UNIVERSITÀ DI PARMA

UNIVERSITA’ DEGLI STUDI DI PARMA

Dottorato di Ricerca in Ingegneria Civile e Architettura
Ciclo XXX

# 2D Shallow Water modelling of flood propagation: GPU parallelization, non-uniform grids, porosity, reverse flow routing

Relatore:
Chiar.mo Prof. Paolo Mignosa

Tutor:
Dott. Ing. Renato Vacondio

Coordinatore del Dottorato:
Chiar.mo Prof. Andrea Carpinteri

Dottoranda:
Alessia Ferrari

Anni 2014/2017

“…the sea of the simulation…”

F. Battiato

# Acknowledgments

Desidero ringraziare in primo luogo il Prof. Paolo Mignosa per la professionalità, il rigore e la curiosità che mette in ogni attività. Ringrazio Renato Vacondio per la disponibilità, le fruttuose idee, i preziosi e concreti suggerimenti. Grazie ad Alessandro Dal Palù per i consigli sulla programmazione in CUDA e a Marco D’Oria per quelli riguardanti i modelli Bayesiani. Grazie a Susanna, Andrea e a tutto il gruppo di ricerca di Idraulica e Costruzioni Idrauliche per i consigli e per il clima positivo che hanno saputo creare. Un grande ringraziamento va ai miei genitori, per il loro immancabile e indescrivibile sostegno: nulla sarebbe stato possibile senza di loro. A Bruno, Luisa, Teresa e a Colui che è con loro.

# Abstract

This thesis concerns different aspects of the numerical modelling of flood propagation, and it addresses both theoretical and practical issues in the framework of explicit finite volume schemes for the Shallow Water Equations (SWEs).

The first part of the thesis is dedicated to investigate the capability of the GPU-parallelized numerical scheme developed at the Department of Engineering and Architecture of the University of Parma to produce real flood events. In particular, the flood occurred on the Secchia River in January 2014 is simulated on a domain of about 180 km$^2$. The adoption of a Cartesian grid of approximately 7.2 M cells (with size 5 m) allows the description of all major embankments that potentially influence the flood evolution inside and outside the river region. The numerical scheme is modified to simulate levee breaches, and the results are validated against aerial and satellite images. Despite the high-resolution adopted, the GPU parallelization guarantees a ratio of simulation to physical times of about 1/15 (about 6 computing hours are necessary to simulate 90 h of physical time).

The simulation of this real flood highlights that the major limitation of the existing numerical scheme is the use of uniform Cartesian grids, which unavoidably limit the maximum size of the domain, and/or prevent the adoption of zones with different resolution. Therefore, this limitation is addressed by implementing a novel grid type named Block Uniform Quadtree that introduces non-uniform grids, while allowing exploiting GPU capability. Theoretical and laboratory tests demonstrate that speed-ups of up to one order of magnitude can be achieved (with comparable level of accuracy) in comparison with uniform Cartesian grids. The capability of the model of simulating large domains is demonstrated by performing a hypothetical flood event induced by a levee breach in a real 83 km-long river reach, considering a domain of 840 km$^2$, with maximum resolution of 5 m. In this simulation, a ratio of physical to computational time of about 12 is obtained.

Another critical aspect of flood simulation is the representation of urban areas. Despite the efficiency of the numerical scheme, the description of buildings and streets in the computational mesh requires high-resolution elements (with size less than 1 m), and thus the simulation of large domains becomes impossible or extremely computationally expensive. To overcome this issue, sub-grid models based on porosity are investigated and implemented in the numerical scheme. In particular, two different SWE formulations based on porosity are analyzed. The first one is obtained by adding a source term and preserving the computational scheme of a classical SWE solver, which inherently guarantees the *C-property*. In a second formulation, a novel augmented Riemann Solver capable of handling porosity discontinuities in 1D and 2D SWE models is derived, and its capability of capturing different wave patterns is assessed against several Riemann Problems with different wave patterns.

After having implemented and validated a stable, accurate and fast numerical model for the solution of the 2D-SWEs, in the last part of the thesis, the model is used in the framework of a Bayesian methodology for solving inverse problems: the goal is the estimation of an unknown inflow hydrograph in an ungauged river section.

The inverse procedure is parallelized as to take advantage of High Performance Computing clusters with GPUs, and the procedure is validated considering real river reaches and different flood wave shapes. The parallel procedure reduces the computational times of about 8 times if compared to a serial procedure.

# Contents

# List of Figures

# List of Tables

# Introduction

In recent years trends in flood frequencies and flooding damage appear to be increasing with a consequent worsening of the social and economic repercussions. Focusing on Europe, flooding events from 2000 to 2012 caused losses (related to climate change and socioeconomic development) for about 4.2 billion euros every year (Jongman et al., 2014).

Still considering European countries, Forzieri et al. (2017) combined disaster records with high-resolution hazard and demographic projection, and obtained that by 2100, two-thirds of the population could be affected by weather-related disasters. Assuming the 1981-2010 as reference period, the authors estimated 3000 deaths and 25 million of population exposed each year. Furthermore, the study highlighted that without introducing adaptation measures, by the period 2070-2100, the number of deaths could increase of about 50 times and the affected population up to 351 million, per year. Among the others, the river flood-related disasters would determine an increase of 105% of the exposed Europeans and of 54% of deaths, if compared to the reference period.

Considering a different temporal interval, Alfieri et al. (2017) highlighted the interconnections between flood risk at global scale and future global warming, and projected for a 1.5°C warming level a 100% increase in damages and affected population, compared with those of the 1976-2005 period.

From the analysis of the cited studies, it clearly emerges that flooding events are one of the most (or maybe the most) natural weather-related phenomenon, which could determine the greatest damages both in the present and even more in the future. Therefore, it is necessary to manage measures and instruments useful for minimizing the risk related to these phenomena: most of the actual strategies aim at reducing flood risk by increasing the resilience, which means the adaptability of population to flooding events.

In this framework, the development and application of mathematical numerical models capable of providing accurate and fast predictions of the flooding evolution is crucial to obtain a reduction of the damages caused by floods.

After the European Union Directive (2007/60/EC), many countries established national programs to define flood risk maps from global to local scale (De Moel et al., 2015). In such programs the numerical modelling of flooding events remains one of the main instruments through which calculate for instance water depth, velocity and extent of inundation, essential to define flood hazard and then flood risk maps.

In the framework of flood mitigation, recently, Dottori et al. (2017) proposed a procedure for rapid flood mapping and risk definition in Europe that combines exposure and vulnerability information, and flood hazard maps.

With the purpose of enhancing the state of the art of flood modelling methods for producing flooding hazard maps, this thesis aims at contributing to the improvement of explicit finite volume numerical methods based on the Shallow Water Equations (SWEs) that allow a large scale modelling of flooding in urban and rural areas.

The first part of the thesis investigates the capability of the GPU-parallelized numerical scheme developed by Vacondio et al. (2014) to produce the January 2014 flood occurred on the Secchia River (Northern Italy).

The numerical computations are carried out on a high-resolution domain of about 180 km$^2$, which is discretized by means of a Cartesian grid of approximately 7.2 M cells with size 5 m. The adoption of a high-resolution mesh is useful in order to capture the natural and artificial elements that influence the flooding dynamics inside and outside the river region. The numerical scheme is modified to simulate levee breaches, and urban areas are considered by means of a higher roughness coefficient. The results of the simulation are validated against several field data and aerial and satellite images. Despite the high-resolution adopted, the parallelization on Graphic Processing Units (GPUs) allows reducing the computational costs, if compared to serial codes.

The issue of modelling large-scale simulations at uniform resolution, and at the same time retaining small-scale structures, poses limitations on the maximum extent of the domain that can be model.

However, the use of non-uniform meshes in GPU-enhanced numerical schemes is not trivial. In fact, classical data structures of unstructured or hierarchical Quadtree meshes (Greaves and Borthwick, 1999; Borthwick et al., 2000; Rogers et al., 2001; Liang et al., 2007, 2008) require a data arrangement in memory, which is opposed to plain matrix memorization necessary for providing the best GPU throughput.

Therefore, a novel type of grid (called Block Uniform Quadtree) that exploits the computational capability of GPUs, while discretizing the domain with non-uniform resolution, is implemented. The key idea of BUQ grids is to discretize the domain using cell blocks with uniform resolution, while allowing different resolutions for different blocks of cells; this means that the Quadtree partition refers to block of cells, instead of a single computational cell.

As a result, the BUQ grids allow resolving small-scale effects in limited areas while simulating large domains.

Since a detailed representation of the flood dynamics around buildings is computationally extremely expensive and cannot be adopted in large domains (order of $10^3$ km$^2$), a sub grid model based on the porosity is investigated. Defina et al. (1994) proposed the first formulation for partially wet/dry areas over irregular domains based on the isotropic porosity, which is defined as the fraction of plan view area available to flow. The SWEs with porosity allow describing building effects in urban areas avoiding the use of high-resolution meshes (for accounting narrow streets and buildings).

In this thesis, the SWEs with isotropic porosity are implemented following two different formulations.

The first one consists in the derivation of a well-balanced scheme, which preserves the computational scheme of the classical SWEs and accounts for the porosity by means of additional source terms.

The second formulation addresses the problem of handling porosity discontinuities, by introducing a Generalized Riemann Problem and theoretically deriving and implementing an augmented Roe Solver in a 1D and 2D Godunov-type finite volume scheme that ensures a robust treatment of transonic flows. An Exact Riemann Solver is derived in order to validate the model.

Since the modelling of flooding events requires, besides accurate, fast and stable numerical models, the use of proper boundary condition (discharge hydrographs), the last argument faced in the thesis is the investigation of a reverse flow routing procedure, which allows estimating discharge hydrographs at upstream-ungauged stations by means of information available at downstream monitored sites. Most of the studies concerning the reconstructing of discharge hydrographs in ungauged river sections require the knowledge of the water level hydrograph in the same section (Aricò et al., 2009; Spada et al., 2017). On the contrary, the Bayesian technique introduced by D’Oria and Tanda (2012), for cases in which a 1D unsteady model can be adopted, presents the relevant advantage of estimating upstream inflow hydrograph without having any measures in the considered section.

However, many real cases of rivers including large floodable areas require the adoption of 2D SWE model to capture the complex hydrodynamic field. Therefore, the last part of the thesis aims at extending the BGA methodology for reverse flow routing to 2D test cases in order to model natural rivers with complex geometry, including flood plains and meanders.

In order to reduce the high computational costs of the Bayesian methodology, a parallel procedure that takes advantage of High Performance Computation clusters with GPUs is implemented. The inverse procedure is tested by estimating the inflow hydrograph with different shapes, in two real river reaches with complex geometry, meanders and floodplains.

The thesis is organized as follows: in Chapter 1, the reconstruction of the real flooding caused by a bank failure in the Secchia River in Northern Italy is presented. Chapter 2 is dedicated to the development of an efficient novel grid type that allows different resolution levels inside the domain, still retaining GPU capability exploitation. The SWEs with porosity are illustrated in Chapters 3 and 4: Chapter 3 describes the isotropic formulation and the well-balanced scheme, whereas Chapter 4 presents the augmented Riemann solver for handling porosity discontinuities. The inflow hydrograph estimation is assessed in Chapter 5, and finally, the conclusions and the future implications of the thesis are outlined.

# Chapter 1

# Reconstruction of the January 2014 flood on the Secchia River

## 1.1. Introduction

In this chapter, high-resolution modelling of the real flood event occurred on the Secchia River (Italy) on January 2014, induced by a levee breach, is performed adopting the GPU-parallelized model implemented by Vacondio et al. (2014), and based on an explicit Shock-Capturing Finite Volume method for the solution of the 2D Shallow Water Equations (SWEs).

The availability of numerical models capable of reproducing such events represents a *sine qua non* for reducing flooding hazard, since these tools are useful for urban planning or as support during the event for civil protection activities.

However, the reconstruction of a real flooding presents several challenges: (1) the adopted/implemented numerical model has to be stable and accurate, (2) a 2D scheme is required for modelling floodplains, (3) high-resolution meshes have to be adopted in order to capture the elements influencing the flooding dynamics, and (4) the computational times have to be reasonable.

In the framework of flooding modeling at large scale, numerical schemes based on the SWEs represent the current state of the art.

Since the prevention of flood damages requires modeling tools capable of simulating the flood propagation not only in the river region but also in the floodplains, and in presence of possible overbanks, a fully 1D approach actually appears outclassed (Costabile et al., 2015).

Focusing on the flooding generated by levee breaches, a different approach consists in describing the river with a 1D model and thus computing the discharge hydrograph flowing through the levee breach, which is then adopted as an inflow boundary condition for the 2D numerical scheme of the flood-prone area. Following this scheme, Masoero et al. (2013) and Di Baldassarre et al. (2009) studied the 1951 and 1879 historical inundations on the Po River in Northern Italy, respectively. More recently, Mazzoleni et al. (2014) analyzed synthetic scenarios to define the flood hazard due to levee breaches in the Po River (Italy).

The choice of adopting a 1D model for the river and further a 2D scheme for the floodplain is less computational expensive than a fully 2D model (of both the river region and the flood–prone area), however, it presents two major limitations: (1) the 1D approximation does not hold in presence of meandering channels, as well as close to the breach, where the velocity in the direction normal to the thalweg is not negligible, and (2) the use of non-coupled numerical schemes excludes the modeling of the backwater effects near the breach failure.

Therefore, some authors have proposed to develop 1D-2D coupled numerical schemes (Gejadze and Monnier, 2007; Bladé et al., 2012; Morales-Hernández et al., 2013, 2014, 2016), which discretize the river region by means of cross-sections, and adopts 2D model to describe the large areas outside the river region, where the 1D assumptions does not hold. However, also the coupled models present a pair of issues: (1) the region where the 1D approximation holds has to be defined a priori, and (2) the coupling between the

1D and 2D models has to be defined in such a way that both global mass and momentum are conserved (Morales-Hernández et al., 2013).

With the aim of overcoming these limitations, fully 2D models of both the river region and the flood prone areas have been developed (among the others Lynch and Gray, 1979; Casulli, 1990; Vacondio et al., 2014; Costabile and Macchione, 2015).

However, the main drawback associated with high-resolution meshes and 2D-SWE model is the high computational cost that makes arduous to run (many) simulations on large domains.

To overcome this issue, parallel numerical schemes have been recently developed. Sanders et al. (2010) presented a parallel SWE numerical model based on Message Passing Interface (MPI) technique, whereas a different approach adopts Graphics Processing Units (GPUs) and parallelizes codes using the Compute Unified Device Architecture (CUDA) programming language (Nvidia C.U.D.A., 2007). The use of GPUs is becoming more common in high parallel computation, mainly due to the efficient on-card communication and to the possibility of drastically reducing the computational times, without increasing the per processor costs: whereas MPI communication requires supercomputers, GPU cards can be installed on standard desktop computers.

With regard to the SWEs schemes, in the last years different authors have already managed to develop GPU models (Crossley et al., 2010; Kalyanapu et al., 2011; Brodtkorb et al., 2012; De la Asunción et al., 2013; Smith and Liang, 2013; Lacasta et al., 2014, 2015a, 2015b).

Focusing on the levee breach modelling, a lot of research is currently being undertaken to achieve a realistic representation of the failure mechanisms of earthen dams and levees. Many studies (Zhu et al., 2004; Macchione, 2008; Macchione and Rino, 2008; Froehlich, 2016; Kim and Sanders, 2016) describe the characteristics of dam breaches by estimating various parameters: geometrical (height, width, failure time), hydraulic (reservoir volume, initial water height), geotechnical (material erodibility). Recently, Viero et al. (2013) proposed the reconstruction of different flooding events due to levee breaches in Northern Italy, considering both overtopping and piping processes, on the basis of a simplified physically based 1D link capable of simulating the breach initiation and growth. In the present chapter, the breach evolution in time has been modelled according to the field data collected during the event.

Model calibration was achieved against the large amount of field data collected during the event, not least the COSMO-SkyMed radar data captured about 60 hours after the beginning of the levee collapse.

This Chapter is derived in part from Vacondio et al. (2016).

## 1.2. Hydraulic model

### 1.2.1. Numerical scheme

The Shallow Water Equations allow mathematically modeling rapidly varying free surface flows, such as dam-break waves, flood waves in river, etc. The equations derive from the physical principles of mass and momentum conservation, which govern the motion of the fluid, under the following assumptions: horizontal length scale much larger than the vertical one, hydrostatic pressure distribution, incompressible fluid, negligible viscosity effects and vertical acceleration, and small channel bed slope (Cunge et al., 1980). As a result, the SWEs are a time-dependent, 2D system of non-linear partially differential equations of hyperbolic type (Toro, 2001).

The numerical model here adopted solves in a finite volume framework the integral form of the 2D-SWEs (e.g. Toro, 2001):

$$\frac{d}{dt}\int_A \mathbf{U}dA + \int_C \mathbf{H}\cdot\mathbf{n}\,dC = \int_A \left(\mathbf{S}_0 + \mathbf{S}_f\right)dA \qquad (1.1)$$

where $A$ denotes the area of the integration element, $C$ the element boundary, **n** the outward unit vector normal to $C$, **U** the vector of the conserved variables and **H** = (**F**,**G**) the tensor of fluxes in the $x$ and $y$ directions, respectively:

$$\mathbf{U}=\begin{bmatrix}h\\uh\\vh\end{bmatrix}\qquad \mathbf{F}=\begin{bmatrix}uh\\u^2h+\frac{1}{2}gh^2\\uvh\end{bmatrix}\qquad \mathbf{G}=\begin{bmatrix}vh\\uvh\\v^2h+\frac{1}{2}gh^2\end{bmatrix}\tag{1.2}$$

where *h* is the flow depth, *u* and *v* are the velocity components in the *x* and *y* directions and *g* is the acceleration due to gravity. The bed ($\mathbf{S}_0$) and friction ($\mathbf{S}_f$) slope source terms are expressed according to the following relations:

$$\mathbf{S}_0=\begin{bmatrix}0\\-gh\dfrac{\partial z}{\partial x}\\-gh\dfrac{\partial z}{\partial y}\end{bmatrix}\qquad \mathbf{S}_f=\begin{bmatrix}0\\-gh\dfrac{n_f^2u\sqrt{u^2+v^2}}{h^{4/3}}\\-gh\dfrac{n_f^2v\sqrt{u^2+v^2}}{h^{4/3}}\end{bmatrix}\tag{1.3}$$

in which *z* is the bed elevation with respect to a horizontal reference plane and $n_f$ is the roughness coefficient according to the Manning equation.

In order to obtain a balance between fluxes and source terms in Equation (1.1), the Liang and Borthwick (2009) formulation, which guarantees the *C-property* regardless of the slope source term discretization, the approximate Riemann solver and with wet-dry interfaces, is adopted (Vacondio et al., 2014). Hence, the terms of Equations (1.2) are modified as follows:

$$\mathbf{U}=\begin{bmatrix}\eta\\uh\\vh\end{bmatrix}\qquad \mathbf{F}=\begin{bmatrix}uh\\u^2h+\frac{1}{2}g(\eta^2-2\eta z)\\uvh\end{bmatrix}\qquad \mathbf{G}=\begin{bmatrix}vh\\uvh\\v^2h+\frac{1}{2}g(\eta^2-2\eta z)\end{bmatrix}\tag{1.4}$$

where $\eta = h + z$ represents the free surface elevation above datum.

The bottom source term $\mathbf{S}_0$, which is discretized using a centered formulation, since the modified form of the SWEs reported in Equation (1.4) inherently guarantees the *C-property*, becomes:

$$\mathbf{S}_0=\begin{bmatrix}0\\-g\eta\dfrac{\partial z}{\partial x}\\-g\eta\dfrac{\partial z}{\partial y}\end{bmatrix}\tag{1.5}$$

For the friction source term, the implicit discretization of Caleffi et al. (2003) is adopted in order to prevent the formation of spurious oscillations in the presence of very small depths (Vacondio et al., 2014).

Adopting a Finite Volume (FV) numerical approximation of the SWEs, the partial differential equations (1.1) are solved on a structured Cartesian grid, which discretizes the domain with cells having sizes $\Delta x$, $\Delta y$, in $x$ and $y$ directions, respectively, and assigns the conserved variables to the center of the cells.

Then, the conserved variables **U** ($\eta$, $uh$, $vh$) and the water depth $h$ are extrapolated at the four faces of each cell, adopting the linear Monotone Upwind Schemes for Scalar Conservation Laws (MUSCL) with minmod limiter (Toro, 1999); this reconstruction allows achieving a second order of accuracy in space.

The procedure that computes the Riemann states preserves water depth positivity in order to ensure the *C-property*; with this aim, the formulation proposed by Liang and Marche (2009) for 1D grid has been extended. Given cell $i$, the reconstructed values at the left $L$ and right $R$ hand side interfaces, at a generic time level and with reference to the $x$-axis direction, are:

$$\eta^L_{i+\frac{1}{2},j} = \eta_{i,j} + \frac{1}{2}\psi_{i-\frac{1}{2},j}\left(\eta_{i,j} - \eta_{i-1,j}\right) \tag{1.6}$$

$$uh^L_{i+\frac{1}{2},j} = uh_{i,j} + \frac{1}{2}\psi_{i-\frac{1}{2},j}\left(uh_{i,j} - uh_{i-1,j}\right) \tag{1.7}$$

$$vh^L_{i+\frac{1}{2},j} = vh_{i,j} + \frac{1}{2}\psi_{i-\frac{1}{2},j}\left(vh_{i,j} - vh_{i-1,j}\right) \tag{1.8}$$

$$h^L_{i+\frac{1}{2},j} = h_{i,j} + \frac{1}{2}\psi_{i-\frac{1}{2},j}\left(h_{i,j} - h_{i-1,j}\right) \tag{1.9}$$

$$\eta^R_{i-\frac{1}{2},j} = \eta_{i,j} - \frac{1}{2}\psi_{i+\frac{1}{2},j}\left(\eta_{i+1,j} - \eta_{i,j}\right) \tag{1.10}$$

$$uh^R_{i-\frac{1}{2},j} = uh_{i,j} - \frac{1}{2}\psi_{i+\frac{1}{2},j}\left(uh_{i+1,j} - uh_{i,j}\right) \tag{1.11}$$

$$vh^R_{i-\frac{1}{2},j} = vh_{i,j} - \frac{1}{2}\psi_{i+\frac{1}{2},j}\left(vh_{i+1,j} - vh_{i,j}\right) \tag{1.12}$$

$$h^R_{i-\frac{1}{2},j} = h_{i,j} - \frac{1}{2}\psi_{i+\frac{1}{2},j}\left(h_{i+1,j} - h_{i,j}\right) \tag{1.13}$$

where $\psi_{i,j}$ denotes a slope limiter function (here minmod) separately evaluated for each considered variable.

If the cell *i* is dry, the velocities at the cell boundaries are forced to zero, whereas if *i* is wet, they are calculated from the conservative variables as:

$$u^L_{i+\frac{1}{2},j} = \frac{uh^L_{i+\frac{1}{2},j}}{h^L_{i+\frac{1}{2},j}} \qquad u^R_{i+\frac{1}{2},j} = \frac{uh^R_{i+\frac{1}{2},j}}{h^R_{i+\frac{1}{2},j}} \tag{1.14}$$

$$v^L_{i+\frac{1}{2},j} = \frac{vh^L_{i+\frac{1}{2},j}}{h^L_{i+\frac{1}{2},j}} \qquad v^R_{i+\frac{1}{2},j} = \frac{vh^R_{i+\frac{1}{2},j}}{h^R_{i+\frac{1}{2},j}} \tag{1.15}$$

In order to ensure the non-negativity of water depths, for each cell interface *i*+1/2 a unique bed elevation value is defined (Audusse et al., 2004):

$$z_{i+\frac{1}{2},j} = max\left[z^L_{i+\frac{1}{2},j}, z^R_{i+\frac{1}{2},j}\right] \tag{1.16}$$

where the bed elevation at the left and right hand side of the same cell interface are calculated as:

$$z^L_{i+\frac{1}{2},j} = \eta^L_{i+\frac{1}{2},j} - h^L_{i+\frac{1}{2},j} \qquad z^R_{i+\frac{1}{2},j} = \eta^R_{i+\frac{1}{2},j} - h^R_{i+\frac{1}{2},j} \tag{1.17}$$

Therefore, the water depths at either side of the interfaces are corrected to ensure their positivity (Liang and Marche, 2009):

$$h^{L*}_{i+\frac{1}{2},j} = max\left[0, \eta^L_{i+\frac{1}{2},j} - z_{i+\frac{1}{2},j}\right] \tag{1.18}$$

$$h^{R*}_{i+\frac{1}{2},j} = max\left[0, \eta^R_{i+\frac{1}{2},j} - z_{i+\frac{1}{2},j}\right] \tag{1.19}$$

Finally, the Riemann states are calculated as follows:

$$\eta^{L*}_{i+\frac{1}{2},j} = h^{L*}_{i+\frac{1}{2},j} + z_{i+\frac{1}{2},j} \tag{1.20}$$

$$uh^{L*}_{i+\frac{1}{2},j} = u^L_{i+\frac{1}{2},j} \cdot h^{L*}_{i+\frac{1}{2},j} \tag{1.21}$$

$$vh^{L*}_{i+\frac{1}{2},j} = v^{L}_{i+\frac{1}{2},j} \cdot h^{L*}_{i+\frac{1}{2},j} \tag{1.22}$$

$$\eta^{R*}_{i+\frac{1}{2},j} = h^{R*}_{i+\frac{1}{2},j} + z_{i+\frac{1}{2},j} \tag{1.23}$$

$$uh^{R*}_{i+\frac{1}{2},j} = u^{R}_{i+\frac{1}{2},j} \cdot h^{R*}_{i+\frac{1}{2},j} \tag{1.24}$$

$$vh^{R*}_{i+\frac{1}{2},j} = v^{R}_{i+\frac{1}{2},j} \cdot h^{R*}_{i+\frac{1}{2},j} \tag{1.25}$$

The procedure illustrated for the cell interface *i*+1/2 can be similarly applied to the cell borders *i*-1/2 in *x* direction, and *j*+1/2 and *j*-1/2 in *y* direction.

Then, the states at the cell faces are employed in the Harten, Lax and van Leer approximate Riemann solver with the Contact wave restored HLLC (Toro, 2001) to compute the numerical interface fluxes **F** and **G**, in *x* and *y* directions, respectively.

Finally, the conserved variables are updated at each time step, following the second order Runge-Kutta method that allows achieving a second order of accuracy in time. The explicit time-marching conservative finite volume formula states:

$$\mathbf{U}^{n+1}_{i,j} = \mathbf{U}^{n}_{i,j} + \frac{1}{2}\Delta t^{n}\left[\mathbf{D}_i\left(\mathbf{U}^{n}_{i,j}\right) + \mathbf{D}_i\left(\mathbf{U}^{n+\frac{1}{2}}_{i,j}\right)\right] \tag{1.26}$$

where the superscript *n* represents the time level, the subscripts *i*, *j* are the cell positions in *x* and *y* directions, respectively, and $\Delta t^n$ is the time step.

The term $\mathbf{U}_{i,j}^{n+1/2}$ is obtained as:

$$\mathbf{U}^{n+\frac{1}{2}}_{i,j} = \mathbf{U}^{n}_{i,j} + \Delta t^{n}\mathbf{D}_i\left(\mathbf{U}_{i,j}\right) \tag{1.27}$$

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

$$\mathbf{D}_i\left(\mathbf{U}_{i,j}\right) = -\frac{\left(\mathbf{F}_{i+\frac{1}{2},j} - \mathbf{F}_{i-\frac{1}{2},j}\right)}{\Delta x} - \frac{\left(\mathbf{G}_{i,j+\frac{1}{2}} - \mathbf{G}_{i,j-\frac{1}{2}}\right)}{\Delta y} + \mathbf{S}_b + \mathbf{S}_f \tag{1.28}$$

where the numerical fluxes **F** and **G** are evaluated at the intercells *i* ± ½ and *j* ± ½, respectively. The bed slope source term $\mathbf{S}_b$ for the momentum equation is calculated as to guarantee a stable and well-balanced scheme also in presence of wet-dry fronts (Liang and Marche, 2009):

$$S_{b_{i,j}} = S_{0_{i,j}} + \bar{S}_{0_{i-\frac{1}{2},j}} + \bar{S}_{0_{i+\frac{1}{2},j}} \tag{1.29}$$

where the first term represents the central discretization of the bed slope source term in Equation (1.5), whereas, the other two terms are added in order to preserve water at rest also between wet and dry cell interfaces.

The two additional terms, which are useful only in presence of dry areas, are calculated as:

$$\bar{S}_{0_{i-\frac{1}{2},j}} = g\Delta z_{i-\frac{1}{2},j} \left[ \frac{z_{i+\frac{1}{2},j} - \left( z_{i-\frac{1}{2},j} - \Delta z_{i-\frac{1}{2},j} \right)}{2\Delta x_{i,j}} \right] \tag{1.30}$$

$$\bar{S}_{0_{i+\frac{1}{2},j}} = g\Delta z_{i+\frac{1}{2},j} \left[ \frac{\left( z_{i+\frac{1}{2},j} - \Delta z_{i+\frac{1}{2},j} \right) - z_{i-\frac{1}{2},j}}{2\Delta x_{i,j}} \right] \tag{1.31}$$

where

$$\Delta z_{i+\frac{1}{2},j} = max\left[0, -\left(\eta^{L}_{i+\frac{1}{2},j} - z_{i+\frac{1}{2},j}\right)\right] \tag{1.32}$$

$$\Delta z_{i-\frac{1}{2},j} = max\left[0, -\left(\eta^{R}_{i-\frac{1}{2},j} - z_{i-\frac{1}{2},j}\right)\right] \tag{1.33}$$

Dealing with an explicit scheme, the time step $\Delta t$ is calculated as to fulfill the Courant–Friedrichs-Lewy condition, which guarantees the stability of the numerical model, according to the following relation (Toro, 2001):

$$\Delta t = \frac{C_r \cdot \Delta x}{\lambda} \leq 1 \tag{1.34}$$

where $C_r$ is the Courant number, here set equal to 0.8, and $\lambda$ is the maximum wave speed.

Finally, the numerical model presents a few techniques, which prevent the formation of high velocities and instabilities during the computation. Particularly, if the water depth $h_{i,j}$ is lower than a very small threshold $h_\varepsilon$, the cell is dried ($h_{i,j} = 0$). However, as to prevent the introduction of any relevant mass conservation error, and using single precision variables, $h_\varepsilon$ has been defined equal to $1\times10^{-5}$ m, which is close to the machine precision.

Moreover, in order to avoid spurious non-physical velocities close to the wet-dry front, the specific discharge is corrected according to the following relation (Kurganov and Petrova, 2007):

$$uh_c = uh \frac{\sqrt{2}}{\sqrt{1 + max\left[1, \left(\frac{\varepsilon}{h}\right)^4\right]}} \tag{1.35}$$

where $uh_c$ is the corrected specific discharge in $x$ direction and $\varepsilon$ is a threshold value for the water depth. An identical correction is applied to the specific discharge in $y$ direction $vh$.

## 1.2.2. CUDA implementation

The 2D-SWE model is implemented in a CUDA/C++ code that exploits parallel computation offered by NVIDIA™ GPUs, by taking advantage of both the host (CPU) and the device (GPU). The code runs on the CPU and it calls a kernel, which is the piece of code that will be executed on the GPU, and once it has received the results it goes over with the computation.

In CUDA, a thread represents the basic work unit; many threads are grouped into blocks and many blocks form a grid (Nvidia C.U.D.A., 2007) (Figure 1.1-a). The crux of the CUDA language is that all the threads inside a block are executed in parallel and perform the same instructions. In the present model, each thread corresponds to a computational cell used to discretize the physical domain (the grid), and $M \times M$ cells form a regular block (here 8×8 or 16×16), which can be processed in parallel by a multicore of the GPU taking advantage of fast memory communication. With the aim of ensuring a coalesced access to memory (contiguous threads read contiguous areas of memory), the domain is discretized with a Cartesian grid, and data are stored as a simple two-dimensional array: the coordinate system used in the program corresponds to the matrix indexes used to store the array.

Even if the CUDA code allows handling both the host and the device, special attention has to be paid to the GPU memory architecture, in order to effectively achieve good speed-up. As illustrated in Figure 1.1-b, a GPU presents six memory levels with different scope, location, lifetime and read/write access (Vacondio et al., 2014). The register and the local memory are visible only inside a thread, whereas all the threads inside a block have access to the shared memory, and finally the global, the constant and the texture memory are visible from the host and the device. According to this memory hierarchy, the best way of reducing the computational costs is by preferring the use of register and shared memory instead of global memory, which presents higher latency.

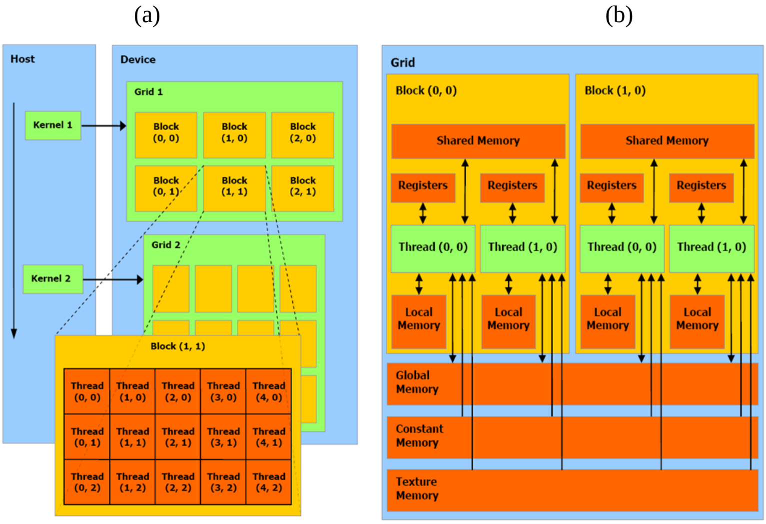


*Figure 1.1. CUDA work units (a) and GPU memory architecture (b) (image from http://cuda-programming.blogspot.it).*

After having separately analyzed both the 2D-SWE formulation (Section 1.2.1.) and the GPU features, the interconnections between the two are now briefly described (see also Vacondio et al., 2014).

Assuming a second order accurate scheme, the procedure that leads to the update of the conservative variables **U**, from time $t^n$ to time $t^{n+1} = t^n + \Delta t$ (Equation 1.26), consists in the following sequentially steps:

1. Definition of the boundary conditions;
2. Sorting among wet and dry blocks;
3. MUSCL reconstruction (1st half step);
4. Flux computation (1st half step);
5. MUSCL reconstruction (2nd half step);
6. Flux computation (2nd half step);
7. Sum of the 1st and the 2nd half steps.

The CPU controls the iteration evolution and invokes the various tasks (points 1 to 7), which correspond to distinct CUDA kernels. In order to achieve a good efficiency of the code, the information exchange between host and device are limited.

The model presents several optimization procedures, which include the implicit local ghost approach for the assignment of boundary conditions, a careful implementation to reduce the data transfer between CPU and GPU, and an efficient Block Deactivation Optimization (BDO) (Vacondio et al., 2014). This last optimization (point 2) allows the reduction of the blocks to process: for each time step only the wet (with at least one wet cell) or the potentially wet (with a dry cell near a wet one) blocks are computed, whereas the dry ones are excluded, with a consequent reduction of the computational times.

## 1.3. Study area and available data

During the flood event of January 19th 2014 on the Secchia River, at around 6:00 A.M. a levee breach occurred in the right bank near Modena (North-Central Italy, Figure 1.2); the triggered flooding caused the displacement of thousands of people and one death, 75 km$^2$ of flooded lands, and estimated overall losses for about 400 million euros.

The area involved in the flooding is characterized by the presence of two main villages (Bastiglia and Bomporto) and it is bounded at east by the left levee of the Panaro River, and at west by the right levee of the Secchia River (Figure 1.2).

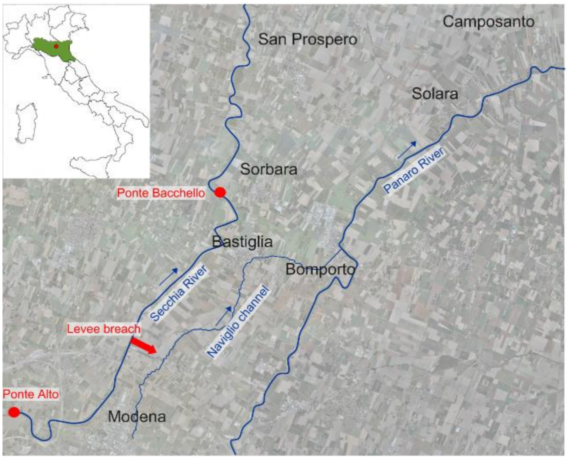


*Figure 1.2. Study area with indicated Ponte Alto and Ponte Bacchello gauge stations.*

A digital terrain model (1 m resolution), based on a LiDAR survey, was available to describe the geometry of the riverbeds and of the flooded area. Since the acquisition was performed during summer, when the Secchia River is almost totally dried up, no other information was necessary to correctly describe the bathymetry.

In order to reach a good compromise between an accurate description of all the main characteristics, which influence the flooding dynamics (i.e. embankments of roads, railways, artificial channels) and the computational time, a Cartesian mesh with cell size equal to 5 m was adopted. To preserve the crest elevation of the artificial embankments after the downsampling of the DTM, each 5x5 m cell crossed by an embankment was identified and its elevation was set equal to the maximum value of the original 25 points belonging to that cell. As shown in Figure 1.3 for a complex highway junction, the 5 m resolution mesh still correctly describes the terrain elevations, analogously to the original 1 m DTM.

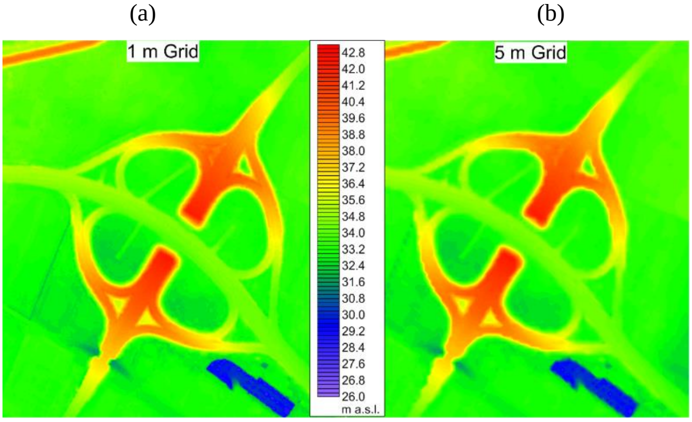


*Figure 1.3. Bathymetry detail of highway intersection: comparison between the original 1 m DTM (a) and the corrected 5 m resolution mesh (b).*

The Secchia River was modeled from Ponte Alto until Ponte Bacchello gauging stations (Figure 1.2): in this 14 km-long reach, the river is bounded by artificial earthen levees of remarkable height above the surrounding lands.

Even if water levels in the river were far from overtopping the crest, a small portion of the levee body collapsed, due to the presence of burrowing animals dens, triggering the flooding (D'Alpaos et al., 2014; Orlandini et al., 2015). Progressively, the breach widened and deepened, reaching a maximum width of about 80 m at around 3:00 P.M. of January 19th 2014; the information about timing, geometry and evolution of the levee breach was provided through direct observations and aerial photos (Figure 1.4).

*Figure 1.4. Above (a) and lateral (b) views of the fully developed levee breach.*

Field information about flood propagation time and flooding extent, useful for calibration purposes, was achieved through press services, media and people directly involved or interested in the event.

Satellite synthetic Aperture Radar (SAR) observations, acquired at 3 m spatial resolution during the flooding event, were also obtained from the COSMO-SkyMed Mission archive and allowed a synoptic view of the flooding extent about 60 hours after the beginning of the breaching process.

## 1.4. Model calibration

The calibration of the flood model aimed at the definition of four unknown quantities: (1) the flow hydrograph to impose as inflow boundary condition, (2) the roughness coefficient inside and outside the riverbed, distinguishing in this last case between (3) rural and (4) urban areas. Actually, these parameters are correlated one another, and thus the calibration was carried out following three different sequential steps.

Firstly, the roughness inside the riverbed was estimated by means of a historical event, which did not involve the regions outside the levees. Then, once the roughness coefficient in the riverbed was defined, the inflow hydrograph of the January 2014 event was calculated. Finally, the propagation of the computed flood wave in the floodplain areas allowed estimated the rural and then the urban roughness coefficients.

### 1.4.1. Riverbed roughness

The considered reach presents only few discontinuous floodplains of relatively small extent between the river levees, and thus, the river region can be characterized by means of a unique roughness coefficient.

The riverbed was calibrated adopting the most significant and well document event occurred in the last ten years (23rd - 30th December 2009). As upstream boundary condition (Ponte Alto), the discharge hydrograph was calculated from the recorded levels by means of the station stage-discharge relationship published by ARPA (Regional Agency for Environmental Protection in the Emilia-Romagna region), while downstream, the stage hydrograph at Ponte Bacchello, was imposed (Figure 1.5). It is noteworthy that the discharge hydrograph was imposed about 1 km upstream of Ponte Alto, in order to allow the model regulation and thus to compare the resulted water levels with those registered at the gauging station.

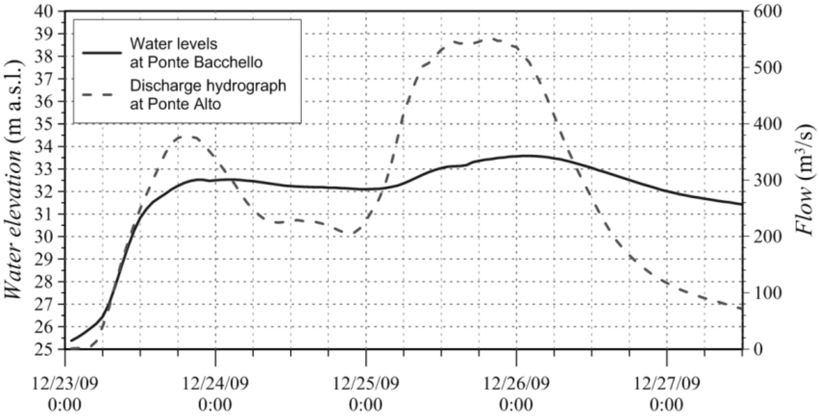


*Figure 1.5. Flow and stage hydrographs for the 2009 flood event at the gauging stations of Ponte Alto and Ponte Bacchello.*

The riverbed resistance parameter was evaluated executing three runs, each characterized by a different uniform value of $n_f$ equal to 0.067, 0.05 and 0.03 $\mathrm{m}^{-1/3}\cdot\mathrm{s}$, respectively.

Then, the Nash-Sutcliffe efficiency criterion (Nash and Sutcliffe, 1970) $E_h$ was adopted in order to estimate the most suitable roughness coefficient:

$$E_h = \left[1 - \frac{\sum_{i=1}^{N}(h_i^{act} - h_i^{est})^2}{\sum_{i=1}^{N}\left(h_i^{act} - \bar{h}^{act}\right)^2}\right] \cdot 100 \tag{1.36}$$

where $N$ is the total number of stage hydrograph data, $h_i^{act}$ and $h_i^{est}$ are the $i$-th observed and estimated water level values (at Ponte Alto gauging station) respectively, and $\bar{h}^{act}$ is the mean value of the actual hydrograph.

Table 1.1 reports the resulted values of $E_h$ for the three considered Manning coefficients.

*Table 1.1. Nash-Sutcliffe efficiency for the considered roughness coefficients.*

| $n_f$ (m$^{-1/3}$·s) | 0.067 | 0.05 | 0.03 |
|---|---|---|---|
| $E_h$ (-) | 86.4 | 94.2 | 87.1 |

Analogously, Figure 1.6 shows the comparison between computed and observed water levels at Ponte Alto. Both from Table 1.1 and Figure 1.6, the best agreement between simulated and registered data was achieved with $n_f$ = 0.05 m$^{-1/3}$·s, which was therefore adopted to characterize the riverbed roughness.

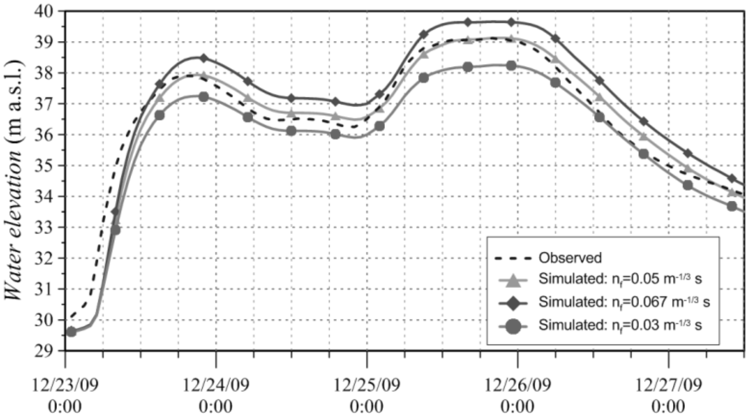


*Figure 1.6. Stage hydrographs at Ponte Alto for the 2009 event: comparison between observed and simulated data with different roughness values.*

## 1.4.2. Inflow boundary condition

The inflow discharge of the January 2014 event (upstream boundary condition) was firstly obtained by converting the stage hydrograph recorded at Ponte Alto gauging station, with the available stage-discharge relationship. However, this rating curve was expected to underestimate the inflow discharges after the occurrence of the breach, which is located only 6 km downstream of the gauging station itself. The breach gives origin to a drawdown profile capable of exerting a significant influence on the rating curve at Ponte Alto; this behavior is confirmed by comparing the registered and simulated water elevations, which satisfactory agree before the breach formation, whereas afterwards the computed water elevations systematically underestimate the registered ones (Figure 1.7).

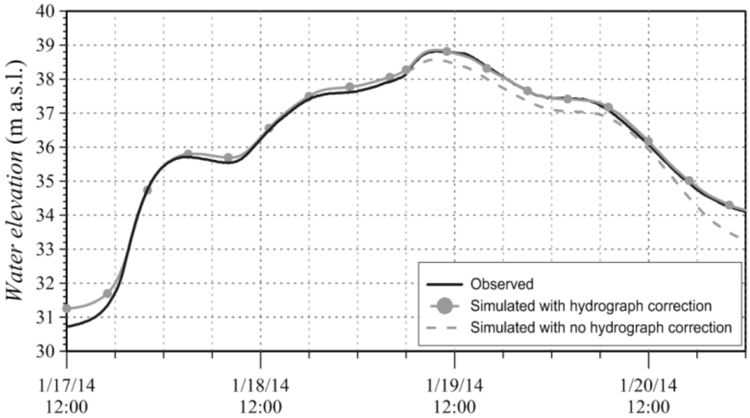


*Figure 1.7. 2014 event: observed and simulated stage hydrographs at Ponte Alto.*

In order to assess this problem, a synthetic triangular flood wave was routed as upstream boundary condition in presence of the fully developed breach. Then, the stage-discharge values extracted at Ponte Alto, were fitted by a power law function to obtain a new rating curve for the station. Finally, for the 2014 event, this rating curve was adopted to convert water elevations into discharges only after the complete development of the breach, whereas the original stage-discharge relationship was used before the beginning of the breach formation. During the breach evolution (between 6.00 A.M. and 3.00 P.M. of the 19th January 2014) a linear interpolation between the two hydrographs was adopted in order to take into account the progressive opening of the breach.

The described methodology allowed the estimation of the inflow hydrograph during the levee breaching. However, other methods can be adopted: in Chapter 5, the estimation of an unknown inflow hydrograph in an ungauged river section will be assessed by means of a Bayesian methodology.

With reference to the uncorrected hydrograph, the new one presents greater discharges, and particularly an increase in the peak value of about 80 $m^3/s$ after the failure (Figure 1.8). The new simulated water levels at Ponte Alto agree with the registered data (see Figure 1.7), and the issue of underestimating the water levels after the failure is addressed.

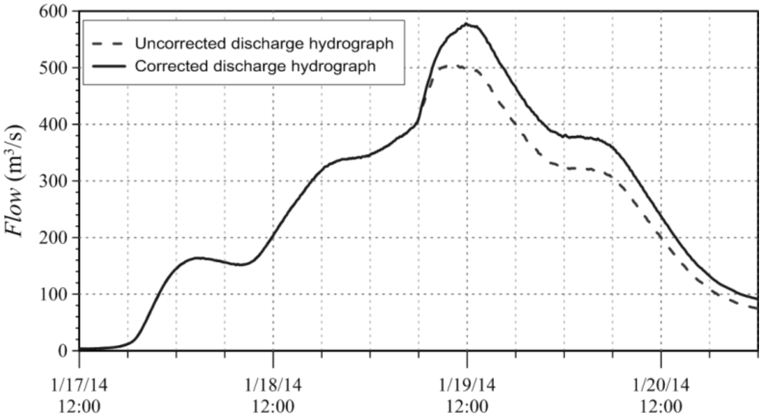


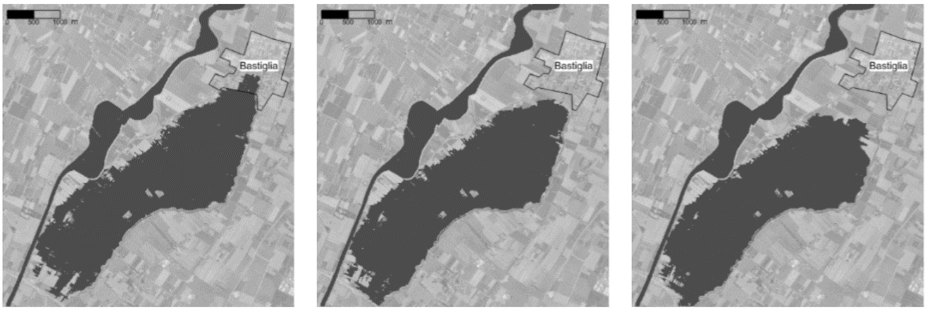
*Figure 1.8. Upstream boundary condition (Ponte Alto) for the 2014 flood event.*

### 1.4.3. Rural and urban floodplain area roughness

Calibration of the roughness values in the flooded area outside the river region was accomplished in two steps, with the aim of evaluating two different Manning’s values, suitable to represent the roughness of rural and urban areas, respectively. With regard to the rural areas, the adoption of a unique roughness coefficient was fairly justified by the almost homogeneous land use characterizing the cropland in the low plain near Modena. As described in the Introduction, urban areas were considered as regions of higher resistance, characterized by their specific Manning’s coefficient to be estimated afterwards the calibration of the rural areas.

