INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2010; **62**:1299–1331
Published online 20 May 2009 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/fld.2066

An approximate-state Riemann solver for the two-dimensional shallow water equations with porosity

P. Finaud-Guyot[1, 2, *, †], C. Delenne[2], J. Lhomme[2, 3], V. Guinot[2] and C. Llovel[1]

[1]*GEI (Ginger Environnement et Infrastructures) société du groupe GINGER, France*
[2]*HydroSciences Université Montpellier 2, France*
[3]*HR Wallingford, U.K.*

SUMMARY

PorAS, a new approximate-state Riemann solver, is proposed for hyperbolic systems of conservation laws with source terms and porosity. The use of porosity enables a simple representation of urban floodplains by taking into account the global reduction in the exchange sections and storage. The introduction of the porosity coefficient induces modified expressions for the fluxes and source terms in the continuity and momentum equations. The solution is considered to be made of rarefaction waves and is determined using the Riemann invariants. To allow a direct computation of the flux through the computational cells interfaces, the Riemann invariants are expressed as functions of the flux vector. The application of the PorAS solver to the shallow water equations is presented and several computational examples are given for a comparison with the HLLC solver. Copyright © 2009 John Wiley & Sons, Ltd.

Received 3 November 2008; Revised 20 March 2009; Accepted 20 March 2009

KEY WORDS: approximate-state Riemann solver; porosity; two-dimensional shallow water; PorAS solver; Godunov-type schemes; source terms

1. INTRODUCTION

The complexity of a two-dimensional flood simulation in urban area has recently led to research on the insertion of porosity in the shallow water models. In that sense, porosity can be used to 'represent the effect that the area subject to flooding is only a fraction of the total surface area' [1]. This approach has also been used to model gas flow across grids [2]. The first known derivation of shallow water flow [3] was very simple in that porosity was not fully taken into account in

*Correspondence to: P. Finaud-Guyot, GEI (Ginger Environnement et Infrastructures) société du groupe GINGER, France.
†E-mail: finaud@msem.univ-montp2.fr

Contract/grant sponsor: GEI (Ginger Environnement et Infrastructures)
Contract/grant sponsor: GINGER Group company
Contract/grant sponsor: Association Nationale de la Recherche et de la Technologie; contract/grant number: 2006/838

the equations. Thus, Hervouet *et al*. [4] have proposed a more detailed set of equations that were neither written nor solved in conservative form. Some solvers have already been proposed for hyperbolic systems of conservation laws (HSCLs) with porosity, such as a Finite Volume Roe solver for two-dimensional Euler equations with porosity [2] and an HLLC-modified Riemann solver for the two-dimensional shallow water equations with porosity [5]. An approximate-state (AS) solver was then proposed [6] for HSCLs with source terms. The present paper aims to propose a new AS solver, named PorAS, to solve HSCLs involving topography- and porosity-driven source terms.

In 1959, Godunov [7] proposed to solve HSCLs by considering finite volumes where the calculated value of each variable is assumed to be the average of the variable over the volume. The fluxes between the volumes can be calculated by solving a Riemann problem at the interface. Theoretically, an exact solution of the Riemann problem can be computed but this is a time-consuming procedure, which is not used effectively even with the development of powerful computers. Therefore, many different approaches have been proposed in the last 50 years to solve the Riemann problem. The different developed solvers can be classified into two categories: those solving a simplified Riemann problem in an exact way and those solving the exact Riemann problem with simplified equations.

Considering simplified Riemann problem solved in an exact way, Roe's solver is based on the assumption that the Jacobian matrix is constant [8]. The matrix is calculated using the consistency and conservation conditions. Since the calculation of the the matrix terms can be difficult, a new algorithm was proposed [9] to solve the Riemann problem without computing the whole Jacobian matrix. A second improvement has been provided to the initial Roe's solver in order to enforce the entropy condition [9–12].

The primitive variable Riemann solver (PVRS) [12] uses the so-called primitive variables (which are most of the time not the conserved variables) to express the Riemann problem to solve. This method is based on the Rankine–Hugoniot relation to compute the flow through the discontinuities and the generalized Riemann invariants for rarefaction waves.

For approximate Riemann solvers—as for many others—being quite inefficient close to strong gradients of the variables, the logical rationale is to combine simple Riemann solvers in regions of smooth flow and more sophisticated ones elsewhere in an 'Adaptive Riemann solver'. One of the first proposed [13] involves the PVRS and the exact Riemann solver. It was first implemented for steady supersonic solutions of the Euler equations [14] and later adapted to an MUSCL scheme [15].

