Experiments by Brown (1985): test conditions and results.

Type	Test	Q (m^3/s)	η_d (m)	η_u exp (m)	η_u sim (m)	η_u error
HR1	2E	0.01	0.1849	0.1875	0.1880	0.3%
	2F	0.01	0.2108	0.2136	0.2139	0.1%
	3D	0.025	0.1807	0.1989	0.2009	1.0%
	3E	0.025	0.2175	0.2379	0.2395	0.7%
	4C	0.035	0.1698	0.2043	0.2099	2.7%
	4D	0.035	0.1957	0.2363	0.2384	0.9%
HR2	14E	0.0104	0.1796	0.1847	0.1849	0.1%
	14F	0.0104	0.2134	0.2192	0.2189	-0.1%
	14G	0.0103	0.2386	0.2447	0.2441	-0.2%
	15E	0.0265	0.1773	0.2115	0.2127	0.6%
	15F	0.0264	0.1988	0.2362	0.2350	-0.5%
	15G	0.0262	0.2105	0.2483	0.2466	-0.7%
	16E	0.029	0.1794	0.2203	0.2220	0.8%
	16F	0.0285	0.2050	0.2482	0.2477	-0.2%
	17D	0.035	0.1674	0.2276	0.2292	0.7%
	17E	0.0347	0.1876	0.2465	0.2501	1.5%
	18E	0.038	0.1656	0.2385	0.2388	0.1%
	19C	0.04	0.1611	0.2467	0.2417	-2.0%
HR3	21D	0.0029	0.2011	0.2022	0.2016	-0.3%
	21E	0.0029	0.2444	0.2449	0.2450	0.0%
	23C	0.0253	0.1906	0.1942	0.1933	-0.5%
	23D	0.0257	0.2289	0.2333	0.2319	-0.6%
	24C	0.035	0.1813	0.1863	0.1864	0.1%
	24D	0.034	0.2152	0.2218	0.2204	-0.6%
	24E	0.0358	0.2417	0.2494	0.2477	-0.7%
	25C	0.0443	0.1695	0.1768	0.1775	0.4%
	25D	0.0441	0.2048	0.2143	0.2135	-0.4%
	25E	0.0441	0.2344	0.2453	0.2435	-0.7%
	26C	0.0617	0.1933	0.2115	0.2101	-0.7%
	26D	0.0608	0.2211	0.2413	0.2382	-1.3%
	27B	0.0795	0.1704	0.1976	0.1973	-0.2%
	27C	0.0792	0.2053	0.2391	0.2341	-2.1%
	29B	0.11	0.1721	0.2270	0.2262	-0.4%

3.2. Unsteady flow experiments

The performance of the model in case of unsteady flows is assessed by simulating some of the experimental dam-break test cases reported by Ratia et al. (2014). The laboratory set-up consisted of a 0.84 m-wide reservoir, filled with still water, and a 6 m-long, 0.24 m-wide channel, closed downstream (Figure 8a); the instantaneous lifting of a gate between channel and reservoir generated a dam-break wave. A bridge was positioned approximately 1.65 m downstream of the gate, and different bridge shapes were employed (Figures 8b-d). The water level was measured at two locations, upstream and downstream of the bridge (G1 and G2 in Figure 8a).

Table 3 summarizes the test conditions for the three cases considered in this work. Numerical simulations are performed adopting a uniform 0.015 m grid resolution, while the value 0.0125 $m^{-1/3}s$ for Manning's coefficient was calibrated to fit the arrival time of the dam-break wave at the upstream gauge.

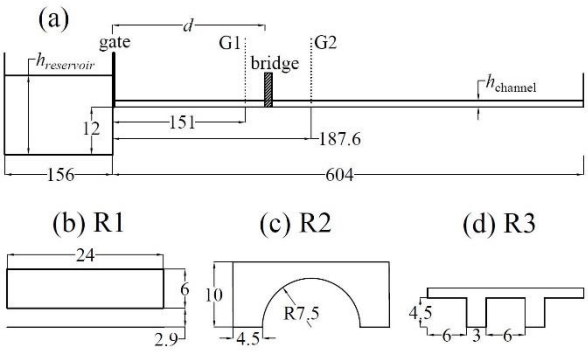

Figure 8. Experimental setup for tests by Ratia et al. (2014). All dimensions are in cm. (a) Lateral view: the reservoir and channel widths are 84 cm and 24 cm, respectively; G1 and G2 are the upstream and downstream water level gauges. (b-d) Bridge geometry (length is 6 cm).

Table 3. Experiments by Ratia et al. (2014): test conditions.