The resistance parameter of the rural areas outside the riverbed was evaluated holding the in-channel roughness at 0.05 $m^{-1/3}\cdot s$ and executing three different runs, each characterized by a different uniform value of $n_f$ equal to 0.07, 0.05 and 0.03 $m^{-1/3}\cdot s$, respectively. The target for the result evaluation was the arrival time of the flooding at the southwestern boundary of Bastiglia urban area, which actually occurred 6.75 hours after the bank failure: the choice of this time guaranteed that the flood propagation did not pose an interest to any urban area yet.

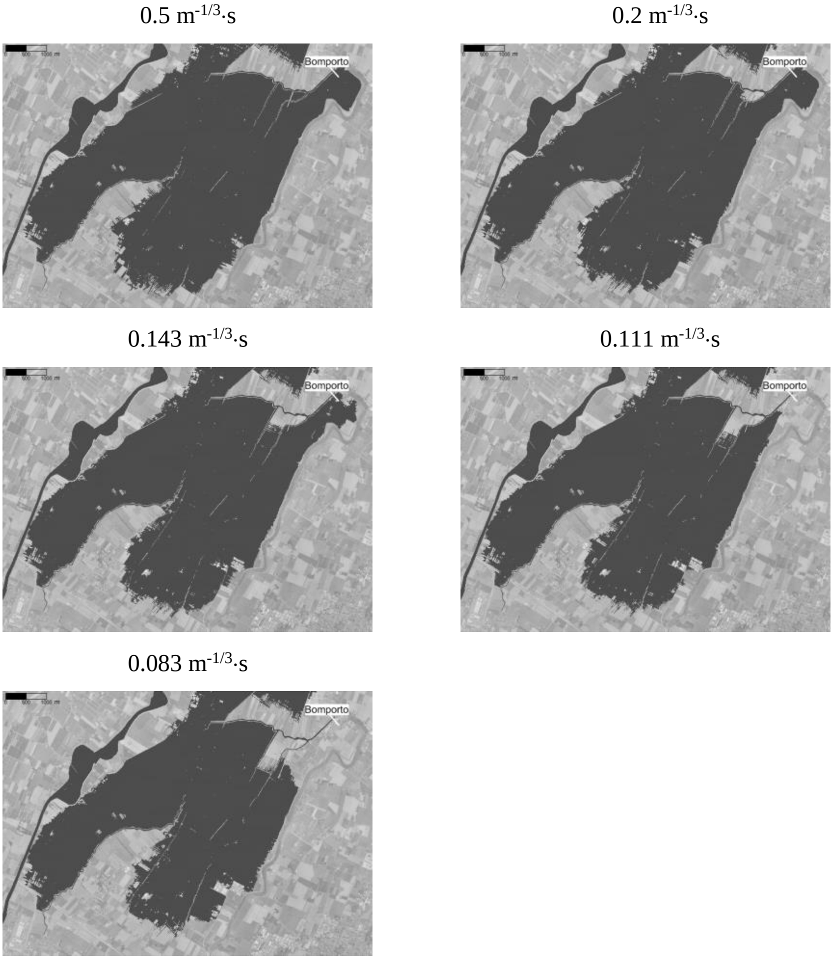
As depicted in Figure 1.9, the roughness value $n_f = 0.05\ \mathrm{m^{-1/3}\cdot s}$ allows again correctly reproducing the arrival time of the wetting front at the entrance of the urban area of Bastiglia, while the arrival is anticipated and delayed of about half an hour adopting the lower and the higher roughness coefficients, respectively. The obtained results show that the flooding dynamics is not dramatically sensitive to the variation of the resistance parameter assigned to the rural areas.



$0.03\ \mathrm{m^{-1/3}\cdot s}$ $0.05\ \mathrm{m^{-1/3}\cdot s}$ $0.07\ \mathrm{m^{-1/3}\cdot s}$

*Figure 1.9. Calibration of the rural areas: maps of the flooding extent at 0:45 P.M. on 19th January 2014 for the three investigated roughness values.*

Assuming the roughness in rural areas equal to $0.05\ \mathrm{m^{-1/3}\cdot s}$, the urban areas resistance parameter was calibrated focusing on the arrival time of the flooding at Bomporto and investigating five different values of $n_f$, equal to 0.083, 0.111, 0.143, 0.2 and $0.5\ \mathrm{m^{-1/3}\cdot s}$. As shown in Figure 1.10, the value $n_f = 0.143\ \mathrm{m^{-1/3}\cdot s}$ allows correctly reproducing the arrival of the flood front at Bomporto, while the flood arrival is anticipated by 2.75 and 1 hours considering $n_f$ equal to 0.5 and $0.2\ \mathrm{m^{-1/3}\cdot s}$ respectively, and it is delayed by 1.5 and 4.5 hours when $n_f$ is equal to 0.111 and $0.083\ \mathrm{m^{-1/3}\cdot s}$, respectively.



$0.5\ m^{-1/3}\cdot s$

$0.2\ m^{-1/3}\cdot s$

$0.143\ m^{-1/3}\cdot s$

$0.111\ m^{-1/3}\cdot s$

$0.083\ m^{-1/3}\cdot s$

*Figure 1.10. Calibration of the urban areas: maps of the flooding extent at 10:30 A.M. on 20th January 2014 for the five investigated roughness values.*

## 1.5. Model results

The flooding event was simulated from 12:00 P.M. of January 17th 2014, when the registered water levels at the gauging stations started to increase significantly, till 6:00 A.M. of January 21th 2014 when the flood event might be considered as concluded (Figure 1.11).

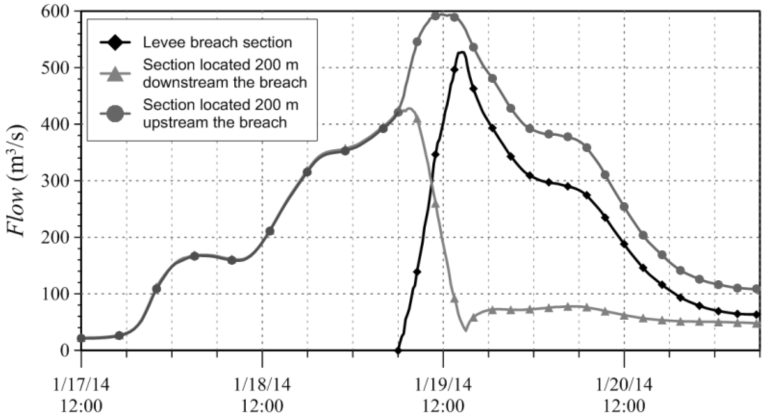


*Figure 1.11. Discharge hydrographs at the sections of the breach, 200 m upstream and 200 m downstream along the river.*

The 2D simulations entailed a storage of about 2671 MB on the K40 Tesla graphic card, using 7.2 million computing cells and required about 6 computing hours to simulate the 90 hours period.

### 1.5.1. Flow and stage hydrographs near the breach

According to field observations, the breach was assumed to progressively evolve during a period of 9 hours reaching a maximum width of about 80 m. Figure 1.11 shows the flow hydrographs computed at the breach section and in other two sections along the river reach, 200 m upstream and downstream of it, respectively. At the end of the breach evolution process, about 525 $m^3/s$, which correspond to more than 87% of the upstream discharge, are flowing through the breach, whereas only the remaining 13% proceed downstream.

Figure 1.12 shows the stage hydrograph close to the breach. Here the spatial variability of the water stage is high, due to the drawdown effect exerted by the outflowing discharge, so the figure is just indicative. After the beginning of the breaching process (about 6:00 A.M. of January 19th) water levels still increase, reaching the highest value of 36.05 m a.s.l. around 8:00 A.M. Then the levels start to decrease quite abruptly reaching the value of about 33 m a.s.l. in the following 7 hours. After the stabilization of the breach opening, the water stages continue to decrease, with a further lowering of 2 meters in about 33 hours.

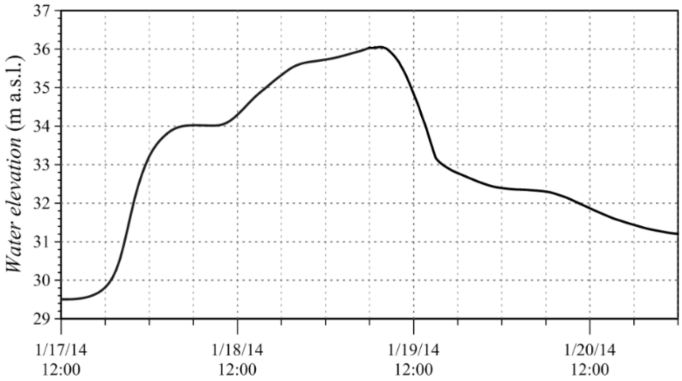
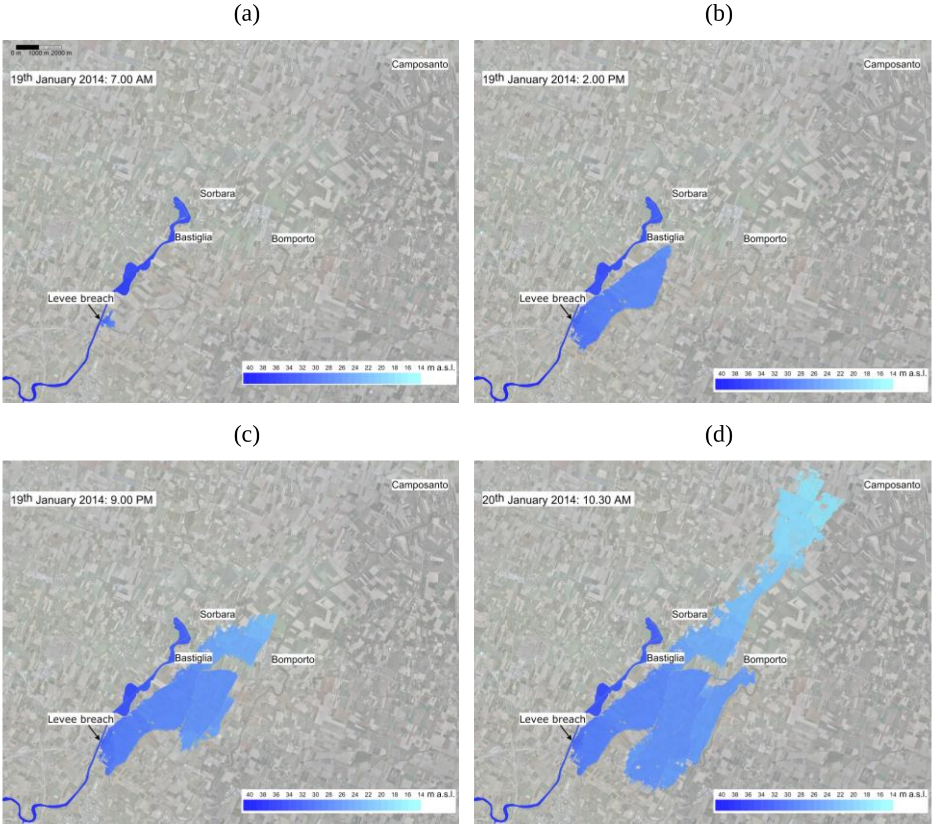


*Figure 1.12. Stage hydrograph referred to a point in the river close to the breach.*

### 1.5.2. Chronological reconstruction of the flooding

After having analyzed the discharge and stage hydrographs near the breach in the previous section, Figure 1.13 highlights some significant frames of the flooding evolution. According to the available observations collected during the event, the timing of the simulated inundation is in good agreement with the real flood event evolution. About 1 hour after the levee breach (7.00 A.M. of 19th January 2014), water quickly flooded near lowlands in northeast direction (Figure 1.13-a). At 2.00 P.M. the flood wave reached the center of the urban area of Bastiglia (Figure 1.13-b), where most people were already evacuated. A few hours later (9.00 P.M.), the countryside of Sorbara was flooded (Figure 1.13-c). At 10:30 A.M. of 20th January 2014, water reached the central Matteotti square in Bomporto (Figure 1.13-d). At 2:00 P.M. of 20th January 2014 the flood wave reached the SP5 road near Camposanto (Figure 13-e) that was overtopped only about 4 hours later (6:00 P.M. of 20th January 2014) (Figure 1.13-f).


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

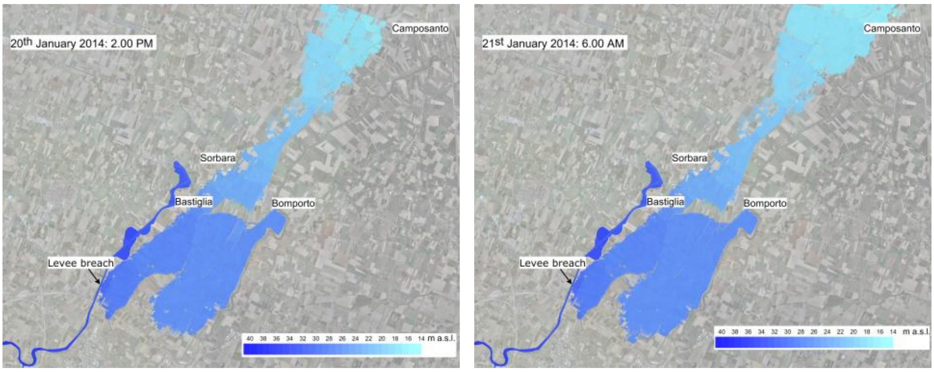

(e) (f)

*Figure 1.13. Numerical reconstruction of the flooding event: 1 hour after the bank failure (a), flooding of Bastiglia (b), 15 hours after the bank failure (c), flood wave arrival at Bomporto (d), 32 hours (e) and 48 hours (f) after the bank failure.*

The chronological evolution of the flooding over the entirely analyzed domain highlights the blocking/delaying effects that embankments of artificial channels, roads and railways exerted on the flooding dynamics. As an example, points A and B in the map of the computed flood arrival times (Figure 1.14) are located 1 km apart but on two opposite sides of a channel embankment, and they were reached by the flooding 10.5 and 24.5 hours after the levee failure, respectively.

Additionally, the map illustrates that some areas in the most southern part of the domain, where the ground elevation is higher, were back flooded only 40-45 hours after the levee collapse (blue colors), and a long delay (more than 24 hours, green colors) occurred between the breach triggering and the flooding of Bomporto.

Focusing on the maximum-modeled water depths, Figure 1.15 depicts that apart from the river and channel beds, the maximum values occurred in the southern portion of the flooded region, where water depths raised up to 2.6-2.8 m, by virtue of the blocking effect of the several embankments crossing this area; these values agree with the observations. Smaller depths (0-1 m) were computed (and observed) in the northern portion of the flooded region, with the exception of the area just upstream the SP5 road near

Camposanto, where maximum depths of 1.4-1.6 m resulted, owing to the retaining effect of the road embankment.

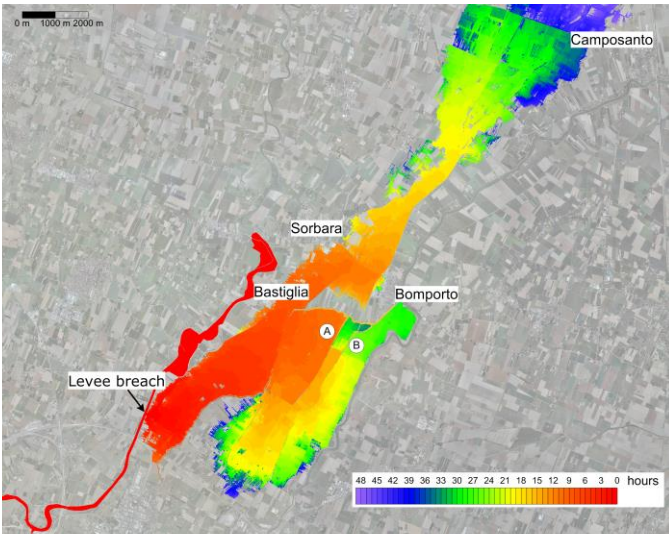


*Figure 1.14. Map of flooding arrival times. Time zero referees to the breach opening.*

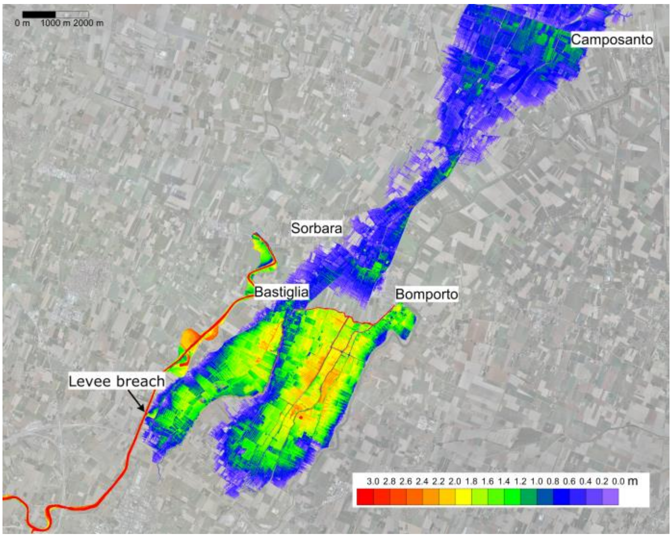


*Figure 1.15. Map of maximum simulated water depths.*

### 1.5.3. Validation of the results

Having assessed the chronological evolution of the simulated event, the resulted flooding extent is now compared with aerial images acquired on the 22th January 2014 at 1.30 PM. As illustrated in Figure 1.16, the numerical model correctly reproduced the flooding of Bomporto, where the east portion is flooded, whereas the west one is not (Figure 1.16 a-b), and also the influence that the disused railway embankment (North-East of Bastiglia) exerted on the flooding of the surrounding cultivated lands (Figure 1.16 c-d).

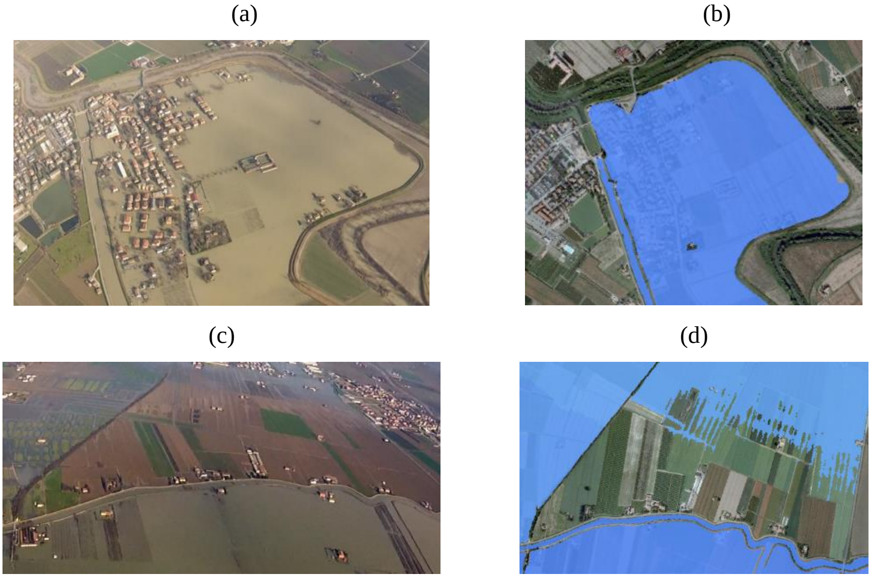


*Figure 1.16. Bomporto town, 22th January 2014 at 1.30 P.M.: comparison among simulated flooded areas (b, d) and aerial views (a, c).*

As the COSMO-SkyMed 4 satellite crossed the areas of Modena on the 21th January 2014, acquiring a 3 m resolution image at 5:20 P.M., the comparison between numerical results and SAR data was also performed (Figure 1.17). Even if, due to the satellite flight path, the extreme northeast portion of the investigated domain is missing, the inundation extent is well enough captured by the numerical model, especially in the south and northwest ends.

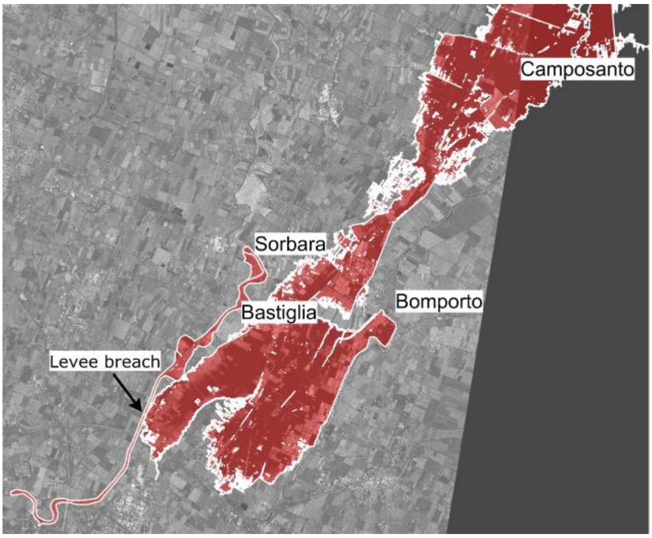


*Figure 1.17. Comparison between the SAR image (foreground) and the simulated flooding (red area).*

Despite the high-resolution information of the flood extent available in the SAR image, in a number of land parcels, located at the edge of the flooded area, the image is biased due to multiple reflections between the water and the emerging vegetation, which causes an increased backscatter compared to that from a smooth open water surface (Mason et al., 2012a, 2012b). This is confirmed by the available aerial views captured during the event, one of which is shown in Figure 1.18. Flooded land parcels, characterized by low backscatter, are apparently contiguous to high backscatter areas, which would be considered as dry on the basis of direct analysis of the SAR image only. This issue makes impossible the evaluation of quantitative metrics, like the one proposed by Bates and De Roo (2000), to measure the capability of the numerical model to reproduce the observed inundation extent. However, the availability of aerial photos depicting the land parcels of doubtful characterization confirms unequivocally that the numerical simulations correctly reproduced the extent of the inundation.

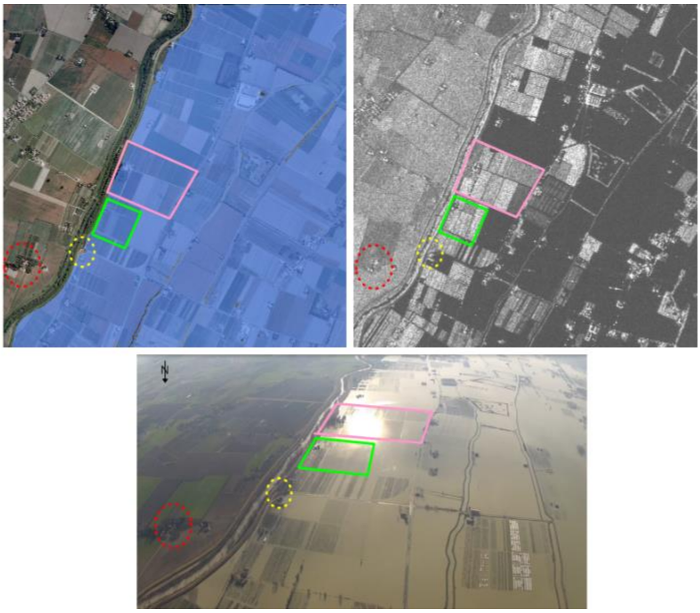


*Figure 1.18. Comparison among computed results, SAR image and aerial image for some land parcels detected as not flooded in the binary SAR.*

## 1.6. Concluding remarks

In this chapter, the real flooding caused by the 2014 levee breach on the Secchia River has been simulated adopting a fully 2D-SWE parallelized model.

The validation of the model results through aerial images and field data has demonstrated that either the adopted calibration procedure (estimation of different roughness coefficients for rural and urban areas), the upstream boundary definition (calculation of an *ad hoc* rating curve with the developed breach) and the 2D-SWE model have successfully contributed to the correct reconstruction of the studied event.

Moreover, dealing with high-resolution meshes and fully 2D-SWE model, another relevant aspect is the computational time: the adoption of the GPU model allowed achieving a ratio of physical time to runtime greater than 15 (about 6 computing hours are necessary to simulate 90 hours of physical time).

These positive results lead to consider the possibility of adopting this kind of instruments in the framework of flood risk management.

On one side, flooding modelling results could be used during the urban development planning in order to define the hazard of some specific zones, and therefore decrease the country vulnerability related to levee breaches. Moreover, the creation of a database of potentially flooding scenarios appears as a useful tool for defining *a priori* evacuation plans, by evaluating for example the flooded and non-flooded streets.

On the other side, the results of flooding modelling could be used also in case of emergency for the civil protection activities; this is especially true with respect to those territories that, for the relative distance from the breach site, would have been involved by the flooding many hours after the levee collapse. Particularly, the knowledge of simulated maximum water depth may suggest how to adopt some countermeasures during the event (sandbags, inflatable rubber dams, etc.), in order to avoid or to reduce the water depths in different urban areas.

# Chapter 2

## A non-uniform efficient grid type: the Block Uniform Quadtree grid

### 2.1. Introduction

The issue of decreasing the computational costs, in modelling large-scale simulations at high-resolution, is relevant also in the framework of parallelized models, which commonly achieve good speed-up, if compared to serial codes. Nevertheless, simulations including wide flood plains typically require high-resolution only in some specific areas, which actually is a portion of the whole domain. Therefore, meshes with local refinements are created in order to capture the elements relevant for the flood propagation (i.e. levees, embankments, etc.) while simulating large domains. Conversely, the adoption of uniform resolution everywhere prevents the possibility of simulating floodings over larger areas (order of $10^3$ $km^2$), if somewhere in the domain a resolution of 1 - 2 m has to be adopted.

According to these reasons, uniform Cartesian grids, mostly adopted in GPU-accelerated models, have two major limitations: (1) the same resolution has to be adopted in the whole domain and (2) the shape of the domain has to be rectangular.

In order to resolve small-scale effects in limited areas while simulating large domains, non-uniform grids are widely adopted for non-parallel numerical models, either in the form of unstructured mesh (Begnudelli and Sanders, 2006; Brufau et al., 2004;) or hierarchical Quadtree (Greaves and Borthwick, 1999; Borthwick et al., 2000; Rogers et al., 2001; Liang et al., 2007, 2008). In the framework of GPU codes, Lacasta et al. (2014) developed several optimizations for a flood simulation model based on unstructured triangular grids that was further adopted for landslide (Lacasta et al., 2015a), sediment transport (Juez et al., 2016) and rainfall-runoff (Lacasta et al., 2015b) simulations. A full GPU implementation of a SWE model based on Adaptive Mesh Refinement has also been presented by Sætra et al. (2014). Despite all these contributions, the use of non-uniform meshes in GPU-enhanced numerical schemes still remains an ongoing challenge.

The principal reason is that the classical data structures of unstructured or Quadtree meshes cannot exploit the computational capability of GPUs; the main limitation is the arrangement in memory of data and, consequently, the cost of accessing graph-like data structures (as in typical Quadtree meshes) as opposed to plain matrix memorization, where regularity in accesses provides the best GPU throughput.

With the goal of providing a data structure that combines the benefits of non-uniform resolution meshes and the performances of plain matrix accesses, in this chapter a novel Block Uniform Quadtree grid (BUQ) for the 2D-SWE GPU code described in Section 1.2 is introduced. The key idea of BUQ grids is to discretize the domain using cell blocks with uniform resolution, while allowing different resolutions for different blocks of cells.

This Chapter is derived in part from Vacondio et al. (2017).

## 2.2. Block Uniform Quadtree grid

### 2.2.1. CUDA memory arrangement

In this section, some basic concepts of the CUDA implementation are recalled and some insights into the novel contribution, namely the capability of handling multiresolution grids, are provided.

The key idea of BUQ grids is to extend the scenario of Cartesian grid memorization to a multiresolution system, while preserving the computational organization of a data block. Therefore, each block, regardless its cell dimensions, contains $M \times M$ cells, but various resolution levels are used: level 1 with cell size $\Delta_1$, level 2 with cell size $\Delta_2=2\times\Delta_1$ up to level $n$ with cell size $\Delta_n=2^{n-1}\times\Delta_1$. In the present thesis, $M$ has been assumed equal to 8 or 16, but it can be set equal to any power of two.

This data organization requires a different type of storage of the multiresolution grid, with reference to the Cartesian one: blocks with various resolutions are tiled according to their code index and therefore the original neighborhood relationships among blocks is not maintained in the final tiling. Because of GPU limitations on threads organization, blocks are actually tiled into a square with a power of two side, large enough to contain all the blocks.

Figure 2.1 illustrates an example of arrangement, where 10 blocks at three different levels of resolution are rearranged into a 4×4 matrix of blocks. In Figure 2.1-a, the block position in the physical space is represented, whereas the same blocks are allocated in memory as shown in Figure 2.1-b (in the images different colors correspond to different resolution levels). Blocks with different resolution have different size in the physical space (Figure 2.1-a), whereas they have the same size in the memory space (Figure 2.1-b), since each block contains information about the same number of cells (8×8 or 16×16).

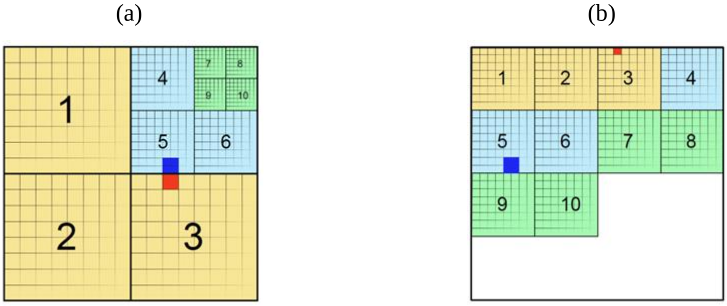


*Figure 2.1. Example of a BUQ grid with 8x8 cell blocks and three resolution levels: the representation in the physical space (a) and the memory allocation (b).*

### 2.2.2. Grid generation procedure

The BUQ grids are introduced for simulating flood propagation over large high-resolution domains, still preserving fine resolution in specific zones to accurately reproduce local phenomena (i.e. the area close to a dam in a dam-break problem, the region of a levee near a breach or a motorway cloverleaf interchange).

With this aim, the user can impose different resolution zones before the simulation starts, by defining seeding points, which are listed in an input file containing their planimetric coordinates and the associated resolution level.

Then, the grid is generated through an automatic procedure, which determines the set of blocks (patches of $M \times M$ cells) at different resolutions that partition the original high-resolution grid. Given $n$ resolution levels, the algorithm firstly identifies (Algorithm 2.1 - line 2) the cells that are required to have the highest resolution and it recognizes the blocks with resolution 1 (grid size $\Delta_1$) that contain such cells (line 3-4).

Then, the algorithm iterates among the lower resolution levels by enforcing three rules:

1. Halving of the resolution from one level to the next lower one;
2. Respect of the quadtree block division. A square region (i.e. a block with lower resolution) can be divided into 4 quadrants (i.e. 4 blocks with higher

resolution) and thus, when a subset of these quadrants is identified, the remaining quadrants are forced at the same resolution, in order to properly tile the area (line 5). Consider as an example the case in Figure 2.2, with four different resolution levels and one seeding point at resolution 1. Due to this partition rule, the algorithm enforces resolution 1 not just to the block that contains the seeding point (the traced one) but also to the other three neighboring blocks that, together with the original one, form a possible block at resolution 2 (Figure 2.2-a).

It is to be noted that the quadtree spatial partitioning associates each node of the quadtree to a specific block of cells, instead of a single cell, as to guarantee a uniform GPU computation within each block, while keeping an adaptive multi-resolution division of the grid;

3. Control in the resolution decrease between adjacent blocks. During the partition, two neighboring blocks (including diagonal adjacency) cannot differ more than one level in resolution. Therefore, for each selected block at level 1 (line 8), the corresponding block that includes it at resolution level 2 is retrieved (line 9), and the neighbors of such block are forced to resolution level 2, if not already used at level 1 (line 11). Due to this rule, 8 blocks at resolution 2 are identified in Figure 2.2-a.

The same procedure is then repeated for each resolution level $i$=2…$n$ (loop at line 1 of Algorithm 2.1), considering the constraint assigned by the seeding points and those resulted from the resolution level $i$-1. This means that, in the considered example, when the iteration with resolution levels $i$=2 and 3 are operated, the third rule forces 12 blocks to be at resolution level 3 (Figure 2.2-b) and 4 (Figure 2.2-c), respectively.

It is noteworthy that the partition process ensures keeping minimal the number of blocks, while preserving the structural constraint (the three rules) and data dependent constraints (e.g. the resolution imposed by the user): this provides the largest reduction in cells to be computed.

*Algorithm 2.1: pseudocode to create blocks at different resolution levels*.

```
for each level i=1..n
  for each c in cell_forced(level)
    b=find_block(c,level)
    set_block(b,level)
    force_quadrants(b,level)
  end
  if (level<n)
  for each b in set_block(level)
    b1=compute_next_level_block(b,level+1)
    for each b2=neighbor(b1)
      if (b2 not used at level) set_block(b2,level1)
    end
  end
end
```

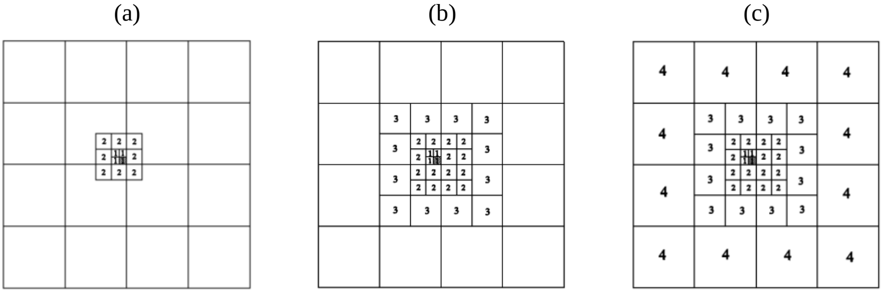


*Figure 2.2. Example of the grid partitioning using 1seeding point at resolution 1 and 4 resolution levels.*

Besides the downsampling process previously described, the grid generation procedure excludes all the blocks outside the domain, which are not even loaded in the GPU memory. Operationally, the domain is identified by means of a closed polyline (Figure 2.3-a), and the information related to the blocks that is located outside are simply not taken into account. In comparison with Cartesian grids, where rectangular domains have to be considered, this leads to a considerable reduction of the allocated memory on the GPU, as it will be shown in Section 2.3.

In order to summarize the overall procedure that leads to the grid generation, Figure 2.3 exemplifies the three relative steps: in (a) the selected domain and the two seeding points, in which the resolution levels are enforced equal to 1 and 3, respectively, are shown. In (b) the grey blocks outside the domain are not loaded in the memory, whereas blocks that are partially inside the polyline are taken into account in the computation. In (c), the resolution levels around the seeding points and in the adjacent blocks are defined according to the three rules described before.

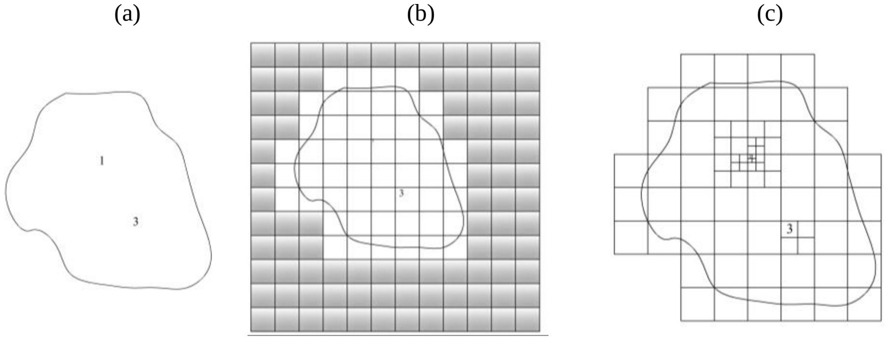


*Figure 2.3. Example of the BUQ grid generation procedure: the domain and two seeding points (a), the blocks that are not loaded in the memory (b) and the resolution level definition (c).*

### 2.2.3. Information exchange between cells at different resolution

During the computation, a SWE model based on a finite volume scheme requires simultaneously the knowledge of the conserved variables in different cells that therefore have to be read from the memory; focus for instance on the core of the scheme, the flux computation, which requires information about the conserved variables of the four neighboring cells.

However, the arrangement of the multiresolution data structure into memory (Section 2.2.1.) determines a crucial issue for the exchange of information between cells: close-by blocks in the domain are not necessarily stored in close-by memory locations. As an example, the red cell in Figure 2.1-a belongs to the block 3 and has as north neighbor, two blue cells at lower resolution, belonging to block 5.

In order to retrieve the actual memory location of the required cell(s), topological information about neighbors for each block is stored during the grid partitioning, namely the Cartesian position, the eight neighboring blocks references and their resolution levels.

Moreover, dealing with multi-resolution, the neighbors blocks may have a higher, same, or lower resolution (Figure 2.4).

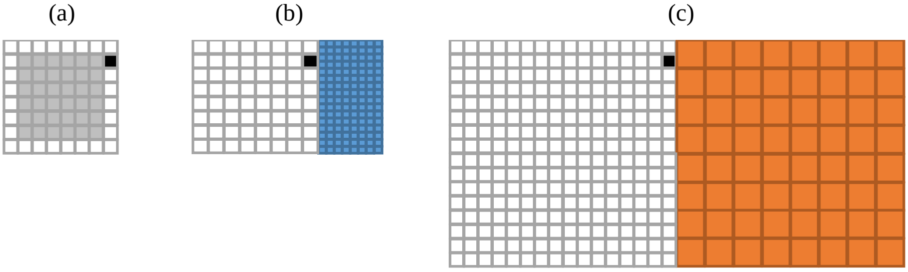


*Figure 2.4. Neighboring research for a cell (black one) in an 8x8 block (a): the neighbor has higher (b) or lower (c) resolution.*

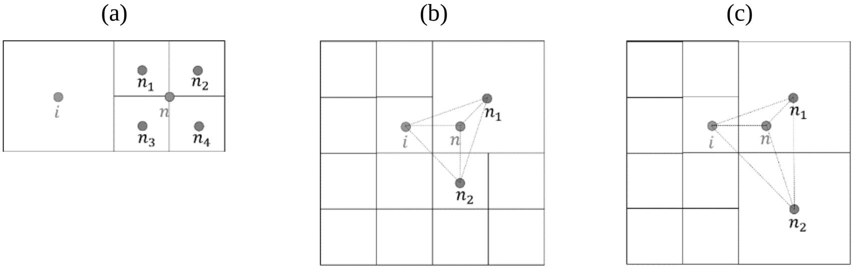
In case of a change in resolution between adjacent blocks, two special sub-cases may arise: two higher resolution blocks face a lower resolution block (two neighbors in place of one, Figure 2.4-b), and the symmetric case, where a higher resolution block face half lower resolution block (Figure 2.4-c). These two cases require extra information to be stored, in order to reconstruct the topological relationships between cells during computation.

When information about a neighboring cell on a different block is requested during the computation, the algorithm reconstructs the neighbor block memory address from the precomputed information; a few tests and arithmetic operations are needed to recover the data and thus to optimize the neighbors search algorithm.

Moreover, depending on the position of the original cell, it adapts the neighbor cell address in order to select the correct cell position within the block. If a higher resolution is defined in the neighbor block, two different cells need to be addressed.

It is worth noting that the block discretization reduces the overhead by transferring the problem only to cells lying on the block borders, whereas the inner cells have neighbors of the same resolution (grey cells in Figure 2.4-a).

After having clarified the procedure that retrieves the cells in memory, the focus shifts now on the related conserved variables. If the neighbor block has a different resolution, the conserved variables at neighbor grid point *n*, which actually does not exists, are reconstructed according to the natural neighboring interpolation procedure (Liang, 2011), with configurations shown in Figure 2.5.



Legend:

● Cell *i*: cell requiring information from its east neighboring cell

● Cell *n*: east neighbor of cell *i* that does not exist and therefore need to be interpolated

● Cells $n_{1,2,3,4}$: real east neighboring cells of cell *i*

*Figure 2.5. Neighbor conserved variables reconstruction for cell i: the east neighbor has a higher resolution (a), or a lower one with the southeast at same (b) or lower (c) resolution.*

Assuming the quantities on cell *i* to be updated, and a higher resolution east block, then the following formula applies (Figure 2.5-a):

$$\mathbf{U}_n = \frac{1}{4}\left(\mathbf{U}_{n_1} + \mathbf{U}_{n_2} + \mathbf{U}_{n_3} + \mathbf{U}_{n_4}\right) \tag{2.1}$$

Conversely, when the neighbor block has a lower resolution two different sub-cases are possible. If the southeast block has the same resolution of the one where cell *i* is located (Figure 2.5-b) the following relation holds:

$$\mathbf{U}_n = \frac{\mathbf{U}_i}{4} + \frac{\mathbf{U}_{n_1}}{2} + \frac{\mathbf{U}_{n_2}}{4} \tag{2.2}$$

Finally, if the southeast block has the same resolution of the east one (Figure 2.5-c), the natural neighboring procedure leads to the following relation:

$$\mathbf{U}_n = \frac{\mathbf{U}_i}{3} + \frac{\mathbf{U}_{n_1}}{2} + \frac{\mathbf{U}_{n_2}}{6} \tag{2.3}$$

The interpolations in west, north and south directions can be obtained analogously by rotating Figure 2.5.

## 2.2.4. Flux and source term computation between cells at different resolution

As explained in Section 1.2.1, the Cartesian model preserves water depth positivity in a uniform 2D scheme and ensures the *C-property* in presence of wet-dry fronts.

However, while adopting non-uniform resolution grids, special attention has to be devoted to ensure mass and momentum conservation between cells laying on block borders at different resolution. This means, for instance, that the computation of the flux across the east interface of cell *i* in Figure 2.6, where two higher resolution neighboring cells are present, requires the computation of two distinct fluxes $F_E'$ and $F_E''$ outward directed (Liang and Borthwick, 2009). Then, the numerical flux $F_E$, used in the updating Equation (1.26), is evaluated as the sum of these two contributes.

It is relevant to remark that the computation of a unique $F_E$ value, without the splitting technique previously described, would lead to a violation of the mass conservation principle.

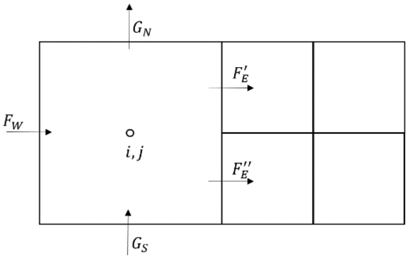


*Figure 2.6. Fluxes definition for cell i,j with higher resolution neighbors on the east edge.*

Dealing with a second order accurate in space scheme, the definition of the fluxes $F_E'$ and $F_E''$ requires the reconstruction of the conserved variables at the cell borders and hence, for the considered case, two reconstructed values are present at the east interface (Figure 2.7): $\mathbf{U}_{i+\frac{1}{2},j}^{L'_*}$ and $\mathbf{U}_{i+\frac{1}{2},j}^{L''_*}$.

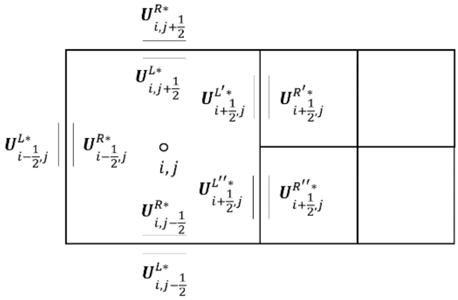


*Figure 2.7. Reconstructed conserved variables* **U** *(η, uh, vh) at cell i,j edges.*

Consistently, in order to ensure a stable and well-balanced scheme also in presence of wet-dry fronts, the slope source term $\mathbf{S}_{b_{i,j}}$ for cell *i,j* has to be calculated as to balance the flux contributions. Therefore, $\mathbf{S}_{b_{i,j}}$ for the momentum equation along the *x* direction is calculated as:

$$S_{b_{i,j}} = \frac{1}{2}\left(S'_{b_{i,j}} + S''_{b_{i,j}}\right) \tag{2.4}$$

where the two split values are:

$$S'_{b_{i,j}} = g\bar{\eta}'_{i,j}\left(\frac{z'_{i+\frac{1}{2},j} - z_{i-\frac{1}{2},j}}{\Delta x_{i,j}}\right) + \bar{S}'_{0_{i-\frac{1}{2},j}} + \bar{S}'_{0_{i+\frac{1}{2},j}} \tag{2.5}$$

$$S''_{b_{i,j}} = g\bar{\eta}''_{i,j}\left(\frac{z''_{i+\frac{1}{2},j} - z_{i-\frac{1}{2},j}}{\Delta x_{i,j}}\right) + \bar{S}''_{0_{i-\frac{1}{2},j}} + \bar{S}''_{0_{i+\frac{1}{2},j}} \tag{2.6}$$

where the first terms represent the central discretization of the bed slope source term (Equation 1.5) and the other ones are added to deal with dry areas:

$$\bar{\eta}'_{i,j} = \frac{1}{2}\left(\eta^{R*}_{i-\frac{1}{2},j} + \eta^{L'*}_{i+\frac{1}{2},j}\right) \tag{2.7}$$

$$\bar{\eta}''_{i,j} = \frac{1}{2}\left(\eta^{R*}_{i-\frac{1}{2},j} + \eta^{L''*}_{i+\frac{1}{2},j}\right) \tag{2.8}$$

and:

$$\bar{S}'_{0_{i-\frac{1}{2},j}} = g\Delta z_{i-\frac{1}{2},j}\left[\frac{z'_{i+\frac{1}{2},j} - \left(z_{i-\frac{1}{2},j} - \Delta z_{i-\frac{1}{2},j}\right)}{2\Delta x_{i,j}}\right] \tag{2.9}$$

$$\bar{S}'_{0_{i+\frac{1}{2},j}} = g\Delta z'_{i+\frac{1}{2},j}\left[\frac{\left(z'_{i+\frac{1}{2},j} - \Delta z'_{i+\frac{1}{2},j}\right) - z_{i-\frac{1}{2},j}}{2\Delta x_{i,j}}\right] \tag{2.10}$$

$$\bar{S}''_{0_{i-\frac{1}{2},j}} = g\Delta z_{i-\frac{1}{2},j}\left[\frac{z''_{i+\frac{1}{2},j} - \left(z_{i-\frac{1}{2},j} - \Delta z_{i-\frac{1}{2},j}\right)}{2\Delta x_{i,j}}\right] \tag{2.11}$$

$$\bar{S}''_{0_{i+\frac{1}{2},j}} = g\Delta z''_{i+\frac{1}{2},j}\left[\frac{\left(z''_{i+\frac{1}{2},j} - \Delta z''_{i+\frac{1}{2},j}\right) - z_{i-\frac{1}{2},j}}{2\Delta x_{i,j}}\right] \tag{2.12}$$

where

$$\Delta z'_{i+\frac{1}{2},j} = max\left[0, -\left(\eta^{L'}_{i+\frac{1}{2},j} - z'_{i+\frac{1}{2},j}\right)\right] \tag{2.13}$$

$$\Delta z_{i-\frac{1}{2},j} = max\left[0, -\left(\eta^{R}_{i-\frac{1}{2},j} - z_{i-\frac{1}{2},j}\right)\right] \tag{2.14}$$

$$\Delta z''_{i+\frac{1}{2},j} = max\left[0, -\left(\eta^{L''}_{i+\frac{1}{2},j} - z''_{i+\frac{1}{2},j}\right)\right] \tag{2.15}$$

In this way, the *C-property* is guaranteed for BUQ grids, also in presence of wet/dry fronts.

## 2.3. Assessment of the model performances

This section focuses on the validation and the evaluation of the model performances, which is assessed through four test cases characterized by different issues: a theoretical case with analytical solution, a dam-break laboratory experiment, a real and a hypothetical flood event.

All the simulations were run using a K40 Tesla® GPU. For the first three test cases, accuracy and efficiency of the simulations are compared against the results obtained with uniform Cartesian grids. Particularly, the computational efficiency is evaluated by means of two indicators, the compression rate $C_R$ and the speed-up $S_U$, defined as:

$$C_R = \frac{N_C}{N} \qquad S_U = \frac{T_C}{T} \tag{2.16}$$

where $N$ and $T$ are the number of allocated cells and the computational time, respectively, for the considered simulation with the BUQ grid; $N_c$ and $T_c$ are the same quantities referred to an identical simulation run using a Cartesian Grid with grid size equal to the maximum resolution adopted in the BUQ grid $\Delta_1$.

Due to the memory requirements, the fourth test case cannot be simulated with uniform Cartesian grid and thus compression rate $C_R$ and speed-up $S_U$ cannot be computed.

## 2.3.1. Analytical test

A steady vortex circulating clockwise on a frictionless and horizontal bed was performed and the modeled results were compared against the analytical solution, which is described in terms of water depth and velocities by the relations (Sanders and Bradford, 2006):

$$h = h_0 + \frac{U_0^2}{4g}\left[1 - \frac{2r}{r_0} exp\left(\frac{-2r}{r_0}\right) - exp\left(\frac{-2r}{r_0}\right)\right] \tag{2.17}$$

$$u = \frac{U_0(y - y_0)}{r_0} exp\left(\frac{-r}{r_0}\right) \tag{2.18}$$

$$v = \frac{-U_0(x - x_0)}{r_0} exp\left(\frac{-r}{r_0}\right) \tag{2.19}$$

where $x_0$ and $y_0$ are the coordinates of the vortex center, $r = \sqrt{x^2 + y^2}$, $h_0$ is the water depth at the center of the vortex, $U_0$ and $r_0$ are a characteristic velocity and length scales of the vortex, respectively. The parameters were set at the following values: $h_0 = 10$ m, $U_0 = 1.5$ m/s, $r_0 = 100$ m, $x_0 = 0$ m, $y_0 = 0$ m, $r = 2000$ m. The velocity magnitude is equal to zero in the center and reaches the maximum value equal to $U_0 e^{-1}$ at $r = r_0$.

Eighteen simulations have been performed (Table 2.1) by varying:

- The minimum cell size, that is the maximum resolution:
    - $\Delta x_{min} = 2$ m (z-y-1 in Table 2.1);
    - $\Delta x_{min} = 4$ m (z-y-2 in Table 2.1);
    - $\Delta x_{min} = 8$ m (z-y-3 in Table 2.1);
- The type of the adopted mesh:
    - Cartesian with size $\Delta x_{min}$ (z-C-x in Table 2.1);
    - BUQ with uniform resolution $\Delta x_{min}$ in the whole domain (z-U-x in Table 2.1);
    - BUQ with $\Delta x_{min}$ corresponding to the resolution level 1 and varying the resolution inside the domain (z-Q-x in Table 2.1);
- The block size:
    - 8×8 cells (8-y-x in Table 2.1);
    - 16×16 cells (16-y-x in Table 2.1).