Introduced by Harten *et al*. [16], the HLL solver solves an exact Riemann problem with simplified equations. It is based on the assumption of two waves separating three regions of constant state with an *a priori* determination of the waves' speed. Direct estimation of the flux through the interface can be established by the Rankine–Hugoniot relations through each discontinuity. As noticed by Guinot [17], the HLL solver can be applied to any HSCL, whatever its size. However, contact discontinuities are smoothed for the 3×3 HSCLs. An efficient adaptation of the HLL solver is the HLLC solver [18] that uses three waves separating two intermediate regions of constant state from the left and right states of the Riemann problem. The treatment of the discontinuities has also been improved to reduce the numerical diffusion as much as possible.

Osher [19] splits the Jacobian matrix into two matrices containing, respectively, the positive and the negative eigenvalues, and computes the flux vector as the sum of two vector-valued functions.

Another solver, independent of any physical relation [20], uses the Rankine–Hugoniot relation to establish the flux through the left and the right discontinuities; a heuristic relation being used to establish the missing equation in the system to solve.

DOI: 10.1002/fld

Unlike the previously quoted solvers, AS solvers do not determine the wave configuration to use the appropriate relation through each discontinuity. In [21] is shown that the relations used for the shock waves differ from the ones used for the rarefaction waves only by a third-order term with respect to the conservative variable. The same relations are thus used regardless of the nature of the waves: the Rankine–Hugoniot relation is used when all the waves are assumed to be shocks, whereas the generalized Riemann invariants are used when rarefaction waves are considered. Riemann solvers based on a two-shock assumption have been proposed in [22, 23] for the Euler equations of gas dynamics. The assumption of two rarefaction waves has been used for example to solve the water hammer equations [24], a system of equations describing two-phase flows in pipes [25], shock wave propagation in liquids [26] or the Euler equations of gas dynamics [27].

All the above-mentioned solvers are designed for HSCLs without source terms. Consequently, instability may arise in the presence of source terms induced by the porosity or the bottom slope for the shallow water equation. Different methods have been developed to deal with such issues. The time-splitting method is a multi-step algorithm that first solves the hyperbolic system without source terms (i.e. the conservation part of the equation) and then corrects the computed value by taking into account the source terms. Since the time-splitting method is quite efficient except when the solution is close to a steady state, LeVeque [28] proposes the Quasi-Steady Wave-Propagation Algorithm. This algorithm modifies the Riemann problems at the cell edges by introducing an extra Riemann problem at the centre of the cell. Another splitting method (Source term upwinding) consists of assigning the source terms to both cells across the interface according to the wave celerity in the cells [29]. The topography can also be incorporated as part of the conserved variables vector via a partial differential equation with respect to time [30]. This leads to new Riemann problems that are solved in an exact way [1, 30] or with approximate Riemann solvers [31]. The Quasi-Steady algorithm and the Source term upwinding technique are examples of well-balanced schemes.

The PorAS solver presented in this paper is based on the Riemann invariants through a two rarefaction wave assumption, as proposed by Lhomme and Guinot [6] for shallow water equations without porosity. The developments with porosity were initiated in [32] but did not succeed. To allow a direct computation of the flux through the interface, the Riemann invariants are expressed as functions of the flux vector. The following section describes the governing equations and the solution principle applied to the shallow water equations. Computational examples are given in Section 3 to compare the results obtained using the PorAS and HLLC solvers. Conclusion and discussion are given in Section 4.

2. GOVERNING EQUATIONS

2.1. General framework

The overall goal of the proposed solver is to solve the hyperbolic system of conservation laws for the one-dimensional projection of the two-dimensional shallow water equations with source terms arising from non-uniform porosity and bottom elevation. This system can be written in vector form as

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}}{\partial x} = \mathbf{S} \tag{1}$$

DOI: 10.1002/fld

where **U** is the conserved variable, **F** is the flux, **S** the source term and t and x are, respectively, the time and the space coordinates. The shallow water equations with porosity (i.e. the continuity and the momentum equation in each direction) written in the form of (1) leads to define **U**, **F** and **S** as

$$\mathbf{U}=\begin{bmatrix}\phi h\\ \phi hu\\ \phi hv\end{bmatrix},\quad \mathbf{F}=\begin{bmatrix}\phi hu\\ \phi hu^2+\frac{1}{2}g\phi h^2\\ \phi huv\end{bmatrix},\quad \mathbf{S}=\begin{bmatrix}0\\ S_0-S_\mathrm{f}+\frac{1}{2}gh^2\frac{\partial\phi}{\partial x}\\ 0\end{bmatrix} \tag{2}$$

where ϕ is the porosity, g the gravitational acceleration, h the water depth, u and v the velocities in the x and y directions, respectively, S_0 and S_f the source terms corresponding to the bottom slope and to the friction in the x direction, respectively. The detailed calculation has been explained in [5, 32]. The source terms S_0 and S_f are defined as

$$S_0=-gh\phi\frac{\partial z_\mathrm{b}}{\partial x},\quad S_\mathrm{f}=gh\phi\frac{(u^2+v^2)^{1/2}}{K^2h^{4/3}}u \tag{3}$$

where z_b is the bottom elevation and K is the Strickler coefficient. The definition of the friction term S_f accounts only for friction on the river bed. In previous works on the subject [5, 33, 34], the friction term S_f was modified to account also for energy losses due to flow regime variations, multiple reflections waves, etc. As shown in Section 2.2, the PorAS solver do not take the friction term into account because its computation is made in a different step, using the time-splitting algorithm. Therefore, the friction term definition does not matter for the purpose of the present work. It can be proved that the system of conservation laws (1) is hyperbolic, i.e. the Jacobian matrix of the flux **F** with respect to **U** exists and has three real and distinct eigenvalues.