Test	$h_{reservoir}$ (cm)	$h_{channel}$ (cm)	d (cm)
R1	21.0	0.9	166.1
R2	22.7	1.0	164.6
R3	22.5	0.9	163.6

The first test case (R1) concerns a deck bridge without piers, and the results of numerical simulations, including or excluding IBCs, are compared with experimental measurements in Figure 9. The description of the bridge through an IBC allows predicting the backwater upstream, which, as expected, cannot be modelled in the "base" simulation without IBC. The discharge coefficient in Eq. (6) was set equal to 0.75 after calibration. In particular, its value was varied in the range 0.65-0.85, and the envelope of the obtained results is represented in Figure 9 as a grey band. The predicted maximum water depth at gauge G1 varies in the range ±7% (compared to the simulation with the calibrated value). It should be kept in mind that discharge formulae used at the IBC are valid for steady flow (or slowly varying flow), while the impact of a dam-break wave is highly unsteady and entails dissipations that cannot be easily included in a shallow water model. Despite this, the IBC model can capture the overall process adopting any value for C_Q, while tuning this parameter allows improving the agreement with experimental data.

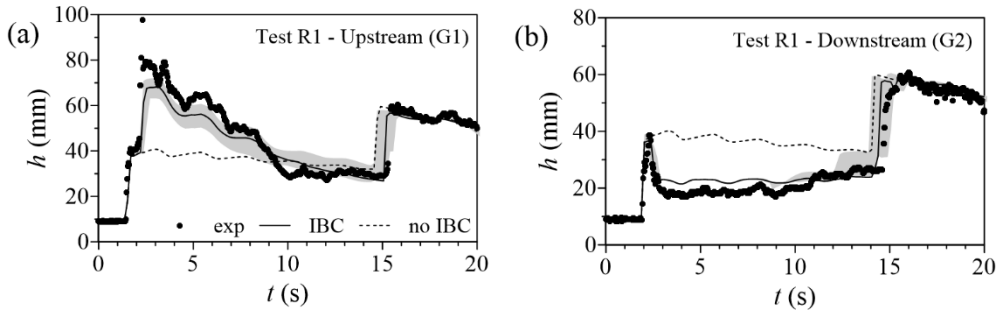

Figure 9. Experimental water depth measurements at gauges G1 and G2 for test R1 by Ratia et al. (2014), and comparison with numerical results (with and without IBC). The grey band represents the sensitivity of IBC model results to C_Q.

Two additional test cases were considered, concerning an arch bridge (R2) or a deck with two piers (R3). For these tests, three approaches were compared: (1) piers as holes in the mesh (MD approach), neglecting the influence of the deck; (2) IBC approach, where piers are included as IBC cells with "zero opening"; and (3) a combination of MD and IBC, where piers are holes in the mesh, but the deck is also included as an IBC. Notice that, in these experimental setups, the main flow obstruction is represented by the bridge piers, and good numerical results may be expected even adopting the MD approach. Moreover, for these test cases, the discharge coefficient does not influence the results, since the bridge barely becomes pressurized during the process.

Figure 10 reports the numerical results along with experimental data. The upstream water levels are reproduced quite well in all simulations, even if the peak is slightly underestimated. Conversely, only approaches (1) and (3) allow a good prediction of the downstream water levels, because the flow contraction due to the piers is relevant in these configurations, but it cannot be described properly assuming a "thin" structure, as in approach (2). This limitation is common to other methods (e.g. HL described by Ratia et al., 2014), and is related to the impossibility of modelling the supercritical flow downstream of the bridge without a detailed reproduction of contraction and expansion through the piers (especially for Test R2). Despite this, the overall process is in reasonable agreement with experiments even with this simplified approach for describing bridge piers. Small oscillations downstream at $t \approx 14$ s are related to the complex flow field due to the interaction of the flow passing through the bridge with the surge wave travelling upstream (generated after the dam-break wave impacts the wall at the end of the channel).