In the non-uniform BUQ grids (z-Q-x) the highest resolution has been imposed in the region close to the center of the vortex (dashed white circle in Figure 2.8), where highest velocity and water depths gradients occur, while decreasing values have been allowed reaching the edge of the domain, up to resolution level 4.

The detail of the vortex center shown in Figure 2.8, gives evidence of the resolution decrease for the 8-Q-1 grid: the 1st level corresponds to cells of $\Delta x_1 = 2$ m, the 2nd level to $\Delta x_2 = 4$ m, the 3rd level to $\Delta x_3 = 8$ m and the 4th level to $\Delta x_4 = 16$ m.

With reference to the entire resulted non-uniform BUQ grids, Figure 2.9 illustrates the configurations 8-Q-1 (a) and 16-Q-1 (b). It is noteworthy that the region with the lower resolution is bigger using blocks of 8×8 cells than 16×16 cells; this is confirmed also by the compression rate, equal to 21 and 18, respectively.

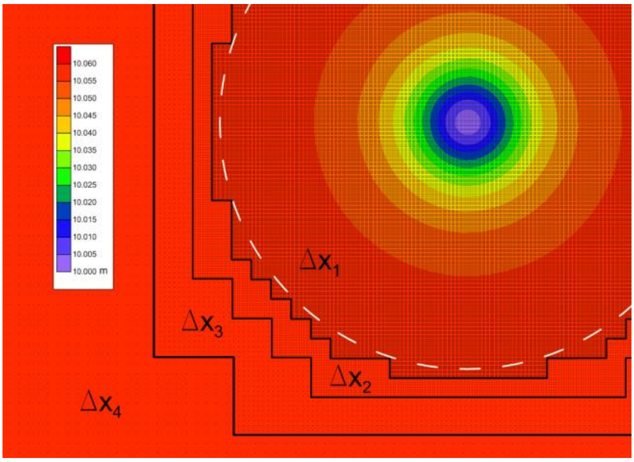


*Figure 2.8. Detail of the cell sizes* $\Delta x_1$*,* $\Delta x_2$*,* $\Delta x_3$*,* $\Delta x_4$ *in the resulted 8-Q-1 BUQ grid. In background the map of the water levels at the end of the simulation.*

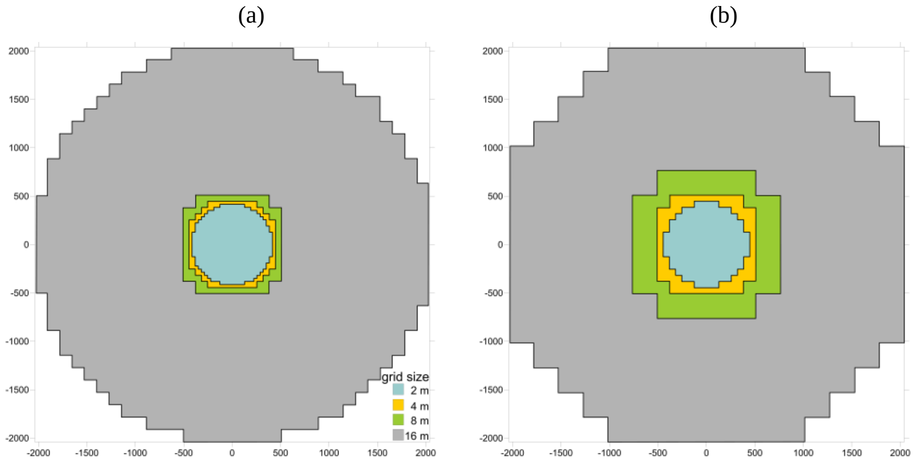


*Figure 2.9. Resulted non-uniform BUQ grids for the case 8-Q-1 (a) and 16-Q-1 (b).*

As suggested by An and Yu (2014), the simulation was run for 1000 s of physical time, and then the results were compared against the analytical solution.

The qualitative comparison shown in Figures 2.10 and 2.11, for water depth and velocity magnitude, along $y = 0$ and $y = x$ lines, respectively, highlights that both numerical results (for 16-Q-1 and 8-Q-1 grids) agree very well with the analytical solution.

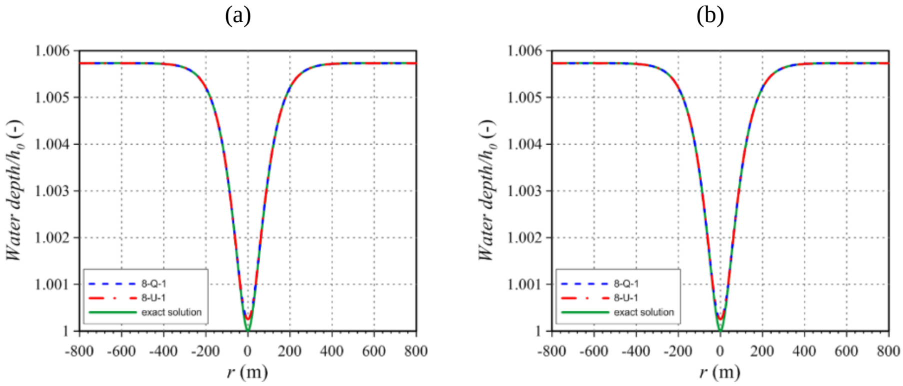


*Figure 2.10. Uniform (8-U-1) and non-uniform (8-Q-1) BUQ grids: comparison of the water depth trends along the y=0 (a) and y=x (b) lines.*

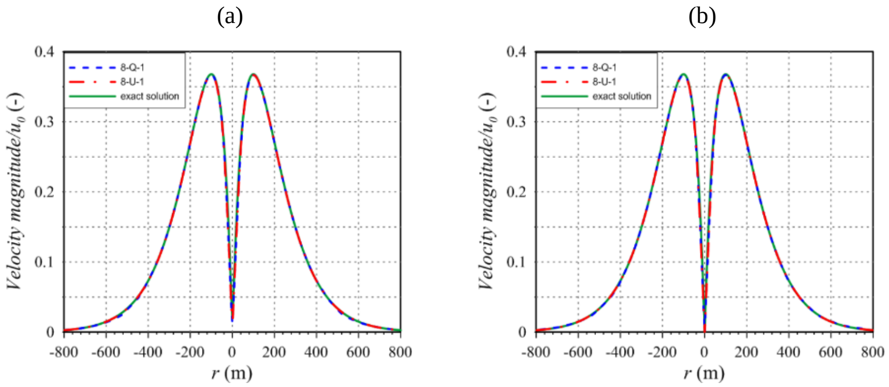


*Figure 2.11. Uniform (8-U-1) and non-uniform (8-Q-1) BUQ grids: comparison of the velocity magnitude trends along the y=0 (a) and y=x (b) lines.*

In order to provide quantitative information about the accuracy of the results, the $L_2$ non-dimensional error norm of water depth and velocity components has been estimated as follows:

$$L_2(f) = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left[\frac{f_{num}^{i} - f_{exact}^{i}}{f_0}\right]^2} \tag{2.20}$$

where $N$ is the total number of cells in the domain, $f$ is the variable of interest (water depth, velocity in $x$ and $y$ direction), $f_{num}$ and $f_{exact}$ are the numerical and analytical solutions and, finally, $f_0$ is a reference value, assumed equal to the parameters $h_0$ and $U_0$.

As outlined in Table 2.1, the $L_2$ error norms are in the range of $10^{-5}$-$10^{-6}$ (close to the machine precision), and $10^{-3}$-$10^{-4}$ for the water depth and the velocity ($v_x$ or $v_y$) variables, respectively. Moreover, the analysis confirms that uniform BUQ grids produce the same results of the Cartesian ones, whereas the adoption of non-uniform BUQ grids leads to a slight increase in the computed norm values, which is essentially due to the unavoidable approximation introduced by interpolating the conserved variables for cells at the interface between different resolution blocks (Section 2.2.3).

*Table 2.1. Vortex simulations: ID, minimum and maximum cell sizes ($\Delta x_{min}$, $\Delta x_{max}$), number of cells $N$, compression rate $C_R$, run times $R$, speed-up $S_U$ and $L_2$ norms for water depth and velocity components.*

| **ID** | **$\Delta x_{min}$ (m)** | **$\Delta x_{max}$ (m)** | **N ($10^6$)** | **$C_R$ (-)** | **R (s)** | **$S_U$ (-)** | **$L_2$ (*h*)** | **$L_2(v_x)=L_2(v_y)$** |
|---|---|---|---|---|---|---|---|---|
| 16-C-1 | 2 | 2 | 4.194 | 1.00 | 501.38 | 1.00 | 4.59E-06 | 2.45E-04 |
| 8-C-1 | 2 | 2 | 4.194 | 1.00 | 583.91 | 1.00 | 4.59E-06 | 2.45E-04 |
| 16-U-1 | 2 | 2 | 3.201 | 1.31 | 670.81 | 0.75 | 4.59E-06 | 2.45E-04 |
| 8-U-1 | 2 | 2 | 3.164 | 1.33 | 691.86 | 0.84 | 4.59E-06 | 2.45E-04 |
| 16-Q-1 | 2 | 16 | 0.240 | 17.50 | 54.63 | 9.18 | 9.55E-06 | 9.12E-04 |
| 8-Q-1 | 2 | 16 | 0.201 | 20.84 | 48.30 | 12.09 | 1.03E-05 | 1.03E-03 |
| 16-C-2 | 4 | 4 | 1.049 | 1.00 | 74.16 | 1.00 | 9.94E-06 | 7.75E-04 |
| 8-C-2 | 4 | 4 | 1.049 | 1.00 | 84.08 | 1.00 | 9.94E-06 | 7.75E-04 |
| 16-U-2 | 4 | 4 | 0.816 | 1.29 | 89.65 | 0.83 | 9.94E-06 | 7.75E-04 |
| 8-U-2 | 4 | 4 | 0.797 | 1.31 | 97.29 | 0.86 | 9.94E-06 | 7.75E-04 |
| 16-Q-2 | 4 | 32 | 0.083 | 12.64 | 11.70 | 6.34 | 2.89E-05 | 2.44E-03 |
| 8-Q-2 | 4 | 32 | 0.060 | 17.39 | 8.74 | 9.62 | 3.36E-05 | 2.87E-03 |
| 16-C-3 | 8 | 8 | 0.262 | 1.00 | 12.16 | 1.00 | 2.98E-05 | 2.28E-03 |
| 8-C-3 | 8 | 8 | 0.262 | 1.00 | 13.12 | 1.00 | 2.98E-05 | 2.28E-03 |
| 16-U-3 | 8 | 8 | 0.209 | 1.25 | 14.23 | 0.85 | 2.98E-05 | 2.28E-03 |
| 8-U-3 | 8 | 8 | 0.202 | 1.30 | 15.67 | 0.84 | 2.98E-05 | 2.28E-03 |
| 16-Q-3 | 8 | 32 | 0.040 | 6.56 | 3.60 | 3.37 | 6.82E-05 | 5.24E-03 |
| 8-Q-3 | 8 | 64 | 0.022 | 12.19 | 2.57 | 5.10 | 9.09E-05 | 7.02E-03 |

The efficiency of the BUQ grids, against the Cartesian ones, is assessed by investigating the runtimes and number of cells used to discretize the domain (Table 2.1).

For the simulations with $\Delta x_{min}$ = 2 m, Figure 2.12 shows the number of cells and runtimes; in order to facilitate the comparison, data have been scaled on the basis of the 16-C-1 simulation results. The adoption of non-uniform BUQ grids (16-Q-1 and 8-Q-1) remarkably reduces the memory allocated and the runtimes, in comparison with the analogous simulations with Cartesian grids: the simulation with 8×8 cell blocks (8-Q-1) achieves the best reduction both in terms of number of cells (0.05) and runtime (0.10). Slightly higher (but substantially similar) values are obtained in the (16-Q-1) simulation.

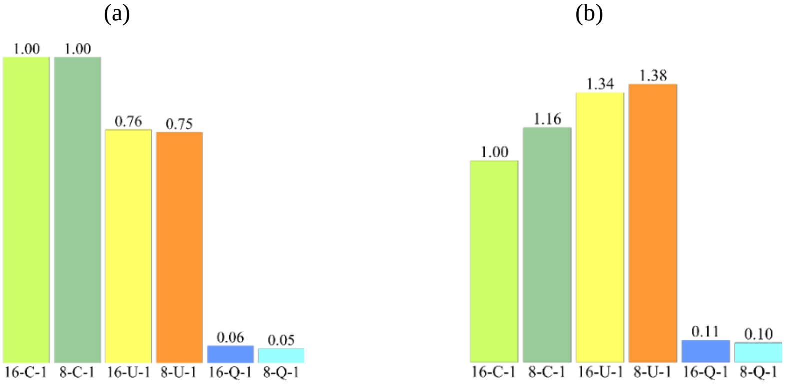


*Figure 2.12. Vortex test case with $\Delta x_{min}$ = 2 m: number of cells (a) and non-dimensional computational times (b) rescaled on 16-C-1 simulation.*

Besides the evaluation of the efficiency, the overhead produced by the data structure of the BUQ grid is investigated by comparing the runtimes of the BUQ grids with uniform resolution and the Cartesian ones, as for a given $\Delta x_{min}$ the two grids produce the same results (as confirmed by the $L_2$ error results in Table 2.1).

As it should be expected, with reference to simulations with $\Delta x_{min}$ = 2 m, the analysis of the runtimes highlights that simulation 8-U-1 is 19% slower (1.38/1.16) than 8-C-1. As explained in Section 2.2.3, this slight increase in the computational costs is due to the procedure that identifies the neighbors and exchanges information between cells located at the edge of the blocks, which are not close in memory as in the Cartesian grids. This behavior is further confirmed by the 16-U-1 mesh, which having a smaller ratio between the number of cells on the edges and the total cells of the block is more efficient than the

8-U-1 one. Similar results can be found by analyzing the data of a simulation performed with different resolution (Table 2.1).

In order to evaluate the scalability of the non-uniform BUQ grid implementation, the normalized runtime $T_N$ is defined as:

$$T_N = \frac{T}{n \cdot N} \tag{2.21}$$

where $T$ denotes the total computational time (s), $n$ the number of iterations in time and $N$ the total number of cells. The ratio has been calculated separately for 8×8 and 16×16 blocks and the results are plotted in Figure 2.13 with the total number of cells. In both cases, the scalability typical of GPUs process is achieved: the more cells are computed, the less time is required for each cell. For more than $0.2 \cdot 10^6$ cells, the normalized computational time remains almost constant, showing that the numerical scheme is able to absorb the overheads for communication and thread setup/switch.

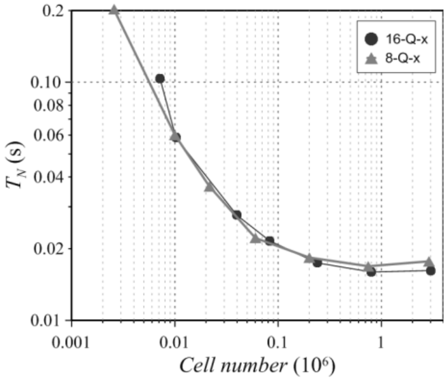
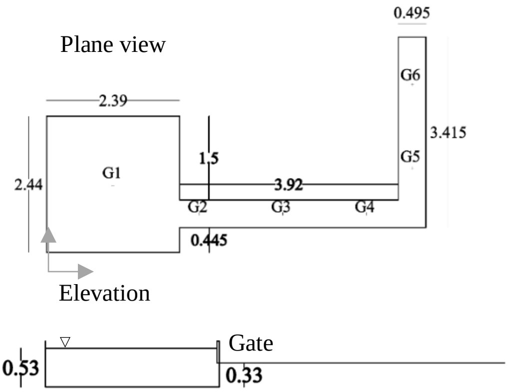


*Figure 2.13. Analysis of the code scalability using non-uniform BUQ grids.*

## 2.3.2. Laboratory test

In this section, the numerical model is tested on a laboratory case concerning a dam-break flow that was carried out by the CADAM (Concerted Action on Dam Break Modelling) group (Soares-Frazão et al., 1999; Soares-Frazão and Zech, 2002).

The laboratory facility consists in a reservoir connected to a rectangular channel, which is characterized by a sharp 90° bend and flat bottom, through a 0.33 m positive step (Figure 2.14). At the beginning of the experiment, the channel is dry and the reservoir presents a still water level of 0.53 m (0.2 m upon the downstream channel bed level); then, the instantaneous removal of the gate, which separates the reservoir from the downstream channel, triggers the flow.



| | *x* (m) | *y* (m) |
|---|---|---|
| *G1* | 1.19 | 1.20 |
| *G2* | 2.74 | 0.69 |
| *G3* | 4.24 | 0.69 |
| *G4* | 5.74 | 0.69 |
| *G5* | 6.56 | 1.51 |
| *G6* | 6.56 | 3.01 |

*Figure 2.14. CADAM laboratory facility schematization (dimension in m).*

In order to reproduce the advancing of the wetting front, the Manning coefficient was set equal to 0.012 s/m$^{1/3}$ value, which is an average between bottom and walls roughness (Soares-Frazão et al., 1999; Aureli et al., 2004). Moreover, a short, fictitious steep lengthening of the channel bottom was added at the outflow, in order to distance the necessarily approximated boundary condition (far-field); this allows a better reproduction of the cross waves up to the end of the physical channel and, particularly, the time series recorded at the G6 gauge.

Six simulations have been performed (Table 2.2) by varying:

- The type of the adopted mesh:
    - Cartesian with size $\Delta x_{min}$ = 0.5 cm (z-C in Table 2.2);
    - BUQ with uniform resolution $\Delta x_{min}$ = 0.5 cm in the whole domain (z-U in Table 2.2);
    - BUQ with $\Delta x_{min}$ = 0.5 cm corresponding to the resolution level 1 and varying the resolution inside the domain (z-Q in Table 2.2);
- The block size:
    - 8×8 cells (8-y in Table 2.2);
    - 16×16 cells (16-y in Table 2.2).

*Table 2.2. CADAM simulations: ID, minimum and maximum cell sizes ($\Delta x_{min}$, $\Delta x_{max}$), number of cells N, compression rate $C_R$, run times R and speed-up $S_U$.*

| **ID** | **$\Delta x_{min}$ (cm)** | **$\Delta x_{max}$ (cm)** | **N ($10^3$)** | **$C_R$ (-)** | **R (s)** | **$S_U$ (-)** |
|---|---|---|---|---|---|---|
| 16-C | 0.5 | 0.5 | 1115.48 | 1.00 | 318.09 | 1.00 |
| 8-C | 0.5 | 0.5 | 1115.48 | 1.00 | 367.56 | 1.00 |
| 16-U | 0.5 | 0.5 | 418.30 | 2.67 | 350.50 | 0.91 |
| 8-U | 0.5 | 0.5 | 400.32 | 2.79 | 366.63 | 1.00 |
| 16-Q | 0.5 | 4 | 189.95 | 5.87 | 167.70 | 1.90 |
| 8-Q | 0.5 | 4 | 169.34 | 6.59 | 159.48 | 2.30 |

In the BUQ grids with non-uniform resolution (16-Q and 8-Q), the highest resolution level ($\Delta x_1$ = 5 mm) was set only in the channel, while in the upstream reservoir the computed grid size gradually increases to 4 cm (Figure 2.15).

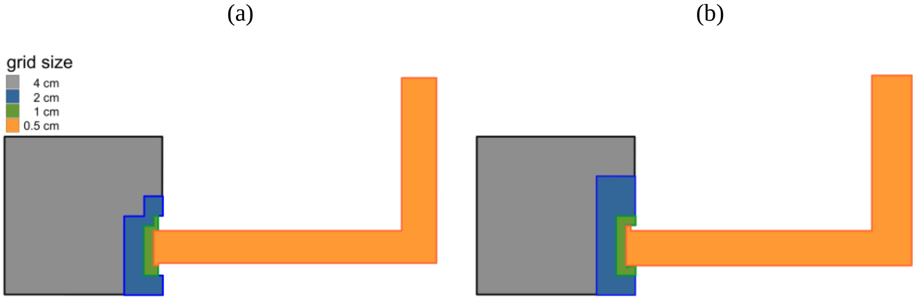


*Figure 2.15. Resulted non-uniform BUQ grid using 8×8 (a) and 16×16 (b) blocks.*

Figure 2.16 shows the water elevation 1 s (a) and 10 s (b) after the gate opening, respectively; in (b) the bore reflected by the sharp bend is travelling upstream toward the reservoir and cross-waves characterize the flow downstream the bend.

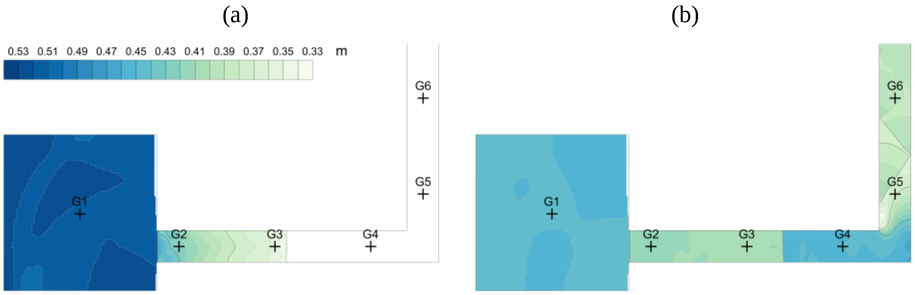


*Figure 2.16. Water surface elevation 1 s (a) and 10 s (b) after the gate opening.*

The simulations were run for 40 s of physical time and the computed time series at the six gauging points G1÷G6 along the channel were compared against the experimental ones. As illustrated in Figure 2.17, the phenomenon is correctly described with both non-uniform BUQ (8-Q) and Cartesian grids (8-C). The water elevation at G1 gauge shows the decrease of the initial water elevation in the reservoir (20 cm above the channel flat bottom), initially halved in 14 s and then gradually decreasing until reaching the lowest value of 6 cm at 40 s. At the G2÷G4 gauging points both the initial increase of the water level due to the arrival of the wetting front and the reflected front, caused by the arrival of the bore at the elbow (which determines an instantaneous raise of the water level), are correctly reproduced. Only the decrease of water levels at G2, before the backward arrival of the bore reflected by the sharp bend, is overestimated by the model. However, this behavior is similar to that obtained by many other authors (Soares-Frazão et al., 1999; Liang et al., 2004). In addition, the numerical results obtained downstream the bend well reproduce the experimental data with the exception of a slight underestimation of the peak of the water depth at gauge G5.

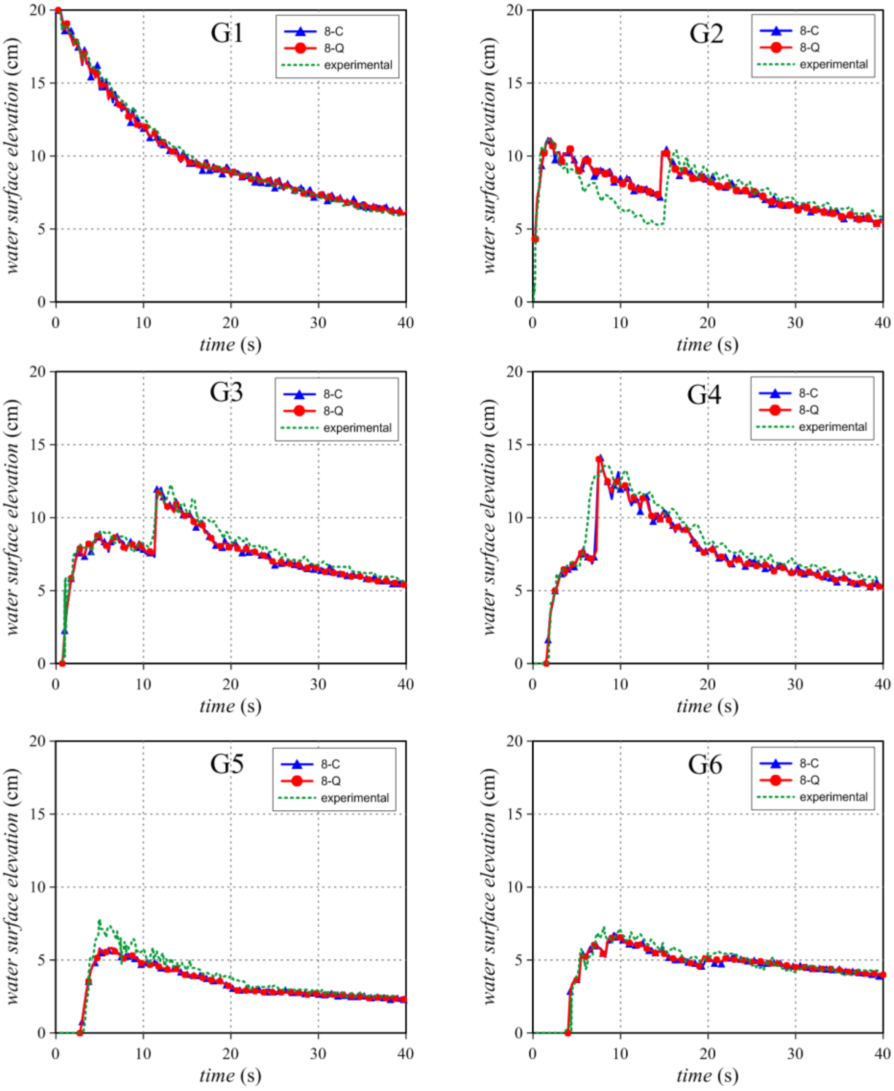


*Figure 2.17. Comparison between registered and simulated water levels at the six gauging points.*

The efficiency of the numerical scheme is investigated by comparing the computational time and the number of cells of the Cartesian and BUQ grids (Figure 2.18).

With reference to uniform BUQ grids, a remarkable reduction of the allocated memory is achieved due to the exclusion of grid blocks located outside the domain; the compression rate $C_R$ is equal to 2.67 and 2.79, adopting 16×16 (16-U) and 8×8 (8-U) cell blocks, respectively. Nevertheless, the computational times do not present an analogous decrease, as the ones with uniform BUQ grids (8-U and 16-U) are similar to the ones obtained with Cartesian grids (8-C and 16-C). This behavior is due the Block Deactivation optimization, which guarantees that blocks completely dry or located outside the domain, although allocated in memory, are not processed (Section 1.2.2).

On the other side, the use of non-uniform BUQ grids allows a further reduction of the number of cells used to discretize the domain, reaching the compression rate $C_R$ of 5.87 and 6.59 for the 16×16 (16-Q) and 8×8 (8-Q) cell blocks, respectively. The simulations with non-uniform grids are also the most efficient in terms of computational times, reaching the speed-up $S_U$ of 1.9 and 2.3 for the 16×16 (16-Q) and 8×8 (8-Q) cell blocks, respectively.

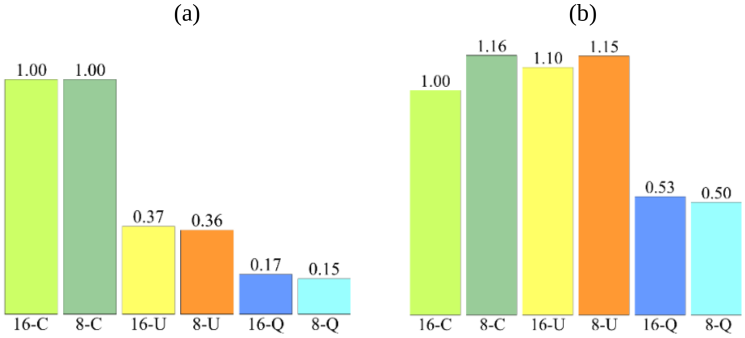


*Figure 2.18. CADAM test case: number of cells (a) and non-dimensional computational times (b) rescaled on 16-C simulation.*

## 2.3.3. Real test

In this section the simulation of the 13th October 2014 flood occurred on the Parma River (Northern Italy), between the homonymous town and the confluence in the Po River, is presented (Figure 2.19). This 38 km-long reach of the Parma River is characterized by a deep main channel, which meanders within two artificial earthen levees of remarkable height above the surrounding lands. In some places, flood plains are directly connected with the main channel; elsewhere one or more dikes must be overtopped before the inundation of the flood plain commences.

The whole bathymetry (main channel and floodplains) was set up starting from a 1 m resolution DTM obtained through a LiDAR survey carried out during the dry season, in drought condition.

As upstream boundary condition the discharge hydrograph, obtained through a stage-discharge relationship from the recorded water levels at Parma Ponte Verdi gauging station, was imposed. The downstream boundary condition was the reconstructed time-series of water levels at the confluence with the Po River.

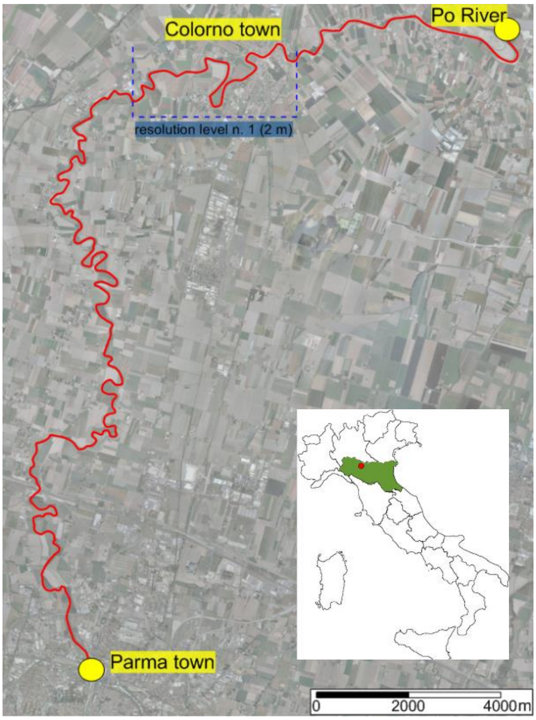


*Figure 2.19. Parma River reach from Parma to the confluence in the Po River: the locations of the upstream and downstream boundary conditions are indicated.*

Six simulations have been performed (Table 2.3) by varying:

- The type of the adopted mesh:
    - Cartesian with size $\Delta x_{min} = 2$ m (z-C in Table 2.3);
    - BUQ with uniform resolution $\Delta x_{min} = 2$ m in the whole domain (z-U in Table 2.3);
    - BUQ with $\Delta x_{min} = 2$ m corresponding to the resolution level 1 and varying the resolution inside the domain (z-Q in Table 2.3);
- The block size:
    - 8×8 cells (8-y in Table 2.3);
    - 16×16 cells (16-y in Table 2.3).

*Table 2.3. October 2014 Parma flooding event simulations: ID, minimum and maximum cell sizes ($\Delta x_{min}$, $\Delta x_{max}$), number of cells N, compression rate $C_R$, runtimes R, speed-up $S_U$ and RMSE.*

| **ID** | **$\Delta x_{min}$ (m)** | **$\Delta x_{max}$ (m)** | **N ($10^6$)** | **$C_R$ (-)** | **R (hours)** | **$S_U$ (-)** | ***RMSE* (m)** |
|---|---|---|---|---|---|---|---|
| 16-C | 2 | 2 | 39.81 | 1.00 | 8.73 | 1.00 | 0.319 |
| 8-C | 2 | 2 | 39.81 | 1.00 | 11.52 | 1.00 | 0.326 |
| 16-U | 2 | 2 | 1.75 | 22.76 | 10.28 | 0.85 | 0.316 |
| 8-U | 2 | 2 | 1.61 | 24.67 | 10.71 | 1.08 | 0.323 |
| 16-Q | 2 | 8 | 0.64 | 62.09 | 3.64 | 2.40 | 0.306 |
| 8-Q | 2 | 16 | 0.36 | 109.41 | 2.60 | 4.43 | 0.335 |

Adopting a non-uniform BUQ grid, the resolution level in the main channel was forced to 2 m close to Colorno (Figure 2.19) and 4 m elsewhere. As an example, Figure 2.20-a depicts the seeding points used to set the resolution level to 4 m in the main channel and the resulted non-uniform BUQ grid generated with 8×8 (Figure 2.20-b) and 16×16 (Figure 2.20-c) cell blocks, respectively. In this particular meandering reach of the Parma River, the adoption of 16×16 cell blocks constrains the same resolution level (4 m) almost everywhere, whereas 8×8 cell blocks allow varying the resolution outside the main channel, with further reduction of the total number of cells.

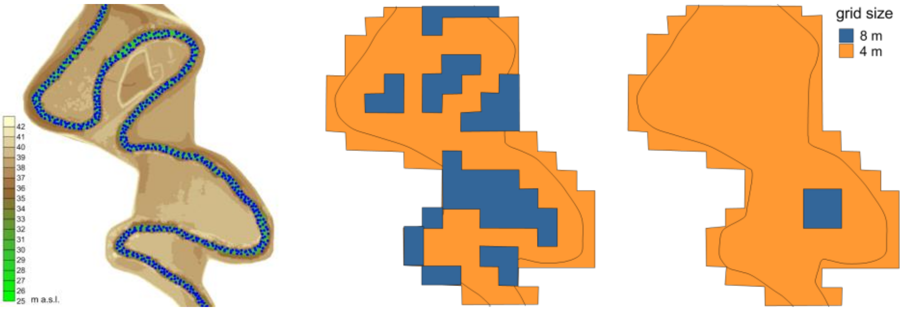


*Figure 2.20. Detail of the non-uniform BUQ grid in a typical meander of the river: the seed points used to set the resolution level in the main channel (a), the resulted grids adopting 8×8 (b) and 16×16 (c) cell blocks. In background the bathymetry.*

The flooding event was simulated from 12 A.M. of the 13th October 2014, till 2.30 P.M. of the following day, for a total duration of 26.5 hours.

Because of the particular river morphology previously highlighted, the flow field strongly varies at low (Figures 2.21-a and 2.22-a), and high discharge values (Figures 2.21-b and 2.22-b). At the highest levels most of the meanders are cut by the flow, flood plains are inundated and contribute significantly to convey the total discharge; see for example the C and D areas in Figures 2.21-b and 2.22-b. Under these conditions, it is quite clear that at least a 2D model is necessary to simulate the flood propagation.

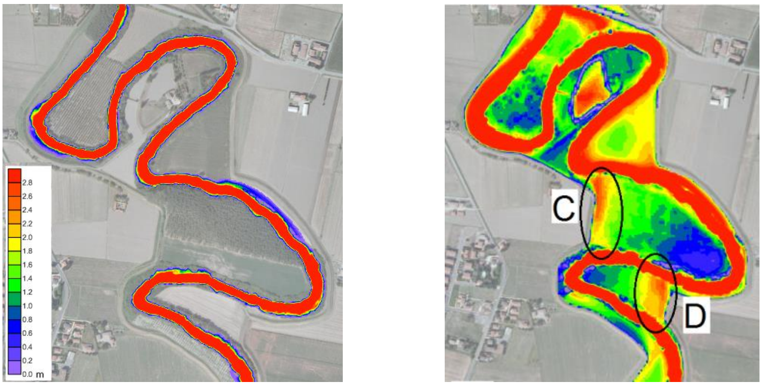

(a) (b)

*Figure 2.21. 8-Q simulation: water depth at low (a) and high (b) discharges in some typical meanders.*

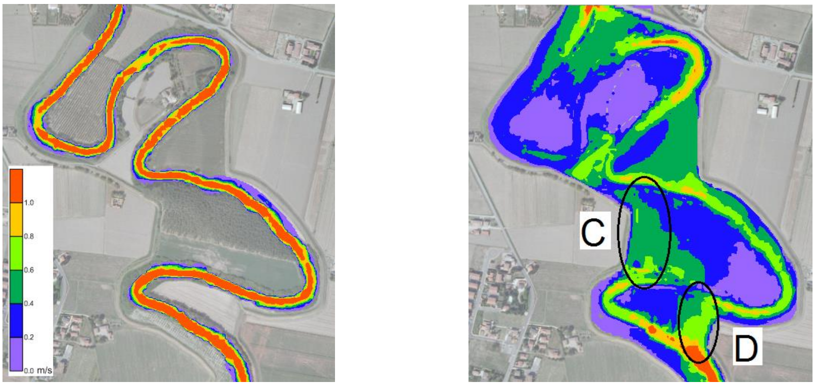

(a) (b)

*Figure 2.22. 8-Q simulation: velocity magnitude at low (a) and high (b) discharges in some typical meanders.*

As during and soon after the flood, the Interregional Agency for the Po River (AIPo) collected the maximum water levels reached in $N$=110 locations along the river main levees, the quantitative evaluation of the model results was assessed through the root mean square error, *RMSE*:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}(WSE_i^{obs} - WSE_i^{mod})^2}{N}} \tag{2.22}$$

where $WSE_i^{obs}$ and $WSE_i^{mod}$ denote the observed and modeled maximum water surface elevations, respectively.

As reported in Table 2.3, the differences of *RMSE* values obtained in the six simulations are almost negligible, confirming that the use of non-uniform BUQ grids does not determine an appreciable loss of accuracy.

Focusing on the model efficiency, this real test case emphasizes the advantage of the BUQ grids in allocating domains, which not necessary have to be rectangular. The considered river course resembles the shape of an overturned "L" (Figure 2.19); whereas Cartesian grids allocate almost $40 \cdot 10^6$ cells, uniform BUQ grids achieve compression rate $C_R$ of 23 or 25, with 16×16 or 8×8 cell blocks, respectively (Table 2.3). However, most of the cells allocated in the Cartesian grids are outside the main river levees and thus dry (assuming neither levee overtopping nor breaches); as in the CADAM test case, the Block Deactivation Optimization procedure excludes these dry blocks from the computation, and this justifies the similar runtimes between Cartesian and uniform BUQ grids (Figure 2.23).

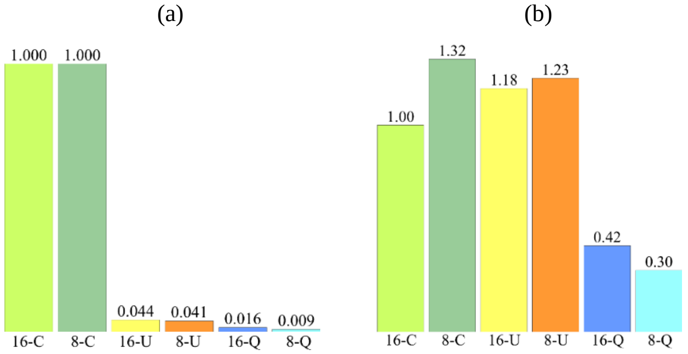


*Figure 2.23. October 2014 Parma flooding test case: number of cells (a) and non-dimensional computational times (b) rescaled on 16-C simulation.*

On the other side, the adoption of non-uniform BUQ grids allows significantly reducing the number of computed cells: the 8-Q grid processes less than the 1% (0.009) of the cells computed in the Cartesian grid 8-C and achieves a compression rate equal to 109 (Figure 2.23-a).

The analysis of the computational time (Figure 2.23-b) shows that 8.73 hours or 11.52 hours are required for the simulation with Cartesian grid and 16×16 cell blocks or 8×8 cell blocks, respectively. The increase of the total computational time in simulation 8-C is due essentially to the code routine that identifies wet blocks. This time, as expected, roughly increases by a factor of 4 in the 8×8 Cartesian case, introducing an overhead that slows down the computation by a 1.3 factor.

For the same reason the 8-U simulation performs slightly better than the 8-C one.

As shown in Table 2.3, the choice of non-uniform BUQ grids allows simulating the flood event in about 3.64 (16×16 cell blocks) or 2.60 (8×8 cell blocks) hours, with speed-ups equal to 2.4 or 4.4, respectively, and a ratio of physical to computational time of about 7.3 or 10.2.

## 2.3.4. Hypothetical flood scenario

In order to assess the model performance in presence of flooding events over large areas, the numerical results concerning the flooding scenario induced by a hypothetical levee breaching in the Secchia River (Northern Italy) are presented in this section. The domain of approximately 840 km$^2$ includes the 83 km-long reach of the Secchia River between the town of Modena and the confluence in the Po River, and the potentially inundated region located on its right bank, which is enclosed by the Panaro River (to the east) and by a railway embankment (south), as shown in Figure 2.24-a. The bathymetry, derived from a 1 m-resolution DTM, is also reported in Figure 2.24-a.

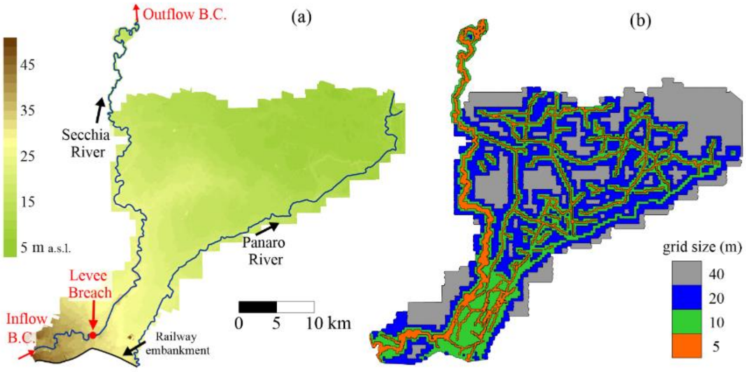


*Figure 2.24. Levee break on the Secchia River: bathymetry (a) and grid size distribution (b).*

A non-uniform BUQ grid with 16×16 cell blocks was adopted for the domain definition (Figure 2.24-b); the resolution level was forced to $\Delta x_{\mathrm{min}} = 5$ m along the Secchia River (levees included) and along the main road embankments and channel dikes in the flooding plain, while it was allowed gradually decreasing to $\Delta x_{\mathrm{max}} = 40$ m elsewhere. The domain was discretized by means of $7.3 \cdot 10^6$ cells, whereas the same area would require $99 \cdot 10^6$ and $33 \cdot 10^6$ cells if a Cartesian grid or a BUQ grid with uniform resolution (equal to $\Delta x_{min}$) were used, respectively. It is worth noting that this test case cannot be simulated with uniform resolution (both Cartesian grid and BUQ grid) because the memory requirements exceed the capability of the K40 Tesla® GPU here adopted for the numerical computations.

An inflow discharge hydrograph (50 years-return period) was imposed upstream, while a rating curve was set as downstream boundary condition (very far from the breach location). The levee breach was generated on the right levee at the location indicated in Figure 2.24-a after 22 h of simulation, when the peak value of discharge was observed at that cross-section. The simulation was prolonged for 58 h after the breach opening, so that the entire event lasted 80 h. The Manning roughness coefficient was assumed equal to 0.05 s/m$^{1/3}$ everywhere. The first-order accurate version of the scheme was used in this case.

Figure 2.25 shows a detail of the flow field around the breach location 2 h after the breach opening. In particular, the water depth contour map is represented in Figure 2.25-a, while Figure 2.25-b reports both the velocity vector and the velocity magnitude contour map. Different resolution zones are also sketched in Figure 2.25-a. It can be noticed that part of the water coming out from the breach moves eastward (and then northward) "canalized" between the levee and a road embankment, while the rest propagates southward. The subsequent flooding evolution in time is outlined in Figure 2.26, which reports a selection of water surface elevation maps. An area of approximately 62 km$^2$ is flooded at the end of the simulation. The total run time is equal to 6.32 h, hence the ratio of physical to computational time is approximately 12.6.

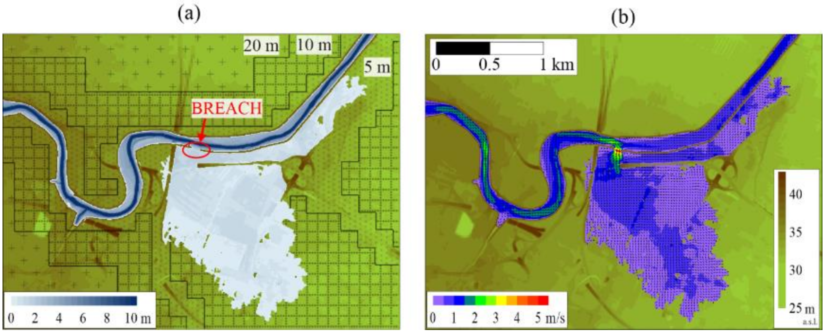


*Figure 2.25. Flow field around the levee breach: water depth (a) and velocity (b) contour maps.*

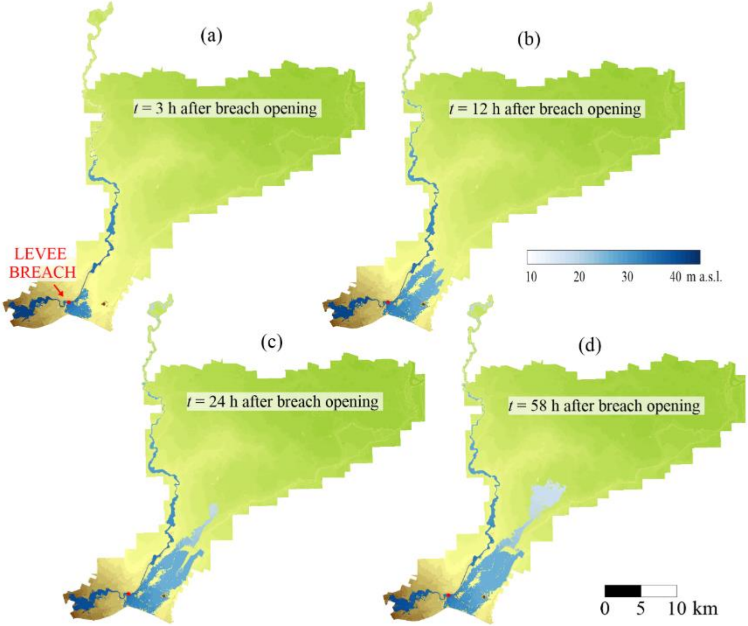


*Figure 2.26. Water surface elevation maps of the flooding event at selected times: 3 h (a), 12 h (b), 24 h (c) and 58 h (d) after the breach opening.*

## 2.4. Concluding remarks

In this chapter a novel Block Uniform Quadtree (BUQ) grid has been implemented in a CUDA code solving the 2D-SWE.

In the framework of large-scale simulations, the BUQ grids overcome two of the main limitations of the GPU codes based on Cartesian grids, by achieving the following goals. Firstly, the adoption of multi-resolution has allowed reproducing small-scale effects while simulating large domains, up to areas of 850 km$^2$. Moreover, the implemented data structure exploits the computational capability of GPUs with minimum overheads. Furthermore, this reduces the computational burden, with a substantial decrease of the number of stored cells and of the runtimes.

As outlined by a theoretical and a laboratory test case, the BUQ grids allow reaching significant speed-ups (2-10), if compared to the Cartesian grids. Moreover, the simulation of a real flood event on a 38 km long river reach, reproduced using a maximum resolution of 2 m, achieved a ratio of physical to computational time of about 10, without appreciable losses of accuracy with respect to a uniformly fine Cartesian grid. Furthermore, in the simulation of a hypothetical flooding over a domain with extension equal to 840 km$^2$ the ratio of physical to computational time equal to 12 was obtained, confirming the capability of the model to provide quite fast flood simulations. Finally, this last test could have been performed only by means of the non-uniform BUQ grids, as the adoption of Cartesian ones was not enable by the memory requirement of the Tesla K40 GPU.

# Chapter 3

## The Shallow Water Equations with porosity (Part I): a well-balanced scheme

### 3.1. Introduction

In the framework of Shallow Water flooding models, different approaches consider the building effects in urban areas avoiding the use of high-resolution meshes, which lead to high computational times (Schubert and Sanders, 2012; Sanders, 2017; Costanzo and Macchione, 2006; Costabile et al., 2016; Viero and Valipour, 2017). Among the others, the most promising method is a sub-grid model based on a porosity coefficient, which considers the effect of buildings without describing in detail their geometry. Therefore, the adoption of the SWEs with porosity is particularly suitable for reducing the run times of large-scale simulations, where urban environments are present, and when the goal is to consider the overall presence of constructions and not to obtain a detailed information of the flow dynamics near the buildings.

First attempts to develop SWE models with porosity were based on an isotropic approach, which defines porosity (storage/isotropic/global porosity) as the fraction of plan view area available to flow. A formulation for partially wet/dry areas over irregular domains was introduced by Defina et al. (1994), later improved (Defina, 2000; Viero et al., 2014; Casulli, 2015) and found promising applications in urban areas (Yu and Lane, 2006; McMillan and Brasington, 2007). In particular, Guinot and Soares-Frazão (2006) derived a differential formulation on a finite volume scheme with a new technique for describing head losses, which was further investigated by Soares-Frazão et al. (2008). Velickovic et al. (2017) theoretically and experimentally investigated the resistance offered by buildings, focusing on drag coefficient calibrations. Finaud-Guyot et al. (2010) proposed a new approximate Riemann solver for porosity model (PorAS), while Cea and Vázquez-Cendón (2010) derived a modified version of the SWEs, in which the porosity is included adding to the classical SWEs two source terms, related to the effect of porosity variation in space and to non-resolved obstructions. Recently, Benkhaldoun et al. (2016) proposed a predictor-corrector procedure for solving the SWEs with porosity on unstructured grids.

Shifting the focus from isotropic to anisotropic models, Sanders et al. (2008) derived the integral formulation of the SWEs with porosity and introduced the water conveyance porosity parameter in order to capture the preferential flow directions due to the buildings alignment and shapes; this approach was recently further extended by introducing a Dual Integral Porosity model (Guinot et al., 2017). Guinot (2017a) carried out a consistency and characteristic analysis of the Integral Porosity (IP) and Dual Integral Porosity (DIP) approaches and demonstrated that the mesh design is a crucial aspect for those models. Recently, the same author (Guinot, 2017b) presented a comparison among the isotropic and the IP and DIP methods based on several networks and building configurations and initial value problems. For an experimental anisotropic test case, Kim et al. (2015) investigated the porosity models errors and showed that they are lower for anisotropic models than for the isotropic ones.

Focusing on anisotropic models adopting Cartesian grids, Bruwier et al. (2017) introduced different edge porosity parameters and presented a merging technique related to the time step definition, for increasing the computational efficiency.

In the framework of a diffusive model, Chen et al. (2012) introduced the building coverage ratio (area occupied by buildings) and the conveyance reduction factors (cross sections occupied by buildings on the cell interfaces) so as to consider the building effects on large 2D simulations with coarse grids.

Merging the isotropic and anisotropic models, Guinot (2012) introduced a multiple porosity approach, which divides the studied domain into five different areas, each governed by a different set of equations. Özgen et al. (2016a,b) derived and tested a new integral formulation, that is focused on depth-dependent anisotropic porosity, in which the porosity parameters depend on the water elevation inside a given cell.