Equation (1) is discretized as follows:

$$\mathbf{U}_i^{n+1}=\mathbf{U}_i^n-\frac{\Delta t}{A_i}\sum_{j\in N(i)}[\mathbf{P}_{i,j}\mathbf{F}_{i,j}^{n+1/2}w_{i,j}+(\mathbf{S}_{i,j}^{n+1/2})_i] \tag{4}$$

where

$$\mathbf{P}_{i,j}=\begin{pmatrix}1 & 0 & 0\\ 0 & n(x) & n(y)\\ 0 & -n(y) & n(x)\end{pmatrix}$$

is the matrix that accounts for the coordinate change from the global (x, y) coordinate system to the local one (ξ, ψ), attached to the interface (see Figure 1), $\mathbf{U}_i^n$ is the average of **U** over the cell i at the time step n, Δt is the computational time step, A_i is the area of the cell i, $N(i)$ is the set of neighbouring cells of i, $\mathbf{F}_{i,j}^{n+1/2}$ is the average of **F** between the time step n and $n+1$ in the normal direction, $w_{i,j}$ is the width of the interface between i and j, $\mathbf{S}_{i,j}^{n+1/2}$ is the contribution of the source terms to the cell i computed from **U** between the cells i and j.

In one dimension, this discretization can be simplified as:

$$\mathbf{U}_i^{n+1}=\mathbf{U}_i^n+\frac{\Delta t}{\Delta x_i}(\mathbf{F}_{i-1/2}^{n+1/2}-\mathbf{F}_{i+1/2}^{n+1/2})+\Delta t(\mathbf{S}_{i-1/2,i}^{n+1/2}+\mathbf{S}_{i+1/2,i}^{n+1/2}) \tag{5}$$

DOI: 10.1002/fld

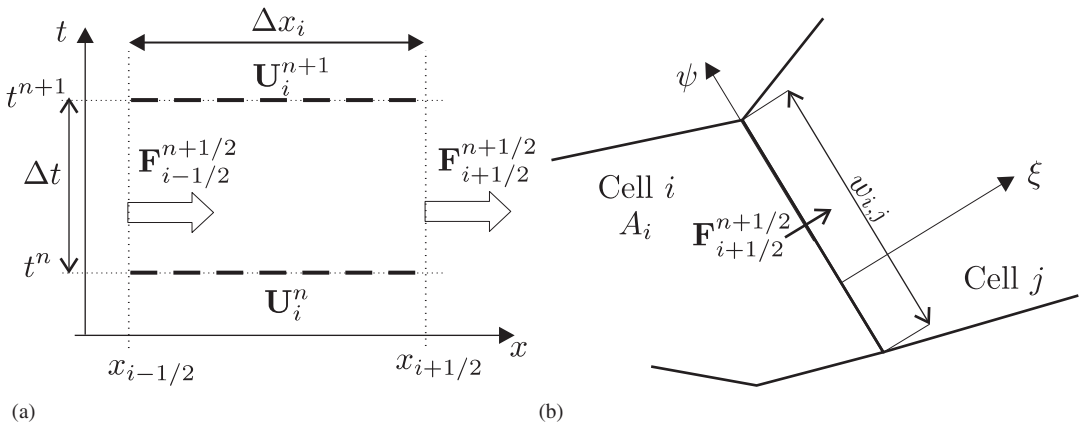

Figure 1. Definition sketch for the finite volume formalism in one dimension (a) and two dimensions (b).

If an explicit method is used to estimate the fluxes (i.e. the fluxes $\mathbf{F}$ are computed using the values of $\mathbf{U}$ at the time step n), the Courant–Friedrichs–Lewy condition must be used to ensure the stability of the numerical solution

$$\frac{\max_p(|\lambda_p|\Delta t)}{\Delta x_i} \leqslant 1 \quad \forall i \tag{6}$$

where λ_p is the pth eigenvalue of the Jacobian matrix $\partial\mathbf{F}/\partial\mathbf{U}$.