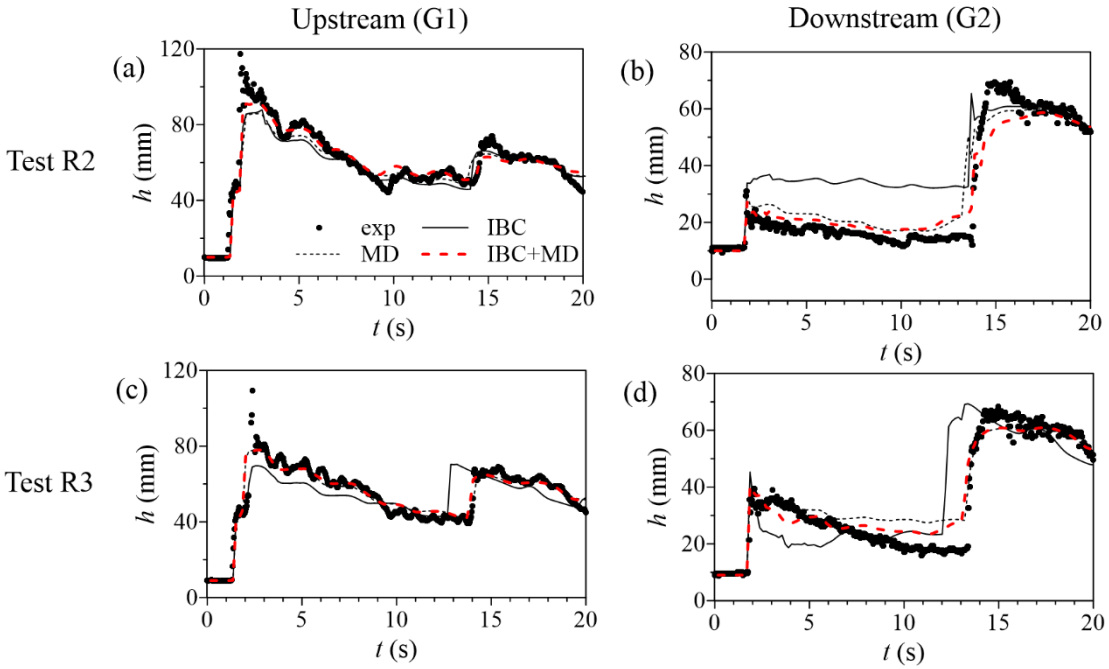

Figure 10. Comparison between experimental measurements and numerical results for the water depth time series at gauges G1 and G2 for tests (a-b) R2, and (c-d) R3 by Ratia et al. (2014).

4. Application

The application of the proposed model to a field test case is here presented. The middle reach of the Parma River (Northern Italy), which flows through the city of Parma, is selected as case study. A sketch of the study area is represented in Figure 11. Two sets of tests are carried out: the first (tests labelled “F”) concerns the simulation of the 2014 flood event in the urban reach of the river, which is crossed by five arched bridges with piers; the second one shows the use of an internal rating curve to represent the flow-through dam of the flood detention basin which protects the town.

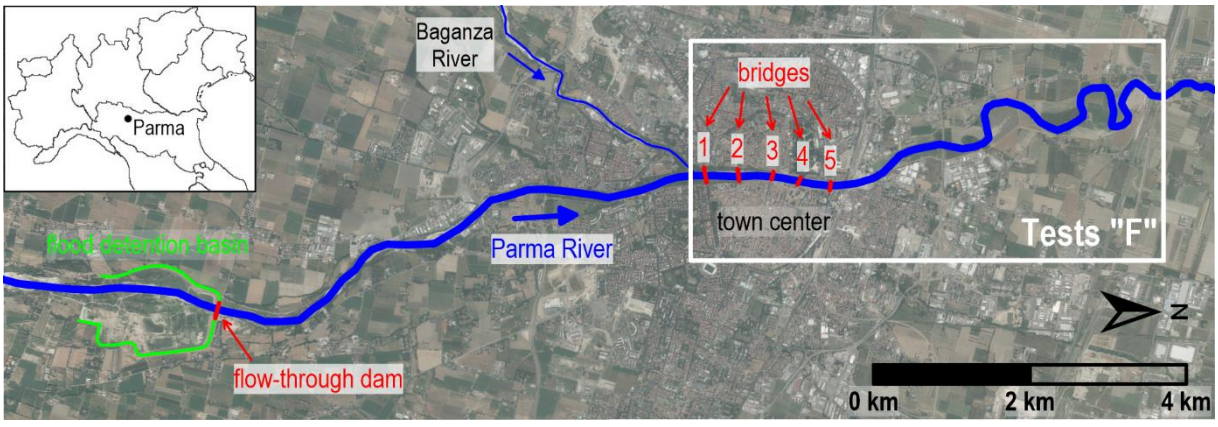

Figure 11. Study area for the Parma River test cases.

4.1. 2014 flood event

The model is used to simulate the flood event which occurred on October 13th, 2014 on the Parma and Baganza Rivers. This first application (tests labelled "F") is focused on the flow interaction with the five historical bridges on the Parma River, hence the domain is limited to the 6 km-long river reach identified in Figure 11. The upstream boundary condition is the estimated discharge hydrograph immediately downstream of the confluence of the two rivers, with a peak value equal to 820 m^3/s (Figure 12a), while a rating curve is imposed as downstream boundary condition at the highway crossing. The geometry of bridge #2 is reported in Figure 12b as an example.