In this thesis, the most established isotropic porosity formulation of SWEs (Guinot and Soares-Frazão, 2006) has been further manipulated, following the idea of Cea and Vázquez-Cendón (2010), as to obtain a well-balanced scheme, which inherently guarantees the *C-property*. In order to test the implemented formulation, 1D analytical solutions are derived and 2D reference solutions are defined.

This Chapter is derived in part from Ferrari et al. (2017).

## 3.2. Governing equations

The 2D Shallow Water Equations with porosity are derived, analogously to the classical SWE formulation, by applying the physical principles of mass and momentum conservation, which govern the motion of the fluid, to a generic control volume with sizes $\Delta x$ and $\Delta y$ along the $x$ and $y$-direction, respectively, and whose left corner has coordinates $x_0$ and $y_0$ (Figure 3.1-b).

In addition to the De Saint Venant hypothesis for SWEs (Section 1.2.1), the derivation of the SWEs with porosity is illustrated assuming that the conveyance (fraction of the cross section available to the flow) and the storage (fraction of the plan view area available to the flow) porosity coincide. The derivation of the 2D-SWEs with porosity presented in the next section refers to the one proposed by Guinot and Soares-Frazão (2006).

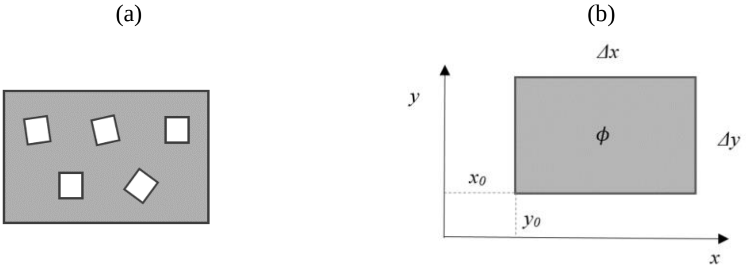


*Figure 3.1. Example of an urban scheme (a) and corresponding control volume with isotropic porosity $\phi$ (b).*

### 3.2.1. The continuity equation

The mass conservation principle states that the variation of mass inside the control volume is due only to the passage of fluid through the interfaces (west, east, north, south) according to the relation:

$$\frac{\partial V}{\partial t} = F_{west} - F_{east} + F_{south} - F_{north} \tag{3.1}$$

Recalling that the porosity $\phi$ is the fraction of plan view area available to flow and in a physical visualization the ratio between the volume available to flow and the total volume, the volume $V$ of water inside the control volume reads:

$$V = \int_{y_0}^{y_0+\Delta y} \int_{x_0}^{x_0+\Delta x} (\phi h)(x,y) dx\, dy \tag{3.2}$$

where $h$ is the water depth.

The volume fluxes $F_{west}, F_{east}, F_{south}, F_{north}$ that cross the west, east, north and south sections, respectively are calculated as:

$$F_{west} = \int_{y_0}^{y_0+\Delta y} (\phi u h)(x_0, y) dy \tag{3.3}$$

$$F_{east} = \int_{y_0}^{y_0+\Delta y} (\phi uh)(x_0 + \Delta x, y)dy \tag{3.4}$$

$$F_{south} = \int_{x_0}^{x_0+\Delta x} (\phi vh)(x, y_0)dx \tag{3.5}$$

$$F_{north} = \int_{x_0}^{x_0+\Delta x} (\phi vh)(x, y_0 + \Delta y)dx \tag{3.6}$$

where *u* and *v* are the velocity components in the *x* and *y*-direction, respectively.

Substituting equations (3.2)-(3.6) into (3.1), the mass balance becomes:

$$\begin{aligned}\frac{\partial}{\partial t}\int_{y_0}^{y_0+\Delta y}\int_{x_0}^{x_0+\Delta x} (\phi h)(x, y)dx\,dy - \int_{y_0}^{y_0+\Delta y} (\phi uh)(x_0, y)dy \\ + \int_{y_0}^{y_0+\Delta y} (\phi uh)(x_0 + \Delta x, y)dy - \int_{x_0}^{x_0+\Delta x} (\phi vh)(x, y_0)dx \\ + \int_{x_0}^{x_0+\Delta x} (\phi vh)(x, y_0 + \Delta y)dx = 0\end{aligned} \tag{3.7}$$

After simple manipulations and assuming an infinitesimal control volume, so that $\Delta x$ $\Delta y$ tend to zero, equation (3.7) reduces to:

$$\frac{\partial}{\partial t}(\phi h)\Delta x\Delta y + \frac{\partial}{\partial x}(\phi uh)\Delta x\Delta y + \frac{\partial}{\partial y}(\phi vh)\Delta x\Delta y = 0 \tag{3.8}$$

Finally, dividing by the area of the control volume the continuity equation results:

$$\frac{\partial}{\partial t}(\phi h) + \frac{\partial}{\partial x}(\phi uh) + \frac{\partial}{\partial y}(\phi vh) = 0 \tag{3.9}$$

### 3.2.2. The momentum equation

The momentum balance in the *x*-axis states that the variation of mass inside the control volume is due to the passage of fluid through the cross sections and to the external forces acting on the edges, according to the relation:

$$\frac{\partial M_x}{\partial t} = F_{west} - F_{east} + F_{south} - F_{north} + P_{west} - P_{east} + R_{p_x} + R_{b_x} + R_{f_x} + R_{w_x} \tag{3.10}$$

The total momentum $M$ is calculated as:

$$M_x = \rho \int_{y_0}^{y_0+\Delta y} \int_{x_0}^{x_0+\Delta x} (\phi u h)(x, y) dx\, dy \tag{3.11}$$

where $\rho$ is the density of the fluid.

The momentum fluxes $F_{west}, F_{east}, F_{south}, F_{north}$ that cross the west, east, north and south sections, respectively are calculated as:

$$F_{west} = \rho \int_{y_0}^{y_0+\Delta y} (\phi u^2 h)(x_0, y) dy \tag{3.12}$$

$$F_{east} = \rho \int_{y_0}^{y_0+\Delta y} (\phi u^2 h)(x_0 + \Delta x, y) dy \tag{3.13}$$

$$F_{south} = \rho \int_{x_0}^{x_0+\Delta x} (\phi u v h)(x, y_0) dx \tag{3.14}$$

$$F_{north} = \rho \int_{x_0}^{x_0+\Delta x} (\phi u v h)(x, y_0 + \Delta y) dx \tag{3.15}$$

The external forces $P_{west}$, $P_{east}$, $R_{p_x}, R_{b_x}, R_{f_x}, R_{w_x}$ that act on the control volume are defined as following.

Under the hypothesis of hydrostatic pressure distribution, the pressure forces exerted on the west and east sections are:

$$P_{west} = \frac{\rho g}{2} \int_{y_0}^{y_0+\Delta y} (\phi h^2)(x_0, y) dy \tag{3.16}$$

$$P_{east} = \frac{\rho g}{2} \int_{y_0}^{y_0+\Delta y} (\phi h^2)(x_0 + \Delta x, y) dy \tag{3.17}$$

where $g$ is the gravitational acceleration.

The reaction due to porosity variation in *x*-direction $R_{p_x}$ represents the force exerted by the walls on the fluid:

$$R_{p_x} = \frac{\rho g}{2} \int_{y_0}^{y_0+\Delta y} \int_{x_0}^{x_0+\Delta x} h^2 \frac{\partial \phi}{\partial x} dxdy \tag{3.18}$$

The reaction $R_{b_x}$that the bed exerts on the fluid per unit surface is:

$$R_{b_x} = -\rho g \int_{y_0}^{y_0+\Delta y} \int_{x_0}^{x_0+\Delta x} \phi h(x,y) \frac{\partial z}{\partial x} dxdy \tag{3.19}$$

where *z* represents the bottom elevation.

The friction resistance $R_{f_x}$expressed according to Strickler formula is:

$$R_{f_x} = -\rho g h \int_{y_0}^{y_0+\Delta y} \int_{x_0}^{x_0+\Delta x} \left[ \frac{\sqrt{u^2+v^2}}{k^2 h^{4/3}} \phi u \right] (x,y)\, dxdy \tag{3.20}$$

where *k* denotes the Strickler coefficient.

Finally, the term $R_{w_x}$ accounting for the wave reflections against unresolved obstacles reads:

$$R_{w_x} = -\rho g h s_x \int_{y_0}^{y_0+\Delta y} \int_{x_0}^{x_0+\Delta x} \left[ \phi u \sqrt{u^2+v^2} \right] (x,y)\, dxdy \tag{3.21}$$

where $s_x$ is the head loss coefficient in *x*-direction, which may be expressed according to different formulations (Cea and Vázquez-Cendón, 2010; Guinot and Soares-Frazão, 2006; Soares-Frazão et al., 2008).

Substituting equations (3.11)-(3.21) into (3.10), and after simple manipulations the momentum balance becomes:

$$\begin{aligned}
&\rho\int_{y_0}^{y_0+\Delta y}\int_{x_0}^{x_0+\Delta x}\frac{\partial}{\partial t}(\phi u h)(x,y)dx\,dy \\
&\qquad= \rho\int_{y_0}^{y_0+\Delta y}[(\phi u^2 h)(x_0,y) \\
&\qquad- (\phi u^2 h)(x_0+\Delta x,y)]dy \\
&\qquad- \rho\int_{x_0}^{x_0+\Delta x}[(\phi u v h)(x,y_0+\Delta y) \\
&\qquad- (\phi u v h)(x,y_0)]dx \\
&\qquad+ \frac{\rho g}{2}\int_{y_0}^{y_0+\Delta y}[(\phi h^2)(x_0,y)-(\phi h^2)(x_0+\Delta x,y)]dy \\
&\qquad+ \frac{\rho g}{2}\int_{y_0}^{y_0+\Delta y}\int_{x_0}^{x_0+\Delta x}h^2\frac{\partial\phi}{\partial x}dxdy \\
&\qquad- \rho g\int_{y_0}^{y_0+\Delta y}\int_{x_0}^{x_0+\Delta x}\phi h(x,y)\frac{\partial z}{\partial x}dxdy \\
&\qquad- \rho g h\int_{y_0}^{y_0+\Delta y}\int_{x_0}^{x_0+\Delta x}\left[\frac{\sqrt{u^2+v^2}}{k^2 h^{4/3}}\phi u\right](x,y)\,dxdy \\
&\qquad- \rho g h s_x\int_{y_0}^{y_0+\Delta y}\int_{x_0}^{x_0+\Delta x}\left[\phi u\sqrt{u^2+v^2}\right](x,y)\,dxdy
\end{aligned} \tag{3.22}$$

Assuming the flow incompressible and with constant flow density, and considering an infinitesimal control volume, so that $\Delta x$ $\Delta y$ are close to zero, the equation reduces to:

$$\begin{aligned}
\frac{\partial}{\partial t}(\phi u h)\Delta x\Delta y &= -\frac{\partial}{\partial x}(\phi u^2 h)\Delta x\Delta y - \frac{\partial}{\partial y}(\phi u v h)\Delta x\Delta y \\
&\quad- \frac{g}{2}\frac{\partial}{\partial x}(\phi h^2)\Delta x\Delta y + \frac{g}{2}h^2\frac{\partial\phi}{\partial x}\Delta x\Delta y - g\phi h\frac{\partial z}{\partial x}\Delta x\Delta y \\
&\quad- gh\frac{\sqrt{u^2+v^2}}{k^2 h^{4/3}}\phi u\Delta x\Delta y - g\phi h s_x u\sqrt{u^2+v^2}\Delta x\Delta y
\end{aligned} \tag{3.23}$$

Finally dividing by the area of the control volume the momentum equation in *x*-direction results:

$$\begin{aligned}\frac{\partial}{\partial t}(\phi uh)+\frac{\partial}{\partial x}(\phi u^2h)+\frac{\partial}{\partial y}(\phi uvh)+\frac{g}{2}\frac{\partial}{\partial x}(\phi h^2)\\ =\frac{g}{2}h^2\frac{\partial\phi}{\partial x}-g\phi h\frac{\partial z}{\partial x}-g\phi hu\frac{\sqrt{u^2+v^2}}{k^2h^{4/3}}\\ -g\phi hs_xu\sqrt{u^2+v^2}\end{aligned} \tag{3.24}$$

Similarly, the momentum equation in $y$-direction is defined as:

$$\begin{aligned}\frac{\partial}{\partial t}(\phi vh)+\frac{\partial}{\partial x}(\phi uvh)+\frac{\partial}{\partial y}(\phi v^2h)+\frac{g}{2}\frac{\partial}{\partial y}(\phi h^2)\\ =\frac{g}{2}h^2\frac{\partial\phi}{\partial y}-g\phi h\frac{\partial z}{\partial y}-g\phi hv\frac{\sqrt{u^2+v^2}}{k^2h^{4/3}}\\ -g\phi hs_yv\sqrt{u^2+v^2}\end{aligned} \tag{3.25}$$

Merging the results of the derived mass and momentum balance equations, the 2D Shallow Water Equations with isotropic porosity written in conservation form are (Guinot and Soares-Frazão, 2006):

$$\begin{cases}\frac{\partial}{\partial t}(\phi h)+\frac{\partial}{\partial x}(\phi uh)+\frac{\partial}{\partial y}(\phi vh)=0\\ \frac{\partial}{\partial t}(\phi uh)+\frac{\partial}{\partial x}\left(\phi u^2h+\frac{1}{2}g\phi h^2\right)+\frac{\partial}{\partial y}(\phi uvh)=\\ \qquad\frac{1}{2}gh^2\frac{\partial\phi}{\partial x}-g\phi h\frac{\partial z}{\partial x}-g\phi h\frac{u\sqrt{u^2+v^2}}{k^2h^{4/3}}-g\phi hs_xu\sqrt{u^2+v^2}\\ \frac{\partial}{\partial t}(\phi vh)+\frac{\partial}{\partial x}(\phi uvh)+\frac{\partial}{\partial y}\left(\phi v^2h+\frac{1}{2}g\phi h^2\right)=\\ \qquad\frac{1}{2}gh^2\frac{\partial\phi}{\partial y}-g\phi h\frac{\partial z}{\partial y}-g\phi h\frac{v\sqrt{u^2+v^2}}{k^2h^{4/3}}-g\phi hs_yv\sqrt{u^2+v^2}\end{cases} \tag{3.26}$$

If compared to the classical 2D SWEs, the system (3.26) is characterized by the presence of the porosity parameter $\phi$ in the conserved variables and by two additional source terms in the momentum equations (first and fourth terms in the RHS of second and third Equations of (3.26)).

## 3.3. A well-balanced scheme

Following the idea of Cea and Vázquez-Cendón (2010), the system of 2D-SWEs with porosity (3.26) can be further algebraic modified.

After the expansion of the partial derivatives, the continuity equation results:

$$\phi\frac{\partial h}{\partial t}+h\frac{\partial \phi}{\partial t}+\phi\frac{\partial}{\partial x}(uh)+(uh)\frac{\partial \phi}{\partial x}+\phi\frac{\partial}{\partial y}(vh)+(vh)\frac{\partial \phi}{\partial y}=0 \tag{3.27}$$

Then, assuming the porosity constant in time ($\frac{\partial \phi}{\partial t}=0$) and dividing each term in (3.27) by the porosity parameter, Equation (3.27) becomes:

$$\frac{\partial h}{\partial t}+\frac{\partial}{\partial x}(uh)+\frac{\partial}{\partial y}(vh)=-\frac{1}{\phi}\left((uh)\frac{\partial \phi}{\partial x}+(vh)\frac{\partial \phi}{\partial y}\right) \tag{3.28}$$

The formulation in Equation (3.28) presents the classical 2D-SWE on the left hand-side and includes the porosity only by means of two additional source terms on the right hand-side.

With the aim of preserving a well-balanced scheme also in presence of the 2D-SWEs with porosity, in the present chapter further manipulations are performed in order to still adopt the formulation of Liang and Borthwick (2009) for the classical SWEs, illustrated in Section 1.2.1. Recalling the relation among water depth, bottom and water surface elevation $h=\eta-z$, Equation (3.28) results:

$$\frac{\partial \eta}{\partial t}-\frac{\partial z}{\partial t}+\frac{\partial}{\partial x}(uh)+\frac{\partial}{\partial y}(vh)=-\frac{1}{\phi}\left((uh)\frac{\partial \phi}{\partial x}+(vh)\frac{\partial \phi}{\partial y}\right) \tag{3.29}$$

Finally, assuming a fixed bottom ($z$ = constant), the manipulated continuity equation states:

$$\frac{\partial \eta}{\partial t}+\frac{\partial}{\partial x}(uh)+\frac{\partial}{\partial y}(vh)=-\frac{1}{\phi}\left((uh)\frac{\partial \phi}{\partial x}+(vh)\frac{\partial \phi}{\partial y}\right) \tag{3.30}$$

Focusing on the momentum equation, the expansion of the partial derivatives leads to the following relation:

$$\begin{aligned}\phi\frac{\partial}{\partial t}(uh)+(uh)\frac{\partial\phi}{\partial t}+\phi\frac{\partial}{\partial x}(u^2h)+(u^2h)\frac{\partial\phi}{\partial x}+\frac{1}{2}g\phi\frac{\partial h^2}{\partial x}+\frac{1}{2}gh^2\frac{\partial\phi}{\partial x}&\\ +\phi\frac{\partial}{\partial y}(uvh)+(uvh)\frac{\partial\phi}{\partial y}&\\ =\frac{1}{2}gh^2\frac{\partial\phi}{\partial x}-g\phi h\frac{\partial z}{\partial x}-g\phi h\frac{u\sqrt{u^2+v^2}}{k^2h^{4/3}}&\\ -g\phi hs_xu\sqrt{u^2+v^2}&\end{aligned}\tag{3.31}$$

Then, assuming the porosity constant in time and dividing each term in (3.31) by the porosity parameter ($\phi\neq 0$), Equation (3.31) becomes:

$$\begin{aligned}\frac{\partial}{\partial t}(uh)+\frac{\partial}{\partial x}\left(u^2h+\frac{1}{2}gh^2\right)+\frac{\partial}{\partial y}(uvh)&\\ =-\frac{1}{\phi}\left[(u^2h)\frac{\partial\phi}{\partial x}+(uvh)\frac{\partial\phi}{\partial y}\right]-gh\frac{\partial z}{\partial x}-gh\frac{u\sqrt{u^2+v^2}}{k^2h^{4/3}}&\\ -ghs_xu\sqrt{u^2+v^2}&\end{aligned}\tag{3.32}$$

By substituting $h^2=(\eta-z)^2$ in the second term, after simple manipulations Equation (3.32) reads:

$$\begin{aligned}\frac{\partial}{\partial t}(uh)+\frac{\partial}{\partial x}\left(u^2h+\frac{1}{2}g(\eta^2-2\eta z)\right)+\frac{\partial}{\partial y}(uvh)&\\ =-\frac{1}{\phi}\left[(u^2h)\frac{\partial\phi}{\partial x}+(uvh)\frac{\partial\phi}{\partial y}\right]-gz\frac{\partial z}{\partial x}-gh\frac{\partial z}{\partial x}&\\ -gh\frac{u\sqrt{u^2+v^2}}{k^2h^{4/3}}-ghs_xu\sqrt{u^2+v^2}&\end{aligned}\tag{3.33}$$

Finally, the momentum equation in *x*-direction states:

$$\begin{aligned}\frac{\partial}{\partial t}(uh)+\frac{\partial}{\partial x}\left(u^2h+\frac{1}{2}g(\eta^2-2\eta z)\right)+\frac{\partial}{\partial y}(uvh)&\\ =-\frac{1}{\phi}\left[(u^2h)\frac{\partial\phi}{\partial x}+(uvh)\frac{\partial\phi}{\partial y}\right]-g\eta\frac{\partial z}{\partial x}-gh\frac{u\sqrt{u^2+v^2}}{k^2h^{4/3}}&\\ -ghs_xu\sqrt{u^2+v^2}&\end{aligned}\tag{3.34}$$

The momentum equation in *y*-direction can be evaluated in a similar way.

Hence, the modified 2D-SWEs with porosity result:

$$\begin{cases} \dfrac{\partial \eta}{\partial t} + \dfrac{\partial}{\partial x}(uh) + \dfrac{\partial}{\partial y}(vh) = -\dfrac{1}{\phi}\left((uh)\dfrac{\partial \phi}{\partial x} + (vh)\dfrac{\partial \phi}{\partial y}\right) \\ \dfrac{\partial}{\partial t}(uh) + \dfrac{\partial}{\partial x}\left(u^2 h + \dfrac{1}{2}g(\eta^2 - 2\eta z)\right) + \dfrac{\partial}{\partial y}(uvh) = \\ -\dfrac{1}{\phi}\left[(u^2 h)\dfrac{\partial \phi}{\partial x} + (uvh)\dfrac{\partial \phi}{\partial y}\right] - g\eta\dfrac{\partial z}{\partial x} - gh\dfrac{u\sqrt{u^2+v^2}}{k^2 h^{4/3}} - ghs_x u\sqrt{u^2+v} \\ \dfrac{\partial}{\partial t}(vh) + \dfrac{\partial}{\partial x}(uvh) + \dfrac{\partial}{\partial y}\left(v^2 h + \dfrac{1}{2}g(\eta^2 - 2\eta z)\right) = \\ -\dfrac{1}{\phi}\left[(uvh)\dfrac{\partial \phi}{\partial x} + (v^2 h)\dfrac{\partial \phi}{\partial y}\right] - g\eta\dfrac{\partial z}{\partial x} - gh\dfrac{v\sqrt{u^2+v^2}}{k^2 h^{4/3}} - ghs_y v\sqrt{u^2+v} \end{cases} \quad (3.35)$$

The formulation in (3.35) has the advantage of considering the porosity by introducing an additional source term in the continuity equation and two additional ones in the two momentum equations (first and fourth terms in RHS), while preserving the adopted classical SWE scheme. Moreover, the derived system (3.35) is well balanced. For a given condition of water at rest ($u = v = 0$), system (3.35) intrinsically guarantees the *C-property* also in presence of porosity, regardless the adopted discretization of the slope source term:

$$\begin{cases} \dfrac{\partial \eta}{\partial t} = 0 \\ \dfrac{\partial}{\partial x}\left(\dfrac{1}{2}g(\eta^2 - 2\eta z)\right) = -g\eta\dfrac{\partial z}{\partial x} \\ \dfrac{\partial}{\partial y}\left(\dfrac{1}{2}g(\eta^2 - 2\eta z)\right) = -g\eta\dfrac{\partial z}{\partial x} \end{cases} \quad (3.36)$$

## 3.4. Analytical solutions for 1D tests

With the aim of creating 1D reference solutions for numerical models, MacDonald et al. (1997) provided benchmark solutions holding in non-prismatic cross sections, for transitions between subcritical and supercritical flows and with non-uniform bed slope. The analytical solutions were derived by solving an inverse problem, which led to the definition of the bed slope once fixed the water depth and flow rate.

In this section, the procedure of MacDonald et al. (1997) for solving the inverse problem is extended to 1D SWE with porosity.

Firstly, assuming a 1D scheme and a steady flow, and defining $q = uh$, the modified momentum equation in (3.32) is rewritten as follows:

$$\frac{d}{dx}\left(\frac{q^2}{h}+\frac{1}{2}gh^2\right) = -\frac{1}{\phi}\frac{q^2}{h}\frac{d\phi}{dx} - gh\frac{dz}{dx} - gh\frac{u^2}{k^2h^{4/3}} - ghs_x u^2 \tag{3.37}$$

Then, expanding the derivatives at the right hand side results:

$$\left(-\frac{q^2}{h^2}+gh\right)\frac{dh}{dx} + \frac{2q}{h}\frac{dq}{dx} = -\frac{1}{\phi}\frac{q^2}{h}\frac{d\phi}{dx} - gh\frac{dz}{dx} - gh\frac{u^2}{k^2h^{4/3}} - ghs_x u^2 \tag{3.38}$$

By applying the same hypotheses to the continuity equation (3.28), the following relation holds:

$$\frac{dq}{dx} = -\frac{q}{\phi}\frac{d\phi}{dx} \tag{3.39}$$

Therefore, substituting Equation (3.39) in (3.38), and dividing each term by the factor $gh$, the momentum equation becomes:

$$\left(-\frac{q^2}{gh^3}+1\right)\frac{dh}{dx} = -\frac{q^2}{gh^2}\frac{d\phi}{dx} + S_0 - \frac{u^2}{k^2h^{4/3}} - s_x u^2 \tag{3.40}$$

where $S_0$ denotes the bed slope term.

Finally, by inverting Equation (3.40), the bed slope term reads:

$$S_0 = \left(-\frac{q^2}{gh^3}+1\right)\frac{dh}{dx} + \frac{q^2}{gh^2}\frac{d\phi}{dx} + \frac{u^2}{k^2h^{4/3}} + s_x u^2 \tag{3.41}$$

The inverse problem is solved by arbitrarily choosing the specific discharge $\hat{q}$, the roughness coefficient $\hat{k}$, and the water depth $\hat{h}$ and porosity $\hat{\phi}$ profiles ($\hat{h}$ and $\hat{\phi}$ are assumed differentiable everywhere). Therefore, the channel bed slope is given by:

$$S_0 = \left(-\frac{\hat{q}^2}{g\hat{h}^3}+1\right)\frac{d\hat{h}}{dx}+\frac{\hat{q}^2}{g\hat{h}^2}\frac{d\hat{\phi}}{dx}+\frac{\hat{u}^2}{\hat{k}^2\hat{h}^{4/3}}+\widehat{s_x}\hat{u}^2 \tag{3.42}$$

Hence, if $h=\hat{h}$, the differential equation (3.41) is satisfied for the entire reach.

## 3.5. Reference solutions for 2D tests

Moving to unsteady test cases, the reference solution of a 2D circular dam-break with variable porosity can be obtained by solving the 1D system written in cylindrical coordinates, due to the radial symmetry.

Assuming a frictionless and horizontal bed ($\eta = h$) and neglecting the head loss term due to obstructions, the 2D-SWEs with isotropic porosity in Equation (3.35), can be written as:

$$\begin{cases}\dfrac{\partial h}{\partial t}+\dfrac{\partial(uh)}{\partial x}+\dfrac{\partial(vh)}{\partial y}=-\dfrac{h}{\phi}\left(u\dfrac{\partial\phi}{\partial x}+v\dfrac{\partial\phi}{\partial y}\right)\\ \dfrac{\partial(uh)}{\partial t}+\dfrac{\partial}{\partial x}\left(u^2h+\dfrac{1}{2}gh^2\right)+\dfrac{\partial(uvh)}{\partial y}=-\dfrac{uh}{\phi}\left(u\dfrac{\partial\phi}{\partial x}+v\dfrac{\partial\phi}{\partial y}\right)\\ \dfrac{\partial(vh)}{\partial t}+\dfrac{\partial(uvh)}{\partial x}+\dfrac{\partial}{\partial y}\left(v^2h+\dfrac{1}{2}gh^2\right)=-\dfrac{vh}{\phi}\left(u\dfrac{\partial\phi}{\partial x}+v\dfrac{\partial\phi}{\partial y}\right)\end{cases} \tag{3.43}$$

The system in Equation (3.43) can be written in Lagrangian coordinates as:

$$\begin{cases}\dfrac{Dh}{Dt}+h\nabla\cdot\mathbf{v}=-\dfrac{h}{\phi}\dfrac{D\phi}{Dt}\\ \dfrac{D(\mathbf{v}h)}{Dt}+\mathbf{v}h\nabla\cdot\mathbf{v}=-\dfrac{1}{2}g\nabla h^2-\dfrac{\mathbf{v}h}{\phi}\dfrac{D\phi}{Dt}\end{cases} \tag{3.44}$$

where $\frac{D}{Dt}$ represents the Lagrangian derivative, and $\boldsymbol{v}=[u,v]$ the velocity vector.

Shifting from a Cartesian to a cylindrical reference frame with radial distance $r$, rotation angle $\theta$ and distance $z$ along the considered axis, and unit vectors $\mathbf{i}_r$, $\mathbf{i}_\theta$ and $\mathbf{i}_z$, the total time derivative, the divergence and the gradient of a generic field $A$ (assuming $\mathbf{i}_z = 0$) can be written as:

$$\frac{DA}{Dt} = \frac{\partial A}{\partial t} + u_r \frac{\partial A}{\partial r} + \frac{u_\theta}{r} \frac{\partial A}{\partial \theta} \tag{3.45}$$

$$\nabla \cdot \mathbf{A} = \frac{1}{r} \frac{\partial}{\partial r}(rA_r) + \frac{1}{r} \frac{\partial}{\partial \theta}(A_\theta) \tag{3.46}$$

$$\nabla \mathrm{A} = \frac{\partial A}{\partial r} \mathbf{i}_r + \frac{1}{r} \frac{\partial A}{\partial \theta} \mathbf{i}_\theta \tag{3.47}$$

Recalling that the porosity is constant in time and that the system presents a radial symmetry, and assuming that the tangential velocity is null ($u_\theta = 0$), the system of Equation (3.44) can be written in cylindrical coordinates as:

$$\begin{cases} \dfrac{\partial h}{\partial t} + u_r \dfrac{\partial h}{\partial r} + \dfrac{h}{r} \dfrac{\partial (r u_r)}{\partial r} = -\dfrac{h}{\phi}\left(u_r \dfrac{\partial \phi}{\partial r}\right) \\ \dfrac{\partial (u_r h)}{\partial t} + u_r \dfrac{\partial (u_r h)}{\partial r} + \dfrac{u_r h}{r} \dfrac{\partial (r u_r)}{\partial r} = -\dfrac{1}{2} g \dfrac{\partial h^2}{\partial r} - \dfrac{u_r h}{\phi}\left(u_r \dfrac{\partial \phi}{\partial r}\right) \end{cases} \tag{3.48}$$

After simple manipulations, the system in Equation (3.48) reduces to the following one:

$$\begin{cases} \dfrac{\partial h}{\partial t} + \dfrac{\partial (u_r h)}{\partial r} = -\dfrac{u_r h}{r} - \dfrac{h}{\phi}\left(u_r \dfrac{\partial \phi}{\partial r}\right) \\ \dfrac{\partial (u_r h)}{\partial t} + \dfrac{\partial}{\partial r}\left({u_r}^2 h + \dfrac{1}{2} g h^2\right) = -\dfrac{{u_r}^2 h}{r} - \dfrac{{u_r}^2 h}{\phi} \dfrac{\partial \phi}{\partial r} \end{cases} \tag{3.49}$$

The porosity field is now assumed to be defined as follows:

$$\phi = \frac{L_0 \phi_b(r)}{r} \tag{3.50}$$

where $L_0$ denotes a reference length and $\phi_b(r)$ a generic radial porosity field.

Substituting Eq. (3.50) and its derivative $\frac{\partial \phi}{\partial r}$, in the second right hand side terms of Equations (3.49), the 2D SWEs of Equation (3.43) finally reduce to:

$$\begin{cases} \dfrac{\partial h}{\partial t} + \dfrac{\partial (u_r h)}{\partial r} = -\dfrac{u_r h}{\phi_b} \dfrac{\partial \phi_b}{\partial r} \\ \dfrac{\partial (u_r h)}{\partial t} + \dfrac{\partial}{\partial r}\left({u_r}^2 h + \dfrac{1}{2} g h^2\right) = -\dfrac{{u_r}^2 h}{\phi_b} \dfrac{\partial \phi_b}{\partial r} \end{cases} \tag{3.51}$$

It is relevant to notice that the system of Equations (3.51) is identical to the 1D-SWEs. This means that, if Equation (3.50) holds, then the 2D-SWEs with variable porosity $\phi(r)$ are equivalent to the 1D-SWEs with porosity $\phi_b(r)$. Moreover, if $\phi_b$ is constant, both source terms in the system of Equation (3.51) vanish. Therefore, the reference solutions available for homogenous 1D-SWEs can be adopted also for 2D-SWEs with porosity (under the assumptions of radial symmetry and null tangential velocity).

## 3.6. Numerical tests

In this section the validation of the implemented numerical model is achieved by performing different test cases. Firstly, the capability of the scheme of preserving the condition of water at rest is numerically investigated. Secondly, the solutions deriving from the inverse approach of MacDonald are adopted for testing 1D configurations. Finally, a circular test with radial symmetry is considered as to validate the 2D scheme.

All the simulations were performed setting the gravitational acceleration as $g = 9.81$ m/s$^2$ and assuming the Courant number for the time step computation as $Cr = 0.8$.

### 3.6.1. Test with water at rest

The first test concerns the preservation of the water at rest condition and numerically investigates the capability of the implemented scheme of guarantying the *C-property*, as theoretically derived in Section 3.3. Therefore, the test presents a rectangular domain with square cell sizes of 1 m, a quiescent water depth of 10 m, a Manning roughness coefficient of 0.04 s/m$^{1/3}$ and a linearly varying porosity field (Figure 3.2). As shown in Figure 3.2, the results confirm that the model preserves water at rest also in presence of variable porosity.

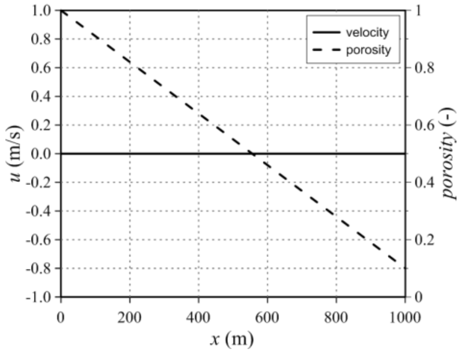


*Figure 3.2. Test with water at rest: initial porosity and resulted velocity.*

## 3.6.2. 1D tests based on MacDonald benchmarks

In the following tests, the inverse procedure described in Section 3.4 is applied as to obtain 1D analytic solutions. With reference to the parameters in Equation (3.42), all the tests were performed assuming the specific discharge $q = \sqrt{g}$ m$^2$/s, and the Strickler coefficient $k$ = 25 m$^{1/3}$/s. The rectangular domain with length $L$ = 1000 m was discretized using mesh size equal to 1 m. In the tests, the effects related to non-resolved obstacles were neglected ($\widehat{s_x \hat{u}^2} = 0$).

In the inverse procedure, three different combinations of water depth and porosity trends were considered as reported in Table 3.1. Test 1 concerned a linear variation of water depth and porosity, test 2 assumed a uniform water depth and defined the porosity with five sinusoidal waves, and test 3 considered both water depth and porosity with sinusoidal variations.

Figures 3.3-3.5 illustrate the assumed water depth and porosity trends and the bed slope resulted from the solution of the Equation (3.42). In the same figures (a), the resulted water depths are depicted, and they match the reference solutions, for all the considered tests. The differences between the analytic and the numerical water depths are shown in Figures 3.3-3.5 (b): it is noteworthy that the highest misfit (about 5 cm) recorded in Test 1 ($x$ = 750 m) is due to the flow passage through the critical state.

*Table 3.1. MacDonald tests: definition of the water depth and porosity trends.*

| | $h$ (m) | $\phi$ (-) |
|---|---|---|
| *Test 1* | Linear | Linear |
| | $y = 2 - 0.0012x$ | $y = 1 - 0.0002x$ |
| *Test 2* | Uniform | Sinusoidal |
| | $y = 1.5$ | $y = 0.8 + 0.2\sin\left(\frac{\Pi x}{100}\right)$ |
| *Test 3* | Sinusoidal | Sinusoidal |
| | $y = 2 + 0.2\sin\left(\frac{\Pi x}{100}\right)$ | $y = 1 + 0.4\sin\left(\frac{\Pi x}{100}\right)$ |

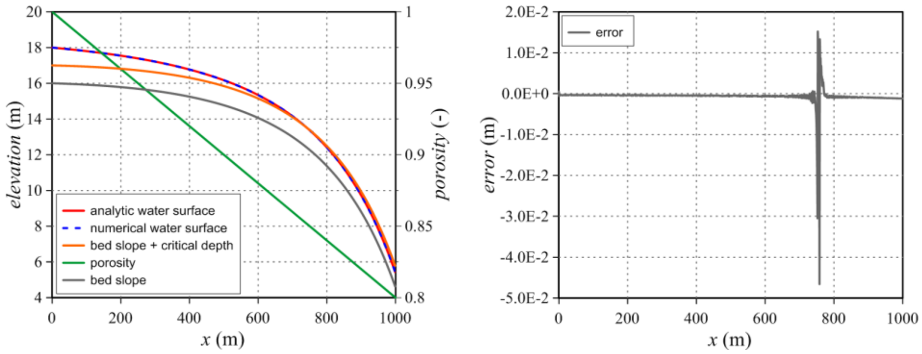


*Figure 3.3. Test 1: numerical vs analytic solution (a) and water depth errors (analytic-resulted)(b).*

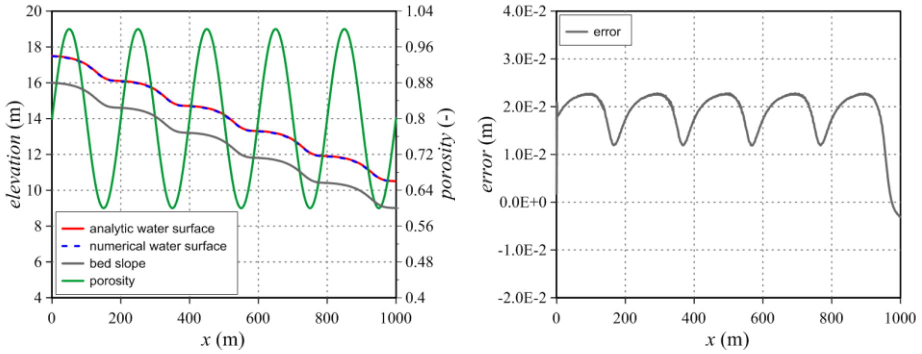


*Figure 3.4. Test 2: numerical vs analytic solution (a) and water depth errors (analytic-resulted)(b).*

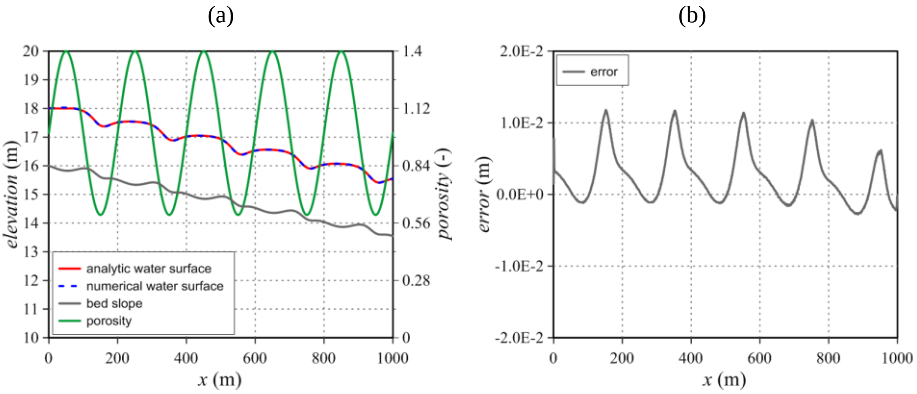


*Figure 3.5. Test 3: numerical vs analytic solution (a) and water depth errors (analytic-resulted)(b).*

Quantitative evaluation of the results was assessed by calculating the L-infinity norms as follows:

$$L_\infty = max_{i=1,N}\left|h_{i,num} - h_{i,act}\right| \tag{3.52}$$

where $N$ represents the number of cells discretizing the domain length $L$, and $h_{i,num}$ and $h_{i,act}$ the numerical and the actual water depth, respectively. The resulted $L_\infty$ values are reported in Table 3.2, and they confirm the capability of the numerical model of reproducing the analyzed 1D configurations.

*Table 3.2. MacDonald tests: $L_\infty$ norms.*

| | *Test 1* | *Test 2* | *Test 3* |
|---|---|---|---|
| $L_\infty$ $(m)$ | 0.047 | 0.023 | 0.012 |

### 3.6.3. Circular dam-break

With the aim of testing the 2D implemented scheme, a circular dam-break with variable porosity was performed, similarly to Guinot and Soares-Frazão (2006). The test was characterized by the presence of a 9 m column of still water, enclosed in a dam with radius *r* = 20 m, which separates the internal initial state from the external one, where a layer of 1 m water depth at rest is present. The dam instantaneous collapses at time $t = 0$ s. The horizontal and frictionless domain was discretized with cell sizes $\Delta x = \Delta y = 0.1$ m, and the porosity varied in space according to Equation (3.50) (Figure 3.6-a). As demonstrated in Section 3.5, the reference solution of a 2D dam-break problem with porosity is obtained by means of a classical Exact Riemann solver for the analogous 1D dam-break with constant porosity (equal to 1), that is the Stoker's solution (Stoker, 2011).

Figure 3.6 illustrates the comparison between numerical and exact water depth (a) and velocity (b) at time $t = 1$ s. In both cases, the numerical solutions capture the rarefaction wave and the shock position, as well as the intermediate state value; only the velocity trend slightly overestimates the intermediate state value near the shock wave.

The reason of this mismatch may be related to the discretization of the porosity source term. In order to avoid these possible negative effects, the presence of porosity could be considered inside the Riemann solver: this will be the topic of the next chapter.

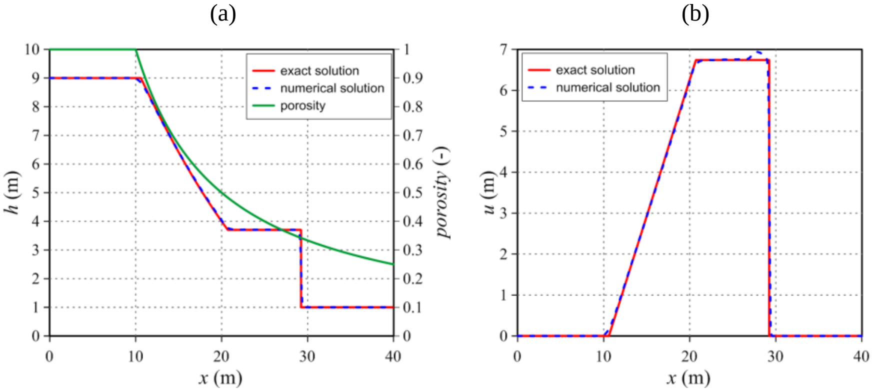


*Figure 3.6. Circular dam-break: water depth (a) and velocity (b) comparison between exact and numerical solutions at time 1 s.*

## 3.7. Concluding remarks

In this chapter the SWEs with isotropic porosity have been investigated and a 2D model capable of guarantying the *C-property* has been derived.

The implemented scheme presented two main advantages.

Firstly, in the framework of finite volume schemes, the model allowed considering the porosity only by means of two additional source terms. This means that a stable and accurate 2D-SWE numerical model can be transformed in a 2D-SWE one with porosity, without changing the conserved variables and the way the fluxes are discretized.

Secondly, the derived scheme inherently preserved the condition of water at rest.

With the aim of testing the implemented model, 1D and 2D reference solutions were derived. The solution of an inverse problem with porosity allowed the definition of 1D analytical benchmarks, whereas it has been demonstrated that 2D test cases characterized by radial symmetry and a proper porosity field can be compared against exact classical 1D Riemann solver solutions (Stoker’s solutions).

Different test cases were performed and the numerical results showed the capability of the model in reproducing the exact solutions.

As this chapter mostly focused on the derivation of a well-balanced scheme and of the reference solutions, future implications may concern more practical aspects, such as the modelling of a real urban flooding.

# Chapter 4

# The Shallow Water Equations with porosity (Part II): a Generalized Riemann solver for discontinuous porosity field

## 4.1. Introduction

As described in Section 3.1, literature presents a various debate on porosity models, concentrating above all on the adopted spatial parameters (storage, conveyance), on the derivation, on the head-loss formulation and experimental validation, on the porosity characteristic (depth-independent or depth-dependent) and on the general approach (single, multiple porosity). While some discretization formulas are available for the non-resolved obstacle source term, the treatment of the source term due to porosity variation in space has still not received much attention. Moreover, the problem of dealing with porosity discontinuities has not been investigated as the analogous bottom step problem related to the bed slope source term (Alcrudo and Benkhaldoun, 2001; Bernetti et al., 2008; Rosatti and Begnudelli, 2010; Cozzolino et al., 2011; Valiani and Caleffi, 2017). However, porosity is intrinsically discontinuous in real bathymetries, and a robust and accurate discretization of the porosity source term is fundamental to guarantee the accuracy and the stability of the numerical schemes. With the purpose of investigating the porosity variation source term, in this chapter a novel Generalized Riemann Problem (GRP), which incorporates porosity, is defined. Then, a possible efficient solution method using an approximate Roe Riemann solver is proposed. This solver has been introduced adopting the most established isotropic porosity formulation of SWEs (Guinot and Soares-Frazão, 2006), but it can be extended to different formulations, including those based on anisotropic porosity.

The capability of handling transonic waves across the discontinuity has been obtained by deriving an entropy fix, which is based on the eigenstructure of the augmented SWEs system. A general procedure for both 1D and 2D schemes is presented, which allows capturing the wave pattern generated by a Riemann problem due to a discontinuity in the porosity field. Moreover, a 1D Exact Riemann solver is derived for the GRP with the aim of obtaining reference solutions for different test cases.

This Chapter is derived in part from Ferrari et al. (2017).

## 4.2. The Riemann Problem with a porosity discontinuity

Following the idea originally proposed for the other source terms (LeRoux, 1998), the effect of the porosity variation and thus the presence of discontinuities in the porosity field is considered inside the Riemann solver, with the advantage of avoiding a special discretization procedure for these source terms.

With the aim of creating an augmented system of partial differential equations, the porosity $\phi$ is considered as an additional fictitious conserved variable, and a new equation, stating the invariance of porosity in time, is enclosed to the system (3.26). It is worth noting that this last equation is introduced as to create the GRP, but it is not actually integrated in time.

Assuming a frictionless and horizontal bed, and neglecting the head loss terms due to obstructions, the governing system of Equations (3.26) becomes:

$$\begin{cases} \dfrac{\partial}{\partial t}(\phi h) + \dfrac{\partial}{\partial x}(\phi u h) + \dfrac{\partial}{\partial y}(\phi v h) = 0 \\ \dfrac{\partial}{\partial t}(\phi u h) + \dfrac{\partial}{\partial x}\left(\phi u^2 h + \dfrac{1}{2} g \phi h^2\right) + \dfrac{\partial}{\partial y}(\phi u v h) = \dfrac{1}{2} g h^2 \dfrac{\partial \phi}{\partial x} \\ \dfrac{\partial}{\partial t}(\phi v h) + \dfrac{\partial}{\partial x}(\phi u v h) + \dfrac{\partial}{\partial y}\left(\phi v^2 h + \dfrac{1}{2} g \phi h^2\right) = \dfrac{1}{2} g h^2 \dfrac{\partial \phi}{\partial y} \\ \dfrac{\partial \phi}{\partial t} = 0 \end{cases} \tag{4.1}$$

## 4.2.1. Eigenstructure analysis for the 1D scheme

The augmented SWE porosity system requires the definition of a new set of eigenvalues and eigenvectors, which will not necessarily coincide with those of the classical SWE system. With the aim of studying the eigenstructure, system of Equations (4.1) is rewritten for the 1D case in the following form:

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}(\mathbf{U})}{\partial x} + \mathbf{H}\frac{\partial \mathbf{U}}{\partial x} = 0 \tag{4.2}$$

where the vectors **U** and **F(U)** of the conserved variables and fluxes, and the matrix **H** of the non-conservative fluxes due to the porosity variation, are defined as:

$$\mathbf{U} = \begin{bmatrix} \phi h \\ \phi u h \\ \phi \end{bmatrix} \qquad \mathbf{F}(\mathbf{U}) = \begin{bmatrix} \phi u h \\ \phi u^2 h + \dfrac{1}{2} g \phi h^2 \\ 0 \end{bmatrix} \qquad \mathbf{H} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -\dfrac{1}{2} g h^2 \\ 0 & 0 & 0 \end{bmatrix} \tag{4.3}$$

By adopting the chain rule, Equation (4.2) can be written in quasi-linear form as:

$$\frac{\partial \mathbf{U}}{\partial t} + \mathbf{A}\frac{\partial \mathbf{U}}{\partial x} = 0 \tag{4.4}$$

where $\mathbf{A} = \mathbf{J} + \mathbf{H}$, and the Jacobian matrix of fluxes vector **J** is calculated as:

$$\mathbf{J} = \frac{\partial \mathbf{F}}{\partial \mathbf{U}} = \begin{bmatrix} 0 & 1 & 0 \\ gh - u^2 & 2u & -\dfrac{1}{2} g h^2 \\ 0 & 0 & 0 \end{bmatrix} \tag{4.5}$$

and **A** becomes:

$$\mathbf{A} = \mathbf{J} + \mathbf{H} = \begin{bmatrix} 0 & 1 & 0 \\ gh - u^2 & 2u & -gh^2 \\ 0 & 0 & 0 \end{bmatrix} \tag{4.6}$$

By computing the roots of the characteristic polynomial $|\mathbf{A} - \lambda\mathbf{I}| = 0$ (Toro, 2001), the eigenvalues of matrix **A** $\lambda_1$, $\lambda_2$ and $\lambda_3$ result as follows:

$$\lambda_1 = u - \sqrt{gh} \qquad \lambda_2 = 0 \qquad \lambda_3 = u + \sqrt{gh} \tag{4.7}$$

As the eigenvalues are real and distinct, the augmented system of Shallow Water Equations is still hyperbolic. Moreover, the characteristic fields $\lambda_1$ and $\lambda_3$ are genuinely non-linear and equal to those of the Shallow Water Equations without porosity, while the characteristic field $\lambda_2$ is linearly degenerate. Consequently, the waves that can develop from the eigenvalues $\lambda_1$ and $\lambda_3$ can be shocks or rarefactions, while the wave associated with the eigenvalue $\lambda_2$ is a contact wave.