It is also possible to generalize (1) to multi-dimensional space problem, but in all cases, the flow variable in a given cell is computed from the balance between the fluxes at the interfaces of the cell and from the contributions of the source terms. As the flux and source terms are frequently computed by solving the one-dimensional equations in the local coordinate system (ξ, ψ), the computation of the one-dimensional equations is a key issue. In [7] is proposed the computation of the flux $\mathbf{F}_{i+1/2}^{n+1/2}$ by solving a Riemann problem at the interface $i+\frac{1}{2}$ between the time step n and $n+1$. The Riemann problem is defined as follows:

$$\begin{aligned} &\frac{\partial\mathbf{U}}{\partial t} + \frac{\partial\mathbf{F}}{\partial x} = \mathbf{S} \\ &\mathbf{U}(x, t^n) = \begin{cases} \mathbf{U}_\mathrm{L} & \text{for } x < x_{i+1/2} \\ \mathbf{U}_\mathrm{R} & \text{for } x > x_{i+1/2} \end{cases} \end{aligned} \tag{7}$$

$\mathbf{U}_\mathrm{L}$ and $\mathbf{U}_\mathrm{R}$ are, respectively, called the left and right state of the Riemann problem.

In most cases, the Riemann solvers do not take the source terms into account in the conservation part of the equation. In such solvers, equilibrium conditions may not be verified. This is the case for instance for water at rest over a non-horizontal bottom. The water depth is not the same on each side of the interface. In the Riemann problem for the shallow water equations, the flux vector computation leads to non-zero discharge and impulse (i.e. the two first components of the flux vector are not nil) [6]. The first component of the source term vector being nil by definition, the non-zero discharge cannot be compensated by the source term discretization. In the second equation, the non-zero impulse can be compensated by a correct discretization of the second component of

DOI: 10.1002/fld

the source term vector (for instance, by using source term upwinding [29, 35]), but this is not the case in the continuity equation. Since equilibrium conditions are not verified, spurious oscillation may arise. In [36] it is proposed to avoid this problem by considering water surface elevation instead of water depth in the shallow water equations. Problems still remain if the variation of the channel width is taken into account in the source terms. For the PorAS solver, equilibrium conditions are satisfied by taking the source terms into account in the Riemann solver. The nature of the waves in the Riemann problem is determined *a priori*. Since characteristic relationships across the rarefaction waves can be approximated by the jump relationships [21, 23], the jump relationships or the Riemann invariants may be used indifferently to provide an approximation of the intermediate states in the solution. The PorAS solver assumes that all waves are rarefaction waves, which simplifies the source terms treatment. This assumption is used to write the Riemann invariant relationships across the waves. Such relationships are generally used to compute the value of **U** and then to estimate **F**. Problems may arise since such a rationale requires the estimation of **S** at the interface; **S** being not necessarily constant on both sides of the interface. Examples with the shallow water equations show that **S** depends on the value of **U** on both sides of the interface. This burden is eliminated in the proposed approach by computing **F** directly at the interface.

2.2. *Flux and source term computation*

This subsection presents the developments for the proposed solver. The solution **U** and the flux **F** are supposed to be differentiable. Equation (1) is multiplied on the left-hand side by the Jacobian matrix **A**. This leads to

$$\mathbf{A}\frac{\partial \mathbf{U}}{\partial t}+\mathbf{A}\frac{\partial \mathbf{F}}{\partial x}=\mathbf{AS} \tag{8}$$

where **A** is given by

$$\mathbf{A}=\begin{pmatrix} 0 & 1 & 0 \\ c^2-u^2 & 2u & 0 \\ -uv & v & u \end{pmatrix} \tag{9}$$

which can also be written as

$$\frac{\partial \mathbf{F}}{\partial t}+\mathbf{A}\frac{\partial \mathbf{F}}{\partial x}=\mathbf{AS} \tag{10}$$

where c is the propagation speed of the pressure waves defined as:

$$c=\left[\frac{\mathrm{d}}{\mathrm{d}h}\left(\frac{gh^2}{2}\right)\right]^{1/2}=(gh)^{1/2} \tag{11}$$

The matrix $\mathbf{\Lambda}$ of the eigenvalues of **A**, the matrix **K** of the eigenvectors of **A** and the inverse matrix of **K** are given by:

$$\mathbf{\Lambda}=\begin{pmatrix} u-c & 0 & 0 \\ 0 & u & 0 \\ 0 & 0 & u+c \end{pmatrix},\quad \mathbf{K}=\begin{pmatrix} 1 & 0 & 1 \\ u-c & 0 & u+c \\ v & 1 & v \end{pmatrix},\quad \mathbf{K}^{-1}=\frac{1}{2c}\begin{pmatrix} u+c & -1 & 0 \\ -2cv & 0 & 2c \\ -(u-c) & 1 & 0 \end{pmatrix} \tag{12}$$

DOI: 10.1002/fld

The wave celerities are therefore given by:

$$\begin{aligned}\lambda_1 &= u-c\\ \lambda_2 &= u\\ \lambda_3 &= u+c\end{aligned} \tag{13}$$