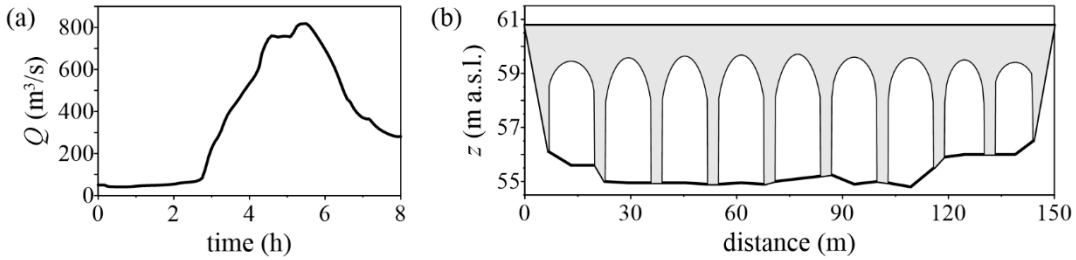

Figure 12. Parma River test case: (a) Inflow hydrograph for the 2014 flood event; (b) geometry of bridge #2.

Four simulations were performed, adopting either the MD approach or the IBC method for representing the bridges (see test conditions in Table 4), and discretizing the domain with square cells of resolution equal to 1 m (5.5 million cells) or 2 m (1.4 million cells). Manning's coefficient in the urban reach of the river was set equal to 0.03 $m^{-1/3}s$. For the IBC tests, the discharge coefficient in Eq. (6) was set equal to 0.9. The physical time of simulation is 8 h.

Table 4. 2014 flood event on the Parma River: test conditions, RMSE of the maximum water surface profile with reference to the surveyed profile, and computational times.

Test	Approach	Δ*x* (m)	RMSE (m)	Computational time (min)	Physical/computational time ratio
F1	MD	1	0.23	21.5	22
F2	IBC	1	0.19	24.9	19
F3	MD	2	0.39	3.8	128
F4	IBC	2	0.18	5.1	93

Figure 13a compares the longitudinal profiles of the maximum water surface elevations along the river centerline obtained from different simulations. The surveyed maximum levels are also reported. The differences between simulated and observed maximum water surface elevations are plotted in Figure 13b for better comparison. Results show that, during this event, four bridges were partially pressurized during the peak flow, and this is predicted correctly by the IBC model adopting both 1 m and 2 m meshes (Tests F2 and F4). In fact, the simulated profiles of the maximum levels are in good

agreement with the high water marks. This is confirmed by the evaluation of the RMSE of the simulated maximum water surface elevations with reference to the surveyed maximum levels (reported in Table 4): the RMSE is below 0.20 m for these tests. The MD simulation with the finest resolution (Test F1) also provides good results (RMSE = 0.23 m), even though the maximum profile is slightly lower than the IBC tests, because in this case the flow interaction with the decks is not modelled. Conversely, the MD simulation with the coarse mesh (Test F3), which is characterized by the highest value of the RMSE (0.39 m), overestimates the backwater upstream of bridges. This is due to the fact that bridge piers are artificially "widened" adopting a 2 m-resolution: for example, the piers of bridge #2 are 3 m-wide (Figure 12b), and cannot be represented adequately unless the finer resolution is used.

Computational times for all tests are reported in Table 4. It can be noticed that, for Tests F1 and F2 (1 m grid), the simulation time required by the IBC method is roughly 15% higher compared to the MD approach. The overhead is due to the execution of additional kernels during the computations, but the IBCs do not restrict the allowable time step. However, the IBC model provides good results even for the coarser resolution (Test F4), which is four times faster, hence it can be considered an efficient alternative to the MD approach.

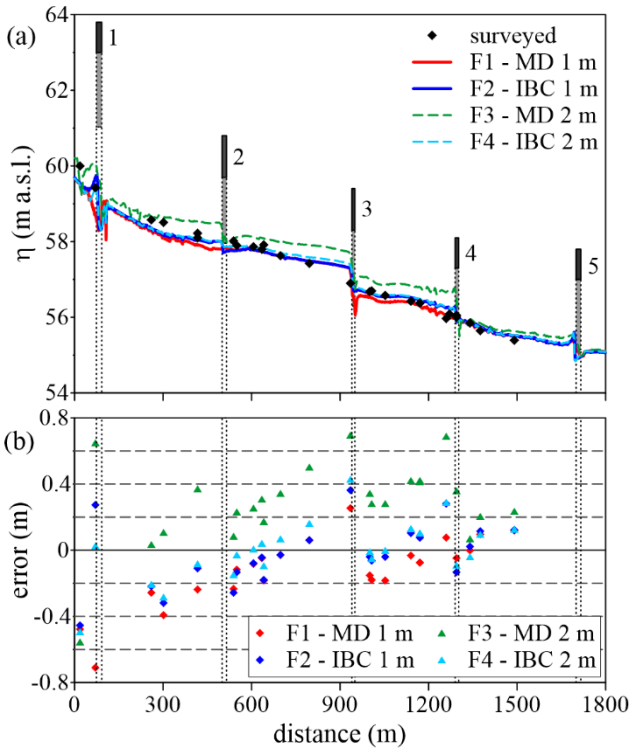

Figure 13. (a) Simulated longitudinal profiles of maximum water surface elevations and surveyed high water marks. The position of bridges is also reported: the elevations corresponding to partial and full bridge pressurization are filled with grey and black, respectively. (b) Differences between simulated and surveyed maximum water surface elevations.