Recalling that a right eigenvector $\mathbf{R}=[\mathbf{R}_1, \mathbf{R}_2, \mathbf{R}_3]^T$ corresponds to each eigenvalue $\lambda$ of matrix **A** according to the relation $\mathbf{AR} = \lambda\mathbf{R}$ (Toro, 2001), the set of the resulting right eigenvectors is:

$$\mathbf{R}_1 = \begin{bmatrix} 1 \\ u - \sqrt{gh} \\ 0 \end{bmatrix} \qquad \mathbf{R}_2 = \begin{bmatrix} 1 \\ 0 \\ \dfrac{gh - u^2}{gh^2} \end{bmatrix} \qquad \mathbf{R}_3 = \begin{bmatrix} 1 \\ u + \sqrt{gh} \\ 0 \end{bmatrix} \tag{4.8}$$

## Definition of the Riemann Invariants

With the purpose of calculating the Riemann Invariants, the quasi-linear 1D system in Equation (4.2) is rewritten according to the chain rule:

$$\frac{\partial \mathbf{U}}{\partial \mathbf{W}}\frac{\partial \mathbf{W}}{\partial t} + \frac{\partial \mathbf{F}(\mathbf{U})}{\partial \mathbf{W}}\frac{\partial \mathbf{W}}{\partial x} + \mathbf{H}\frac{\partial \mathbf{U}}{\partial \mathbf{W}}\frac{\partial \mathbf{W}}{\partial x} = 0 \tag{4.9}$$

and defining $\mathbf{D} = \frac{\partial \mathbf{U}}{\partial \mathbf{W}}$ and $\mathbf{J} = \frac{\partial \mathbf{F}(\mathbf{U})}{\partial \mathbf{W}}$ Equation (4.9) becomes:

$$\mathbf{D}\frac{\partial \mathbf{W}}{\partial t} + \mathbf{J}\frac{\partial \mathbf{W}}{\partial x} + \mathbf{HD}\frac{\partial W}{\partial x} = 0 \tag{4.10}$$

Multiplying each term by the inverse of matrix **D**, the formula results:

$$\frac{\partial \mathbf{W}}{\partial t} + (\mathbf{D}^{-1}\mathbf{J} + \mathbf{D}^{-1}\mathbf{H}\mathbf{D}) \frac{\partial \mathbf{W}}{\partial x} = 0 \tag{4.11}$$

Assuming the primitive variables vector as $\mathbf{W}$=$[h, u, \phi]^{\mathrm{T}}$, the terms in Equation (4.11) are defined as follows:

$$\mathbf{D} = \begin{bmatrix} \phi & 0 & h \\ \phi u & \phi h & uh \\ 0 & 0 & 1 \end{bmatrix} \qquad \mathbf{D}^{-1} = \begin{bmatrix} \frac{1}{\phi} & 0 & -\frac{h}{\phi} \\ -\frac{u}{\phi h} & \frac{1}{\phi h} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{J} = \begin{bmatrix} \phi u & \phi h & uh \\ \phi u^2 + g\phi h & 2\phi uh & u^2 h + \frac{1}{2} g h^2 \\ 0 & 0 & 0 \end{bmatrix} \qquad \mathbf{H} = -\frac{1}{2} g h^2 \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \tag{4.12}$$

The eigenvalues of system (4.11) are calculated as the roots of the characteristic polynomial $|\mathbf{D}^{-1}\mathbf{J} + \mathbf{D}^{-1}\mathbf{H}\mathbf{D} - \lambda\mathbf{I}| = 0$, and they coincide with those presented in Equation (4.7).

The set of the right eigenvectors, which derives from the solution of the relation $(\mathbf{D}^{-1}\mathbf{J} + \mathbf{D}^{-1}\mathbf{H}\mathbf{D})\mathbf{R} = \lambda\mathbf{R}$ (Toro, 2001), results:

$$\mathbf{R}_1 = \begin{bmatrix} 1 \\ -\frac{\sqrt{gh}}{h} \\ 0 \end{bmatrix} \qquad \mathbf{R}_2 = \begin{bmatrix} 1 \\ -\frac{g}{u} \\ \frac{\phi g}{u^2} - \frac{\phi}{h} \end{bmatrix} \qquad \mathbf{R}_3 = \begin{bmatrix} 1 \\ \frac{\sqrt{gh}}{h} \\ 0 \end{bmatrix} \tag{4.13}$$

Since rarefactions, as well as shock waves, can develop exclusively in genuinely non-linear fields, only the first and third eigenvectors will be studied.

Focusing on the eigenvector $\mathbf{R}_1$ the following relation can be written (Toro, 2001):

$$\frac{dh}{1} = -\frac{du}{\frac{\sqrt{gh}}{h}} = \frac{d\phi}{0} \tag{4.14}$$

From the integration of the first equality in the phase space, it results that $u + 2\sqrt{gh}$ is constant and, thus, the left Riemann Invariant between states $\mathbf{U}_L(h_L,u_L,\phi_L)$ and $\mathbf{U}_R(h_R,u_R,\phi_R)$ can be written as:

$$u_L + 2\sqrt{gh_L} = \mathrm{u}_R + 2\sqrt{gh_R} \tag{4.15}$$

Analogously, the following relation subsists for the eigenvector **R**$_3$ (Toro, 2001):

$$\frac{dh}{1} = \frac{du}{\frac{\sqrt{gh}}{h}} = \frac{d\phi}{0} \tag{4.16}$$

The integration of the first equality in the phase space leads to define $u - 2\sqrt{gh}$ constant. Hence, the right Riemann Invariant asserts that the following relation is verified between states **U**$_L(h_L, u_L, \phi_L)$ and **U**$_R(h_R, u_R, \phi_R)$:

$$u_L - 2\sqrt{gh_L} = \mathrm{u}_R - 2\sqrt{gh_R} \tag{4.17}$$

The third ratios in Equations (4.14) and (4.16) state that the Riemann Invariants hold only if $\phi$ is constant across the wave, and this implies that these relations can not be adopted across a porosity discontinuity.

## 4.2.2. Equations across the porosity discontinuity

The eigenstructure analysis of the augmented 1D Shallow Water Equations showed that the eigenvalue $\lambda_2$ is null and thus the associated wave is a contact wave. While for conservative systems both Riemann Invariants and Rankine-Hugoniot conditions hold across contact waves, this is not true for non-conservative systems such as the one of Equation (4.2). Particularly, the derivation of the Riemann Invariants demonstrated that these relations are not constant across the contact wave associated with the eigenvalue $\lambda_2$.

Therefore, across the porosity discontinuity only the Rankine-Hugoniot conditions, which relate the jump of the conserved quantities across a shock wave travelling with velocity $s$ and the jump of the flux over the discontinuity, hold:

$$\begin{cases} \phi_L u_L h_L - \phi_R u_R h_R = s(\phi_L h_L - \phi_R h_R) \\ \left(\phi_L {u_L}^2 h_L + \frac{1}{2} g \phi_L {h_L}^2\right) - \left(\phi_R {u_R}^2 h_R + \frac{1}{2} g \phi_R {h_R}^2\right) \\ \qquad\qquad = s(\phi_L u_L h_L - \phi_R u_R h_R) + P_r \\ s(\phi_L - \phi_R) = 0 \end{cases} \tag{4.18}$$

where $s$ denotes the shock speed, and $P_r$ the term due to the presence of the porosity discontinuity.

Focusing on the third Equation in (4.18), it clearly emerges that in presence of a porosity discontinuity the shock velocity is null, and consequently the mass and momentum equations reduce to:

$$\begin{cases}\phi_L u_L h_L = \phi_R u_R h_R \\ \phi_L {u_L}^2 h_L + \dfrac{1}{2} g \phi_L {h_L}^2 - \phi_R {u_R}^2 h_R - \dfrac{1}{2} g \phi_R {h_R}^2 = P_r\end{cases} \tag{4.19}$$

For GRP derived for slope source terms, it has been demonstrated that the effect of the bottom discontinuity can be physically seen as the reaction exerted by the bottom step onto the fluid (Rosatti and Begnudelli, 2010; Rosatti and Fraccarollo, 2006). A similar physical interpretation can be assumed for the porosity discontinuity. Recalling that $\phi$ is defined as the fraction of plan view area available to flow, the discontinuity in the porosity field between $\phi_L$ and $\phi_R$ can be considered analogously to the width variation along a channel (Figure 4.1). In a one-dimensional framework, the contraction is characterized by only one geometric parameter, namely the width ratio $\phi_R/\phi_L$ (Defina and Viero, 2010).

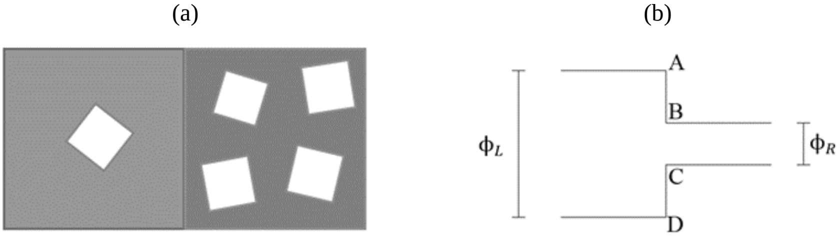


*Figure 4.1. Example of how a porosity discontinuity, with $\phi_L > \phi_R$ (a), can be seen as a reduction in the cross section width along a channel (b).*

If this physical interpretation holds, the term $P_r$ is the reaction that the walls AB and CD exert on the fluid (dimensionally $P_r$ is a force per unit width divided by the fluid density). Then, assuming a hydrostatic pressure distribution, $P_r$ is equal to:

$$P_r = -\frac{1}{2} g|\phi_L - \phi_R||h_j|h_j \qquad \text{where} \begin{cases} h_j = -h_L & \text{if} \quad \phi_L \geq \phi_R \\ h_j = h_R & \text{if} \quad \phi_L < \phi_R \end{cases} \tag{4.20}$$

Therefore, the set of equations that can be written across a porosity discontinuity is as follows:

$$\begin{cases} \Phi_L u_L h_L = \phi_R u_R h_R \\ \phi_L {u_L}^2 h_L + \frac{1}{2} g \phi_L {h_L}^2 - \phi_R {u_R}^2 h_R - \frac{1}{2} g \phi_R {h_R}^2 = -\frac{1}{2} g|\phi_L - \phi_R||h_j|h_j \end{cases} \tag{4.21}$$

# 4.3. 1D Approximate Riemann Solver

## Augmented Roe Solver: eigenvalues, eigenvectors, wave strengths

The first order Godunov method allows updating the approximate solution of a generic non-linear hyperbolic system to the next time-step by means of the finite volume formula (Toro, 2001):

$$\mathbf{U}_i^{n+1} = \mathbf{U}_i^n - \frac{\Delta t}{\Delta x}\left(\mathbf{F}_{i+\frac{1}{2}} - \mathbf{F}_{i-\frac{1}{2}}\right) \tag{4.22}$$

where *i* is the considered cell, $\mathbf{F}_{i+\frac{1}{2}}$ and $\mathbf{F}_{i-\frac{1}{2}}$ are the intercell fluxes and $\mathbf{U}_i^n$ is the cell-averaged values of the conserved variables at time *n*. For a discontinuity located at *x* = 0, the initial value local Riemann Problem can be linearly approximated as follows (Toro, 2001):

$$\begin{cases} \frac{\partial \widehat{\mathbf{U}}}{\partial t} + \tilde{\mathbf{J}}(\mathbf{U}_L, \mathbf{U}_R) \frac{\partial \widehat{\mathbf{U}}}{\partial x} = 0 \\ \widehat{\mathbf{U}}(x, 0) = \begin{cases} \mathbf{U}_L^n & \text{if} \quad x < 0 \\ \mathbf{U}_R^n & \text{if} \quad x > 0 \end{cases} \end{cases} \tag{4.23}$$

where, according to the Generalized Roe scheme (Rosatti et al., 2008), the constant matrix $\tilde{\mathbf{J}}(\mathbf{U}_L, \mathbf{U}_R)$ is defined as:

$$\tilde{\mathbf{J}}(\mathbf{U}_L, \mathbf{U}_R) = (\mathbf{A}' + \mathbf{A}'')\mathbf{B}^{-1} \tag{4.24}$$

The adoption of the approximate Roe solver, which locally linearizes the problem, requires that the integrals on the same control volume of the approximated solution ($\widehat{\mathbf{U}}$)

and of the exact solution must be equal (consistency condition). In order to fulfil this condition, the following relations must hold:

$$\mathbf{A}'(\mathbf{W}_R-\mathbf{W}_L)=\mathbf{F}(\mathbf{U}_R)-\mathbf{F}(\mathbf{U}_L) \tag{4.25}$$

$$\mathbf{A}''(\mathbf{W}_R-\mathbf{W}_L)=-\breve{\mathbf{P}}_r \tag{4.26}$$

$$\mathbf{B}(\mathbf{W}_R-\mathbf{W}_L)=\mathbf{U}_R-\mathbf{U}_L \tag{4.27}$$

where $\mathbf{W}=[h,\ u,\ \phi]^{\mathrm{T}}$ is the vector of primitive variables, $\breve{\mathbf{P}}_r=[0,\ P_r,\ 0]^{\mathrm{T}}$, and the Jacobian matrixes of the fluxes $\mathbf{A}'$ and of the conserved variables $\mathbf{B}$ and the matrix $\boldsymbol{A}''$ are defined as follows:

$$\mathbf{A}'=\left.\frac{\partial\mathbf{F}}{\partial\mathbf{W}}\right|_{\widetilde{\mathbf{W}}}=\begin{bmatrix}\phi u & \phi h & uh\\ \phi u^2+g\phi h & 2\phi uh & u^2h+\dfrac{1}{2}gh^2\\ 0 & 0 & 0\end{bmatrix} \tag{4.28}$$

$$\mathbf{B}=\left.\frac{\partial\mathbf{U}}{\partial\mathbf{W}}\right|_{\widetilde{\mathbf{W}}}=\begin{bmatrix}\phi & 0 & h\\ \phi u & \phi h & uh\\ 0 & 0 & 1\end{bmatrix} \tag{4.29}$$

$$\mathbf{A}''=\begin{bmatrix}0 & 0 & 0\\ 0 & 0 & a_{23}\\ 0 & 0 & 0\end{bmatrix}=\begin{bmatrix}0 & 0 & 0\\ 0 & 0 & -\dfrac{1}{2}g\dfrac{|\phi_L-\phi_R|}{\phi_R-\phi_L}|h_j|h_j\\ 0 & 0 & 0\end{bmatrix}\quad\begin{cases}h_j=-h_L & \phi_L\geq\phi_R\\ h_j=h_R & \phi_L<\phi_R\end{cases} \tag{4.30}$$

The averaged values of primitive variables $\tilde{h}$ and $\tilde{u}$ are defined as (LeVeque, 2002):

$$\tilde{h}=\frac{h_L+h_R}{2}\qquad\qquad \tilde{u}=\frac{\sqrt{\phi_L h_L}u_L+\sqrt{\phi_R h_R}u_R}{\sqrt{\phi_L h_L}+\sqrt{\phi_R h_R}} \tag{4.31}$$

Using $\tilde{h}$ and $\tilde{u}$, the eigenvalues $[\tilde{\lambda}_1,\tilde{\lambda}_2,\tilde{\lambda}_3]$ of the matrix $\tilde{\mathbf{J}}(\mathbf{U}_L,\mathbf{U}_R)$ can be computed using Equation (4.7), whereas the corresponding eigenvectors, which are different from the ones presented in Equation (4.8), result:

$$\widetilde{\mathbf{R}}_1=\begin{bmatrix}1\\ \tilde{u}-\sqrt{g\tilde{h}}\\ 0\end{bmatrix}\qquad \widetilde{\mathbf{R}}_2=\begin{bmatrix}a_{23}-\dfrac{1}{2}g\tilde{h}^2\\ 0\\ \tilde{u}^2-g\tilde{h}\end{bmatrix}\qquad \widetilde{\mathbf{R}}_3=\begin{bmatrix}1\\ \tilde{u}+\sqrt{g\tilde{h}}\\ 0\end{bmatrix} \tag{4.32}$$

The Roe solver approximates the differences between conserved variables $\boldsymbol{\delta}=\mathbf{U}_R-\mathbf{U}_L$ as follows (LeVeque, 2002):

$$\boldsymbol{\delta} = \sum_{p=1}^{m} (\tilde{\alpha}_p)_{i+1/2} (\widetilde{\mathbf{R}}_p)_{i+1/2} \tag{4.33}$$

where $m$ represents the number of the waves separating the two initial states, and $\tilde{\alpha}_p$ the wave strengths. Inverting the matrix of the right eigenvectors $\widetilde{\mathbf{R}}$ and multiplying it by vector $\boldsymbol{\delta}$, the following expression for $\tilde{\alpha}_p$ is derived:

$$\tilde{\alpha}_1 = -\frac{\tilde{\lambda}_3}{\tilde{\lambda}_1 - \tilde{\lambda}_3}\delta_1 + \frac{1}{\tilde{\lambda}_1 - \tilde{\lambda}_3}\delta_2 + \frac{2a_{23} - g\tilde{h}^2}{2\tilde{\lambda}_1(\tilde{\lambda}_1 - \tilde{\lambda}_3)}\delta_3 \tag{4.34}$$

$$\tilde{\alpha}_2 = \frac{1}{\tilde{\lambda}_1\tilde{\lambda}_3}\delta_3 \tag{4.35}$$

$$\tilde{\alpha}_3 = \frac{\tilde{\lambda}_1}{\tilde{\lambda}_1 - \tilde{\lambda}_3}\delta_1 - \frac{1}{\tilde{\lambda}_1 - \tilde{\lambda}_3}\delta_2 - \frac{2a_{23} - g\tilde{h}^2}{2\tilde{\lambda}_3(\tilde{\lambda}_1 - \tilde{\lambda}_3)}\delta_3 \tag{4.36}$$

## Approximate intermediate state values

The approximate solution of the Riemann Problem consists of a sum of jumps across the $p$-waves ($\lambda$ eigenvalues), which separate inner constant states between the two initial states $\mathbf{U}_L$ and $\mathbf{U}_R$ (Equation (4.33)).

In presence of a stationary contact wave, located at $x = 0$, the state immediately to the left $\mathbf{U}_i^-$ and the one immediately to the right $\mathbf{U}_{i+1}^+$ can be approximated by the formulas (Murillo and Navas-Montilla, 2016; Rosatti et al., 2008):

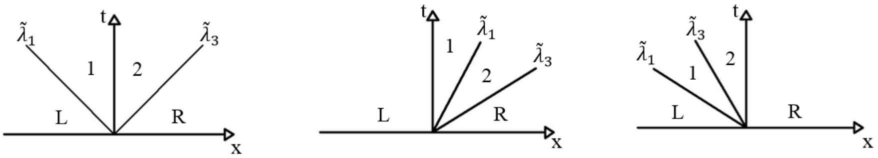
$$\mathbf{U}_i^- = \mathbf{U}_L + \sum_{p=1}^{m_{neg}} (\tilde{\alpha}_p \widetilde{\mathbf{R}}_p)_{i+\frac{1}{2}} \tag{4.37}$$

$$\mathbf{U}_{i+1}^+ = \mathbf{U}_R - \sum_{p=m_{neg}+1}^{m_{pos}} (\tilde{\alpha}_p \widetilde{\mathbf{R}}_p)_{i+\frac{1}{2}} \tag{4.38}$$

where $\tilde{\alpha}_p$ and $\widetilde{\mathbf{R}}_p$ are the wave strength and the eigenvector referred to the $p$ wave, $m_{neg}$ and $m_{pos}$ are the number of negative and positive waves, respectively.

According to the eigenstructure analysis of the augmented SWEs system with porosity, the problem presents two distinct eigenvalues $\lambda_1$ and $\lambda_3$, and a third one $\lambda_2 = 0$, to which a contact wave is associated. Therefore, assuming that no transonic waves are present (this case is described in next section), the approximate solution of the Riemann Problem with initial states $\mathbf{U}_L(h_L,u_L,\phi_L)$ and $\mathbf{U}_R(h_R,u_R,\phi_R)$ consists of two inner constant states $\widetilde{\mathbf{U}}_1$ and $\widetilde{\mathbf{U}}_2$, which can be calculated using Equations (4.37) and (4.38).

Figure 4.2 shows in the *x*-*t* plane the three configurations that may occur depending on the sign of the eigenvalues $\tilde{\lambda}_1$ and $\tilde{\lambda}_3$, and the formulas adopted to calculate the intermediate approximate states $\widetilde{\mathbf{U}}_1$ and $\widetilde{\mathbf{U}}_2$ for each scheme: subcritical condition with $\tilde{\lambda}_1 < 0$ and $\tilde{\lambda}_3 > 0$ (Figure 4.2-a), supercritical condition with $\tilde{\lambda}_1 > 0$ and $\tilde{\lambda}_3 > 0$ (Figure 4.2-b), and supercritical condition with $\tilde{\lambda}_1 < 0$ and $\tilde{\lambda}_3 < 0$ (Figure 4.2-b).



(a) $\tilde{\lambda}_1 < 0$ and $\tilde{\lambda}_3 > 0$

(b) $\tilde{\lambda}_1 > 0$ and $\tilde{\lambda}_3 > 0$

(c) $\tilde{\lambda}_1 < 0$ and $\tilde{\lambda}_3 < 0$

$$\widetilde{\mathbf{U}}_1 = \mathbf{U}_L + \tilde{\alpha}_1\widetilde{\mathbf{R}}_1 \qquad \widetilde{\mathbf{U}}_2 = \mathbf{U}_R - \tilde{\alpha}_3\widetilde{\mathbf{R}}_3 \qquad \widetilde{\mathbf{U}}_1 = \mathbf{U}_L + \tilde{\alpha}_1\widetilde{\mathbf{R}}_1$$

$$\widetilde{\mathbf{U}}_2 = \mathbf{U}_R - \tilde{\alpha}_3\widetilde{\mathbf{R}}_3 \qquad \widetilde{\mathbf{U}}_1 = \widetilde{\mathbf{U}}_2 - \tilde{\alpha}_1\widetilde{\mathbf{R}}_1 \qquad \widetilde{\mathbf{U}}_2 = \widetilde{\mathbf{U}}_1 + \tilde{\alpha}_3\widetilde{\mathbf{R}}_3$$

*Figure 4.2. The approximate solution for the states $\widetilde{\boldsymbol{U}}_1$ and $\widetilde{\boldsymbol{U}}_2$ in subcritical (a), supercritical with positive eigenvalues (b) and supercritical with negative eigenvalues (c) condition.*

## Entropy fix

Since the approximate Roe Solver defines the solution of a Riemann Problem as a sum of discontinuities, this can lead to some disadvantages when the solution includes a rarefaction wave, which spreads in *x* = 0 partly to the left and to the right. Particularly, in presence of a transonic rarefaction, the approximate solution obtained with Equations (4.37)-(4.38) would determine a violation of the Lax entropy condition, which requires the left characteristic speed greater than the propagation speed of the discontinuity and of the right characteristic speed (LeVeque, 2002). In order to solve this problem, the entropy fix procedure proposed by LeVeque (1990, 2002) is here adopted.

Firstly, a transonic rarefaction across a $p$-wave occurs when the eigenvalue $\lambda_p$ calculated in the state $(\mathbf{U}_p)_L$ to the left of the $p$-wave assumes opposite sign to the one computed in the right state $(\mathbf{U}_p)_R$:

$$\left(\lambda_p\right)_L < 0 < \left(\lambda_p\right)_R \tag{4.39}$$

When the condition that leads to an entropy violation is verified (Equation (4.39)), the entropy fix splits the transonic eigenvalue $\lambda_p$ into a positive $\lambda_p^+$ and negative $\lambda_p^-$ term, according to the relation:

$$\lambda_p = \lambda_p^+ + \lambda_p^- = (1-\beta)\left(\lambda_p\right)_R + \beta\left(\lambda_p\right)_L \tag{4.40}$$

where the $\beta$ parameter is calculated as:

$$\beta = \frac{\left(\lambda_p\right)_R - \tilde{\lambda}_p}{\left(\lambda_p\right)_R - \left(\lambda_p\right)_L} \tag{4.41}$$

For sake of clarity, if considering the $\tilde{\lambda}_1$ eigenvalue separating states $\mathbf{U}_L$ and $\mathbf{U}_1$ in Figure 4.3-a, a transonic rarefaction occurs when:

$$u_L - \sqrt{gh_L} < 0 < u_1 - \sqrt{gh_1} \tag{4.42}$$

where the approximate intermediate state values $u_1$ and $h_1$ are calculated following the procedure previously illustrated in Figure 4.2.

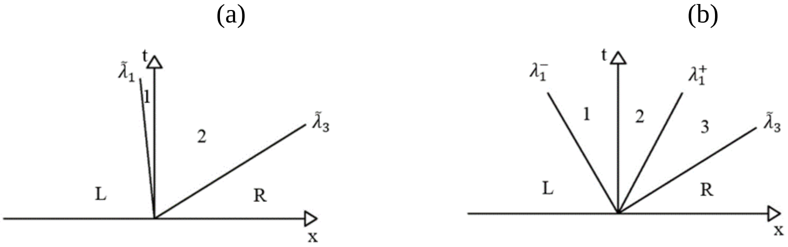


*Figure 4.3. Example of transonic rarefaction: eigenvalue $\tilde{\lambda}_1$ close to zero (a) and its splitting (b).*

Assuming that Equation (4.42) is verified, the transonic eigenvalue $\tilde{\lambda}_1$ is split as follows (Figure 4.3-b):

$$\tilde{\lambda}_1 = \lambda_1^+ + \lambda_1^- = \left(1 - \frac{(\lambda_1)_1 - \tilde{\lambda}_1}{(\lambda_1)_1 - (\lambda_1)_L}\right)(\lambda_1)_1 + \frac{(\lambda_1)_1 - \tilde{\lambda}_1}{(\lambda_1)_1 - (\lambda_1)_L}(\lambda_1)_L \tag{4.43}$$

Otherwise, if there is no transonic rarefaction, the eigenvalue $\tilde{\lambda}_1$ resulting from the Roe solver is preserved, and a positive $\lambda_p^+$ and negative $\lambda_p^-$ are calculated as follows:

$$\lambda_1^+ = max(\tilde{\lambda}_1, 0) \tag{4.44}$$

$$\lambda_1^- = min(\tilde{\lambda}_1, 0) \tag{4.45}$$

An analogous formulation can be derived considering the $\tilde{\lambda}_3$ eigenvalue separating states $\mathbf{U}_R$ and $\mathbf{U}_2$ in Figure 4.4-a.

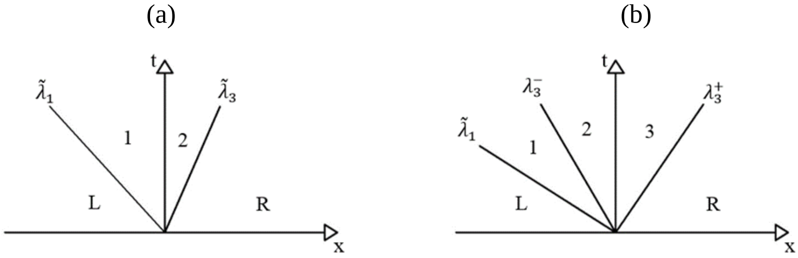


*Figure 4.4. Example of a subcritical case with $h_2<0$ (a) and a transonic rarefaction across $\tilde{\lambda}_3$ (b).*

As previously illustrated, the entropy fix procedure requires the approximate values of the intermediate states $\widetilde{\mathbf{U}}_1$ and $\widetilde{\mathbf{U}}_2$. However, if a transonic rarefaction wave is present, one of these values can be non-physical (negative water depth), because of the presence of an additional state $\widetilde{\mathbf{U}}_3$.

Focusing for instance on the subcritical case shown in Figure 4.4-a, with $\phi_L > \phi_R$, and assuming that a transonic rarefaction is present across the $\tilde{\lambda}_3 = \tilde{u} + \tilde{c}$ eigenvalue, the intermediate right state should be calculated as $\widetilde{\mathbf{U}}_2 = \mathbf{U}_R - \tilde{\alpha}_3\widetilde{\mathbf{R}}_3$ (Equation (4.38)). However, this is not correct, because the true wave pattern for this case is the one shown in Figure 4.4-b, where an additional state $\mathbf{U}_3$ separates the intermediate state $\mathbf{U}_2$ from $\mathbf{U}_R$.

Therefore, while state **U**$_1$ can be computed from Equation (4.37), four unknowns characterize states **U**$_2$ and **U**$_3$, and the entropy fix of Equation (4.40) cannot be adopted.

However, recalling that only transonic rarefactions and not transonic shocks exist, the Riemann Invariant relation between states **U**$_3$ and **U**$_R$ can be written:

$$u_R - 2\sqrt{gh_R} = u_3 - 2\sqrt{gh_3} \tag{4.46}$$

Additionally, as a transonic rarefaction determines the transition from subcritical (**U**$_2$) to supercritical (**U**$_R$) states, **U**$_3$ is assumed as the critical state and for the present example, the following relation holds:

$$u_3 = -\sqrt{gh_3} \tag{4.47}$$

Therefore, state **U**$_3$ can be computed from Equations (4.46)-(4.47).

Finally, state **U**$_2$ is calculated by applying the mass and momentum equations holding across the porosity discontinuity between states **U**$_2$ and **U**$_3$:

$$\begin{cases} \phi_L u_2 h_2 = \phi_R u_3 h_3 \\ \phi_L {u_2}^2 h_2 + \frac{1}{2} g \phi_L h_2^2 - \phi_R {u_3}^2 h_3 - \frac{1}{2} g \phi_R h_3^2 = \frac{1}{2} g (\phi_L - \phi_R) h_2^2 \end{cases} \tag{4.48}$$

Obviously, the same procedure can be applied if a transonic rarefaction is present across the $\tilde{\lambda}_1$ eigenvalue.

This simple procedure allows overcoming the problem of negative depths and obtaining approximate values for the intermediate states, necessary for the entropy fix step without requiring the solution of a non-linear system. This is a fundamental feature to preserve computational efficiency particularly on parallel architecture such as the GPUs (Vacondio et al., 2017).

### Flux computation

The split eigenvalues resulted from the entropy fix procedure, together with the eigenvectors and wave strengths deriving from the Roe solver, are finally adopted for calculating the fluxes $\boldsymbol{F}_{i+\frac{1}{2}}, \boldsymbol{F}_{i-\frac{1}{2}}$ in the updating formula of Equation (4.22) as (Roe, 1981; LeVeque, 2002; Murillo and Navas-Montilla, 2016):

$$\mathbf{F}_{i-\frac{1}{2}} = \sum_{p=1}^{m} \left[ \left(\lambda_p\right)_{i-\frac{1}{2}} \right]^{+} \left(\tilde{\alpha}_p\right)_{i-\frac{1}{2}} \left(\widetilde{\mathbf{R}}_p\right)_{i-\frac{1}{2}} \tag{4.49}$$

$$\mathbf{F}_{i+\frac{1}{2}} = -\sum_{p=1}^{m} \left[ \left(\lambda_p\right)_{i+\frac{1}{2}} \right]^{-} \left(\tilde{\alpha}_p\right)_{i+\frac{1}{2}} \left(\widetilde{\mathbf{R}}_p\right)_{i+\frac{1}{2}} \tag{4.50}$$

The splitting of the transonic eigenvalue allows increasing the numerical viscosity in its characteristic field, avoiding the entropy violation, which would occur if the no-split eigenvalue close to zero was used in Equations (4.49)-(4.50), determining a too small viscous term in the flux (LeVeque, 2002).

A flow chart representing the overall procedure of this numerical model based on the Roe solver for porosity discontinuities is shown in Figure 4.5. It is relevant to notice that the computation of the intermediate states can be seen as a preparatory step for the entropy fix, as they do not directly appear in the flux definition of Equations (4.49)-(4.50).

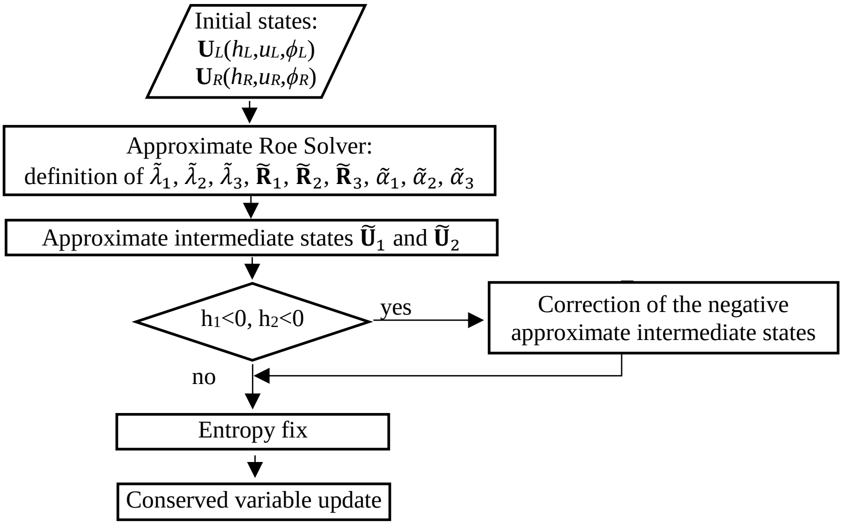


*Figure 4.5. Flow chart representing the numerical Roe solver for porosity discontinuities.*

## 4.4. Exact Riemann Solver

As illustrated before, the solution of the augmented SWE system with porosity consists of two genuinely non-linear characteristic fields associated with the eigenvalues $\tilde{\lambda}_1$ and $\tilde{\lambda}_3$, and a degenerate field due to the null eigenvalue $\tilde{\lambda}_2$, which is associated with a contact wave.

As an example, Figure 4.6 depicts the scheme of the generalized Stoker problem characterized by two initial states $\mathbf{U}_L(h_L,u_L,\phi_L)$ and $\mathbf{U}_R(h_R,u_R,\phi_R)$, with $u_L = u_R = 0$ m/s, $h_L > h_R$ and $\phi_L > \phi_R$. The presence of a porosity discontinuity located at $x=x_0$ determines two intermediate constant states $\mathbf{U}_1$ and $\mathbf{U}_2$, to the left and to the right of the contact wave, respectively (Figure 4.6-a). The water depth profile (Figure 4.6-b) is characterized by a rarefaction wave, which is associated with the eigenvalue $\tilde{\lambda}_1$ and connects the initial state $\mathbf{U}_L$ to the intermediate constant state $\mathbf{U}_1$, and a shock wave across the eigenvalue $\tilde{\lambda}_3$, which connects the initial state $\mathbf{U}_R$ with the intermediate constant state $\mathbf{U}_2$.

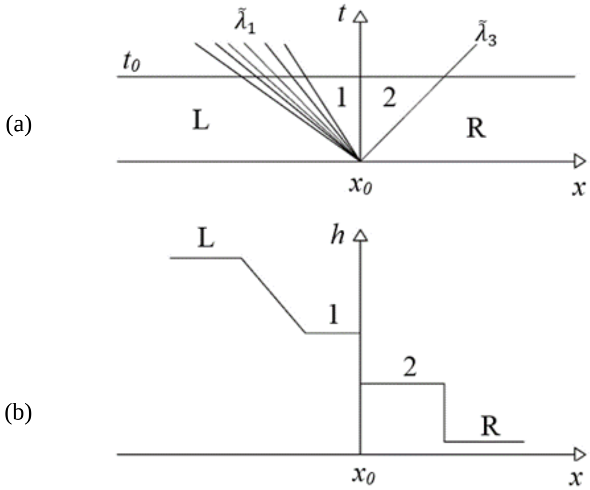


*Figure 4.6. Example of the Stoker test case: wave pattern (a) and water depth profile (b).*

Obviously, different initial states $\mathbf{U}_L$ and $\mathbf{U}_R$ lead to different types of waves. The purpose of the Exact Riemann Solver presented in this section is the definition of the intermediate constant states, which are connected to the initial states $\mathbf{U}_L$ and $\mathbf{U}_R$ through the two non-null eigenvalues. As only the waves developing from genuinely non-linear characteristic fields can be either shocks or rarefactions, the investigation of the wave pattern of a given Riemann problem is restricted to $\lambda_1$ and $\lambda_3$ eigenvalues.

For a given Riemann problem, the Exact Riemann Solver consists of two steps.

### a) Definition of the wave structure

The first step concerns the definition of the wave structure and the computation of the approximate values of the intermediate states.

If no transonic waves are present, the unknowns of the problem are the two intermediate states $\mathbf{U}_1$ and $\mathbf{U}_2$ and their approximate values are calculated according to Equations (4.37)-(4.38). Then the wave type is identified recalling that a shock wave occurs across the *p*-eigenvalue if the following relation holds (Toro, 2001):

$$\left(\lambda_p\right)_L > \left(\lambda_p\right)_R \tag{4.51}$$

Similarly, a rarefaction wave across the *p*-eigenvalue satisfies the formula:

$$\left(\lambda_p\right)_L < \left(\lambda_p\right)_R \tag{4.52}$$

Depending on these results, four different cases are possible: 2 shocks, shock – rarefaction, rarefaction – shock, or 2 rarefactions.

The approximate intermediate states calculated in this step have to be good initial values for the iterative procedure that will be described at step b). Particularly, if the water depth and velocity values are defined according to Equations (4.37)-(4.38), when a shock wave occurs, the related shock speed $s_1$ or $s_2$ is added to the list of the unknown variables and its initial value is defined according to the relations proposed by Toro (2001) for SWEs and here modified for the 1D-SWE with porosity.

**Approximate initial values for shock speeds $s_1$ and $s_2$**

Consider for example a shock wave, moving with speed $s_1$ and separating two states $\mathbf{U}_L(h_L,u_L,\phi_L)$ and $\mathbf{U}_1(h_1,u_1,\phi_1)$, which are to the left and to the right of the wave, respectively. In a reference frame moving with shock speed $s_1$, the velocities of the two states are defined as follows:

$$\hat{u}_L = u_L - s_1 \qquad \hat{u}_1 = u_1 - s_1 \tag{4.53}$$

and thus the vectors of the conserved variables for states $\hat{\mathbf{U}}_L$ and $\hat{\mathbf{U}}_1$ become:

$$\hat{\mathbf{U}}_L = \begin{bmatrix} \phi_L h_L \\ \phi_L \hat{u}_L h_L \\ \phi_L \end{bmatrix} \qquad \hat{\mathbf{U}}_1 = \begin{bmatrix} \phi_1 h_1 \\ \phi_1 \hat{u}_1 h_1 \\ \phi_1 \end{bmatrix} \tag{4.54}$$

In the new reference frame, the Rankine-Hugoniot conditions are defined as:

$$\begin{cases} \phi_1 \hat{u}_1 h_1 = \phi_L \hat{u}_L h_L \\ \phi_1 \hat{u}_1^2 h_1 + \dfrac{1}{2} g \phi_1 h_1^2 = \phi_L \hat{u}_L^2 h_L + \dfrac{1}{2} g \phi_L h_L^2 \end{cases} \tag{4.55}$$

Introducing the term $M_L = \phi_1 \hat{u}_1 h_1 = \phi_L \hat{u}_L h_L$, and substituting it in the second Equation of (4.55), after simple manipulations the following expression is obtained:

$$M_L = \sqrt{\frac{1}{2} g \phi_1 h_1 \phi_L h_L \frac{\phi_L h_L^2 - \phi_1 h_1^2}{\phi_L h_L - \phi_1 h_1}} \tag{4.56}$$

and the velocity $\hat{u}_L$ directly derives from the definition of $M_L$:

$$\hat{u}_L = \frac{M_L}{\phi_L h_L} \tag{4.57}$$

As the states $\mathbf{U}_L(h_L,u_L,\phi_L)$ and $\mathbf{U}_1(h_1,u_1,\phi_1)$ are on the same side of the discontinuity, the porosity values $\phi_L$ and $\phi_1$ coincide. Hence, substituting Equation (4.56)-(4.57) in (4.53), after some algebra the shock speed $s_1$ is defined as:

$$s_1 = u_L - \sqrt{g h_L} \sqrt{\frac{1}{2} \frac{(h_L + h_1) h_1}{h_L^2}} \tag{4.58}$$

The same procedure can be applied to a shock wave moving with speed $s_2$, and separating two states $\mathbf{U}_2(h_2,u_2,\phi_2)$ and $\mathbf{U}_R(h_R,u_R,\phi_R)$ to the left and to the right of the wave, respectively. The resulting shock speed $s_2$ is:

$$s_2 = u_R + \sqrt{gh_R}\sqrt{\frac{1}{2}\frac{(h_2 + h_R)h_2}{h_R^2}} \tag{4.59}$$

### b) Computation of the exact values

The exact values of the intermediate constant states are calculated by solving a non-linear system. Recalling that at the end of step a) the wave pattern is defined, the Riemann Invariants (RI) are used when a rarefaction wave occurs, while the Rankine-Hugoniot conditions (RH) are written for shock waves. Moreover, the presence of the porosity discontinuity requires the application of the mass and momentum balance for the states located immediately to the left and to the right of the porosity step (Equation (4.21)).

When either $\lambda_1$ or $\lambda_3$ corresponds to a transonic wave, then a third unknown state $\mathbf{U}_3$ has to be calculated, therefore Equations (4.46)-(4.47) are also included in order to close the system.

For sake of clarity, two examples of non-linear systems are illustrated.

The generalized Stoker test case in Figure 4.6 (with a rarefaction and a shock wave) presents five unknowns: $h_1$, $u_1$, $h_2$, $u_2$, $s_2$, where $s_2$ is the shock speed connecting states $\mathbf{U}_2$ and $\mathbf{U}_R$. According to the wave structure, the system of equations to be solved is constituted by the Riemann Invariant for the left rarefaction connecting states $\mathbf{U}_L$ and $\mathbf{U}_1$ (4.60), the mass (4.61) and momentum (4.62) equations across the porosity discontinuity between states $\mathbf{U}_1$ and $\mathbf{U}_2$, and the Rankine-Hugoniot conditions (4.63)-(4.64) for the right moving shock connecting states $\mathbf{U}_2$ and $\mathbf{U}_R$:

$$\begin{cases} u_L + 2\sqrt{gh_L} = u_1 + 2\sqrt{gh_1} & (4.60) \\ u_1 h_1 \phi_L = u_2 h_2 \phi_R & (4.61) \\ u_1^2 h_1 \phi_L + \frac{1}{2} g h_1^2 \phi_L - u_2^2 h_2 \phi_R - \frac{1}{2} g h_2^2 \phi_R = P_r & (4.62) \\ u_R h_R \phi_R - u_2 h_2 \phi_R = s_2 (h_R \phi_R - h_2 \phi_R) & (4.63) \\ u_R^2 h_R \phi_R + \frac{1}{2} g h_R^2 \phi_R - u_2^2 h_2 \phi_R - \frac{1}{2} g h_2^2 \phi_R = s_2 (u_R h_R \phi_R - u_2 h_2 \phi_R) & (4.64) \end{cases}$$

The second example concerns a transonic case, where both waves associated with $\lambda_1$ and $\lambda_3$ are rarefactions, and the latter is transonic (Figure 4.4-b). This problem presents six unknowns: $h_1$, $u_1$, $h_2$, $u_2$, $h_3$, $u_3$. According to the wave structure, the system of equations includes the Riemann Invariants for the left rarefaction connecting states $\mathbf{U}_L$ and $\mathbf{U}_1$ (4.65) and for the right rarefaction connecting states $\mathbf{U}_1$ and $\mathbf{U}_2$ (4.66), the mass (4.67) and momentum (4.68) equations across the porosity discontinuity between states $\mathbf{U}_2$ and $\mathbf{U}_3$, the passage through the critical state in $\mathbf{U}_3$ (4.69), and the Riemann Invariant for the right rarefaction connecting states $\mathbf{U}_3$ and $\mathbf{U}_R$ (4.70):

$$\begin{cases} u_L + 2\sqrt{gh_L} = u_1 + 2\sqrt{gh_1} & (4.65) \\ u_1 - 2\sqrt{gh_1} = u_2 - 2\sqrt{gh_2} & (4.66) \\ u_2 h_2 \phi_L = u_3 h_3 \phi_R & (4.67) \\ u_2^2 h_2 \phi_L + \frac{1}{2} g h_2^2 \phi_L - u_3^2 h_3 \phi_R - \frac{1}{2} g h_3^2 \phi_R = P_r & (4.68) \\ u_3 = -\sqrt{gh_3} & (4.69) \\ u_R - 2\sqrt{gh_R} = u_3 - 2\sqrt{gh_3} & (4.70) \end{cases}$$

The choice of good initial values is crucial to obtain the correct solution of the non-linear system of equations by means of the Newton-Raphson iterative procedure. The intermediate state values resulted at step a) are usually a good first guess for the exact ones, because the Roe solver already takes into account the presence of a porosity discontinuity.

Finally, once the constant intermediate states are obtained, the exact solution can be calculated at any arbitrary time *t*, following the procedure proposed by Toro (2001).

## 4.5. 1D numerical tests

The capability of the numerical model of handling different wave configurations is tested in this section by means of a selection of 1D Riemann problems summarized in Table 4.1; the numerical and exact solutions are compared in terms of water depth and velocity. The horizontal and frictionless domain was discretized with cells of size $\Delta x$ = 0.1 m and the discontinuity in the initial values (water depth, velocity, porosity) was set at $x_0$ = 2000 m. The gravitational acceleration was set as $g$ = 9.81 m/s$^2$, and the Courant number $Cr$ was assumed equal to 0.8. The initial conditions of the simulated test cases (Table 4.1) lead to, respectively, Rarefaction-Discontinuity-Shock (RDS), Rarefaction-Discontinuity-Rarefaction (RDR), Shock-Discontinuity-Shock (SDS), Rarefaction-Rarefaction-Discontinuity-Rarefaction (RRDR), and Shock-Discontinuity-Rarefaction-Shock (SDRS).

*Table 4.1. Initial conditions for the 1D test cases.*

| *Test number* | *Test case ID* | $h_L$ (m) | $u_L$ (m/s) | $\phi_L$ (-) | $h_R$ (m) | $u_R$ (m/s) | $\phi_R$ (-) |
|---|---|---|---|---|---|---|---|
| *1* | *RDS* | 8 | 0 | 1 | 3 | 0 | 0.5 |
| *2* | *RDR* | 8 | -4 | 0.9 | 6 | 0 | 0.5 |
| *3* | *SDS* | 4 | 4 | 0.9 | 1.5 | -1.5 | 0.5 |
| *4* | *RRDR* | 6 | -18 | 0.9 | 15 | 0 | 0.7 |
| *5* | *SDRS* | 4 | 7 | 0.9 | 1 | 7 | 0.7 |

## Rarefaction-Discontinuity-Shock test

The Rarefaction-Discontinuity-Shock (RDS) test (Test 1, Table 4.1) is a generalized dam-break case (Stoker solution), whose solution is characterized by the presence of a stationary step in $x_0$, from which a left going rarefaction wave and a right moving shock wave depart.

The RDS exact solver, which presents 5 unknowns ($h_1$, $u_1$, $h_2$, $u_2$, $s_2$), requires the solution of a five-equation system with the following relations:

- RI across the left rarefaction connecting states $\mathbf{U}_L$ and $\mathbf{U}_1$ (one equation);
- Discontinuity equations between states $\mathbf{U}_1$ and $\mathbf{U}_2$ (two equations);
- RH relations across the right shock connecting states $\mathbf{U}_2$ and $\mathbf{U}_R$ (two equations).

As can be seen in Figure 4.7, numerical water depth (a) and velocity (b) obtained through the augmented solver are in very good agreement with the exact one: the intermediate states are correctly approximated, and the shock position is well captured.

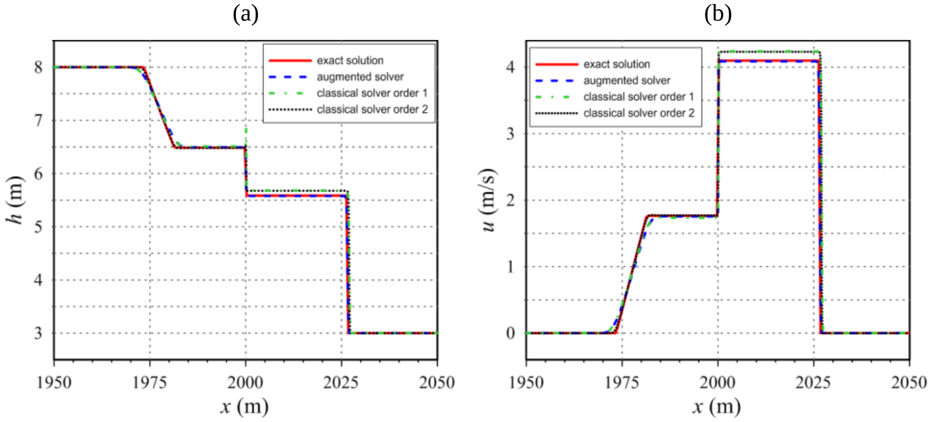


*Figure 4.7. RDS test: augmented solver for porosity discontinuities vs classical solvers (first and second order): water depth (a) and velocity (b) comparison between exact and numerical solutions at time 3 s.*

In the same figure, the results obtained by means of two classical Riemann solvers for SWE, based on first and second order numerical schemes, and discretizing the porosity source term with a central finite difference formulation, are presented for comparison. The results demonstrate that the classical solvers (both first and second order) are not able to capture the contact wave due to the porosity step well, either in terms of water depth or velocity, with an overestimation of state $\mathbf{U}_2$ (Figure 4.7).

## Rarefaction-Discontinuity-Rarefaction test

The Rarefaction-Discontinuity-Rarefaction (RDR) test (Test 2, Table 4.1) simulates a divergent flow characterized by the presence of two rarefactions, one moving to the left and the other to the right, and a contact wave in the discontinuity.

The RDR exact solver presents 4 unknowns ($h_1$, $u_1$, $h_2$, $u_2$) and requires the solution of a four-equation system with the following relations:

- RI across the left rarefaction connecting states $\mathbf{U}_L$ and $\mathbf{U}_1$ (one equation);
- Discontinuity equations between states $\mathbf{U}_1$ and $\mathbf{U}_2$ (two equations);
- RI across the right rarefaction connecting states $\mathbf{U}_2$ and $\mathbf{U}_R$ (one equation).

Figure 4.8 illustrates that the numerical solution well approximates the exact one, by capturing the wave pattern and the intermediate state values, for both water depth (a) and velocity (b).

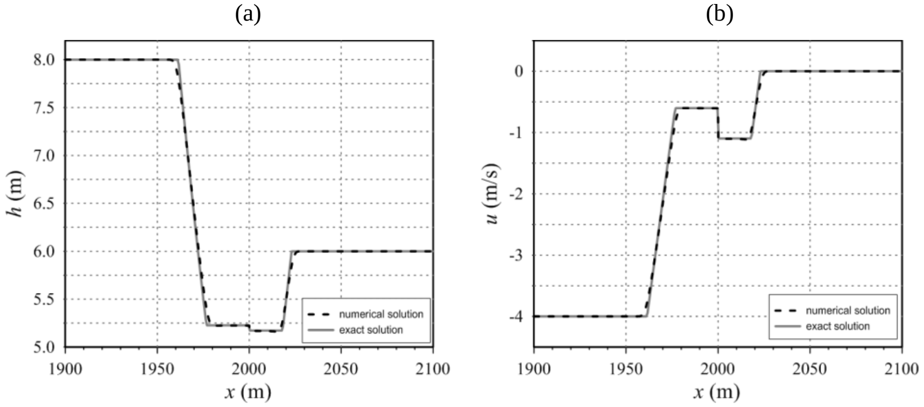

(a) (b)

*Figure 4.8. RDR test: water depth (a) and velocity (b) comparison between exact and numerical solutions at time 3 s.*

## Shock-Discontinuity-Shock test

The Shock-Discontinuity-Shock (SDS) test (Test 3, Table 4.1) simulates a convergent flow, in which two shocks, one going to the left and the other to the right, depart from the porosity discontinuity, where a stationary wave is generated.