Equation (10) is left multiplied by $\mathbf{K}^{-1}$

$$\mathbf{K}^{-1}\frac{\partial \mathbf{F}}{\partial t}+\mathbf{K}^{-1}\mathbf{A}\frac{\partial \mathbf{F}}{\partial x}=\mathbf{K}^{-1}\mathbf{A}\mathbf{S} \tag{14}$$

Introducing the term $\mathbf{\Lambda}=\mathbf{K}^{-1}\mathbf{A}\mathbf{K}$ leads to:

$$\mathbf{K}^{-1}\frac{\partial \mathbf{F}}{\partial t}+\mathbf{\Lambda}\mathbf{K}^{-1}\frac{\partial \mathbf{F}}{\partial x}=\mathbf{\Lambda}\mathbf{K}^{-1}\mathbf{S} \tag{15}$$

Introducing the definition $\mathrm{d}\mathbf{W}=\mathbf{K}^{-1}\mathrm{d}\mathbf{F}$, Equation (15) can be rewritten as

$$\frac{\partial \mathbf{W}}{\partial t}+\mathbf{\Lambda}\frac{\partial \mathbf{W}}{\partial x}=\mathbf{\Lambda}\mathbf{S}' \tag{16}$$

where the source term $\mathbf{S}'$ is defined by:

$$\mathbf{S}'=\mathbf{K}^{-1}\mathbf{S} \tag{17}$$

From (2) and (12), vectors $\mathrm{d}\mathbf{W}$ and $\mathbf{S}'$ are therefore given by:

$$\mathrm{d}\mathbf{W}=\frac{1}{2c}\begin{bmatrix}\lambda_3\,\mathrm{d}F_1-\mathrm{d}F_2\\ -2cv\,\mathrm{d}F_1+2c\,\mathrm{d}F_3\\ -\lambda_1\,\mathrm{d}F_1+\mathrm{d}F_2\end{bmatrix},\quad \mathbf{S}'=\frac{1}{2c}\begin{bmatrix}S_\mathrm{f}-S_0-\frac{1}{2}gh^2\frac{\partial\phi}{\partial x}\\ 0\\ S_0-S_\mathrm{f}+\frac{1}{2}gh^2\frac{\partial\phi}{\partial x}\end{bmatrix} \tag{18}$$

The overall algorithm used to solve the shallow water equations with porosity (1) and (2) is the same as in [5]. Therefore, the present Riemann solver computes only the flux and the part of the source term that arises from the bottom slope and the porosity gradient. The above-mentioned friction term is computed separately in a different step of the overall algorithm. Therefore, the friction terms are not taken into account in the formulation of the Riemann solver; i.e. S_f is zero in (18). Equation (16) is equivalent to the following set of differential relationships:

$$\frac{\mathrm{d}W_k}{\mathrm{d}t}=\lambda_k S'_k \quad \text{along} \quad \frac{\mathrm{d}x}{\mathrm{d}t}=\lambda_k,\quad k=1,\ldots,m \tag{19}$$

where the subscript k corresponds to the kth component of the vector in (16) and where

$$\mathrm{d}W_k=\sum_{l=1}^{3}K^{-1}_{l,k}\,\mathrm{d}F_l \tag{20}$$

where $K^{-1}_{l,k}$ is the value on the lth column and the kth row of the matrix $\mathbf{K}^{-1}$ and F_l is the lth component of the flux vector $\mathbf{F}$. The Riemann invariants of the system are the W_k if the vector

DOI: 10.1002/fld

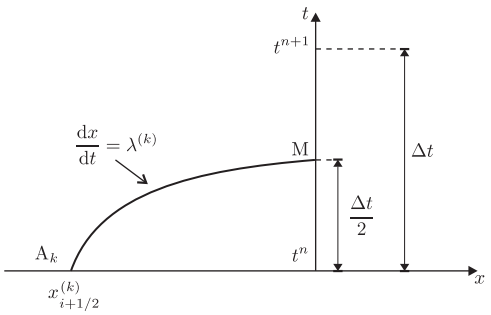

Figure 2. Definition sketch for the integration path along the characteristic.

d**W** can be integrated. It was shown in [37] that it is not possible to integrate d**W** to provide independent Riemann invariants except in particular cases (e.g. for the 2×2 system of conservation laws). The method presented hereafter is thus based on the general formulation (19).