4.2. Hypothetical flood scenario with 500 years-return period

The model is then used to simulate the hypothetical scenario of a 500 years-return period flood along the Parma River. The domain includes the flood detention basin and the 13 km-long river reach downstream (Figure 11). The contribution of the Baganza River is neglected for this application. The domain is discretized with square cells of size 2 m (3.7 million cells), and the bridges in the urban reach are modelled using the IBC approach similarly to Section 4.1.

The flood-control reservoir dam is composed of a central concrete body with two lateral spillways and three bottom openings, equipped with movable gates (D'Oria et al., 2012). The structure is included in the simulation using an internal rating curve as IBC (shown in Figure 14). In this application, gate operations are not considered, and then a unique rating curve is adopted.

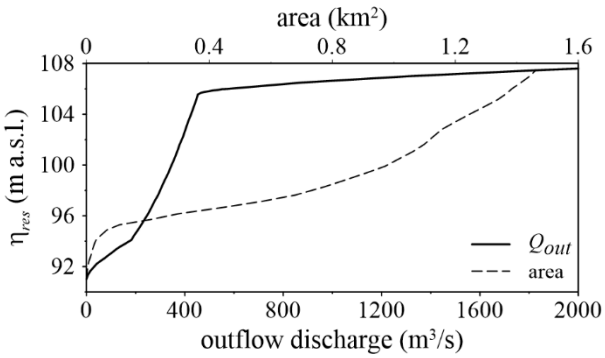

Figure 14. Internal rating curve for the dam on the Parma River, and stage-area curve for the reservoir.

A common approach for simulating the lamination in online flood detention basins is the Level-Pool-Routing (LPR) method (Chow et al., 1988), which also requires the definition of a stage-area relationship for the reservoir (see the curve shown in Figure 14, obtained from the digital terrain model). Figure 15a shows a comparison between the predicted discharge hydrograph flowing through the dam obtained from the LPR method and from the 2D simulation with IBC. The difference between the two predictions is negligible. The same observation is true as regards the stage hydrograph inside the reservoir (Figure 15b). In this application, velocities in the reservoir are close to zero, hence the zero-dimensional approximation of the LPR method is acceptable and provides results similar to a 2D model. However, for more complex cases (e.g. multiple basins, non-negligible dynamics inside the reservoir) the 2D model is expected to provide more accurate results compared to the LPR approach. Moreover, when detention basins are located halfway in the domain, coupling between different models can be avoided if IBCs are used.

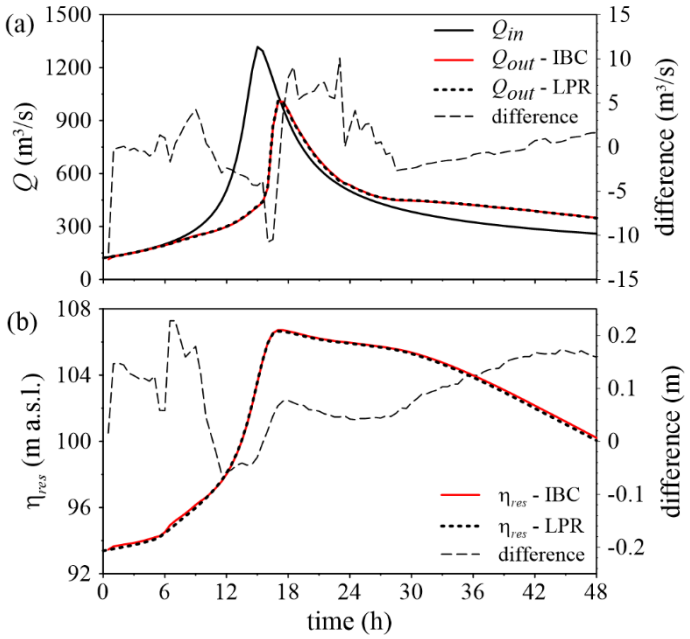

Figure 15. Comparison between numerical results of the 2D model with IBC and LPR method for the flood detention basin: (a) outflow discharge, and (b) water level in the reservoir. Differences between IBC and LPR results are also plotted.