The SDS exact solver, which presents 6 unknowns ($h_1$, $u_1$, $s_1$, $h_2$, $u_2$, $s_2$), requires the solution of a six-equation system with the following relations:

- RH relations across the left shock connecting states $\mathbf{U}_L$ and $\mathbf{U}_1$ (two equations);
- Discontinuity equations between states $\mathbf{U}_1$ and $\mathbf{U}_2$ (two equations);
- RH relations across the right shock connecting states $\mathbf{U}_2$ and $\mathbf{U}_R$ (two equations).

As shown in Figure 4.9, the numerical solution is almost indistinguishable from the exact one.

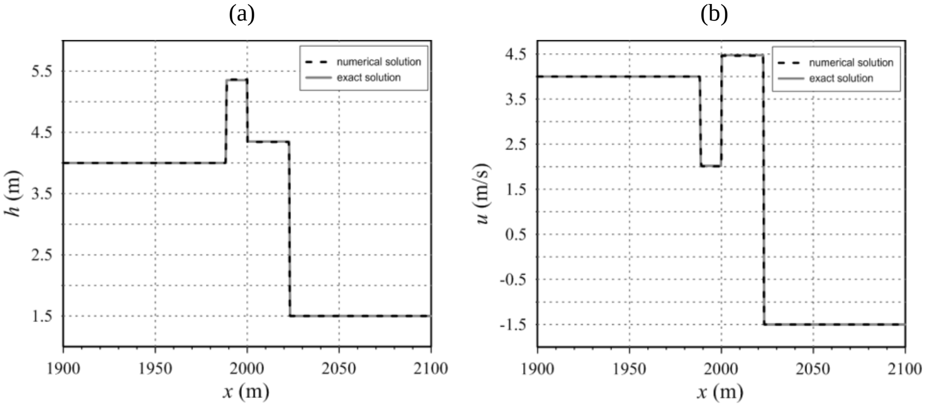


*Figure 4.9. SDS test: water depth (a) and velocity (b) comparison between exact and numerical solutions at time 3 s.*

### Rarefaction-Rarefaction-Discontinuity-Rarefaction test

The Rarefaction-Rarefaction-Discontinuity-Rarefaction (RRDR) Riemann problem (Test 4, Table 4.1) is a transonic test, which differs from the non-transonic RDR one, because of the presence of two rarefactions on the left side of the discontinuity, one going to the left and the other to the right. The wave pattern is then completed with the porosity discontinuity contact wave and a right moving rarefaction that departs at the right side of the discontinuity.

The RRDR exact solver presents 6 unknowns ($h_1$, $u_1$, $h_2$, $u_2$, $h_3$, $u_3$) and requires the solution of a six-equation system with the following relations:

- RI across the left rarefaction connecting states $\mathbf{U}_L$ and $\mathbf{U}_1$ (one equation);
- RI across the right rarefaction connecting states $\mathbf{U}_1$ and $\mathbf{U}_2$ (one equation);
- Discontinuity equations between states $\mathbf{U}_2$ and $\mathbf{U}_3$ (two equations);
- Passage through the critical state in $\mathbf{U}_3$ (one equation);
- RI across the right rarefaction connecting states $\mathbf{U}_3$ and $\mathbf{U}_R$ (one equation).

Figure 4.10 illustrates that the numerical solution favorably agrees with the exact one: the three rarefactions are correctly captured as well as the intermediate states $\mathbf{U}_1$, $\mathbf{U}_2$

and $\mathbf{U}_3$. It is worth noting that, whereas states $\mathbf{U}_1$ and $\mathbf{U}_2$ are easily distinguishable, because of the development in the *x*-direction of their two distinct constant values, the critical state $\mathbf{U}_3$ consists in a single point at the right side of the discontinuity.

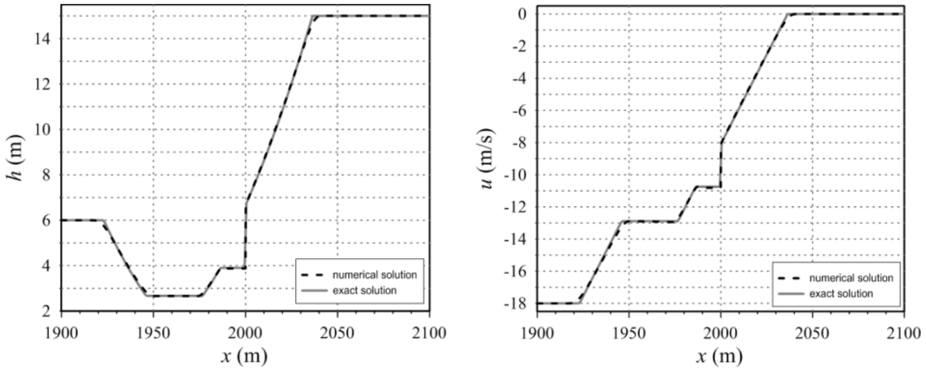

(a) (b)

*Figure 4.10. RRDR test: water depth (a) and velocity (b) comparison between exact and numerical solutions at time 3 s.*

This interesting Riemann problem allowed testing two particular aspects of the numerical model. Firstly, as the test is transonic across the $\tilde{\lambda}_3$ eigenvalue, the entropy fix procedure is necessary in order to obtain good results. Moreover, the approximate value $\tilde{h}_2$ is negative and thus the procedure illustrated in Section 4.3 is fundamental for correcting the initial approximate intermediate states.

## Shock-Discontinuity-Rarefaction-Shock test

The Shock-Discontinuity-Rarefaction-Shock (SDRS) Riemann problem (Test 5, Table 4.1) is a transonic test, which differently from SDS presents a left going rarefaction and a right moving shock wave on the right side of the discontinuity. On the left side of the discontinuity, where a contact wave can be observed, a shock wave travels to the left.

The SDRS exact solver, which presents 8 unknowns ($h_1$, $u_1$, $s_1$, $h_2$, $u_2$, $h_3$, $u_3$, $s_3$), requires the solution of an eight-equation system with the following relations:

- RH relations across the left shock connecting states $\mathbf{U}_L$ and $\mathbf{U}_1$ (two equations);

- Discontinuity equations between states $\mathbf{U}_1$ and $\mathbf{U}_2$ (two equations);
- Passage through the critical state in $\mathbf{U}_2$ (one equation);
- RI across the left rarefaction connecting states $\mathbf{U}_2$ and $\mathbf{U}_3$ (one equation);
- RH relations across the right shock connecting states $\mathbf{U}_3$ and $\mathbf{U}_R$ (two equations);

This test is transonic across the $\tilde{\lambda}_1$ eigenvalue and thus, the entropy fix (see Section 4.3) is necessary to obtain good results: as shown in Figure 4.11, the numerical solution well agrees with the exact one, and the wave pattern is well captured. Analogously to the RRDR test, state $\mathbf{U}_2$ consists in a single point at the left side of the discontinuity.

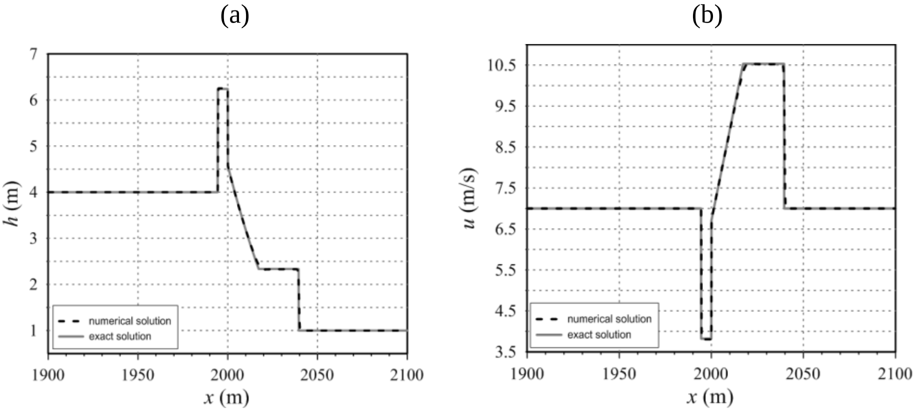


*Figure 4.11. SDRS test: water depth (a) and velocity (b) comparison between exact and numerical solutions at time 3 s.*

## 4.6. Extension to 2D SWEs

In this section the numerical scheme is extended to 2 dimensions: as details about the 1D derivation can be found in Sections 4.2 and 4.3, only the results and the relevant aspects of the 2D approach are presented.

### 4.6.1. Eigenstructure analysis

The augmented 2D SWEs with porosity in Equation (4.1) can be rewritten in the following form:

$$\frac{\partial \mathbf{U}}{\partial t}+\frac{\partial \mathbf{F}(\mathbf{U})}{\partial x}+\frac{\partial \mathbf{G}(\mathbf{U})}{\partial y}+\mathbf{H}\frac{\partial \mathbf{U}}{\partial x}+\mathbf{K}\frac{\partial \mathbf{U}}{\partial y}=0 \tag{4.71}$$

where the vector **U** of the conserved variables, the vectors **F(U)** and **G(U)** of the fluxes in $x$ and $y$ directions, respectively, and the matrixes **H** and **K** of the non-conservative fluxes due to the porosity variation in *x* and *y* directions, are defined as:

$$\mathbf{U}=\begin{bmatrix}\phi h\\ \phi uh\\ \phi vh\\ \phi\end{bmatrix}\qquad \mathbf{F}(\mathbf{U})=\begin{bmatrix}\phi uh\\ \phi u^2h+\frac{1}{2}g\phi h^2\\ \phi uvh\\ 0\end{bmatrix}\qquad \mathbf{G}(\mathbf{U})=\begin{bmatrix}\phi vh\\ \phi uvh\\ \phi v^2h+\frac{1}{2}g\phi h^2\\ 0\end{bmatrix} \tag{4.72}$$

$$\mathbf{H}=-\frac{1}{2}gh^2\begin{bmatrix}0&0&0&0\\0&0&0&1\\0&0&0&0\\0&0&0&0\end{bmatrix}\qquad \mathbf{K}=-\frac{1}{2}gh^2\begin{bmatrix}0&0&0&0\\0&0&0&0\\0&0&0&1\\0&0&0&0\end{bmatrix}$$

Focusing on the $x$-direction and following the Generalized Roe approach illustrated in Section 4.3 the eigenvalues are:

$$\tilde{\lambda}_1=\tilde{u}-\sqrt{g\tilde{h}}\qquad \tilde{\lambda}_2=0\qquad \tilde{\lambda}_3=\tilde{u}\qquad \tilde{\lambda}_4=\tilde{u}+\sqrt{g\tilde{h}} \tag{4.73}$$

and the corresponding right eigenvectors result:

$$\tilde{\mathbf{R}}_1=\begin{bmatrix}1\\ \tilde{u}-\sqrt{g\tilde{h}}\\ \tilde{v}\\ 0\end{bmatrix}\qquad \tilde{\mathbf{R}}_2=\begin{bmatrix}a_{24}-\frac{1}{2}g\tilde{h}^2\\ 0\\ \tilde{v}\left(a_{24}-\frac{1}{2}g\tilde{h}^2\right)\\ \tilde{u}^2-g\tilde{h}\end{bmatrix}\qquad \tilde{\mathbf{R}}_3=\begin{bmatrix}0\\0\\1\\0\end{bmatrix}\qquad \tilde{\mathbf{R}}_4=\begin{bmatrix}1\\ \tilde{u}+\sqrt{g\tilde{h}}\\ \tilde{v}\\ 0\end{bmatrix} \tag{4.74}$$

where the term $a_{24}$ is defined, analogously to the $a_{23}$ term for the 1D scheme (Eq. 4.30), as:

$$a_{24}=-\frac{1}{2}g\frac{|\phi_L-\phi_R|}{\phi_R-\phi_L}|h_j|h_j\quad \text{where}\quad \begin{cases}h_j=-h_L & \text{if}\quad \phi_L\geq\phi_R\\ h_j=h_R & \text{if}\quad \phi_L<\phi_R\end{cases} \tag{4.75}$$

The Roe averages $\tilde{h}$, $\tilde{u}$ are defined according to Equation (4.31) and $\tilde{v}$ is calculated as:

$$\tilde{v} = \frac{\sqrt{\phi_L h_L} v_L + \sqrt{\phi_R h_R} v_R}{\sqrt{\phi_L h_L} + \sqrt{\phi_R h_R}} \tag{4.76}$$

The vector of the conserved variable differences $\boldsymbol{\delta}$ is evaluated as:

$$\boldsymbol{\delta} = \begin{bmatrix} \delta_1 \\ \delta_2 \\ \delta_3 \\ \delta_4 \end{bmatrix} = \begin{bmatrix} \phi_R h_R - \phi_L h_L \\ \phi_R u_R h_R - \phi_L u_L h_L \\ \phi_R v_R h_R - \phi_L v_L h_L \\ \phi_R - \phi_L \end{bmatrix} \tag{4.77}$$

and finally the wave strengths result:

$$\tilde{\alpha}_1 = -\frac{\tilde{\lambda}_4}{\tilde{\lambda}_1 - \tilde{\lambda}_4}\delta_1 + \frac{1}{\tilde{\lambda}_1 - \tilde{\lambda}_4}\delta_2 + \frac{2a_{24} - g\tilde{h}^2}{2\tilde{\lambda}_1(\tilde{\lambda}_1 - \tilde{\lambda}_4)}\delta_4 \tag{4.78}$$

$$\tilde{\alpha}_2 = \frac{1}{\tilde{\lambda}_1 \tilde{\lambda}_4}\delta_4 \tag{4.79}$$

$$\tilde{\alpha}_3 = -\tilde{v}\delta_1 + \delta_3 \tag{4.80}$$

$$\tilde{\alpha}_4 = \frac{\tilde{\lambda}_1}{\tilde{\lambda}_1 - \tilde{\lambda}_4}\delta_1 - \frac{1}{\tilde{\lambda}_1 - \tilde{\lambda}_4}\delta_2 - \frac{2a_{24} - g\tilde{h}^2}{2\tilde{\lambda}_4(\tilde{\lambda}_1 - \tilde{\lambda}_4)}\delta_4 \tag{4.81}$$

Adopting the same procedure for the *y*-direction, the eigenvalues result:

$$\tilde{\lambda}_1 = \tilde{v} - \sqrt{g\tilde{h}} \qquad \tilde{\lambda}_2 = 0 \qquad \tilde{\lambda}_3 = \tilde{v} \qquad \tilde{\lambda}_4 = \tilde{v} + \sqrt{g\tilde{h}} \tag{4.82}$$

and the corresponding right eigenvectors read:

$$\tilde{\mathbf{R}}_1 = \begin{bmatrix} 1 \\ \tilde{u} \\ \tilde{v} - \sqrt{g\tilde{h}} \\ 0 \end{bmatrix} \quad \tilde{\mathbf{R}}_2 = \begin{bmatrix} a_{24} - \frac{1}{2}g\tilde{h}^2 \\ \tilde{u}\left(a_{24} - \frac{1}{2}g\tilde{h}^2\right) \\ 0 \\ \tilde{v}^2 - g\tilde{h} \end{bmatrix} \quad \tilde{\mathbf{R}}_3 = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \quad \tilde{\mathbf{R}}_4 = \begin{bmatrix} 1 \\ \tilde{u} \\ \tilde{v} + \sqrt{g\tilde{h}} \\ 0 \end{bmatrix} \tag{4.83}$$

Assuming the vector of the conserved variable differences $\boldsymbol{\delta}$ as in Equation (4.77), the wave strengths result:

$$\tilde{\alpha}_1 = -\frac{\tilde{\lambda}_4}{\tilde{\lambda}_1 - \tilde{\lambda}_4}\delta_1 + \frac{1}{\tilde{\lambda}_1 - \tilde{\lambda}_4}\delta_3 + \frac{2a_{24} - g\tilde{h}^2}{2\tilde{\lambda}_1(\tilde{\lambda}_1 - \tilde{\lambda}_4)}\delta_4 \tag{4.84}$$

$$\tilde{\alpha}_2 = \frac{1}{\tilde{\lambda}_1 \tilde{\lambda}_4} \delta_4 \tag{4.85}$$

$$\tilde{\alpha}_3 = -\tilde{u}\delta_1 + \delta_2 \tag{4.86}$$

$$\tilde{\alpha}_4 = \frac{\tilde{\lambda}_1}{\tilde{\lambda}_1 - \tilde{\lambda}_4} \delta_1 - \frac{1}{\tilde{\lambda}_1 - \tilde{\lambda}_4} \delta_3 - \frac{2a_{24} - g\tilde{h}^2}{2\tilde{\lambda}_4 \left(\tilde{\lambda}_1 - \tilde{\lambda}_4\right)} \delta_4 \tag{4.87}$$

The 2D augmented SWE system with porosity is characterized by real and distinct eigenvalues, and thus the system is still hyperbolic. Moreover, the characteristic fields $\lambda_1$ and $\lambda_4$ are genuinely non-linear, while the characteristic fields $\lambda_2$ and $\lambda_3$ are linearly degenerate. Consequently, the waves that can develop from the eigenvalues $\lambda_1$ and $\lambda_4$ can be shocks or rarefactions, while the wave associated with the eigenvalue $\lambda_2$ is a contact wave and the one related to eigenvalue $\lambda_3$ is a shear wave.

### 4.6.2. 2D Approximate Riemann Solver

The 2D numerical model adopts a Cartesian mesh with grid size $\Delta x$ and $\Delta y$, along $x$ and $y$ directions, respectively, and the first order approximation explicit finite volume formula for updating the conserved variables from time $t^n$ to $t^{n+1}$ states:

$$\mathbf{U}_{i,j}^{n+1} = \mathbf{U}_{i,j}^{n} - \frac{\Delta t}{\Delta x}\left[\mathbf{F}_{i+\frac{1}{2},j} - \mathbf{F}_{i-\frac{1}{2},j}\right] - \frac{\Delta t}{\Delta \mathrm{y}}\left[\mathbf{G}_{i,j+\frac{1}{2}} - \mathbf{G}_{i,j-\frac{1}{2}}\right] \tag{4.88}$$

where $i$ and $j$ are the cell center coordinates in $x$ and $y$ directions.

According to Equation (4.88), the numerical fluxes **F** and **G** can be calculated separately, distinguishing between $x$ and $y$ directions: therefore, the 2D model can be seen as the combination of two 1D models, one in the $x$-direction and the other in the $y$-direction. On that basis, the procedure for the computation of numerical fluxes **F** is illustrated for the $x$-axis: the same method can then be adopted for fluxes **G** in the $y$-axis.

For a given Riemann problem with two initial states $\mathbf{U}_L(h_L,u_L,v_L,\phi_L)$ and $\mathbf{U}_R(h_R,u_R,v_R,\phi_R)$, the Roe solver determines the averaged eigenvalues ($\tilde{\lambda}_1$, $\tilde{\lambda}_2$, $\tilde{\lambda}_3$, $\tilde{\lambda}_4$), eigenvectors ($\tilde{\mathbf{R}}_1$, $\tilde{\mathbf{R}}_2$, $\tilde{\mathbf{R}}_3$, $\tilde{\mathbf{R}}_4$) and wave strengths ($\tilde{\alpha}_1$, $\tilde{\alpha}_2$, $\tilde{\alpha}_3$, $\tilde{\alpha}_4$).

According to the eigenvalue signs, four different configurations may occur (Figure 4.12): subcritical with positive $\tilde{\lambda}_3$ (a), subcritical with negative $\tilde{\lambda}_3$ (b), supercritical with positive velocities (c), and supercritical with negative velocities (d).

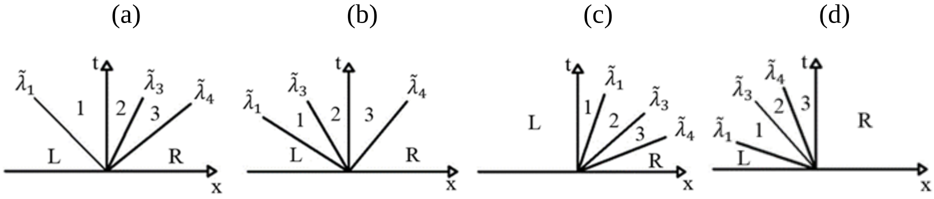


*Figure 4.12. 2D eigenvalue configurations: subcritical with positive $\tilde{\lambda}_3$ (a), subcritical with negative $\tilde{\lambda}_3$ (b), supercritical with positive (c) and negative (d) velocities.*

The identification of the wave configuration is necessary in order to calculate the approximate intermediate states $\widetilde{\mathbf{U}}_1$, $\widetilde{\mathbf{U}}_2$ and $\widetilde{\mathbf{U}}_3$ to be used in the entropy fix procedure. However, due to the fact that the splitting in the entropy fix concerns eigenvalues $\tilde{\lambda}_1$ or $\tilde{\lambda}_4$, and the procedure is illustrated for the *x*-direction, only the *u* velocity is involved. Moreover, the tangential velocity *v* changes across the $\tilde{\lambda}_3$ contact wave, but remains constant across the waves associated with the eigenvalues $\tilde{\lambda}_1$, $\tilde{\lambda}_2$ and $\tilde{\lambda}_4$, therefore it is easily evaluated separately from the other physical quantities. Furthermore, the variables *h* and *u* do not change across the $\tilde{\lambda}_3$ contact wave. This means that, with reference to Figure 4.12, (1D) states $[\tilde{h}_2, \tilde{u}_2]$ and $[\tilde{h}_3, \tilde{u}_3]$ in (a) and (c), $[\tilde{h}_1, \tilde{u}_1]$ and $[\tilde{h}_2, \tilde{u}_2]$ in (b) and (d) coincide. Therefore, the four configurations reduce to the three cases already presented for the 1D approach (see Figure 4.2), and the intermediate states can be calculated following exactly the same procedure described in Section 4.3.

A similar procedure can be followed for the *y*-direction considering *v* instead of *u*.

Finally, the positive and negative split eigenvalues, together with the averaged wave strengths and eigenvectors, are used to compute the fluxes $\mathbf{F}_{i+\frac{1}{2},j}$, $\mathbf{F}_{i-\frac{1}{2},j}$, $\mathbf{G}_{i,j+\frac{1}{2}}$, $\mathbf{G}_{i,j-\frac{1}{2}}$ as follows:

$$\mathbf{F}_{i-\frac{1}{2},j} = \sum_{p=1}^{m} \left[ \left(\lambda_p\right)_{i-\frac{1}{2},j} \right]^{+} \left(\tilde{\alpha}_p\right)_{i-\frac{1}{2},j} \left(\widetilde{\mathbf{R}}_p\right)_{i-\frac{1}{2},j} \tag{4.89}$$

$$\mathbf{F}_{i+\frac{1}{2},j} = -\sum_{p=1}^{m} \left[ \left(\lambda_p\right)_{i+\frac{1}{2},j} \right]^{-} \left(\tilde{\alpha}_p\right)_{i+\frac{1}{2},j} \left(\widetilde{\mathbf{R}}_p\right)_{i+\frac{1}{2},j} \tag{4.90}$$

$$\mathbf{G}_{i,j-\frac{1}{2}} = \sum_{p=1}^{m} \left[ \left(\lambda_p\right)_{i,j-\frac{1}{2}} \right]^{+} \left(\tilde{\alpha}_p\right)_{i,j-\frac{1}{2}} \left(\widetilde{\mathbf{R}}_p\right)_{i,j-\frac{1}{2}} \tag{4.91}$$

$$\mathbf{G}_{i,j+\frac{1}{2}} = -\sum_{p=1}^{m} \left[ \left(\lambda_p\right)_{i,j+\frac{1}{2}} \right]^{-} \left(\tilde{\alpha}_p\right)_{i,j+\frac{1}{2}} \left(\widetilde{\mathbf{R}}_p\right)_{i,j+\frac{1}{2}} \tag{4.92}$$

which are then substituted in Equation (4.88) in order to update the solution.

## 4.7. 2D numerical tests

In order to validate the 2D numerical model with wave configurations similar to those illustrated in Section 4.5, a selection of circular Riemann problems is here presented.

In Section 3.5 it has been demonstrated that due to the radial symmetry and the choice of a proper porosity field, the 2D SWEs with variable porosity $\phi$ are equivalent to the 1D SWEs with porosity $\phi_b(r)$. The adoption of a generic function $\phi_b(r)$ allows obtaining the reference solution of a 2D test case, by solving the 1D system (3.51) on a very fine mesh (Aureli et al., 2008). Whereas, the adoption of a porosity field defined according to Equation (3.50) and shown in Figure 4.13, allows obtaining the reference solution by means of the Exact 1D Riemann Solver described in Section 4.4.

The simulations were performed setting the gravitational acceleration as $g$ = 9.81 m/s$^2$ and assuming the Courant number for the time step computation as $Cr$ = 0.8. The horizontal and frictionless domains were discretized with cell sizes $\Delta x = \Delta y = 0.05$ m. The circular tests were modeled considering a dam with radius $r$ = 120 m, which separates the internal initial state $\boldsymbol{U}_{int}$ from the external state $\boldsymbol{U}_{ext}$ until its instantaneous collapse at time $t$ = 0 s. The porosity field adopted for all the simulations is shown in Figure 4.13: it gradually decreases according to Equation (3.50), and a discontinuity of 0.2 is introduced at $r$ = 120 m.

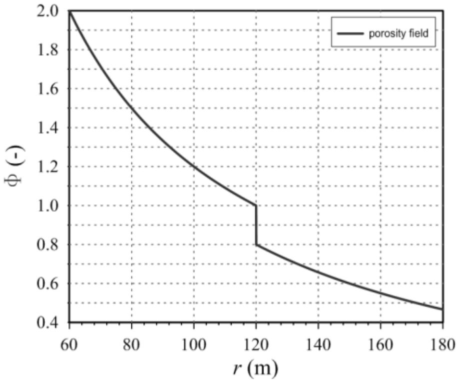


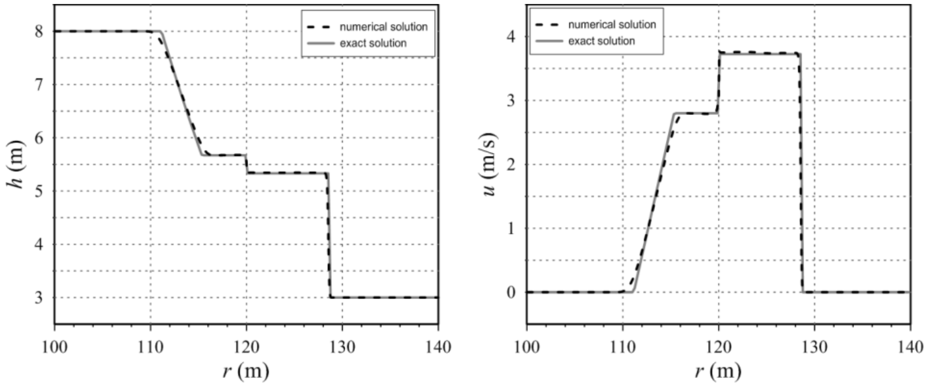
*Figure 4.13. Porosity field of the 2D radial tests.*

The initial conditions of the 2D test cases are summarized in Table 4.2.

*Table 4.2. Initial conditions for the radial test cases.*

| *Test case ID* | $h_{int}$ *(m)* | $u_{int}$ *(m/s)* | $h_{ext}$ *(m)* | $u_{ext}$ *(m/s)* |
|---|---|---|---|---|
| *RDS radial* | 8 | 0 | 3 | 0 |
| *RDR radial* | 8 | -4 | 6 | 0 |
| *SDS radial* | 4 | 4 | 1.5 | -1.5 |
| *RRDR radial* | 6 | -18 | 15 | 0 |
| *SDRS radial* | 4 | 7 | 1 | 7 |

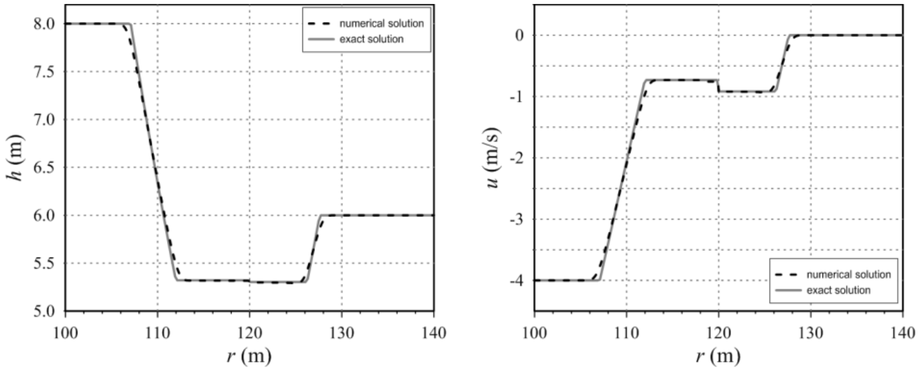
The numerical solutions of the circular Riemann problems, along $y = x$ line, are compared in terms of water depth and velocity with the exact solutions of the corresponding 1D scheme. A very good agreement between the numerical and the reference solution is obtained for the non-transonic test cases: RDS radial (Figure 4.14), RDR radial (Figure 4.15), and SDS radial (Figure 4.16).



(a)

(b)

*Figure 4.14. RDS radial test: water depth (a) and velocity (b) comparison between reference and numerical solutions at time 1 s.*



(a)

(b)

*Figure 4.15. RDR radial test: water depth (a) and velocity (b) comparison between reference and numerical solutions at time 1 s.*

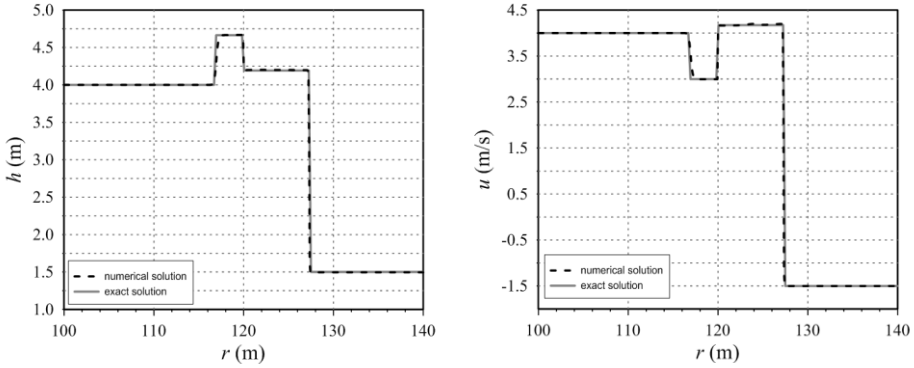

(a) (b)

*Figure 4.16. SDS radial test case: water depth (a) and velocity (b) comparison between reference and numerical solutions at time 1 s.*

The numerical solution favorably approximates the reference one, also in the presence of transonic waves, as shown for the RRDR radial (Figure 4.17) and SDRS radial (Figure 4.18) test cases.

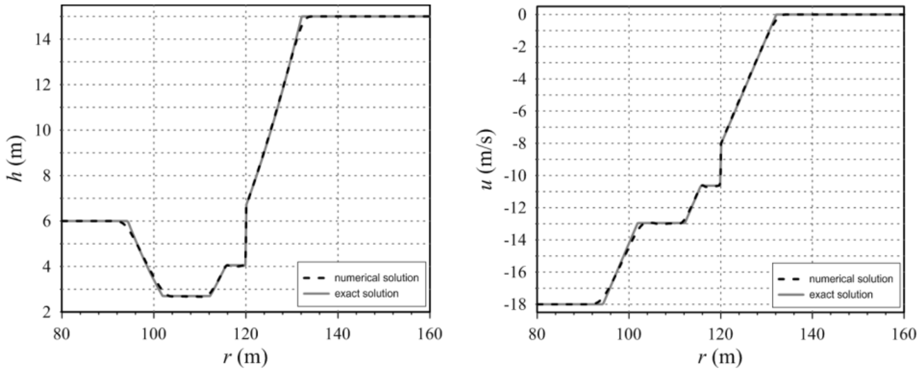

(a) (b)

*Figure 4.17. RRDR radial test case: water depth (a) and velocity (b) comparison between reference and numerical solutions at time 1 s.*

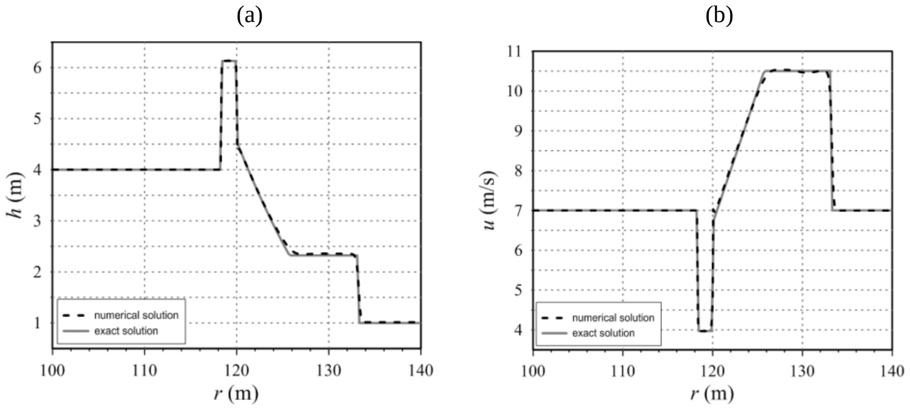


*Figure 4.18. SDRS radial test case: water depth (a) and velocity (b) comparison between reference and numerical solutions at time 1 s.*

## 4.8. Concluding remarks

This chapter focused on the development of a numerical method capable of handling discontinuities in the porosity field, in the framework of SWEs finite volume schemes. With this aim, the porosity source term has been included in the General Riemann Solver, and an augmented Roe Solver has been derived. The *sine qua non* to the Solver development has been the analysis of the physical meaning of a porosity discontinuity and the enforcement of mass and momentum conservation across the porosity discontinuity. It has also been demonstrated that the Riemann Invariants do not hold across the discontinuity, and only the Rankine-Hugoniot conditions can be considered. The implemented entropy fix procedure, necessary in a Roe Solver, has ensured accurate results also in presence of transonic waves, without requiring the solution of any non-linear system. For this reason, the proposed model is expected to exploit the efficiency of numerical models running on GPUs. A 1D Exact Riemann Solver has been derived in order to validate the 1D numerical model by means of several test cases. Moreover, it has been demonstrated that these exact solutions are valid also for 2D test cases characterized by radial symmetry, and by a proper porosity field. The test cases have shown that the proposed numerical procedure is general, as it is capable of capturing different Riemann Problems, involving both shocks, rarefactions and transonic rarefactions. Moreover, the results achieved with the augmented Solver better agree with the exact solution than the ones obtained through a classical solver. Future implications of the augmented Roe Solver can involve two branches: the first leads to the adoption of non-uniform grids, instead of the Cartesian ones here adopted, while the second concerns the modelling of real events involving both rural and urban areas.

# Chapter 5

# A parallel Bayesian methodology for estimating upstream hydrographs in ungauged sections

## 5.1. Introduction

The modelling of flooding events, either in terms of historical reconstructions, hypothetical scenarios for improving resilience, and future projections, requires, besides accurate, fast and stable numerical models, the use of proper input data. If bathymetrical information can be obtained from high-resolution Digital Terrain Models (DTM), and roughness coefficients can be calculated via calibration processes, the definition of the boundary conditions, and particularly the inflow hydrographs, remains a crucial issue.

This challenge derives from the fact that direct discharge measurement is difficult to carry out during flood events in natural rivers; therefore, discharge hydrographs are usually inferred from observed stage hydrographs by means of rating curves. However, this procedure presents several limitations. Firstly, rating curves are seldom well calibrated for high discharge values. Moreover, the stage-discharge relationship is not unique due to inertial and/or backwater effects. Furthermore, rating curves hold only for the considered gauging station and they are influenced by the river geometry, which actually changes from one season to the other and by natural (erosion/deposition) and/or artificial processes. Finally, water levels gauges can be damaged during extreme flooding events, consequently causing corruptions in the historical registrations.

Therefore, the knowledge of discharge hydrographs in ungauged river sections is still a relevant hydraulic problem not only, as mentioned before, for flood modelling purposes, but also for more practical issues related to flood protection measures, hydropower plants, water resource management, design of new structures, etc.

In the framework of numerical models, discharge hydrographs at a specific site can be evaluated by coupling rainfall-runoff and flood propagation schemes. However, rainfall-runoff models (Beven, 2011) present several uncertainties associated for example with the choice of the model for the basin schematization, with the evaluation of the effective rainfall, and with the calibration procedure.

A different method for defining the discharge hydrograph in an ungauged river section concerns the application of a reverse flow routing process: this numerical procedure allows evaluating the upstream flood wave starting from the knowledge of the downstream hydrograph and the hydraulic characteristics of the river reach.

It is noteworthy that the solution of an inverse problem presents three main challenges: the solution may not exist, or it may be non-unique, and during the computation instabilities in the solution may arise.

First attempts of solving the reverse flow routing problem for flood propagation were based on the solution of a reverse form of the Saint Venant equations. Following this approach, Eli et al. (1974) calculated the upstream discharge, once assumed as known quantities the discharge series at the end of the river and the initial conditions. Similar results were obtained by Szymkiewicz (1993) adopting an implicit scheme, under the hypothesis of subcritical flow conditions. However, in both works, the known information is assumed free of errors and instabilities in the discharge computation arise. For a given single shape of inflow hydrograph, Dooge and Bruen (2005) investigated the stability of the reverse routing problem for flood propagation and demonstrated that it is related to the adopted discretization parameters and to the channel bed slope.

Adopting a different approach, Das (2009) solved the reverse stream flow routing by adopting a reverse form of the Muskingum model and highlighted the importance of separately calibrating the Muskingum model parameters. However, in this work no errors corrupting the observations were considered. Furthermore, the Muskingum model provides a simplified flood routing based on the adoption of two parameters that supply a detailed description of the channel characteristics.

A different literature branch uses the water levels measured in two or three gauging stations of a river reach, in order to estimate the discharge hydrograph in one of these sections. For natural and artificial channels, Aricò et al. (2009; 2010) investigated and validated a hydraulic diffusive flow routing model, which requires the knowledge of synchronous water level measurements in two river sections a few kilometers apart, and estimates both the discharge hydrograph in the upstream section and the channel roughness. Perumal et al. (2007) adopted a Muskingum numerical scheme for estimating the discharge hydrograph at the upstream or downstream gauging station, from the knowledge of the stage hydrographs registered at the two gauging stations; recently, Barbetta et al. (2017) extended the method as to take into account later inflows.

Considering the presence of later contributions, and using the water level measurements at two gauged sections, Spada et al. (2017) estimated the discharge hydrograph at the downstream one and the channel roughness.

The cited works require the knowledge of the water levels in two/three river sections in order to estimate (considering or not later contributions) the discharge hydrograph in one of the section with register instruments. However, they do not manage to estimate the flow hydrograph in an ungauged section located upstream a gauging one.

Zucco et al. (2015) investigated the reverse flood routing process in natural channels, and estimated the discharge hydrograph in ungauged sections, by means of a Genetic algorithm. The discharge hydrograph is described by means of a Pearson type III distribution and thus the algorithm estimates three parameters. However, this assumption limits the estimation of real flood waves with irregular shapes. Moreover, the equifinality problem may arise, since different set of parameters can produce the same downstream-observed hydrographs.

Recently, D’Oria and Tanda (2012) solved the reverse flow routing problem adopting a novel Bayesian Geostatistical Approach (BGA), which considers a flow hydrograph as a statistical continuous random function that presents autocorrelation and accounts for uncertainties. For 1D cases, the authors showed the capability of the BGA methodology to estimate the flow discharge in an upstream-ungauged section without having knowledge of the water levels in this section, and having information only in a downstream section: the procedure evidenced no instabilities also in presence of corrupted downstream discharge.

The BGA method was further extended in order to adopt as downstream observations stage hydrographs instead of the discharge ones (D’Oria et al., 2014). Focusing on a real river reach that presents also junctions, the BGA methodology allowed the inflow estimation of the tributary channel, having information about the inflow discharge on the main channel and the stage hydrograph recorded downstream the confluence (D’Oria et al., 2014). Additionally, the same methodology was adopted to estimate the flow through a levee breach assuming the knowledge of the water levels recorded downstream and/or upstream the river bank failure (D’Oria et al., 2015): all these test cases were performed using a 1D forward model.

However, in many real cases of rivers including large floodable areas, it is necessary to adopt a 2D Shallow Water Equation model to capture the complex hydrodynamic field, even if this poses the drawback of the high computation costs, with respect to the 1D scheme.

Therefore, this chapter extends the BGA methodology for reverse flow routing to 2D test cases in order to model natural rivers with complex geometry, including flood plains and meanders. Thus, the goal of the procedure is to estimate the discharge hydrograph in an upstream-ungauged river section, having water level information only in a downstream observation site. Since the inverse procedure requires also a stable, accurate and fast forward numerical model, the 2D GPU finite volume model illustrated in Section 1.2 is thus adopted.

## 5.2. The Bayesian Geostatistical Approach

The software adopted to solve the reverse flow routing is the bgaPEST (Fienen et al., 2013), which implements the Bayesian Geostatistical Approach of Kitanidis (1995) and it is developed according to the PEST (Model Independent Parameter Estimation) parameter estimation software (Doherty 2016). The bgaPEST solves inverse problems (in a context of a highly parametrized inversion), which are characterized by unknown parameters that are correlated one another in space or time, as for example a flow hydrograph.

### 5.2.1. The Bayes' theorem

The crux of the adopted bgaPEST, as well as other methods based on the Bayesian Approach, is the Bayes' theorem. This theorem, which is related to the conditional probability, states that the probability of **s**, for given **y**, is proportional to the probability of **y** for given **s**, and to the probability of **s** independently from **y**, according to the following relation:

$$p(\mathbf{s}|\mathbf{y}) \propto L(\mathbf{y}|\mathbf{s})p(\mathbf{s}) \tag{5.1}$$

where **s** is the vector of the unknown parameters, **y** is the vector of the measured data, $p(\mathbf{s}|\mathbf{y})$ is the posterior probability density function (pdf) of **s** given **y**, $L(\mathbf{y}|\mathbf{s})$ is the likelihood function and $p(\mathbf{s})$ is the prior probability density function of **s**.

Recalling that the aim of the present chapter is the estimation of an upstream hydrograph in an ungauged section, assuming the knowledge of downstream water levels, **s** represents the time values of the inflow hydrograph, whereas **y** denotes the downstream water level observations.

Following Equation (5.1), the posterior pdf, which represents the parameter knowledge after the observations, can be seen as a weighted average between *a priori* knowledge of the parameters (prior pdf), where *a priori* means that the observed data are still not present, and information about parameters contained in the measured data (likelihood function) (Glickman and Van Dyk, 2007). Between the two contributes, the likelihood function has a more relevant role in the posterior pdf definition than the prior information (Fienen et al 2009). In the BGA method proposed by Kitanidis (1995), the prior pdf and the likelihood function are described by means of a Gaussian distribution and the best set of parameter $\boldsymbol{s}$ is obtained by maximizing the posterior pdf.

## The likelihood function

Focusing on the terms of the Bayes' theorem (Equation 5.1), the likelihood function $L(\mathbf{y}|\mathbf{s})$, which in the BGA is assumed to be Gaussian distributed, accounts for the misfit between observed data and model results (Fienen et al 2013). Starting from the results of the forward model, $L(\mathbf{y}|\mathbf{s})$ delineates how a particular set of parameters $\mathbf{s}$ is able to reproduce the observations $\mathbf{y}$ in space and/or time. Particularly, a weighted sum of the squared differences between modeled and measured data, based on a covariance matrix, which takes into account the epistemic errors, is evaluated.

The investigated inverse problem presents different sources of errors, which are related to the conceptual schematization of the inverse procedure, to the numerical forward model and to the data measurements. In the likelihood function, the errors are assumed to be identically and independently distributed, with null mean and covariance matrix, expressed as follows:

$$\mathbf{R} = \sigma_R^2 \mathbf{I} \tag{5.2}$$

where $\sigma_R^2$ denotes the variance and $\mathbf{I}$ the identity matrix;

### The prior probability density function

On the other side, the prior probability density function of **s,** $p(\mathbf{s})$, has the following purposes (Fienen et al., 2006):

1. Provide a soft knowledge of the structure/shape of the unknowns **s**;
2. Provide a regularization of the solution (enforcing smoothness and/or continuity);
3. Can enforce non-negativity to the parameters.

According to the first point, $p(\mathbf{s})$ provides information about the unknown parameters, without considering the observed data. However, this *a priori* knowledge is referred only to the structure of the parameters (and not to their values), and it consists in specifying a variogram or a covariance function, by means of geostatistical tools. Consequently, the solution is still mostly driven by the observations, but it is also influenced by these prior assumptions.

The prior probability density function of **s,** $p(\mathbf{s})$, has been assumed with Gaussian distribution; the vector ***s*** is *a priori* defined by its mean (unknown and estimated during the procedure):

$$E[\mathbf{s}] = \mathbf{X}\beta^{*} \tag{5.3}$$

where $E$ is the expected value, $\beta^{*}$ is the vector of drift coefficients, and $\mathbf{X}$ is the known matrix of basis functions, which link each value of **s** with the appropriate element of $\beta^{*}$ (Fienen et al., 2008).

As a Gaussian distribution is described not only by the mean but also by the covariance, the prior covariance matrix of the unknown parameters $\mathbf{Q}_{ss}$ is defined as:

$$\mathbf{Q}_{ss} = E[(\mathbf{s} - \mathbf{X}\beta^{*})(\mathbf{s} - \mathbf{X}\beta^{*})^{T}] \tag{5.4}$$

In the context of geostatistics, the covariance matrix $\mathbf{Q}_{ss}$ is a function of the separation distance (in time in this case) between the parameters and describes their deviations from the mean behavior. Different models can be adopted to describe the covariance; for example, it can be assumed as a linear function, represented through a limiting case of the exponential covariance model (Fienen et al., 2008), according to the following relation:

$$\mathbf{Q}_{ss}(\theta) = \theta l \cdot exp\left(-\frac{|d|}{l}\right) \tag{5.5}$$

where $d$ represents the separation distance in time between the parameters, $l$ a fixed integral scale ($l$=10 * max($d$)) and $\theta$ the slope structural parameter, which influences the correlation between the discharge values of the unknown hydrograph.

A different formulation (D'Oria et al., 2014) defines the prior covariance matrix $\mathbf{Q}_{ss}$ by means of a Gaussian function:

$$\mathbf{Q}_{ss} = \sigma_s^2 \cdot exp\left(-\frac{d^2}{l^2}\right) \tag{5.6}$$

where $\sigma_s^2$ denotes the variance.

The linear function (Equation (5.5)) enforces only continuity to the solution whereas the Gaussian model (Equation (5.6)) adds also some degree of smoothness, but the final solution is still driven by the observations. Moreover, the structural parameters ($\theta, \sigma_s^2$, $l$), which govern the structure of the general parameters, and not the physical problem, are estimated based on the observations.

### The posterior probability density function

After having separately described the quantities involved in the likelihood function and in the prior probability density function, the terms in the posterior probability density function (pdf) of Equation (5.1) can be rewritten as follows (Fienen et al., 2009; D'Oria and Tanda, 2012; D'Oria et al., 2014):

$$L(\mathbf{y}|\mathbf{s}) = exp\left(-\frac{1}{2}\big(\mathbf{y} - \mathbf{h}(\mathbf{s})\big)^T \mathbf{R}^{-1}\big(\mathbf{y} - \mathbf{h}(\mathbf{s})\big)\right) \tag{5.7}$$

$$p(\boldsymbol{s}) = exp\left(-\frac{1}{2}(\mathbf{s} - \mathbf{X}\beta^*)^T \mathbf{Q}_{ss}{}^{-1}(\mathbf{s} - \mathbf{X}\beta^*)\right) \tag{5.8}$$

The evaluation of the likelihood function requires a forward model of the considered river reach that is able to describe the hydraulic routing process. In fact, the term $\mathbf{h}(\mathbf{s})$ is the function representing the results of this model that provides for an inflow upstream hydrograph the downstream water levels. This means that, for a given set of parameter **s**, the function $\mathbf{h}(\mathbf{s})$ provides the modeled values in the same place and time of the available observations **y**.

Recalling that the aim of the inverse procedure is to obtain the vector of the unknown parameters ***s***, as well as to quantify the uncertainty in the estimation, the procedure maximizes the posterior probability density function. In case of a linear relationship between parameters and observations, the best estimate $\hat{\mathbf{s}}$ of vector **s** is obtained by solving the following linear system of equations (Michalak and Kitanidis, 2003):

$$\begin{cases} \hat{\mathbf{s}} = \mathbf{X}\hat{\boldsymbol{\beta}} + \mathbf{Q}_{ss}\mathbf{H}^{\mathrm{T}}\boldsymbol{\xi} \\ \begin{bmatrix} \mathbf{H}\mathbf{Q}_{ss}\mathbf{H}^{\mathrm{T}} + \mathbf{R} & \mathbf{H}\mathbf{X} \\ \mathbf{X}^{\mathrm{T}}\mathbf{H}^{\mathrm{T}} & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{\xi} \\ \hat{\boldsymbol{\beta}} \end{bmatrix} = \begin{bmatrix} \mathbf{y} \\ 0 \end{bmatrix} \end{cases} \tag{5.9}$$

where **H** is the sensitivity (Jacobian) matrix, representing how observations **y** are influenced by a single unknown parameter $s_i$ (D'Oria et al., 2015).

However, for this particular problem, $\mathbf{h}(\mathbf{s})$ is nonlinear and therefore matrix **H** depends on **s**. Following the quasi-linear geostatistical approach (Kitanidis, 1995), for each iteration *k*, the relationship between observations and parameters is successively linearized about a candidate solution $\mathbf{s}_k$:

$$\mathbf{h}(\mathbf{s}) \approx \mathbf{h}(\mathbf{s}_k) + \widetilde{\mathbf{H}}(\mathbf{s} - \mathbf{s}_k) \tag{5.10}$$

and then a correction to the measurements is applied according to the following relation:

$$\mathbf{y}_k = \mathbf{y} - \mathbf{h}(\mathbf{s}_k) + \widetilde{\mathbf{H}}_{\boldsymbol{k}}\mathbf{s}_k \tag{5.11}$$

Therefore, the sensitivity matrix is evaluated at each iteration, as follows (D'Oria et al., 2014):

$$\widetilde{\mathbf{H}}_{k=} \frac{\partial \mathbf{h}(\mathbf{s})}{\partial \mathbf{s}}\bigg|_{\mathbf{s}_k} \tag{5.12}$$

It is relevant to notice that the sensitivity matrix, is usually evaluated by means of a finite differences method, and it requires as many forward model runs as the number of parameters plus one base run (as will be described in next section).