To estimate the flux $\mathbf{F}^{n+1/2}_{i+1/2}$, (19) must be integrated between the points $\mathbf{M}(x_{i+1/2}, t_{n+1/2})$ and the foot A_k of the kth characteristic leading to M: $\mathrm{d}x/\mathrm{d}t = \lambda_k$ (see Figure 2). Assuming that λ_k is constant along the characteristic, the integration of (19) leads to

$$\int_{\mathrm{A}_k}^{\mathrm{M}} \mathrm{d}W_k = \int_{t^n}^{t^{n+1/2}} \lambda_k S'_k \,\mathrm{d}t \approx \frac{\Delta t}{2}\lambda_k \overline{S'_k}, \quad k=1,2,3 \tag{21}$$

where $\overline{S'_k}$ is the average of S'_k along the integration path. Substituting (20) into (21) leads to:

$$\int_{\mathrm{A}_k}^{\mathrm{M}} \sum_{l=1}^{3} K^{-1}_{l,k}\,\mathrm{d}F_l = \frac{\Delta t}{2}\lambda_k \overline{S'_k}, \quad k=1,2,3 \tag{22}$$

The coefficients of the matrix $\mathbf{K}^{-1}$ are assumed to be constant along the characteristics [6]. The coefficients in the kth row of the matrix are thus taken equal to the coefficients on the side of the interface to which the foot A_k of the kth characteristic belongs:

$$K^{-1}_{l,k} = \begin{cases} K^{-1}_{l,k}(\mathbf{U}_{\mathrm{L}}) & \text{if } \lambda_k > 0 \\ K^{-1}_{l,k}(\mathbf{U}_{\mathrm{R}}) & \text{if } \lambda_k < 0 \end{cases} \tag{23}$$

Since the coefficients of the matrix $\mathbf{K}^{-1}$ and λ_k are assumed to be constant along the integration path, (22) can be written as

$$\int_{\mathrm{A}_k}^{\mathrm{M}} \sum_{l=1}^{3} K^{-1}_{l,k}\,\mathrm{d}F_l \approx \sum_{l=1}^{3} K^{-1}_{l,k} \int_{\mathrm{A}_k}^{\mathrm{M}} \mathrm{d}F_l = \frac{\Delta t}{2}\lambda_k \overline{S'_k}, \quad k=1,2,3 \tag{24}$$

and therefore

$$\sum_{l=1}^{3} K^{-1}_{l,k}\,(F_{l,\mathrm{M}} - F_{l,A_k}) = \frac{\Delta t}{2}\lambda_k \overline{S'_k}, \quad k=1,2,3 \tag{25}$$

DOI: 10.1002/fld

where $F_{l,M}$ and F_{l,A_k} represent the lth component of the vector $\mathbf{F}$, respectively, in M and A_k. A_k is located on the left-hand side of the interface when λ_k is positive and on the right-hand side when λ_k is negative. Therefore,

$$F_{l,A_k} = \begin{cases} F_{l,\mathrm{L}} & \text{if } \lambda_k > 0 \\ F_{l,\mathrm{R}} & \text{if } \lambda_k < 0 \end{cases} \tag{26}$$

The value of $\mathbf{F}$ at $M(x_{i+1/2}, t_{n+1/2})$ is computed by solving the 3×3 system of equations (25). This value is assumed to be equal to the average value $\mathbf{F}_{i+1/2}^{n+1/2}$ over the time step $n,\ n+1$. Prior to resolution, the average source term $\overline{S'_k}$ is estimated by

$$\overline{S'_k} = S'_k\left(\mathbf{U}_{\mathrm{LR}}, \left(\frac{\partial z_\mathrm{b}}{\partial x}\right)_{\mathrm{LR}}, \left(\frac{\partial \phi}{\partial x}\right)_{\mathrm{LR}}\right) \tag{27}$$

where

$$\left(\frac{\partial z_\mathrm{b}}{\partial x}\right)_{\mathrm{LR}} \approx \frac{z_{\mathrm{b_R}} - z_{\mathrm{b_L}}}{(\lambda_3 - \lambda_1)\dfrac{\Delta t}{2}} \tag{28a}$$

$$\left(\frac{\partial \phi}{\partial x}\right)_{\mathrm{LR}} \approx \frac{\phi_\mathrm{R} - \phi_\mathrm{L}}{(\lambda_3 - \lambda_1)\dfrac{\Delta t}{2}} \tag{28b}$$

correspond to the average slope over the domain of dependence of the solution. Combining (28) with (2) and (3) leads to

$$S = -g(\phi h)_{\mathrm{LR}} \frac{z_{\mathrm{b_R}} - z_{\mathrm{b_L}}}{(\lambda_3 - \lambda_1)\dfrac{\Delta t}{2}} + \frac{1}{2} g (h_{\mathrm{LR}})^2 \frac{\phi_\mathrm{R} - \phi_\mathrm{L}}{(\lambda_3 - \lambda_1)\dfrac{\Delta t}{2}} \tag{29}$$

where $(\phi h)_{\mathrm{LR}}$ is estimated at the interface between the left and right cells and h_{LR} is the depth at the interface. These variables may depend on the considered characteristic.