As an example of simulation results downstream, Figure 16 shows the predicted water surface profiles in the urban reach of the river at selected times. The transition from low flow to partially and fully pressurized conditions can be observed at all the bridge sites except bridge #1. In particular, when the peak discharge is observed ($t \approx 18$ h), bridges #2 and #3 become fully pressurized, and the difference between the upstream and downstream water level is roughly 1 m.

A sensitivity analysis on the discharge coefficient C_Q in Eq. (6) was performed in order to verify its influence on the numerical results in real applications. Besides the reference value (0.9), the coefficients 0.8 and 1.0 were considered, and the comparison was performed by analyzing the longitudinal profile of the maximum water surface elevations along the 1.8 km-long urban reach of the Parma River. The RMSE of this profile (vs the reference simulation) is equal to 16 cm and 12 cm when C_Q is 0.8 and 1.0, respectively. The highest differences (±25 cm) are observed upstream of bridge #3. In general, the definition of the most suitable value for C_Q in field studies, for which calibration data are seldom available, is not straightforward. However, this application shows that its variation generates a change in the maximum water levels in the order of 15-25 cm (only upstream of fully pressurized bridges), which is in line with the possible inaccuracy in the numerical results due to the other sources of uncertainty (roughness, topography, boundary conditions, bridge geometry modification due to drift accumulation, etc.).

Finally, the simulation time is approximately 82 minutes (for 48 h of physical time), and the physical/computational time ratio (equal to 35) confirms the good performance of the PARFLOOD code for field simulations even when IBCs are included in the model. In fact, the simulation takes less than 10% more time compared to a companion simulation where bridges are excluded from the computations.

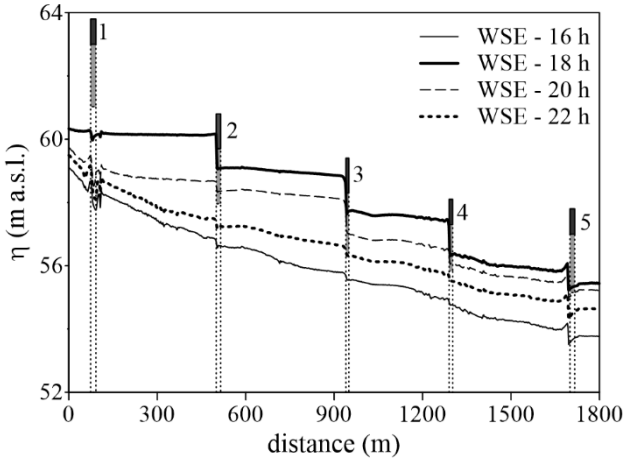

Figure 16. Simulated longitudinal profiles of water surface elevations at selected times. The bridge position is also reported: the elevations corresponding to partial and full bridge pressurization are filled with grey and black, respectively.

5. Conclusions

In this paper, the PARFLOOD model was extended with the implementation of internal boundary conditions to simulate the presence of bridges and other hydraulic structures in 2D domains. The validation tests showed that the adoption of IBCs provides good results both in low and high flow conditions for bridges, while the simulation of a real case confirmed the applicability of this approach for field-scale tests involving not only bridges, but also flood detention basins. The model is computationally efficient (physical/computational time ratio around 20-30), thanks to the GPU implementation, and can therefore be used for real simulations (e.g. for flood hazard maps, levee system verification, etc.) when the backwater upstream of bridges or the presence of structures interfering with the flow needs to be considered. For this kind of analysis, the drawback of not obtaining a very accurate representation of the flow field near the bridge can be surely acceptable. Further applications to real flood events may provide useful guidelines on the definition of the most suitable values for the empirical coefficients required by the model in case of fully pressurized flow.

Appendix A. Analysis of momentum conservation at IBC edges

Let us consider the example of a 1D channel with rectangular section (width B) and horizontal bottom, where a constant unit discharge q is flowing; a gate is placed midstream (Figure A1). The downstream level is high enough to obtain submerged conditions at the gate.