Analogously to the linear system in Equation (5.9), the linearized system is solved according to:

$$\begin{bmatrix} \tilde{\mathbf{H}}_k \mathbf{Q}_{ss} \tilde{\mathbf{H}}_k^T + \mathbf{R} & \tilde{\mathbf{H}}_k \mathbf{X} \\ \mathbf{X}^T \tilde{\mathbf{H}}_k^T & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{\xi}_{k+1} \\ \hat{\boldsymbol{\beta}}_{k+1} \end{bmatrix} = \begin{bmatrix} \mathbf{y}_k \\ 0 \end{bmatrix} \tag{5.13}$$

and the next estimate of the parameters is evaluated by means of:

$$\tilde{\mathbf{s}}_{k+1} = \mathbf{X}\hat{\beta}_{k+1} + \mathbf{Q}_{ss} \tilde{\mathbf{H}}_{k+1}^T \boldsymbol{\xi}_{k+1} \tag{5.14}$$

Dealing with an iterative procedure, the first guess of the structural parameters and the epistemic error variance, which characterize the covariance functions $\mathbf{Q}_{ss}$ and $\mathbf{R}$ in Equations (5.13-5.14), is important for the solution of the inverse problem. However, during the modelling, the structural parameters are estimated from the data using a Bayesian adaptation of the Restricted Maximum Likelihood (RML) method of Kitanidis (1995), which adopts probability functions and allows reaching the best compromise between the fitting of the modeled data and the observations, and the prior information (Fienen et al., 2013).

Finally, at the end of the estimation, the linearized uncertainties of the unknowns can be evaluated in terms of the posterior covariance matrix of the estimated parameters as follows (D'Oria et al., 2014):

$$\mathbf{V} = \mathbf{Q}_{ss} - \mathbf{Q}_{ss}\mathbf{H}^T(\mathbf{H}\mathbf{Q}_{ss}\mathbf{H}^T + \mathbf{R})^{-1}\mathbf{H}\mathbf{Q}_{ss} \tag{5.15}$$

The diagonal of matrix $\mathbf{V}$ represents the variance $\sigma^2$ related to the parameter estimation: considering $\pm 2\sigma^2$ allows defining the 95% credibility interval of the resulted parameters.

## 5.2.2. Operational aspects of the BGA algorithm

As mentioned in Section 5.1 and sketched in Figure 5.1, the goal of the adopted BGA methodology is the estimation of the discharge hydrograph in an upstream-ungauged river section, having information about the downstream boundary condition and the water levels observed in an intermediate downstream section.

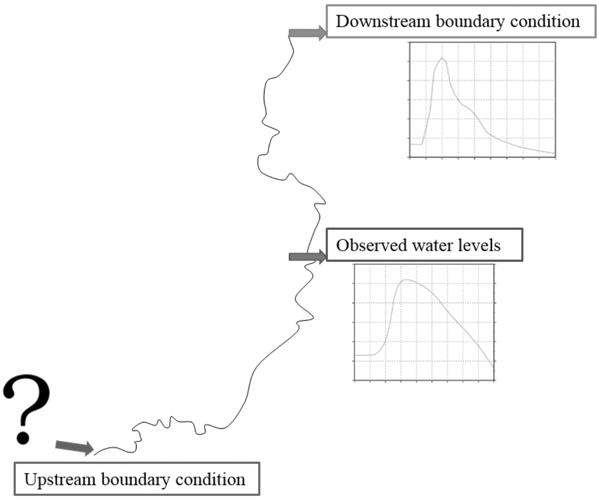


*Figure 5.1. Definition of the reverse flood routing problem.*

After having described the theory of the Bayesian approach in the previous section, some operational information about the bgaPEST software is now illustrated.

Before proceeding, it is important to clarify the physical meaning of the vector of the unknown parameters **s** in Equation (5.1). Assuming the discharge hydrograph in Figure 5.2-a to be estimated, this is actually discretized by means of regular time intervals, in this example equal to 8 hours (Figure 5.2-b). Therefore, a certain number of points/parameters $N_p$, which represent the intersections between the discharge value and the time interval, is identified in Figure 5.2-b. Thus, the definition of a discharge hydrograph actually consists in the estimation of $N_p$ parameters.

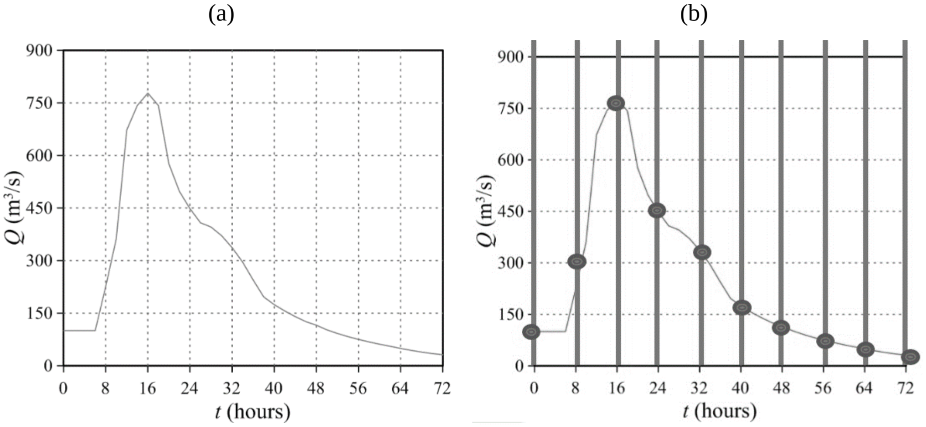


*Figure 5.2. Generic discharge hydrograph (a) and definition of the unknown parameters (b).*

The BGA algorithm solves the inverse problem by means of the following steps.

Firstly, the unknown parameters ***s*** and the structural ones are initialized. The first ones may be all assumed equal to a constant value coherent with the considered river, whereas the structural parameters are usually set as to guarantee a flat solution (complexity is introduced during the optimization process only if supported by the data) and finally the variance of the epistemic errors is assumed close to the expected one.

Secondly, the hydraulic forward model is run by assuming the first guess of the unknown parameters **s** as upstream boundary condition.

Then, the resulted water levels in the observation site are extracted.

The simulation of a base run, once assumed a particular set of parameters (deriving from the initialization or from previous estimation steps), represents a mandatory step for the Jacobian/sensitivity matrix evaluation, which is performed at this point of the procedure.

The Jacobian matrix, which quantifies how each observation is influenced by the variation of each estimable parameter, is calculated using a finite differences method. According to Equation (5.12), each element is evaluated as the ratio between the variation of each observation for given variation of each parameter (numerator) and the variation of the parameter value with reference to the base run (denominator).

Therefore, additionally to the base run, the Jacobian matrix evaluation requires further running the hydraulic forward model as many times as the number of parameters to estimate $N_p$. At each run, a single value of the upstream boundary condition is modified by a known quantity with respect to the previous value (denominator of Eq. 5.12), the hydraulic forward model is run and the output file with the resulted observations is read. Therefore, each simulation tests the sensitivity of the resulted water levels (all the observations at once) to the variation of a single parameter $i$.

After having collect all the observations, the Jacobian/sensitivity matrix is evaluated and a new set of parameters **s** is estimated.

Figure 5.3 exemplifies the sensitivity matrix evaluation for a given parameter $i$. In Figure 5.3-a, the base run is simulated: the discharge depicted with the blue line is imposed as upstream boundary condition and the resulted water levels extracted in the observation site are represented with the red line. Assuming that the Jacobian matrix is testing the sensitivity to parameter $i$ (Figure 5.3-b), this is varied with a known quantity and the orange line represents the parameters imposed as upstream boundary condition; it is worth noting that the blue and the orange line differs only for the parameter $i$. The water levels resulted from this single parameter variation are identical to the base run ones until time $i$-1, whereas after the green line differs from the red one of the base run. The computation of the differences between the resulted water levels of the simulation $i$ and of the base run (green and red lines) and the variation of parameter $i$ allows computing the column $i$ of the Jacobian matrix, which is a $N_{obs} \times N_p$ matrix, where $N_{obs}$ represents the number of the observations.

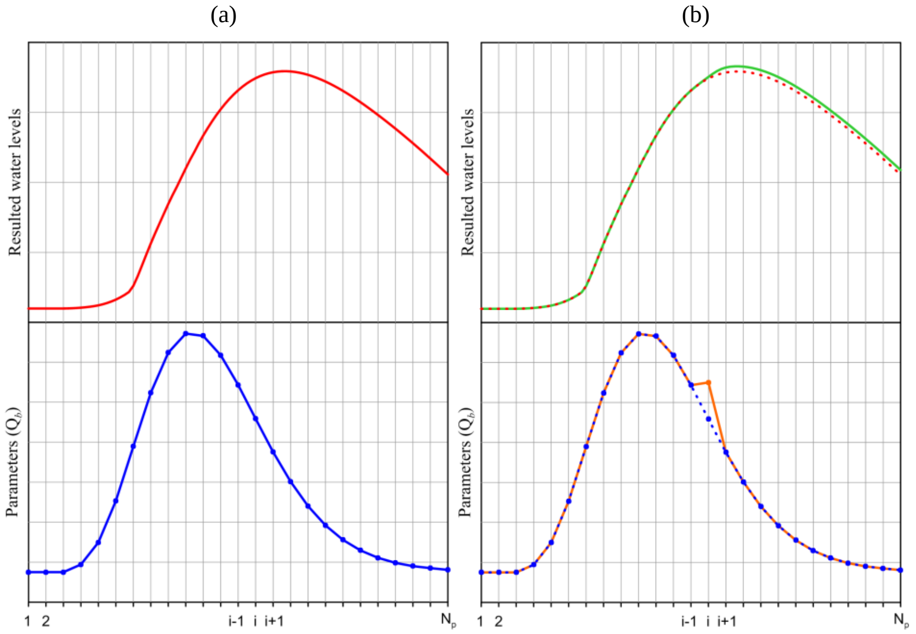


*Figure 5.3. Example of the base run (a) and of the run i for the Jacobian matrix evaluation (b).*

Then the first set of resulted parameters is used for evaluating a second Jacobian matrix and as a result, a second set of parameters is estimated. This procedure is repeated until the convergence or the maximum number of iteration $N_i$ is reached.

Then, holding the last set of parameters **s** constant, the structural parameters are estimated. Due to the non-linearity of the problem, the model and the structural parameter estimation is repeated until convergence of the lasts (or the maximum number of iterations $N_o$ is reached). Therefore, the BGA implementation requires running the forward model for $N_t$ times, according to the following relation (Fienen et al., 2013):

$$N_t = \left(N_p + 1\right) \cdot N_o \cdot N_i + 1 \tag{5.16}$$

The whole BGA procedure previously described is sketched in Figure 5.4.

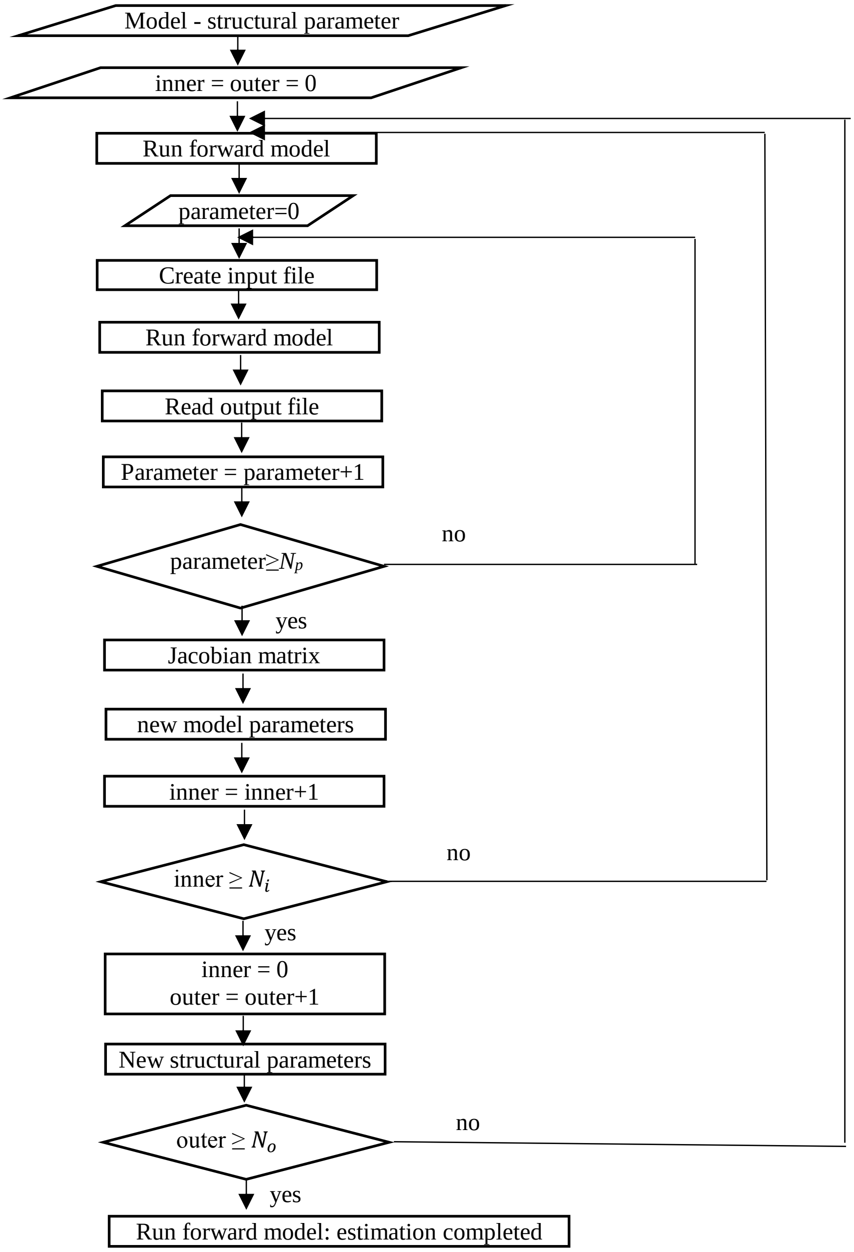


*Figure 5.4. Scheme of BGA algorithm in the serial version.*

## 5.3. Parallelization of the Jacobian matrix evaluation

As described in the previous section, the solution of the inverse problem requires running the forward model $N_t$ times, according to Equation 5.16. The most relevant contribution of this high number of simulations is associated with the evaluation of the Jacobian matrix that requires as many runs as the number of parameters $N_p$ to estimate plus one. Recalling that this last additional run can been seen as the base run, performed with a particular set of parameters, each of the other runs tests the sensitivity of the solution to the variation of a particular parameter (i.e. run 1 tests the sensitivity on parameter 1, run 2 on parameter 2, etc.). Consequently, these runs are independent one another (the solution of a test does not have effects on the solution of other tests) and with the aim at reducing the computational burden they can be performed in a parallel manner.

In this thesis, the 2D-GPU numerical model described in Section 1.2 has been adopted for routing the inflow hydrograph. The bgaPEST routine to evaluate the Jacobian matrix has been parallelized in order to run simulations taking advantage of the computational capability of modern High Performance Computing (HPC) clusters, which are usually equipped with many GPUs.

The implemented parallel procedure handles the parallelism among host and GPUs by means of the Secure Shell network protocol (SSH) and manage the most operative parts of the parallelism (login, run, etc.) outside the bgaPEST code. The essential schemes of the original (serial) and modified (parallel) versions of the code are summarized in Table 5.1. In the serial version, the crucial part of the implementation consists in a do-loop over the parameters. Considering an *i* parameter, firstly the input file that will be read by the forward model is written, then the model is run and finally the resulted values are read. In the modified version, this main loop is split in three parts: firstly, all the input files (equal to $N_p$), inside of which a particular parameter is modified are written, then the forward model is run, and finally a second loop is performed to read all the resulted values.

*Table 5.1. Pseudocode of the serial/original and parallel/modified version.*

*Serial version*

```
DO Loop over parameters i=1, $N_p$
    Write input file
    Run forward model
    Read modeled values
END DO
```

*Parallel version*

```
DO Loop over parameters i=1, $N_p$
    Write input files
END DO
    Run forward model
DO Loop over parameters i=1, $N_p$
    Read modeled values: output_file_i
END DO
```

The overall bgaPEST procedure with the parallel evaluation of the Jacobian matrix is sketched in Figure 5.5.

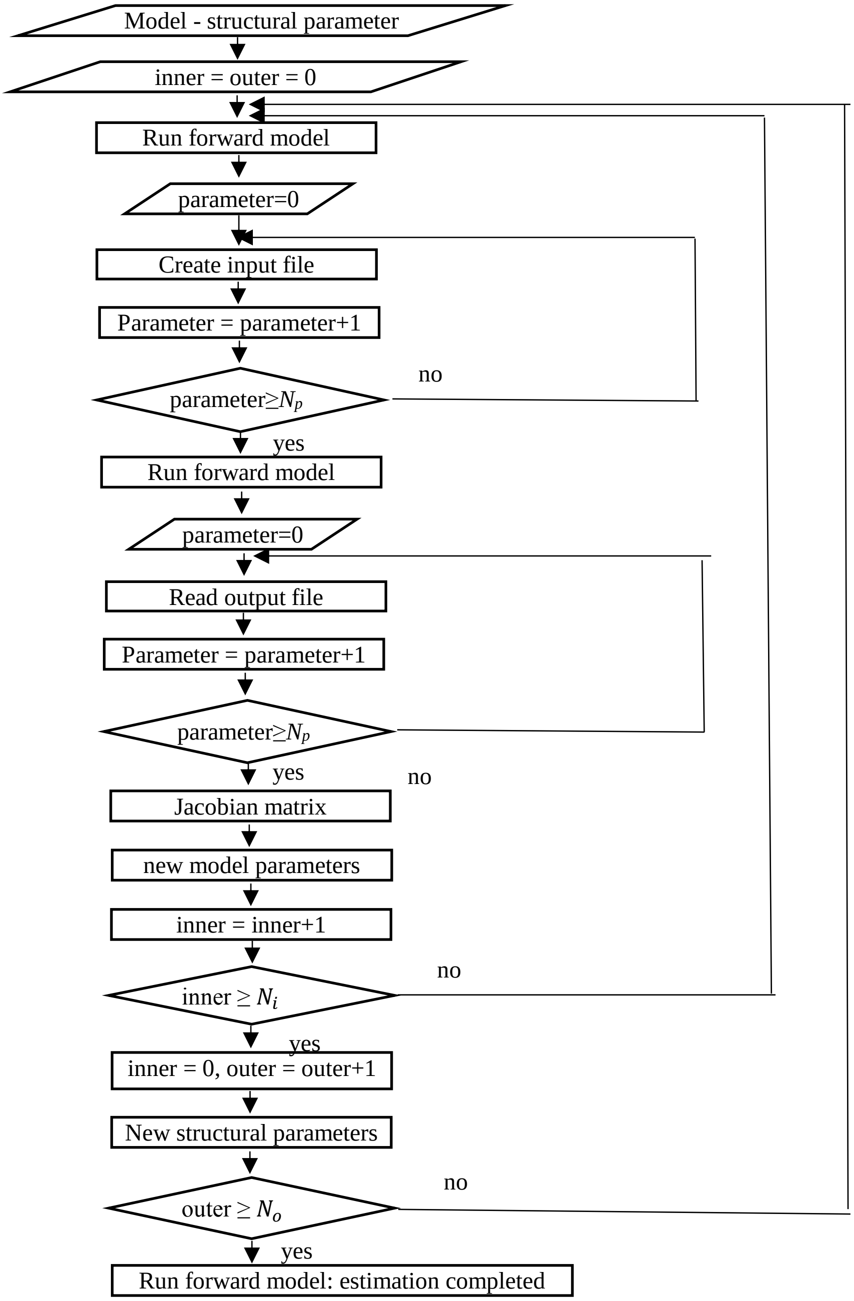


*Figure 5.5. Scheme of BGA algorithm in the parallel version.*

## 5.4. The forward model

In the parallel bgaPEST (Figure 5.5), the “Run forward model” instruction actually runs a shell script, which controls the file transfer between the client (a classical PC or a single node of a cluster), and the HPC platform, the creation of the $N_p$ simulations for the Jacobian matrix evaluation, and the run of the 2D-GPU code on the device (GPU). In the present thesis, the cluster with 10 NVIDIA ® Tesla ® P100 GPUs of the University of Parma was adopted.

As shown in Figure 5.6, the bgaPEST algorithm runs on the CPU of a computer, where the $N_p$ simulations (in Figure 5.6 assumed equal to three) are firstly created and then sent to the server user partition, by means of the SSH protocol. Here, the cluster access node schedules all the jobs submitted by the users, using the HPC scheduler Portable Batch System (PBS). Then, each simulation is assigned to a specific GPU node and it is performed on the device (GPU). At the end of the computation, the observations are extracted and the output files remain on the cluster partition, until the CPU verifies via SSH the end of the simulation and copies the results back.

The procedure sketched in Figure 5.6 and following described represents one of the $N_t$ iterations.

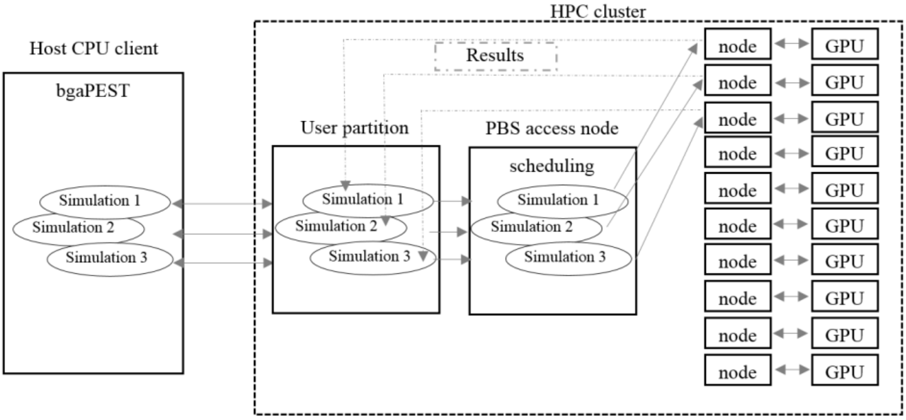


*Figure 5.6. Schematization of the data transfer.*

With the aim of describing in detail the “Run forward model”, after having clarified the data transfer procedure (Figure 5.6), in Algorithm 5.1 the structure of the shell file is presented.

In order to use the Algorithm for different test cases and potentially on different HPC clusters, all the paths are firstly declared together with the involved variables (number of parameters to estimate, time interval among parameters, start/end of the simulation) (line 2). Then, the algorithm (line 3) checks if the considered run is one useful for the Jacobian matrix evaluation, where a given parameter varies, or if it is the base run. Considering the first if condition true (line 3), the script generates and copies all the simulations to the server (lines 5-7). These tests contain the same bathymetrical, initial conditions (water level and velocity), roughness configuration, and different upstream boundary condition; each simulation tests the sensitivity to the variation of a given model parameter.

Moreover, all the simulations present the same grid (Cartesian or multiresolution). Dealing with variable resolution grids, the preprocessing step that creates a BUQ grid (Section 2.2.2) has been modified in order to compute the partitioning once and producing extra files (included in the test folder) useful to reconstruct the position of the blocks and the resolution level. Therefore, a significant reduction of the computational time is obtained by avoiding the calculation of the variable resolution grid for $N_p$ times.

It is relevant to notice that all the $N_p$ simulations have not to be run from time $t_{start}$ to time $t_{end}$. In effect, the variation of parameter $i$ causes effects only after time $t_i$ and thus the results until $t_i$ are still identical to the base run (see Figure 5.3-b). The 2D-GPU parallel numerical model guarantees the possibility of using the results of the base run and starting simulations from time $t_i$.

The theoretical physical time $T$ of simulations run from time $t_{start}$ to time $t_{end}$ is evaluated as follows:

$$T = N_p(N_p - 1)\Delta t \tag{5.17}$$

where $N_p$ denotes the parameter number and $\Delta t$ the time interval between parameters $i$ and $i$+1.

Conversely, the physical time $T^*$ of simulations run from time $t_{restart}$ to time $t_{end}$ reads:

$$T^* = (N_p - 1)\Delta t + \sum_{i=2}^{N_p} [N_p - (N_i - 1)]\Delta t \tag{5.18}$$

As pointed out by Equations (5.17)-(5.18) and exemplified in Figure 5.7 for a test case with 20 parameters, this simply operation allows reaching a relevant decrease of the total computational times.

Therefore, at line 8, the algorithm computes the time useful to restart the simulation.

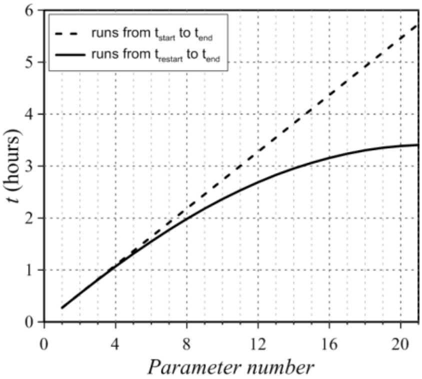


*Figure 5.7. Comparison of the cumulative computational times obtained simulating the $N_p$ runs from time $t_{start}$ until $t_{end}$ or with restart options.*

In order to perform the simulation, the client CPU logins to the HPC cluster server by means of the SSH protocol (line 9) and a sleep condition guarantees the correct realignment of all the copy (line 10). Then the job is submitted to the queue of the cluster using external parameters for passing the name of the simulation folder and the time for restart (line 11): the submitted job contains the reference to the PBS queue and the link to the executable 2D-GPU code. At the end of the simulation, the water levels in the observation site are automatically extracted (the 2D-GPU model is enabled with this post-processing feature). Once the job is submitted, the SSH login is closed (line 12).

After having submitted all the simulations, for each parameter (line 15) the code regularly (line 18) tests via SSH the presence of the *end_file*, which states the end of the simulation (line 20), and waits in case it is missing (line 25). Once the simulation is finished, the resulted observations are copied back to the CPU (line 28) and the folder is removed from the server (line 29).

On the other side, the else condition (line 31) is true for the base run of the bgaPEST. It is noteworthy recalling that the Jacobian matrix calculates the variation of the given parameters and thus a base run assumed as reference is necessary. The simulation folder is copied to the server (line 32) and the job is submitted (line 35). Then, the algorithm tests the presence of the end simulation file and copies the results back to the CPU (lines 39-49).

Even if the standard iteration executes firstly the base run and then the $N_P$ runs, the algorithm tests in the opposite direction only for achieving more computational efficiency.

*Algorithm 5.1: pseudocode of the "Run forward model" for the parallel bgaPEST scheme*

```
1:   #!/bin/bash
2:   Variable and path declaration
3:   if [parameter run];
4:     then
5:       for (( i=1; i<=$N_p$; i++ ))
6:         do
7:           Create and copy the simulation folder to the server (simulation_i)
8:           Compute the time from which restarting the simulation
9:          ssh server name << EOF
10:        sleep 15
11:        ssh submission: frame number for restart , name of the test case, job to submit
12:        exit
13:        EOF
14:      done
15:    for (( i=1; i<=$N_p$; i++ ))
16:      do
17:      end_file=0
18:      while [ end_file -eq 0 ];
```

```
              do
                  ssh server name  find server_path -iname ‘end_file.txt' | wc -l > end_file
                  if [end_file];
                      then
                              continue
                      else
                            sleep 10
                  fi
        done
        Copy the file with observation from server to CPU
        Remove the simulation folder on the cluster
    done
else
      Create and copy the simulation folder to server
      ssh server name << EOF
      sleep 15
      ssh submission: frame number for restart , name of the test case, job to submit
      exit
      EOF
      end_file=0
      while [ end_file -eq 0 ];
              do
                  ssh server name  find server_path -iname ‘end_file.txt' | wc -l > end_file
                  if [end_file];
                            then
                              continue
                            else
                                  sleep 10
                        fi
        done
        Copy the file with observation from server to CPU
      fi
```

## 5.5. Quantitative evaluation of the inverse methodology

The capability of the BGA inverse methodology is investigated in the next sections by estimating the inflow hydrographs in two natural rivers in Northern Italy.

With the purpose of validating the inverse procedure described before, it is worth noting that reference solutions for inverse problems are by definition unavailable: as described in previous sections, the goal of the inverse methodology is the estimation of an upstream inflow hydrograph that is unknown at the beginning of the process. Therefore, with the aim of validating the inverse methodology, in all tests presented in this section the reference solutions were obtained as follows (D'Oria and Tanda, 2012). With reference to the domain in Figure 5.8, a selected inflow discharge $Q^{act}$ is routed from the upstream section A until the downstream boundary D, where a rating curve is imposed. The resulted water level hydrograph is extracted in sites B and C.

The inverse procedure is then applied to the smaller domain sketched with solid line in Figure 5.8, by assuming the water levels in sites B and C (resulted from step 1) as observations and downstream boundary condition, respectively.

The methodology estimates the inflow $Q^{est}$ assuming that no information is available on the discharge at the inflow.

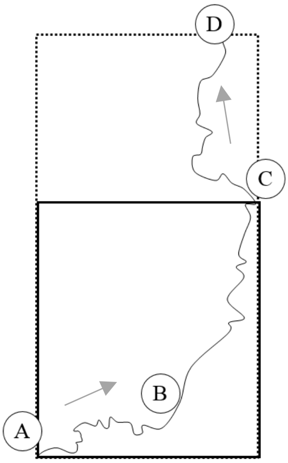


*Figure 5.8. Exemplification of a test case definition.*

The availability of reference solutions allows providing quantitative information about the accuracy of the inverse methodology. Particularly, in this thesis the differences between the actual $Q^{act}$ and the estimated $Q^{est}$ hydrographs have been evaluated by means of three different indicators.

Firstly, the Nash-Sutcliffe efficiency criterion (Nash and Sutcliffe, 1970) $E_h$ was adopted, according to the following relation:

$$E_h = \left[1 - \frac{\sum_{i=1}^{N}(Q_i^{act} - Q_i^{est})^2}{\sum_{i=1}^{N}(Q_i^{act} - \bar{Q}^{act})^2}\right] \cdot 100 \tag{5.19}$$

where $N$ is the total number of parameters (data of the discharge hydrograph), $Q_i^{act}$ and $Q_i^{est}$ are the $i$-th actual and estimated inflow values, respectively, and $\bar{Q}^{act}$ is the mean value of the actual hydrograph.

Then, the root mean square error, *RMSE* was evaluated as follows:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}(Q_i^{act} - Q_i^{est})^2}{N}} \tag{5.20}$$

Finally, the error in the peak discharge $E_p$ was assessed as:

$$E_p = \left[\frac{Q_p^{est}}{Q_p^{act}} - 1\right] \cdot 100 \tag{5.21}$$

where $Q_p^{est}$ and $Q_p^{act}$ denote the peak discharge value of the estimated and actual hydrographs, respectively.

## 5.6. Inflow hydrograph estimation on the Secchia River

The first studied domain includes a 25 km-long reach of the Secchia River between the flood control reservoir of Rubiera-Campogalliano located at West of Modena town (point A) and the gauging station of Ponte Bacchello (point C) and referring the water level observations to the gauging station of Ponte Alto (point B) (Figure 5.9).

The modeled river reach is characterized by the presence of many flood plains that influence the flood propagation. The bathymetry was derived from a 1 m resolution DTM obtained through a LiDAR survey carried out during the dry season, in drought condition.

*Figure 5.9. Bathymetry of the Secchia studied domain with indicated the upstream (A) and downstream (C) boundary conditions and the intermediate observation site (B).*

The domain was discretized by means of a non-uniform BUQ grid and 8×8 cell blocks (see Chapter 2); the resolution level was forced to $\Delta x_{\min} = 10$ m along the main channel, while it was allowed gradually decreasing to $\Delta x_{\max} = 80$ m elsewhere. It is noteworthy that this lowest resolution value was limited to a small portion of flood plain near the reservoir. As a result of the BUQ grid partition, $77 \cdot 10^3$ computing cells discretized the domain.

With reference to the calibration procedure described in Section 1.4.1, the Manning roughness coefficient in the riverbed was assumed equal to 0.05 s/m$^{1/3}$.

All the simulations were performed adopting the first-order accurate version of the hydraulic forward model (parallel 2D-SWE code described in Section 1.2).

The initial conditions were obtained as illustrated in Figure 5.10, where the two green points denote the first unknown parameter to be estimated $p_1$ and the red points the other unknown parameters $N_{p\text{-}1}$. The initial condition is obtained by simulating a steady state condition associated with the first parameter value $p_1$: in the example of Figure 5.10 this condition is achieved after 30 hours.

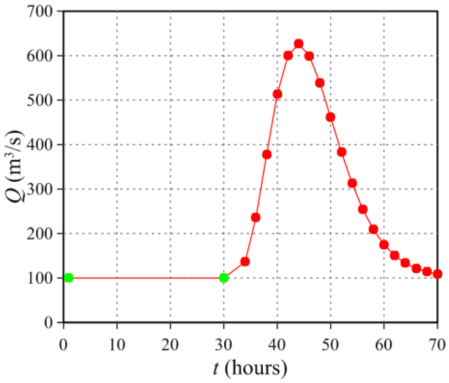


*Figure 5.10. Example of the initial condition definition.*

## 5.6.1. Gamma distributed inflow

The first test concerning the Secchia River imposed as inflow condition a synthetic discharge hydrograph with gamma distribution that was calculated as follows (D'Oria et al., 2015):

$$Q(t) = A + B \cdot f(t, b, k) \tag{5.22}$$

where $t$ denotes the time, $A$ the base flow (constant value), $B$ the volume above the base flow (constant value) and $f$ the gamma distribution, which states:

$$f(t, b, k) = \frac{1}{k^b \Gamma(b)} t^{b-1} e^{-\frac{t}{k}} \tag{5.23}$$

where $\Gamma(b)$ represents the gamma function defined through the parameters $b$ and $k$ that denote the shape and the scale parameter, respectively.

In order to obtain a discharge hydrograph with volume and peak value analogous to the 50 years-return period flood wave of the Secchia River, the parameters of the gamma distribution were set as follows: $b$ = 8.5, $k$ = 9000 s, $A$ = 150 m$^3$/s, $B$ = 7.5·10$^7$ m$^3$. The resulted flow hydrograph was characterized by a peak value of about 1350 m$^3$/s at time $(b-1)k$ = 18.75 hours (Figure 5.11-a).

The inflow hydrograph ended in 72 hours and it was discretized using 2 hours steps, whereas the observation stage hydrograph was discretized every 0.5 hours; the inverse methodology estimated 37 parameters having the availability of 144 water level values.

Focusing on the outflow stage hydrograph, two different conditions were assumed in order to account for the presence of the epistemic errors. Particularly, the first estimation of the inflow hydrograph was performed assuming the observations extracted at Ponte Alto free of errors and with truncation error resulting in a variance of 10$^{-8}$ m$^2$. On the other side, the second estimation was performed corrupting the observed water levels with random errors uniformly distributed with maximum deviations of ± 0.05 m and variance 10$^{-3}$ m (Figure 5.11-b).

Besides the inflow and outflow hydrographs, Figure 5.11-a depicts the discharge hydrograph resulted in the downstream boundary condition section, in order to highlight the delamination effect and the importance of adopting a 2D forward hydraulic model.

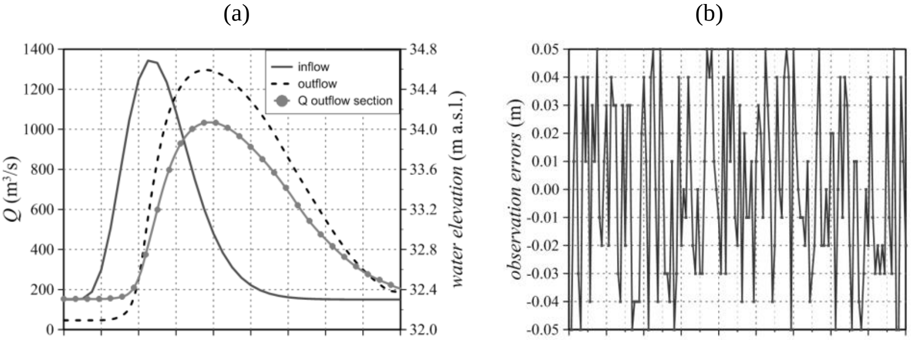


*Figure 5.11. Secchia gamma distributed inflow: inflow and outflow hydrographs (a) and observation error distribution (b).*

### Initial model and structural parameters

In order to better reproduce low values of the discharge hydrograph and enforce non-negativity, the structural and the model parameters were estimated in a logarithmic space.

The initial model parameter values were defined by applying the linesearch tool of the bgaPEST, which damps the solution between successive iterations (Fienen et al., 2013), and avoids numerical instabilities that may occur starting from a worse first choice of the parameters.

For the solution of the inverse problem with no-error observations, a linear variogram for the prior pdf was adopted: the slope $\theta$ and the epistemic uncertainty $\sigma_R^2$ structural parameters were at the beginning set as reported in Table 5.2. The choice of a low initial value of parameter $\theta$ enforced more correlation between contiguous parameters; this avoids the spurious oscillations that may occur in the first iterations between adjacent parameters especially in a logarithmic estimation space. On the other side, the assumption of a low initial epistemic error was due to the absence of errors corrupting the observations.

The estimation of the discharge hydrograph in presence of corrupted observations was assessed by means of a Gaussian variogram, with the initial structural parameters defined as reported in Table 5.2.

### Results of the inverse methodology

Qualitative assessment of the inverse methodology is achieved by comparing the actual with the estimated inflow hydrograph, as well as the observed with the modeled water levels in the observation site. Focusing on the simulation without errors in the observations, Figure 5.12 illustrates that the estimated flood wave agrees with the actual one (a), as well as the modeled water levels overlap the measured ones (b).

The results of the simulation with random errors corrupting the observations are depicted in Figure 5.13. The estimated wave well matches the actual one: only the last values may slightly differ because fewer observations contain information about the unknown inflow hydrograph, as illustrated by the increasing range of the 95% credibility interval.

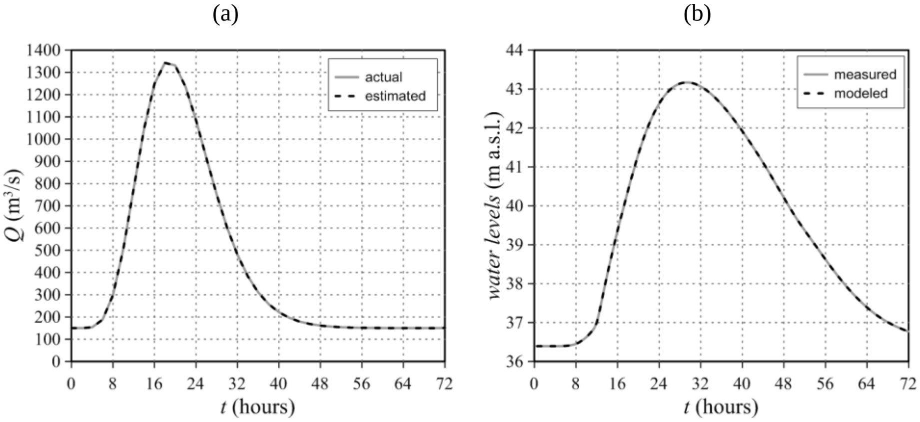


*Figure 5.12. Secchia gamma distributed inflow and uncorrupted observations: actual vs estimated inflow hydrograph (a) and observed vs modeled water levels (b).*

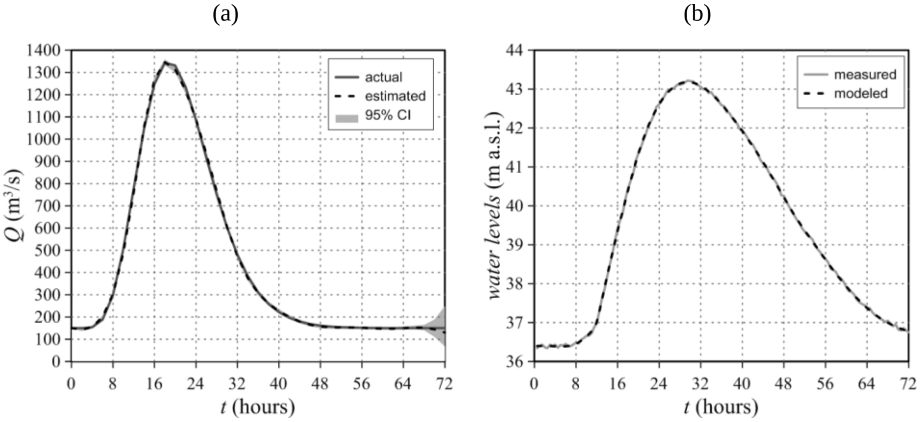


*Figure 5.13. Secchia gamma distributed inflow and corrupted observations: actual vs estimated inflow hydrograph and 95% credibility interval (a) and observed vs modeled water levels (b).*

The structural parameters estimated in presence and absence of corrupted observations are reported in Table 5.2. It is worth noting that in presence of random errors, the estimated epistemic errors correspond to the imposed ones ($10^{-3}$ m), thus confirming the accuracy of the inverse methodology.

*Table 5.2. Secchia gamma distributed inflow: initial and estimated structural parameters.*

| | | *No random errors* | *Random errors* |
|---|---|---|---|
| $\theta$ (m$^6$s$^{-3}$) | *Initial* | 1.0E-10 | - |
| | *Estimated* | 2.3994E-6 | - |
| $\sigma_R^2$(m$^2$) | *Initial* | 1.0E-8 | 1.0E-4 |
| | *Estimated* | 1.0E-8 | 1.0251E-3 |
| $\sigma_S^2$(m$^6$s$^{-2}$) | *Initial* | - | 500 |
| | *Estimated* | - | 1.4659E+1 |
| $l$ (s) | *Initial* | - | 43200 |
| | *Estimated* | - | 5.2456E+4 |

In order to assess quantitative information about the accuracy of the methodology, the resulted Nash-Sutcliffe $E_h$, root mean square error RMSE and error in the peak discharge $E_p$ values are reported in Table 5.3. The $E_h$ values exceed the 99%, the $E_p$ values are almost negligible and the RMSE error is less than 1 m$^3$/s without random errors and reaches the maximum value of 9 m$^3$/s with corrupted observations: these values confirm the accuracy of the procedure in estimating the overall shape and peak of the inflow hydrographs.

*Table 5.3. Secchia gamma distributed inflow: Nash-Sutcliffe $E_h$, root mean square error RMSE and error in the peak discharge $E_p$ values.*

| | $E_h$ (-) | $RMSE$ (m$^3$/s) | $E_p$ (%) |
|---|---|---|---|
| *No random errors* | 99.99984 | 0.490 | -0.014 |
| *Random errors* | 99.94653 | 9.205 | 0.199 |

### Focus on the computational time

With the aim of exemplifying the efficiency of the proposed parallel inverse procedure, some details about the computational times are furnished for this test case, whose main features are reported in Table 5.4.

*Table 5.4. Secchia gamma distributed inflow: characteristics of the simulation.*

| | |
|---|---|
| Number of parameters $N_p$ | 37 |
| Physical total time of the inflow hydrograph | 72 hours |
| Physical total time of the run testing the 1st parameter $p_1$, assuming 100 hours for reaching the steady state condition | 172 hours |
| Computational time of the whole inflow hydrograph simulation (72 hours) | 9.62 minutes |
| Computational time of the run testing the 1st parameter (172 hours) | 19.38 minutes |
| Number of the BGA iterations $N_i$ for the model parameter estimation | 4 |
| Number of the BGA iterations $N_o$ for the structural parameter estimation | 4 |
| Total number of simulations $N_t$ (Eq. 5.16) | 609 |

The computational time of the whole inflow hydrograph simulation (72 hours) is 9.62 minutes, whereas the simulations for evaluating the Jacobian matrix and testing parameters from 2 till 37 required a computational time progressively lower than 9.62 minutes, owning to the restart option illustrated in the Section 5.4.

In order to evaluate the total time required by the inverse procedure, it is noteworthy that dealing with an HPC cluster the global run time depends on the number of the available GPUs. However, this test was performed using 10 GPUs and the computational cost of the 609 runs was about 13 hours.

On the other side, the adoption of the serial bgaPEST procedure and the parallel 2D-SWE code as routing model would have required about 4 days of computations.

Particularly interesting is the hypothetical evaluation of the computational time for a serial BGA procedure and the adoption of a serial CPU code as forward hydraulic model. Vacondio et al. (2014) pointed out that the 2D-SWE GPU code (described in Section 1.2) led to speedup up to two order of magnitude if compared to a serial CPU code. Therefore, if a serial BGA procedure and the GPU forward model would have required about 4 computational days, the inverse problem solution with a serial forward code would ended in 400 computational days, which means that the problem could have not be faced.

## 5.6.2. 20 years-return period inflow

In the second test case, the discharge hydrograph imposed as inflow condition derived from the synthetic flood wave of 20 years-return period of the Secchia River, by discretizing the original wave with 2-hour time steps. Therefore, the modeled flood wave presented a peak value of about 780 $m^3/s$ after 18 hours. In order to increase the non-smoothness of the wave, a quite abrupt increment that separates the initial steady-state condition (100 $m^3/s$) from the rising limb was introduced (Figure 5.14-a).

Two distinct inflow hydrograph estimations were performed in order to consider the epistemic errors related to the observations. The first evaluation assumed the water levels extracted at Ponte Alto free of errors, and with truncation error resulting in a variance of $10^{-8}$ $m^2$, whereas the second one considered the observations corrupted by random errors uniformly distributed with maximum deviations of ± 0.05 m and variance $10^{-3}$ m (Figure 5.14-b).

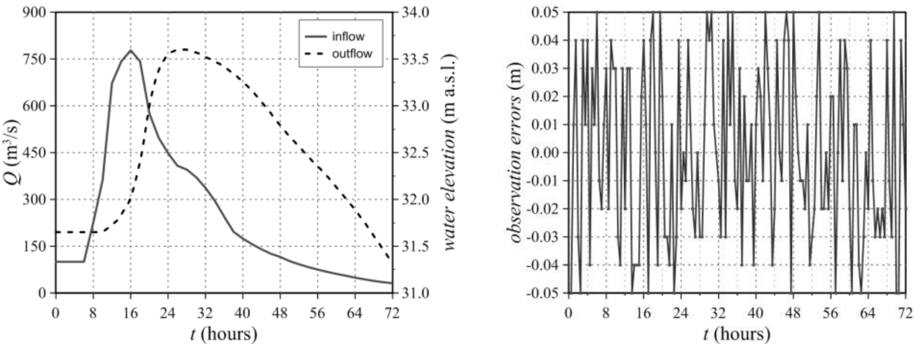

(a) (b)

*Figure 5.14. 20 years-return period inflow: inflow and outflow hydrographs (a) and observation error distribution (b).*

The inflow hydrograph ended in 72 hours and it was discretized using 2 hours steps (37 parameters), whereas the observation stage hydrograph was discretized every 0.5 hours (144 water level values).

## Initial model and structural parameters

The initial model parameter values were calculated adopting the linesearch tool of the bgaPEST (Fienen et al., 2013), whereas the structural parameters (the prior pdf) were described by means of a linear and Gaussian variogram, in the configuration with and without corrupted observations, respectively. The initial values of these last parameters are reported in Table 5.5.

Both the structural and the model parameters were estimated in a logarithmic space.

## Results of the inverse methodology

The results of the hydrograph estimations are qualitatively analyzed by comparing the actual with the estimated inflow hydrographs, and the observed with the modeled water levels in the observation point. As shown in Figure 5.15 for the simulation without corrupted observations, the estimated flood wave matches the actual one, as well as the modeled water levels agree with the measured ones.

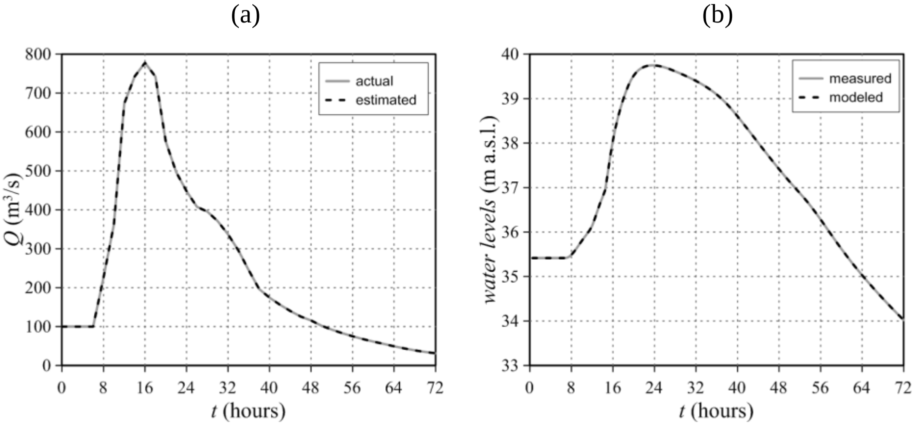


*Figure 5.15. 20 years-return period inflow and uncorrupted observations: actual vs estimated inflow hydrograph (a) and observed vs modeled water levels (b).*

The results of the simulation with random errors corrupting the observations are depicted in Figure 5.16. The estimated peak flood wave presents small discrepancies from the actual one. This essentially owns to the fact that the solution of the inverse problem is non-unique, the portion with the peak is discretized with only a few parameters and the adopted variogram smooths the solution. However, both the shape and the peak value are well captured.

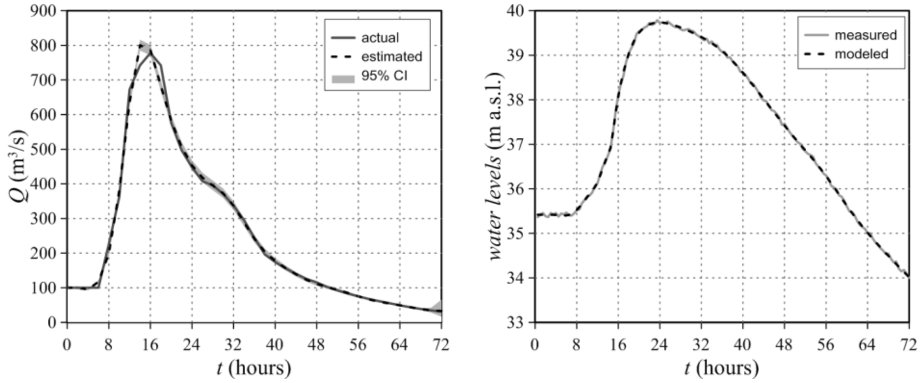

(a) (b)

*Figure 5.16. 20 years-return period inflow and corrupted observations: actual vs estimated inflow hydrograph and 95% credibility interval (a) and observed vs modeled water levels (b).*

The structural parameters estimated in presence and absence of corrupted observations are reported in Table 5.5.