The eigenvalues of $\mathbf{A}$ are assumed to be constant. Substituting (12) and (18) in (25) leads to:

$$\lambda_3(F_1 - F_{1,\mathrm{R}}) - (F_2 - F_{2,\mathrm{R}}) = -\frac{\Delta t}{2}\lambda_1 S \quad \text{for } \frac{\mathrm{d}x}{\mathrm{d}t} = \lambda_1 \tag{30a}$$

$$-v(F_1 - F_{1,A}) + (F_3 - F_{3,A}) = 0 \quad \text{for } \frac{\mathrm{d}x}{\mathrm{d}t} = \lambda_2 \tag{30b}$$

$$-\lambda_1(F_1 - F_{1,\mathrm{L}}) + (F_2 - F_{2,\mathrm{L}}) = \frac{\Delta t}{2}\lambda_3 S \quad \text{for } \frac{\mathrm{d}x}{\mathrm{d}t} = \lambda_3 \tag{30c}$$

with

$$F_{l,A} = \begin{cases} F_{l,\mathrm{L}} & \text{if } \lambda_2 > 0 \\ F_{l,\mathrm{R}} & \text{if } \lambda_2 < 0 \end{cases} \tag{31}$$

Since (30a) and (30c) are written for two different waves, the expressions $(\phi h)_{\mathrm{LR}}$ and (h_{LR}) used in (29) for the definition of S may be different. In the following, the variables at the interface are so written with the superscripts $^{(1)}$, $^{(2)}$ and $^{(S)}$; corresponding, respectively, to the variables at the

DOI: 10.1002/fld

interface for the first and second component of the flux vector and for the source term. The next subsection is devoted to their evaluation. Combining (29) with (30) leads to the following system of equations that is valid for the flux vector in the intermediate region of constant state (marked by the star superscript):

$$F_1^* = \frac{1}{\lambda_3-\lambda_1}(F_{2,\mathrm{L}} - F_{2,\mathrm{R}} - \lambda_1 F_{1,\mathrm{L}} + \lambda_3 F_{1,\mathrm{R}}) + \frac{1}{\lambda_3-\lambda_1}\left[\frac{1}{2}g(h_{\mathrm{LR}}^{(1)})^2(\phi_\mathrm{R}-\phi_\mathrm{L}) - g(\phi h)_{\mathrm{LR}}^{(1)}(z_{\mathrm{b_R}} - z_{\mathrm{b_L}})\right] \tag{32a}$$

$$F_2^* = \frac{1}{\lambda_3-\lambda_1}(\lambda_3 F_{2,\mathrm{L}} - \lambda_1 F_{2,\mathrm{R}} - \lambda_1\lambda_3(F_{1,\mathrm{L}} - F_{1,\mathrm{R}})) + \frac{\lambda_3+\lambda_1}{\lambda_3-\lambda_1}\left[\frac{1}{2}g(h_{\mathrm{LR}}^{(2)})^2(\phi_\mathrm{R}-\phi_\mathrm{L}) - g(\phi h)_{\mathrm{LR}}^{(2)}(z_{\mathrm{b_R}} - z_{\mathrm{b_L}})\right] \tag{32b}$$

$$F_3^* = v(F_1^* - F_{1,A}) + F_{3,A} \tag{32c}$$

2.3. Estimation of the variables at the interface

The estimation of the different variables at the interface is made by using physical cases. One should notice that porosity can be considered as the effective width of the computing cell available for the flow. In a one-dimensional problem, involving a spatial variation of the porosity or the width leads to the same set of equations.

2.3.1. Estimation of $(\phi h)_{\mathrm{LR}}^{(1)}$ *and* $h_{\mathrm{LR}}^{(1)}$. These values are determined using the continuity and momentum equations. The continuity equation can be written as:

$$(\phi hu)_\mathrm{L} = (\phi hu)_\mathrm{R} = (\phi hu)^* = q \tag{33}$$

Combining (33) with the expression of the flux F_1^* in (32a) leads to:

$$F_1^* = \frac{1}{\lambda_3-\lambda_1}\left[q(u_\mathrm{L}-u_\mathrm{R}) + \frac{1}{2}g((\phi h^2)_\mathrm{L} - (\phi h^2)_\mathrm{R})\right] + \frac{1}{\lambda_3-\lambda_1}\left[\frac{1}{2}g(h_{\mathrm{LR}}^{(1)})^2(\phi_\mathrm{R}-\phi_\mathrm{L}) - g(\phi h)_{\mathrm{LR}}^{(1)}(z_{\mathrm{b_R}} - z_{\mathrm{b_L}})\right] + q \tag{34}$$

In a steady-state case, F_1^* is equal to the flow q across the interface and hence:

$$q(u_\mathrm{L}-u_\mathrm{R}) + \tfrac{1}{2}g[(\phi h^2)_\mathrm{L} - (\phi h^2)_\mathrm{R}] + \tfrac{1}{2}g(h_{\mathrm{LR}}^{(1)})^2(\phi_\mathrm{R}-\phi_\mathrm{L}) - g(\phi h)_{\mathrm{LR}}^{(1)}(z_{\mathrm{b_R}} - z_{\mathrm{b_L}}) = 0 \tag{35}$$

The first relation linking the three variables has been established; the momentum equation is used to estimate q. The balance is carried out on two cells on both sides of a porosity and bottom elevation discontinuity. To establish the balance, the forces applied to the fluid have to be estimated along the horizontal axis. Hereafter, the flow is assumed to be directed from left to right. The different sketches used to derive the momentum balance are illustrated in Figure 3.