Assuming steady-state conditions and neglecting friction and head losses, the momentum balance applied to the control volume between section 1 and section 2 (Figure A1) reads:

$$P_1 - P_2 - F_g + M_1 - M_2 = 0 \tag{A1}$$

where P_1 and P_2 are the pressure terms on sections 1 and 2, while M_1 and M_2 represent the convective part of the momentum flux, and F_g is the (net) reaction of the gate. Assuming a hydrostatic pressure distribution, Eq. (A1) becomes:

$$\left(\frac{1}{2}\rho g h_1^2 B - \frac{1}{2}\rho g h_2^2 B\right) + (\rho q B u_1 - \rho q B u_2) - F_g = 0 \tag{A2}$$

where h_1 (h_2) and u_1 (u_2) represent the water depth and mean velocity at section 1 (2), and ρ is the water density. Hence, the reaction of the gate (per unit width per unit density) can be expressed as:

$$\frac{F_g}{\rho B} = \frac{1}{2} g(h_1^2 - h_2^2) + (u_1^2 h_1 - u_2^2 h_2) \tag{A3}$$

It can be easily verified that the term on the right-hand side of Eq. (A3) coincides with the difference between the momentum terms in the physical flux at cells upstream and downstream of the IBC (see second component in Eqs. (9) and (10)) when the bottom is horizontal (z=0, and η=h) and sections 1 and 2 coincide with the centers of cells (i,j) and (i+1,j). The same physical interpretation holds in the case of a non-submerged gate.

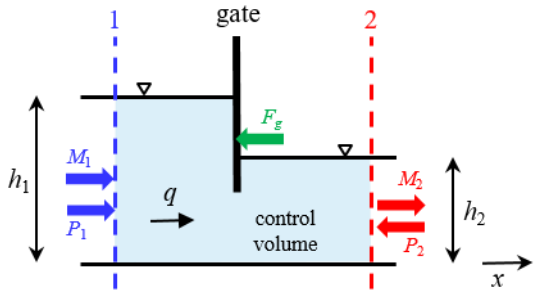

Figure A1. Control volume for momentum balance.

Acknowledgments

The authors wish to thank Prof. T. Picek and Prof. J. Murillo for providing the experimental data used for model validation. The reviewers are also kindly acknowledged for their valuable suggestions. This research benefits from the HPC (High Performance Computing) facility of the University of Parma.

References

Audusse, E., Bouchut, F., Bristeau, M., Klein, R., and Perthame, B. (2004) A fast and stable well-balanced scheme with hydrostatic reconstruction for shallow water flows, SIAM J. Sci. Comput., 25, 2050-2065. https://doi.org/10.1137/S1064827503431090

Bradley, J. N. (1978). Hydraulics of bridge waterways. US Department of Transportation, Federal Highway Administration.

Brown, P. M. (1985). Afflux at British bridges: interim report. Hydraulic Research Wallingford, Wallingford, UK, Report SR, 60.

Brunner, G. W. (2016). HEC-RAS River Analysis System. Hydraulic Reference Manual. Version 5.0. U.S. Army Corps of Engineering - Hydrologic Engineering Center (HEC), Davis CA.

Caleffi, V., Valiani, A., and Zanni, A. (2003). Finite volume method for simulating extreme flood events in natural channels, J. Hydraul. Res., 41, 167-177. https://doi.org/10.1080/00221680309499959

Catella, M., & Bechi, G. (2006). Conservative schemes for flow numerical modeling of submerged bridges. In RiverFlow2006–proceedings of the international conference on fluvial hydraulics (Vol. 1, pp. 747-55). Taylor & Francis London.

Chow, V. T., Maidment, D.R., & Mays, L.W. (1988). Applied Hydrology. McGraw-Hill, New York.

Costabile, P., Macchione, F., Natale, L., & Petaccia, G. (2015). Comparison of scenarios with and without bridges and analysis of backwater effect in 1-D and 2-D river flood modeling. Comput. Modell. Eng. Sci., 109(2), 181-204.

Cozzolino, L., Cimorelli, L., Covelli, C., Della Morte, R., & Pianese, D. (2015). The analytic solution of the shallow-water equations with partially open sluice-gates: The dam-break problem. Advances in Water Resources, 80, 90-102. https://doi.org/10.1016/j.advwatres.2015.03.010

Dazzi, S., Vacondio, R., Dal Palù, A., & Mignosa, P. (2018). A local time stepping algorithm for GPU-accelerated 2D shallow water models. Advances in Water Resources, 111, 274-288. https://doi.org/10.1016/j.advwatres.2017.11.023

Dazzi, S., Vacondio, R., & Mignosa, P. (2019). Integration of a Levee Breach Erosion Model in a GPU-Accelerated 2D Shallow Water Equations Code. Water Resources Research, 55(1), 682-702. https://doi.org/10.1029/2018WR023826

D'Oria, M., Mignosa, P., & Tanda, M. G. (2012). Reverse level pool routing: Comparison between a deterministic and a stochastic approach. Journal of Hydrology, 470, 28-35. https://doi.org/10.1016/j.jhydrol.2012.07.045

Echeverribar, I., Morales-Hernández, M., Brufau, P., & García-Navarro, P. (2019). Use of internal boundary conditions for levees representation: application to river flood management. Environmental Fluid Mechanics, 1-19. https://doi.org/10.1007/s10652-018-09658-6