*Table 5.5. 20 years-return period inflow: initial and estimated structural parameters.*

| *Random errors* | | *No random errors* | *Random errors* |
|---|---|---|---|
| $\theta$ (m$^6$s$^{-3}$) | *Initial* | 1.0E-10 | |
| | *Estimated* | 3.9727E-6 | |
| $\sigma_R^2$(m$^2$) | *Initial* | 1.0E-8 | 1.0E-4 |
| | *Estimated* | 1.0E-8 | 1.1126E-3 |
| $\sigma_S^2$(m$^6$s$^{-2}$) | *Initial* | - | 500 |
| | *Estimated* | - | 1.38296E+1 |
| $l$ (s) | *Initial* | - | 43200 |
| | *Estimated* | - | 3.8821E+4 |

The resulted indicators used for evaluating the accuracy of the methodology are reported in Table 5.6. The Nash-Sutcliff efficiency $E_h$ exceed the 99%, the errors in the peak flow $E_p$ are almost negligible and the RMSE error is less than 1 m$^3$/s without random errors and reaches the maximum value of 16 m$^3$/s with corrupted observations: these values highlight the accuracy of the procedure in estimating the overall shape and peak of the inflow hydrographs.

*Table 5.6. 20 years-return period inflow: Nash-Sutcliffe $E_h$, root mean square error RMSE and error in the peak discharge $E_p$ values.*

| | $E_h$ (-) | $RMSE$ ($m^3/s$) | $E_p$ (%) |
|---|---|---|---|
| *No random errors* | 99.99996 | 0.131 | -0.024 |
| *Random errors* | 99.44149 | 16.567 | 2.893 |

### 5.6.3. Two peak inflow

With the aim of evaluating a more challenging flood wave with more than a rising and recession limb, in this test an inflow hydrograph with two picks is presented. Particularly, the real event occurred in December 2009 on the Secchia River (used in Section 1.4.1. for the roughness calibration), is considered. However, the choice of this flood wave needs some further considerations.

The flood wave shown in Figure 5.17-a derives from a rating curve referred to Ponte Alto gauging station. Dealing with an inverse methodology, the registered water levels at Ponte Alto and Ponte Bacchello could have been used for estimating the inflow hydrograph at the upstream reservoir section. However, as previously described, the aim of the test case section is to validate the inverse methodology against actual/known flood waves, whereas the estimation of real unknown flood wave is beyond the purpose of the present chapter. Therefore, this two-peak flood wave was assumed as to test the implemented procedure with an inflow characterized by shape and total time proxy to a real one.

The discharge hydrograph (Figure 5.17-a) is characterized by a total time of 150 hours and presents a first peak value of about 380 $m^3/s$ and a second one of about 550 $m^3/s$ 45 hours later.

Two distinct inflow hydrograph estimations were performed in order to consider the epistemic errors related to the observations. The first evaluation assumed the water levels extracted at Ponte Alto free of errors, and with truncation error resulting in a variance of $10^{-8}$ $m^2$, whereas the second one considered the observations corrupted by random errors uniformly distributed with maximum deviations of ± 0.05 m and variance $9.7 \cdot 10^{-4}$ m (Figure 5.17-b).

The inflow hydrograph ended in 150 hours and it was discretized using 3 hours steps (51 parameters), whereas the observation stage hydrograph was discretized every 0.5 hours (300 water level values).

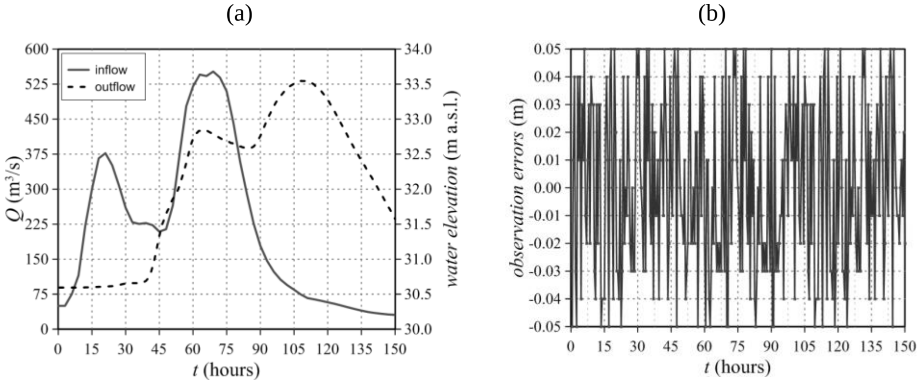

(a) (b)

*Figure 5.17. Two peak inflow: inflow and outflow hydrographs (a) and observation error distribution (b).*

## Initial model and structural parameters

The structural and the model parameters were estimated in a logarithmic space. The prior pdf was defined by means of a Gaussian variogram, and the initial structural parameters were set as reported in Table 5.7. On the other side, the initial model parameter values were calculated adopting the linesearch tool of the bgaPEST (Fienen et al., 2013).

## Results of the inverse methodology

Qualitatively assessment of the inverse methodology results is achieved by comparing the actual with the estimated inflow hydrograph, and the observed with the modeled water levels. Figures 5.18 and 5.19 illustrate the results of the simulation with and without errors in the observations, respectively; in both cases, the estimated hydrographs reproduce the actual ones.

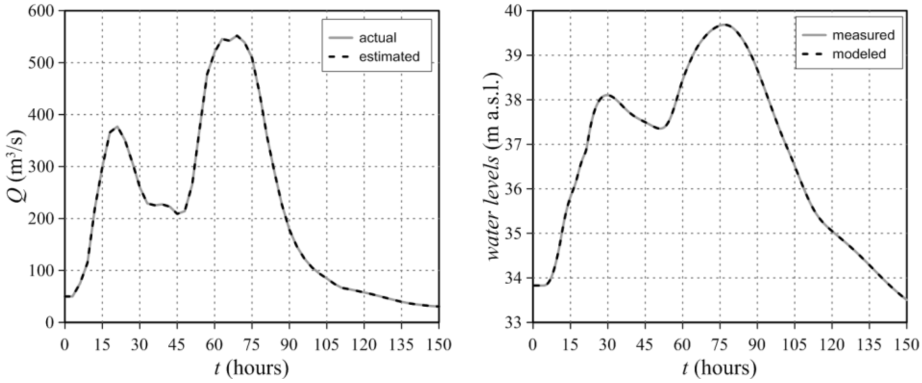

(a) (b)

*Figure 5.18. Two peak inflow and uncorrupted observations: actual vs estimated inflow hydrograph (a) and observed vs modeled water levels (b).*

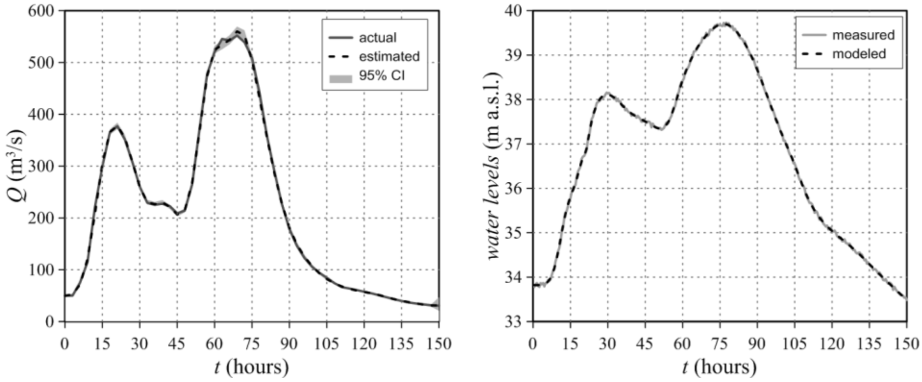

(a) (b)

*Figure 5.19. Two peak inflow and corrupted observations: actual vs estimated inflow hydrograph and 95% credibility interval (a) and observed vs modeled water levels (b).*

The structural parameters estimated in presence and absence of corrupted observations are reported in Table 5.7.

*Table 5.7. Two peak inflow: initial and estimated structural parameters.*

| *Random errors* | | *No random errors* | *Random errors* |
|---|---|---|---|
| $\sigma_R^2$(m$^2$) | *Initial* | 1.0E-8 | 1.0E-4 |
| | *Estimated* | 1.0E-8 | 9.95034E-4 |
| $\sigma_S^2$(m$^6$s$^{-2}$) | *Initial* | 500 | 500 |
| | *Estimated* | 1.99866E+3 | 9.36701E+0 |
| $l$ (s) | *Initial* | 64800 | 64800 |
| | *Estimated* | 3.2038E+4 | 5.28591E+4 |

The accuracy of the inverse methodology is assessed by means of the indicators reported in Table 5.8. In both estimations, the Nash-Sutcliff efficiency $E_h$ exceeds the 99%, the error in the peak flow $E_p$ is around 1% and the RMSE errors about 4 m$^3$/s: these values show the capability of the procedure in reconstructing also unregularly waves with more than a peak.

*Table 5.8. Two peak inflow: Nash-Sutcliffe $E_h$, root mean square error RMSE and error in the peak discharge $E_p$ values.*

| | $E_h$ (-) | $RMSE$ (m$^3$/s) | $E_p$ (%) |
|---|---|---|---|
| *No random errors* | 99.99999 | 0.055 | -0.025 |
| *Random errors* | 99.93457 | 4.294 | 1.447 |

## 5.7. Inflow hydrograph estimation on the Parma River

After having tested different types of waves on the Secchia River reach, this last test estimates a discharge hydrograph in a different river. The studied domain is a portion of the Parma River described in Section 2.3.3. and includes 20-km long reach characterized by several meanders and flood plains. Figure 5.20 illustrates the domain and the locations of the upstream boundary condition A, of the observation site B and of the downstream boundary section C.

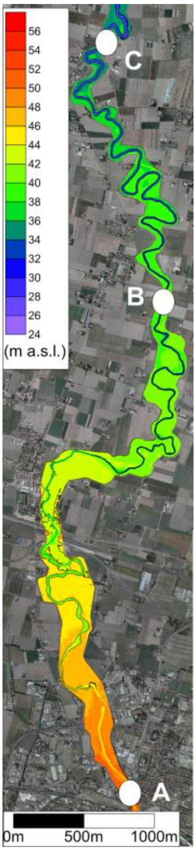


*Figure 5.20. Bathymetry of the Parma studied domain with indicated the upstream (A) and downstream (C) boundary conditions and the intermediate observation site (B).*

The bathymetry was derived from a 1 m resolution DTM obtained through a LiDAR survey carried out during the dry season, in drought condition. In the computation neither levee overtopping nor breaches were assumed.

The domain was discretized by means of a Cartesian grid with cell sizes $\Delta x = \Delta y = 4$ m and about $275 \cdot 10^3$ computing cells were adopted. The Manning roughness coefficient was assumed equal to 0.05 s/m$^{1/3}$.

All the simulations were performed adopting the first-order accurate version of the hydraulic forward model (parallel 2D-SWE code described in Section 1.2).

The initial conditions were obtained, analogously to the example illustrated in Figure 5.10 for the Secchia test cases, by achieving a steady state condition of the first parameter to estimate $p_1$.

The test case presented in this section concerns the estimation of a synthetic hydrograph with gamma distribution imposed as inflow condition on the Parma River. The discharge was calculated by means of Equations (5.22)-(5.23) and assuming the base flow $A$ = 100 m$^3$/s, the volume above the base flow $B$=3·10$^7$ m$^3$, the shape parameter $b$ = 6 and the scale parameter $k$ = 10000 s. The resulted flood wave presented a peak value of about 630 m$^3$/s at time $(b-1)k$ = 14 hours (Figure 5.21-a).

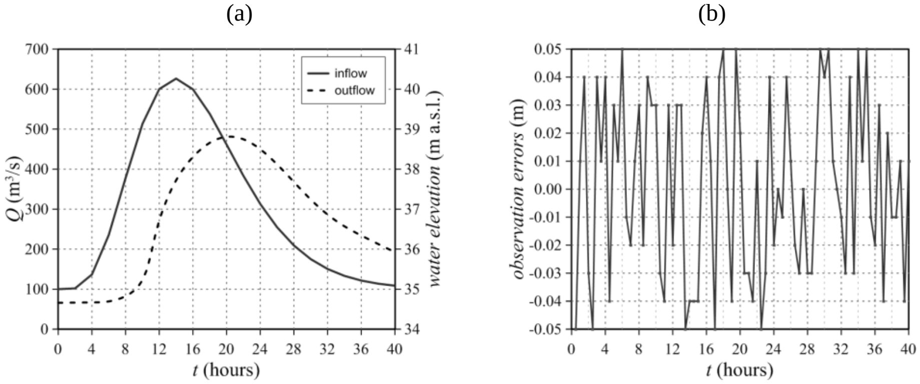


*Figure 5.21. Parma gamma distributed inflow: inflow and outflow hydrographs (a) and observation error distribution (b).*

The inflow hydrograph was estimated firstly considering the observations free of errors and with truncation error resulting in a variance of 10$^{-8}$ m$^2$. Then, the same discharge hydrograph was defined corrupting the observed water levels with random errors uniformly distributed with maximum deviations of ± 0.05 m and variance 10$^{-3}$ m (Figure 5.21-b).

The inflow hydrograph ended in 40 hours and it was discretized using 2 hours steps (21 parameters), whereas the observation stage hydrographs were discretized every 0.5 hours (80 water levels).

The structural and the model parameters were estimated in a logarithmic space. The initial model parameter values were calculated adopting the linesearch tool of the bgaPEST (Fienen et al., 2013), whereas the prior pdf was defined by means of a Gaussian variogram, and the initial structural parameters were set as reported in Table 5.9.

Considering the simulation without errors in the observations, Figure 5.22 shows that the estimated flood wave reproduces the actual one (a), and the modeled water levels agree with the measured ones (b).

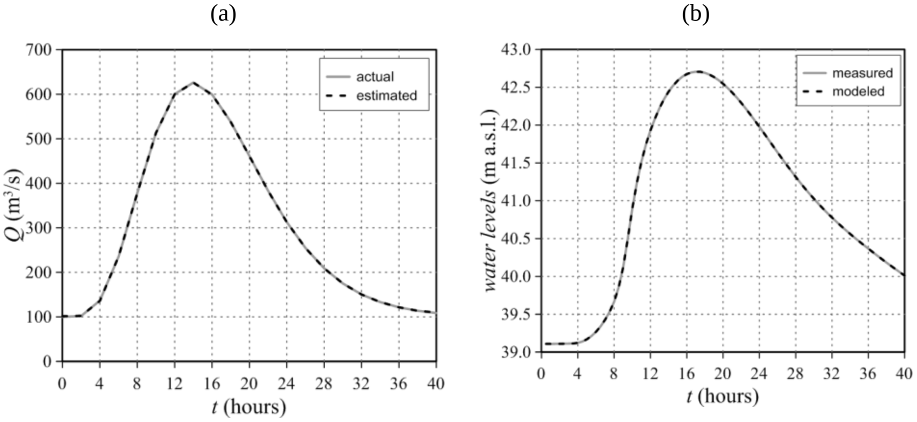


*Figure 5.22. Parma gamma distributed inflow and uncorrupted observations: actual vs estimated inflow hydrograph (a) and observed vs modeled water levels (b).*

The results of the simulation with random errors corrupting the observations are depicted in Figure 5.23. The last value of the reconstructed flood wave slightly overestimates the actual one because fewer observations contain information about the unknown inflow hydrograph. However, the estimated wave is inside the 95% credibility interval, thus confirming the good results of the solution.

It is noteworthy that the same 95% credibility interval curve widens near the final discharge, owning to the previously described issue that the more the tested parameter (in the Jacobian matrix computation) nears the end of the wave, the fewer observations contain information about the related effects.

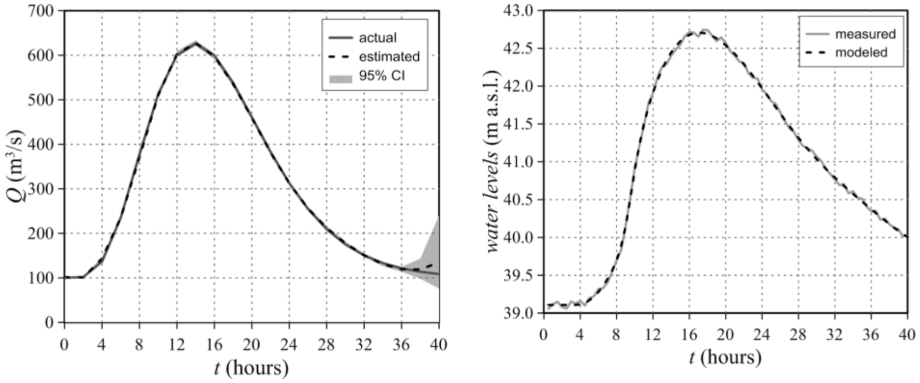

(a) (b)

*Figure 5.23. Parma gamma distributed inflow and corrupted observations: actual vs estimated inflow hydrograph and 95% credibility interval (a) and observed vs modeled water levels (b).*

The structural parameters estimated in presence and absence of corrupted observations are reported in Table 5.9.

*Table 5.9. Parma gamma distributed inflow: initial and estimated structural parameters.*

| | | *No random errors* | *Random errors* |
|---|---|---|---|
| $\sigma_R^2$(m$^2$) | *Initial* | 1.0E-8 | 1.0E-04 |
| | *Estimated* | 1.0E-8 | 1.0957E-3 |
| $\sigma_S^2$(m$^6$s$^{-2}$) | *Initial* | 500 | 500 |
| | *Estimated* | 1.0672E+3 | 5.3585E+1 |
| $l$ (s) | *Initial* | 64800 | 64800 |
| | *Estimated* | 2.8987E+4 | 5.2793E+4 |

Quantitative assessment of the methodology accuracy has been achieved by means of the Nash-Sutcliffe $E_h$, root mean square error RMSE and error in the peak discharge $E_p$ values reported in Table 5.10. The $E_h$ values are over the 99%, the $E_p$ ones are almost negligible and the RMSE error is less than 0.5 m$^3$/s without random errors and reaches the maximum value of 6 m$^3$/s with corrupted observations.

*Table 5.10. Parma gamma distributed inflow: Nash-Sutcliffe $E_h$, root mean square error RMSE and error in the peak discharge $E_p$ values.*

|  | $E_h$ (-) | $RMSE$ ($m^3/s$) | $E_p$ (%) |
|---|---|---|---|
| *No random errors* | 99.99933 | 0.4894 | -0.0407 |
| *Random errors* | 99.87651 | 6.6529 | 0.1451 |

## 5.8. Concluding remarks

In this chapter the inverse problem of estimating the unknown inflow hydrograph in an upstream-ungauged section, from the knowledge of water level observations in downstream sites has been solved by means of a Bayesian methodology.

The key aspects in the solution of this problem have been the adoption of a parallel 2D-SWE code running on GPUs and the performance of the simulations over a HPC cluster.

Firstly, the procedure for computing the Jacobian matrix has been parallelized. This has allowed achieving the highest reduction in the computational costs: the more GPUs are available on the cluster, the less time is required for the parameters estimations. Meanwhile, the adoption of BUQ grids and the optimizations implemented for reducing the grid generation time further decreased the computational costs.

Secondly, an *ad hoc* procedure has been implemented as to take advantage of any HPC cluster with GPUs, by means of the protocol SSH. Therefore, the forward model contains many instructions to prepare, and run the simulations in parallel.

For a considered case, this parallel procedure reduced the computational time of a factor 8 against running the 2D-SWE code on a single GPU. Furthermore, the analysis of the runtimes has highlighted that the use of a parallel hydraulic forward routing model is the *sine qua non* for solving this type of inverse problem, whereas the adoption of a serial code would lead to inadmissible computational times.

The inverse procedure has been applied to two different natural rivers; in both tests, no instabilities, both due to the adopted inverse procedure and to the availability of a stable, fast and accurate forward hydraulic model, arose. Moreover, the obtained results have highlighted that the implemented procedure well estimates the unknown inflow hydrographs with different shapes and in presence of corrupted observations: quantitative indicators have proved the accuracy of the methodology. In all the presented tests, the resulted Nash-Sutcliff efficiency criterion exceeded the 99%, the error in the peak discharge was less than 3% and the RMSE error less than 2%.

Future development of the methodology will focus on the possibility of reconstructing the flood waves also in presence of levee breaches and flooding outside the river region.

# Conclusions

The thesis focused on the numerical modelling of flooding events involving rural and urban areas in the framework of explicit finite volume schemes based on the Shallow Water Equations (SWEs).

Dealing with the modelling of flooding events, the first goal was the reconstruction of a real event. Since a real flood involves not only the river channel but also the flood prone areas, a fully 2D model was necessary: particularly a second order finite volume scheme solving the SWEs on a Cartesian grid was adopted.

With the aim of capturing the natural and/or artificial elements, which influence the flooding dynamics, a high-resolution mesh was used. Meanwhile, the urban areas were described by means of a different roughness coefficient that was obtained through a calibration procedure.

Even if dealing with a 5 m mesh and a fully 2D-SWE scheme, the parallel implementation on GPUs allowed reaching reasonable computational times (ratio of physical time to runtime greater than 15).

The reconstruction of a real flooding event and the addressed challenges related to the adoption of a stable, accurate and fast numerical scheme, as well as proper input data (bathymetry, breach evolution, urban areas representation, etc.), was considered as a positive starting point for more detailed investigations.

Even if the numerical model was implemented in CUDA language as to take advantage of GPUs capabilities, the goal of reducing the computational times, or of simulating larger domains, still remained.

At this aim, the implementation of the multi-resolution Block Uniform Quadtree grids allowed reproducing small-scale effects while simulating large domains, and exploited the computational capability of GPUs with minimum overheads. As a result, a decrease of the computational costs was achieved (2-10 speed-ups for theoretical tests against Cartesian grids) and the flooding simulation over a 840 km$^2$ domain achieved a ratio of physical to computational time equal to 12, whereas the adoption of Cartesian ones was not enable by the memory requirement of the Tesla K40 GPU.

With reference to urban areas, the BUQ grids allowed modelling these zones with a finer mesh, while describing the rural domain with a coarse one.

Focusing on urban environment description, the thesis investigated the SWEs with porosity, which aim at the global representation of building effects, without requiring for example a detailed information about the velocity field near the obstacles.

The first branch concerned the derivation of a well-balanced 2D model based on isotropic porosity that includes porosity by adding to the classical SWEs additional source terms. Different test cases showed the capability of the porosity model in reproducing the 1D and 2D derived reference solutions.

The second branch concerning the SWEs with isotropic porosity focused on the development of a numerical method capable of handling discontinuities in the porosity field. A 1D and 2D augmented Roe Solver was derived analyzing the physical meaning of a porosity discontinuity and enforcing mass and momentum conservation across it. The test cases showed that the proposed numerical procedure is capable of capturing different Riemann Problems, involving both shocks, rarefactions and transonic rarefactions.

Therefore, in the thesis, the issue related to urban area description in flooding models has been investigated or by modelling those regions with different roughness coefficient or BUQ grid type, or by analyzing dedicated approaches, such as the SWEs with porosity.

The last part of the thesis solved the inverse problem related to the inflow hydrograph estimation in an upstream-ungauged river section, from the knowledge of water level observations in downstream sites, adopting a Bayesian methodology, and some of the results previously assessed (a stable and fast 2D-SWE code, BUQ grids).

The implemented procedure well estimated discharge hydrographs with different shapes and in presence of corrupted observations. Moreover, by taking advantage of a HPC cluster with GPUs the parallel scheme reduced the computational time of a factor 8 against running the 2D-SWE code on a single GPU.

The results outlined in this thesis lead to the possibility of adopting a stable, accurate and fast numerical model, enabled with a proper urban area description, in order to produce hazard maps useful for flood risk definition. This kind of simulation tools opens up new perspectives in the devising and implementing of flood event management strategies for civil protection purposes and with the aim of minimizing the economic loss.

Additionally to these implications, further investigations may be performed, particularly on the porosity approaches and on the inverse problem, in order to model real cases and reconstructing the flood waves also in presence of levee breaches and flooding outside the river region, respectively.

Finally, a brief remark is pointed out regarding the urban area analysis in order to clarify that the thesis did not aim at the definition of the best approach, by testing different techniques (roughness calibration, grid type, classical vs porosity SWEs) on the same benchmark. On the contrary, the main purpose was the development of independently branches motivated by different challenges, which were interconnected one another, but each of one presented a theoretical derivation and/or implementation and required a proper validation procedure.

# References

Alcrudo, F., Benkhaldoun, F., 2001. Exact solutions to the Riemann problem of the shallow water equations with a bottom step. *Computers & Fluids*, 30(6), 643-671.

Alfieri, L., Bisselink, B., Dottori, F., Naumann, G., de Roo, A., Salamon, P., Wyser K., Feyen, L., 2017. Global projections of river flood risk in a warmer world. *Earth's Future*, 5(2), 171-182.

An, H., Yu, S., 2014. An accurate multidimensional limiter on quadtree grids for shallow water flow simulation. *Journal of Hydraulic Research*, Vol. 52, n.4, 565-574.

Aricò, C., Nasello, C., Tucciarelli, T., 2009. Using unsteady-state water level data to estimate channel roughness and discharge hydrograph. *Advances in Water Resources*, 32(8), 1223-1240.

Aricò, C., Corato, G., Tucciarelli, T., Meftah, M.B., Petrillo, A.F., Mossa, M., 2010. Discharge estimation in open channels by means of water level hydrograph analysis. *Journal of Hydraulic Research*, 48(5), 612-619.

Audusse, E., Bouchut, F., Bristeau, M.O., Klein, R., Perthame, B.T., 2004. A fast and stable well-balanced scheme with hydrostatic reconstruction for shallow water flows. *SIAM Journal on Scientific Computing*, 25(6), 2050-2065.

Aureli, F., Maranzoni, A., Mignosa, P., 2004. Two dimensional modeling of rapidly varying flows by finite volume schemes. *In River Flow* (Vol. 2004, pp. 837-847).

Aureli, F., Maranzoni, A., Mignosa, P., Ziveri, C., 2008. A weighted surface-depth gradient method for the numerical integration of the 2D shallow water equations with topography. *Advances in Water Resources*, 31(7), 962-974.

Barbetta, S., Moramarco, T., Perumal, M., 2017. A Muskingum-based methodology for river discharge estimation and rating curve development under significant lateral inflow conditions. *Journal of Hydrology*, 554, 216-232.

Bates, P., De Roo, A.P.J., 2000. A simple raster-based model for flood inundation simulation. *Journal of Hydrology*, 236, 54-77.

Begnudelli, L., Sanders, B.F., 2006. Unstructured grid finite-volume algorithm for shallow-water flow and scalar transport with wetting and drying. *Journal of hydraulic engineering*, 132(4), 371-384.

Benkhaldoun, F., Elmahi, I., Moumna, A., Seaid, M., 2016. A non-homogeneous Riemann solver for shallow water equations in porous media. *Applicable Analysis*, 95(10), 2181-2202.

Bernetti, R., Titarev, V.A., Toro, E.F., 2008. Exact solution of the Riemann problem for the shallow water equations with discontinuous bottom geometry. *Journal of Computational Physics*, 227(6), 3212-3243.

Beven, K.J., 2011. Rainfall-runoff modelling: the primer. John Wiley & Sons.

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

Borthwick, A.G.L., Marchant, R.D., Copeland, G. J. M., 2000. Adaptive hierarchical grid model of water-borne pollutant dispersion. *Advances in Water Resources,* 23(8), 849-865.

Brodtkorb, A.R., Saetra, M.L., Altinakar, M., 2012. Efficient shallow water simulations on GPUs: implementation, visualization, verification, and validation. *Computers & Fluids*, 55, 1–12.

Brufau, P., García-Navarro, P., Vázquez-Cendón, M.E., 2004. Zero mass error using unsteady wetting–drying conditions in shallow flows over dry irregular topography. *International Journal for Numerical Methods in Fluids*, 45(10), 1047-1082.

Bruwier, M., Archambeau, P., Erpicum, S., Pirotton, M., Dewals, B., 2017. Shallow-water models with anisotropic porosity and merging for flood modelling on Cartesian grids, *Journal of Hydrology*, doi: https://doi.org/10.1016/j.jhydrol.2017.09.051

Caleffi, V., Valiani, A., Zanni, A., 2003. Finite volume method for simulating extreme flood events in natural channels. *Journal of Hydraulic Research*, 41, 167-177.

Casulli, V., 1990. Semi-implicit finite difference methods for the two-dimensional shallow water equations, *Journal of Computational Physics*, 86(1), 56-74.

Casulli, V., 2015. A conservative semi-implicit method for coupled surface–subsurface flows in regional scale. *International Journal for Numerical Methods in Fluids*, 79(4), 199-214.

Cea, L., Vázquez-Cendón, M.E., 2010. Unstructured finite volume discretization of two-dimensional depth-averaged shallow water equations with porosity. *International Journal for Numerical Methods in Fluids*, 63(8), 903-930.

Chen, A.S., Evans, B., Djordjević, S., Savić, D.A., 2012. A coarse-grid approach to representing building blockage effects in 2D urban flood modelling. *Journal of hydrology*, 426, 1-16.

Costanzo, C., Macchione, F., 2006. Two-dimensional numerical simulation of flood propagation in presence of buildings. *RiverFlow*, 1, 291-301.

Costabile, P., Macchione, F., 2015. Enhancing river model set-up for 2-D dynamic flood modelling. *Environmental Modelling & Software*, 67, 89-107.

Costabile, P., Macchione, F., Natale, L., Petaccia, G., 2015. Flood mapping using LIDAR DEM. Limitations of the 1-D modeling highlighted by the 2-D approach. *Natural Hazards*, 77(1), 181-204.

Costabile, P., Costanzo, C., Macchione, F., 2016. Performances and limitations of the diffusive approximation of the 2-d shallow water equations for flood simulation in urban and rural areas. *Applied Numerical Mathematics*, 116, 141-156.

Cozzolino, L., Della Morte, R., Covelli, C., Del Giudice, G., Pianese, D., 2011. Numerical solution of the discontinuous-bottom shallow-water equations with hydrostatic pressure distribution at the step. *Advances in Water Resources*, 34(11), 1413-1426.

Crossley, A., Lamb, R., Waller, S., 2010. Fast solution of the Shallow Water Equations using GPU technology, *Proceedings of the British Hydrological Society 3rd International Symposium*, Newcastle, UK, 13–19 July 2010.

Cunge, J. A., Holly, F. M., Verwey, A., 1980. Practical aspects of computational river hydraulics, Pitman, London.

D’Alpaos, L., Brath, A., Fioravante, V., Gottardi, G., Mignosa, P., Orlandini, S., 2014. Relazione tecnico-scientifica sulle cause del collasso dell’argine del fiume Secchia avvenuto il giorno 19 gennaio 2014 presso la frazione San Matteo. Regione Emilia-Romagna, Bologna.

Das, A., 2009. Reverse stream flow routing by using Muskingum models. Sadhana, 34(3), 483-499.

De la Asunción, M., Castro, M.J., Fernández-Nieto, E., Mantas, J.M., Acosta, S.O., González-Vida J.M., 2013. Efficient GPU implementation of a two waves TVD-WAF method for the two-dimensional one layer shallow water system on structured meshes. *Computers & Fluids*, 80 (2013), pp. 441–452

De Moel, H., Jongman, B., Kreibich, H., Merz, B., Penning-Rowsell, E., Ward, P.J., 2015. Flood risk assessments at different spatial scales. *Mitigation and Adaptation Strategies for Global Change*, 20(6), 865-890.

Defina, A., D'Alpaos, L., Matticchio, B., 1994. A new set of equations for very shallow water and partially dry areas suitable to 2D numerical models. *In Modelling of Flood Propagation Over Initially Dry Areas*, (pp. 72-81). ASCE.

Defina, A. 2000. Two-dimensional shallow flow equations for partially dry areas. *Water Resources Research*, 36(11), 3251–3264.

Defina A., Viero D.P., 2010. Open channel flow through a linear contraction. *Physics of Fluids*, 22, 036602.

Di Baldassarre, G., Castellarin, A., Brath, A., 2009. Analysis of the effects of levee heightening on flood propagation: example of the River Po, Italy. *Hydrological sciences journal*, 54(6), 1007-1017.

Doherty, J.E., 2016. PEST, Model-Independent Parameter Estimation – User Manual, sixth ed. Watermark Numerical Computing, Brisbane, Australia.

Dooge, J., Bruen, M., 2005. Problems in reverse routing. *Acta Geophysica Polonica*, 53(4), 357.

D’Oria, M., Tanda, M.G., 2012. Reverse flow routing in open channels: A Bayesian Geostatistical Approach. *Journal of Hydrology*, 460, 130-135.

D’Oria, M., Mignosa, P., Tanda, M.G., 2014. Bayesian estimation of inflow hydrographs in ungauged sites of multiple reach systems. *Advances in Water Resources*, 63, 143-151.

D’Oria, M., Mignosa, P., Tanda, M.G., 2015. An inverse method to estimate the flow through a levee breach. *Advances in Water Resources*, 82, 166-175.

Dottori, F., Kalas, M., Salamon, P., Bianchi, A., Alfieri, L., Feyen, L., 2017. An operational procedure for rapid flood risk assessment in Europe. *Natural Hazards and Earth System Sciences*, 17(7), 1111.

Eli, R.N., Wiggert, J.M., Contractor, D.N., 1974. Reverse flow routing by the implicit method. *Water Resources Research*, 10(3), 597-600.

Ferrari, A., Vacondio, R., Dazzi, S., Mignosa, P., 2017. A 1D–2D Shallow Water Equations solver for discontinuous porosity field based on a Generalized Riemann Problem. *Advances in Water Resources*, 107, 233-249.

Fienen, M.N., Luo, J., Kitanidis, P.K., 2006. A Bayesian geostatistical transfer function approach to tracer test analysis. *Water Resources Research*, 42(7).

Fienen, M.N., Clemo, T., Kitanidis, P.K., 2008. An interactive Bayesian geostatistical inverse protocol for hydraulic tomography. *Water Resources Research*, 44(12).

Fienen, M., Hunt, R., Krabbenhoft, D., Clemo, T., 2009. Obtaining parsimonious hydraulic conductivity fields using head and transport observations: A Bayesian geostatistical parameter estimation approach. *Water Resources Research*, 45(8).

Fienen, M.N., D'Oria, M., Doherty, J.E., Hunt, R.J., 2013. Approaches in highly parameterized inversion: bgaPEST, a Bayesian geostatistical approach implementation with PEST: documentation and instructions (No. 7-C9). US Geological Survey.

Finaud-Guyot, P., Delenne, C., Lhomme, J., Guinot, V., Llovel, C., 2010. An approximate-state Riemann solver for the two-dimensional shallow water equations with porosity. *Journal for Numerical Methods in Fluids*, 62(12), 1299-1331.

Forzieri, G., Cescatti, A., e Silva, F.B., Feyen, L., 2017. Increasing risk over time of weather-related hazards to the European population: a data-driven prognostic study. *The Lancet Planetary Health*, 1(5), e200-e208.

Froehlich, D.C., 2016. Predicting peak discharge from gradually breached embankment dam. *Journal of Hydrologic Engineering*, 21(11), 04016041.

Gejadze, I.Y., Monnier, J., 2007. On a 2D ’zoom’ for the 1D shallow water model: coupling and data assimilation. *Computer methods in applied mechanics and engineering*, 196(45), 4628-4643.

Glickman, M.E., Van Dyk, D.A., 2007. Basic bayesian methods. *Topics in Biostatistics*, 319-338.

Greaves, D.M., Borthwick, A.G.L., 1999. Hierarchical tree-based finite element mesh generation. *International Journal for Numerical Methods in Engineering*, 45(4), 447-471.

Guinot, V., Soares-Frazão, S., 2006. Flux and source term discretization in two-dimensional shallow water models with porosity on unstructured grids. *International Journal for Numerical Methods in Fluids*, 50(3), 309-345.

Guinot, V., 2012. Multiple porosity shallow water models for macroscopic modelling of urban floods. *Advances in Water Resources*, 37, 40-72.

Guinot, V., Sanders, B.F., Schubert, J.E., 2017. Dual integral porosity shallow water model for urban flood modelling. *Advances in Water Resources*, 103, 16-31.

Guinot, V., 2017a. Consistency and bicharacteristic analysis of integral porosity shallow water models. Explaining model oversensitivity to mesh design. *Advances in Water Resources*, 107, 43-55.

Guinot, V., 2017b. A critical assessment of flux and source term closures in shallow water models with porosity for urban flood simulations. *Advances in Water Resources*, doi: 10.1016/j.advwatres.2017.09.002

Jongman, B., Hochrainer-Stigler, S., Feyen, L., Aerts, J.C.J.H., Mechler, R., Botzen, W.J.W., Bouwer, L.M., Pflug, G., Rojas, R., Ward, P.J., 2014. Increasing stress on disaster-risk finance due to large floods, *Nature Climate Change*, 4,264-268.

Juez, C., Lacasta, A., Murillo, J., García-Navarro, P., 2016. An efficient GPU implementation for a faster simulation of unsteady bed-load transport. *Journal of Hydraulic Research*, 54(3):275–288.

Kalyanapu, A. J., Shankar, S., Pardyjak, E. R., Judi, D. R., Burian, S. J., 2011. Assessment of GPU computational enhancement to a 2D flood model. *Environmental Modelling & Software*, 26(8), 1009-1016.

Kim, B., Sanders, B.F., Famiglietti, J.S., Guinot, V., 2015. Urban flood modeling with porous shallow-water equations: A case study of model errors in the presence of anisotropic porosity. *Journal of Hydrology*, 523, 680-692.

Kim, B., Sanders, B.F., 2016. Dam-Break Flood Model Uncertainty Assessment: Case Study of Extreme Flooding with Multiple Dam Failures in Gangneung, South Korea. *Journal of Hydraulic Engineering*, 142(5), 05016002.

Kitanidis, P.K., 1995. Quasi-linear geostatistical theory for inversing. *Water Resources Research*, 31(10), 2411–9.

Kurganov, K., 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.

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. *Advances in Engineering Software*, 78, 1-15.

Lacasta, A., Juez, C., Murillo, J., García-Navarro, P., 2015a. An efficient solution for hazardous geophysical flows simulation using GPUs. *Computers & Geosciences*, 78, 63-72.

Lacasta, A., Morales-Hernández, M., Murillo, J., García-Navarro, 2015b. GPU implementation of the 2D shallow water equations for the simulation of rainfall/runoff events. *Environmental Earth Sciences*, 74(11):7295–7305.

LeRoux, A.Y., 1998. Discrétisation des termes sources raides dans les problemes hyperboliques. Systemes hyperboliques: noveau schemas et nouvelles applications. Ecoles CEA-EDFINRIA “problemes non linéaires appliqués”, INRIA Rocquencourt (France).

LeVeque, R.J., 1990. Numerical Methods for Conservation Laws, Birkhäuser.

LeVeque, R.J., 2002. Finite volume methods for hyperbolic problems (Vol. 31). Cambridge university press.

Liang, Q., Borthwick, A.G.L., Stelling, G., 2004. Simulation of dam- and dyke-break hydrodynamics on dynamically adaptive quadtree grids. *International Journal for Numerical Methods in Fluids,* 46(2), 127-162.

Liang, Q., Zang, J., Borthwick, A. G., Taylor, P. H., 2007. Shallow flow simulation on dynamically adaptive cut cell quadtree grids. *International Journal for Numerical Methods in Fluids*, 53(12), 1777-1799.

Liang, Q., Du, G., Hall, J. W., Borthwick, A. G., 2008. Flood inundation modeling with an adaptive quadtree grid shallow water equation solver. *Journal of Hydraulic Engineering*, 134(11), 1603-1610.

Liang, Q., Borthwick, A.G.L., 2009. Adaptive quadtree simulation of shallow flows with wet-dry fronts over complex topography. *Computers & Fluids*, 38, 221-234.

Liang, Q., Marche, F., 2009. Numerical resolution of well-balanced shallow water equations with complex source terms. *Advances in Water Resources*, 32, 873-884.

Liang, Q., 2011. A structured but non-uniform Cartesian grid-based model for the shallow water equations. *International Journal for Numerical Methods in Fluids*, 66, 537 - 554.

Lynch, D.R., Gray, W.G., 1979. A wave equation model for finite element tidal computations. *Computers & Fluids*, 7, 207-228.

Macchione, F., 2008. Model for Predicting Floods due to Earthen Dam Breaching. I: Formulation and Evaluation. *Journal of Hydraulic Engineering*, 134: 1688-1696.

Macchione, F., Rino, A., 2008. Model for Predicting Floods due to Earthen Dam Breaching. II: Comparison with Other Methods and Predictive Use. *Journal of Hydraulic Engineering*, 134: 1697-1707.

MacDonald, I., Baines, M.J., Nichols, N.K., Samuels, P.G., 1997. Analytic benchmark solutions for open-channel flows. *Journal of Hydraulic Engineering*, 123(11), 1041-1045.

Masoero, A., Claps, P., Asselman, N.E.M., Mosselman, E., Di Baldassarre, G., 2013. Reconstruction and analysis of the Po River inundation of 1951. *Hydrological Processes*, 27(9), 1341-1348.

Mason, D.C., Schumann, G.J.P., Neal, J.C., Garcia-Pintado, J., Bates, P.D., 2012a. Automatic near real-time selection of flood water levels from high resolution Synthetic Aperture Radar images for assimilation into hydraulic models: A case study. *Remote Sensing of Environment*. Volume 124. Pages: 705-716.

Mason, D.C., Davenport, I. J., Neal, J.C., Schumann, G.J.P., Bates, P.D., 2012b. Near Real-Time Flood Detection in Urban and Rural Ares Using High-Resolution Synthetic Aperture Radar Images. *IEEE transactions on Geoscience and Remote Sensing*, 50(8), 3041-3052.

Mazzoleni, M., Bacchi, B., Barontini, S., Di Baldassarre, G., Pilotti, M., Ranzi, R., 2014. Flooding Hazard Mapping in Floodplain Areas Affected by Piping Breaches in the Po River, Italy. *Journal of Hydrologic Engineering*, 19(4), 717–731.

McMillan, H.K., Brasington, J., 2007. Reduced complexity strategies for modelling urban floodplain inundation. *Geomorphology*, 90(3), 226–243.

Michalak, A. M., Kitanidis, P.K., 2003. A method for enforcing parameter nonnegativity in Bayesian inverse problems with an application to contaminant source identification. *Water Resources Research*. 39(2).

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

Morales-Hernández, M., Lacasta, A., Murillo, J., Brufau, P., García-Navarro, P., 2014. A Comparative study of accuracy and performance between a fully 2D GPU based and a 1D-2D coupled numerical model in a real river. *Proceedings of International Conference on Hydrodinformatics*, HIC , 2014, New York City, USA.

Morales-Hernández, M., Petaccia, G., Brufau, P., García-Navarro, P., 2016. Conservative 1D–2D coupled numerical strategies applied to river flooding: The Tiber (Rome). *Applied Mathematical Modelling*, 40(3), 2087-2105.

Murillo, J., Navas-Montilla, A., 2016. A comprehensive explanation and exercise of the source terms in hyperbolic systems using Roe type solutions. Application to the 1D-2D shallow water equations. *Advances in Water Resources*, 98, 70-96.

Nash, J.E., Sutcliffe, J.V., 1970. River flow forecasting through conceptual models part I-A discussion of principles. *Journal of Hydrology*, 10(3), 282-290.

Nvidia C.U.D.A., 2007. Compute unified device architecture programming guide.

Orlandini, S., Moretti, G., Albertson, J.D., 2015. Evidence of an emerging levee failure mechanism causing disastrous floods in Italy. *Water Resources Research*, 51(10), 7995-8011.

Özgen, I., Liang, D., Hinkelmann, R., 2016a. Shallow water equations with depth-dependent anisotropic porosity for subgrid-scale topography. *Applied Mathematical Modelling*, 40(17), 7447-7473.

Özgen, I., Zhao, J., Liang, D., Hinkelmann, R., 2016b. Urban flood modeling using shallow water equations with depth-dependent anisotropic porosity. *Journal of Hydrology*, 541, 1165-1184.

Perumal, M., Moramarco, T., Sahoo, B., Barbetta, S., 2007. A methodology for discharge estimation and rating curve development at ungauged river sites. *Water Resources Research*, 43(2).

Roe, P.L., 1981. Approximate Riemann solvers, parameter vectors, and difference schemes. *Journal of Computational Physics*, 43(2), 357-372.

Rogers, B., Fujihara, M., Borthwick, A.G., 2001. Adaptive Q-tree Godunov-type scheme for shallow water equations. *International Journal for Numerical Methods in Fluids*, 35(3), 247-280.

Rosatti, G., Fraccarollo, L., 2006. A well-balanced approach for flows over mobile-bed with high sediment-transport. *Journal of Computational Physics*, 220(1), 312-338.

Rosatti, G., Murillo, J., Fraccarollo, L., 2008. Generalized Roe schemes for 1D two-phase, free-surface flows over a mobile bed. *Journal of Computational Physics*, 227(24), 10058-10077.

Rosatti, G., Begnudelli, L., 2010. The Riemann Problem for the one-dimensional, free-surface Shallow Water Equations with a bed step: Theoretical analysis and numerical simulations. *Journal of Computational Physics*, 229(3), 760-787.

Sætra, M.L., Brodtkorb, A.R., Lie K.-A., 2014. Efficient GPU-Implementation of Adaptive Mesh Refinement for the Shallow-Water Equations. *Journal of Scientific Computing*, 63 (1): 23-48.

Sanders, B.F., Bradford, S.F., 2006. Impact of Limiters on Accuracy of High-Resolution Flow and Transport Models. *Journal of Engineering Mechanics*, 132(1), 87-98.

Sanders, B.F., Schubert, J. E., Gallegos, H.A., 2008. Integral formulation of shallow-water equations with anisotropic porosity for urban flood modeling. *Journal of Hydrology*, 362(1), 19-38.

Sanders, B.F., Schubert, J.E., Detwiler, R.L., 2010. ParBreZo: A parallel, unstructured grid, Godunov-type, shallow-water code for high-resolution flood inundation modeling at the regional scale. *Advances in Water Resources*, 33(12), 1456-1467.

Sanders, B.F. 2017. Hydrodynamic Modeling of Urban Flood Flows and Disaster Risk Reduction, Oxford Research Encyclopedia, Natural Hazard Science, Oxford University Press USA doi:10.1093/acrefore/9780199389407.013.127

Schubert, J.E., Sanders, B.F., 2012. Building treatments for urban flood inundation models and implications for predictive skill and modeling efficiency. *Advances in Water Resources*, 41, 49-64.

Smith, L.S., Liang, Q., 2013. Towards a generalised GPU/CPU shallow-flow modelling tool. *Computers & Fluids*, 88, 334-343.

Soares-Frazão, S., Sillen, X., Zech, Y., 1999. Dam-break Flow through Sharp Bends Physical Model and 2D Boltzmann Model Validation. Proc., CADAM Meeting Wallingford, U.K., 2–3 March 1998, European Commission, Brussels, Belgium, 151–169.

Soares-Frazão, S., Zech, Y., 2002. Dam Break in Channels with 90° Bend. *Journal of Hydraulic Engineering*, 128(11), 956-968.

Soares-Frazão, S., Lhomme, J., Guinot, V., Zech, Y., 2008. Two-dimensional shallow-water model with porosity for urban flood modelling. *Journal of Hydraulic Research*, 46(1), 45-64.

Spada, E., Sinagra, M., Tucciarelli, T., Barbetta, S., Moramarco, T., Corato, G., 2017. Assessment of river flow with significant lateral inflow through reverse routing modeling. *Hydrological Processes*, 31(7), 1539-1557.

Stoker, J.J., 2011. Water waves: The mathematical theory with applications (Vol. 36). John Wiley & Sons.

Szymkiewicz, R., 1993. Solution of the inverse problem for the Saint Venant equations. *Journal of Hydrology*, 147(1-4), 105-120.

Toro, E.F., 1999. Riemann Solvers and Numerical Methods for Fluid Dynamics. Springer.

Toro, E.F., 2001. Shock-capturing methods for free-surface shallow flows. Chichester, UK: Wiley.

Vacondio, R., Dal Palù, A., Mignosa, P., 2014. GPU-enhanced Finite Volume Shallow Water solver for fast flood simulations. *Environmental Modelling & Software*, 57, 60-75.

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. *Natural Hazards*, 80(1), 103-125.

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. *Environmental Modelling & Software*, 88, 119-137.

Valiani, A., Begnudelli, L., 2006. Divergence form for bed slope source term in shallow water equations. *Journal of Hydraulic Engineering*, 132, 652-665.

Valiani, A., Caleffi, V., 2017. Momentum balance in the shallow water equations on bottom discontinuities. *Advances in Water Resources*, 100, 1-13.

Velickovic, M., Zech, Y., Soares-Frazão, S., 2017. Steady-flow experiments in urban areas and anisotropic porosity model. *Journal of Hydraulic Research*, 55(1), 85-100.

Viero, D.P., D'Alpaos, A., Carniello, L., Defina, A., 2013. Mathematical modeling of flooding due to river bank failure. *Advances in Water Resources*, 59, 82-94.

Viero D.P., Peruzzo P., Carniello L., Defina A., 2014. Integrated mathematical modeling of hydrological and hydrodynamic response to rainfall events in rural lowland catchments. *Water Resources Research*, 50(7), 5941-5957.

Viero, D.P., Valipour, M., 2017. Modeling anisotropy in free-surface overland and shallow inundation flows. *Advances in Water Resources*, 104, 1-14.

Yu, D., Lane, S.N., 2006. Urban fluvial flood modelling using a two-dimensional diffusion-wave treatment, part 2: Development of a sub-grid-scale treatment. *Hydrological Processes*, 20(7), 1567-1583.

Zhu, Y.H., Visser, P.J., Vrijling, J.K., 2004. Review on embankment dam breach modeling. Taylor and Francis Group, 1189-1196.

Zucco, G., Tayfur, G., Moramarco, T., 2015. Reverse flood routing in natural channels using genetic algorithm. *Water Resources Management*, 29(12), 4241-4267.