Ferrari, A., Dazzi, S., Vacondio, R., and Mignosa, P. (2020). Enhancing the resilience to flooding induced by levee breaches in lowland areas: a methodology based on numerical modelling, Nat. Hazards Earth Syst. Sci., 20, 59-72. https://doi.org/10.5194/nhess-20-59-2020

Kurganov, A., & Petrova, G. (2007). A second-order well-balanced positivity preserving central-upwind scheme for the Saint-Venant system. Communications in Mathematical Sciences, 5(1), 133-160. https://doi.org/10.4310/CMS.2007.v5.n1.a6

Lacasta, A., Morales-Hernández, M., Murillo, J., García-Navarro, P. (2014). An optimized GPU implementation of a 2D free surface simulation model on unstructured meshes. Adv. Eng. Softw. 78, 1–15. https://doi.org/10.1016/j.advengsoft.2014.08.007

Lastra, M., Mantas, J.M., Ureña, C., Castro, M.J., García-Rodríguez, J.A. (2009). Simulation of shallow-water systems using graphics processing units. Math. Comput. Simul. 80 (3), 598–618. https://doi.org/10.1016/j.matcom.2009.09.012

Liang, Q., & Marche, F. (2009). Numerical resolution of well-balanced shallow water equations with complex source terms. Advances in Water Resources, 32(6), 873-884. https://doi.org/10.1016/j.advwatres.2009.02.010

Maranzoni, A., Dazzi, S., Aureli, F., & Mignosa, P. (2015). Extension and application of the Preissmann slot model to 2D transient mixed flows. Advances in Water Resources, 82, 70-82. https://doi.org/10.1016/j.advwatres.2015.04.010

Maranzoni, A., & Mignosa, P. (2018). Numerical treatment of a discontinuous top surface in 2D shallow water mixed flow modeling. International Journal for Numerical Methods in Fluids, 86(4), 290-311. https://doi.org/10.1002/fld.4418

Martin-Vide, J. P., & Prio, J. M. (2005). Backwater of arch bridges under free and submerged conditions. Journal of Hydraulic Research, 43(5), 515-521. https://doi.org/10.1080/00221680509500149

Morales-Hernández, M., Murillo, J., & García-Navarro, P. (2013). The formulation of internal boundary conditions in unsteady 2-D shallow water flows: Application to flood regulation. Water Resources Research, 49(1), 471-487. https://doi.org/10.1002/wrcr.20062

Natale, L., Petaccia, G., & Savi, F. (2004). Mathematical simulation of the effects of bridges and structures on flood waves propagation. Proceedings of River Flow 2004. Naples, Italy.

Picek, T., Havlik, A., Mattas, D., & Mares, K. (2007). Hydraulic calculation of bridges at high water stages. Journal of Hydraulic Research, 45(3), 400-406. https://doi.org/10.1080/00221686.2007.9521773

Ratia, H., Murillo, J., & García-Navarro, P. (2014). Numerical modelling of bridges in 2D shallow water flow simulations. International Journal for Numerical methods in fluids, 75(4), 250-272. https://doi.org/10.1002/fld.3892

Seckin, G. (2004). A simple formula for estimating backwater at bridge constrictions. Canadian Journal of Civil Engineering, 31(4), 561-568. https://doi.org/10.1139/l04-024

Toro, E. (2001). Shock Capturing Methods for Free Surface Shallow Water Flows. Wiley, Chichester, UK.

Vacondio, R., Dal Palù, A., Mignosa, P. (2014). GPU-enhanced Finite Volume Shallow Water solver for fast flood simulations. Environ. Modell. Softw. 57, 60-75. https://doi.org/10.1016/j.envsoft.2014.02.003

Vacondio, R., Aureli F., Ferrari, A., Mignosa, P., Dal Palù, A. (2016). Simulation of the January 2014 flood on the Secchia River using a fast and high-resolution 2D parallel shallow-water numerical scheme. Nat. Hazards, 80(1), 103-125. https://doi.org/10.1007/s11069-015-1959-4

Vacondio, R., Dal Palù, A., Ferrari, A., Mignosa, P., Aureli, F., Dazzi, S. (2017). A non-uniform efficient grid type for GPU-parallel Shallow Water Equations models. Environ. Modell. Softw. 88, 119-137. https://doi.org/10.1016/j.envsoft.2016.11.012

Zhao, D.H., Shen, H.W., Tabios III, G.Q., Lai, J.S. & Tan, W.Y. (1994). Finite-volume two-dimensional unsteady-flow model for river basins. Journal of Hydraulic Engineering, 120(7), 863-883. https://doi.org/10.1061/(ASCE)0733-9429(1994)120:7(863)