DOI: 10.1002/fld

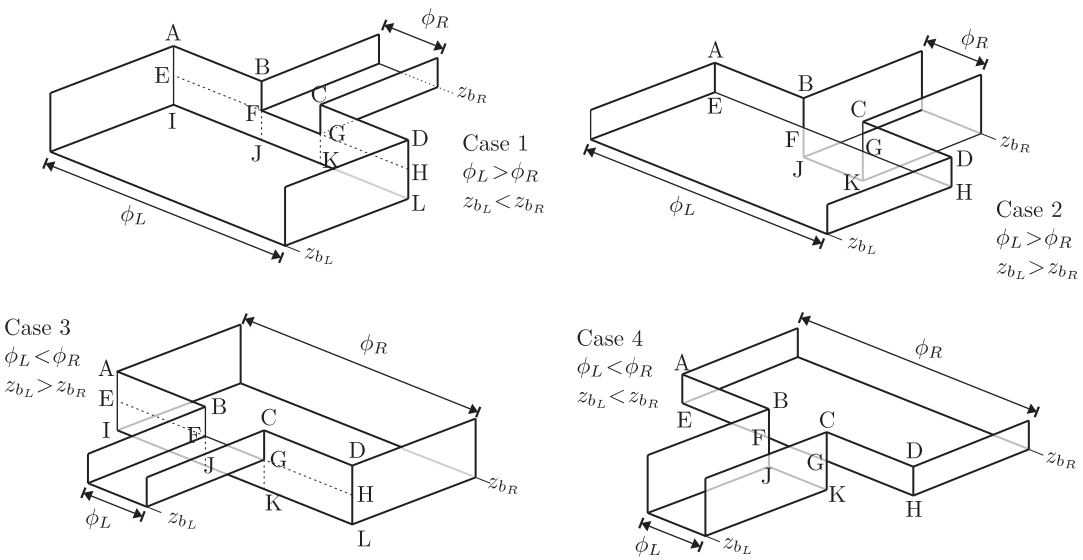

Figure 3. Considered sketches to establish the momentum equation.

The force calculation is detailed for case 1 corresponding to $z_{b_L} < z_{b_R}$ and $\phi_L > \phi_R$.
Force exerted by areas ABFE and CDHG:

$$F_a = -g \int_{\phi_R}^{\phi_L} \left[\int_{z_{b_R}}^{z_L} (z_L - z)\, dz \right] d\phi = -\frac{1}{2} g (\phi_L - \phi_R)(z_L - z_{b_R})^2 \tag{36}$$

Force exerted by area FGKJ:

$$F_b = -g\phi_R \int_{z_{b_L}}^{z_{b_R}} (z_L - z)\, dz = -g\phi_R (z_{b_R} - z_{b_L}) \left(z_L - \frac{z_{b_R} + z_{b_L}}{2} \right) \tag{37}$$

Force exerted by areas EFJI and GHLK:

$$F_c = -g \int_{\phi_R}^{\phi_L} \left[\int_{z_{b_L}}^{z_{b_R}} (z_L - z)\, dz \right] d\phi = -g(\phi_L - \phi_R)(z_{b_R} - z_{b_L}) \left(z_L - \frac{z_{b_R} + z_{b_L}}{2} \right) \tag{38}$$

Total force exerted onto the fluid:

$$F_a + F_b + F_c = -\frac{1}{2} g (\phi_L - \phi_R) h_L^2 - g\phi_R (z_{b_R} - z_{b_L}) \left(z_L - \frac{z_{b_R} + z_{b_L}}{2} \right) \tag{39}$$

The momentum balance should be written for a steady-state regime to ensure static balance:

$$(\phi h u^2 + \tfrac{1}{2} g \phi h^2)_R - (\phi h u^2 + \tfrac{1}{2} g \phi h^2)_L = F_a + F_b + F_c \tag{40}$$

Combining Equations (39) and (40) leads to:

$$q(u_R - u_L) = \frac{1}{2} g [(\phi h^2)_L - (\phi h^2)_R] - \frac{1}{2} g (\phi_L - \phi_R) h_L^2 - g\phi_R (z_{b_R} - z_{b_L}) \left(z_L - \frac{z_{b_R} + z_{b_L}}{2} \right) \tag{41}$$

DOI: 10.1002/fld

Using the same reasoning for cases 2 to 4 leads to the following general expression:

$$q(u_{\mathrm{L}}-u_{\mathrm{R}})=-\frac{1}{2}g[(\phi h^2)_{\mathrm{L}}-(\phi h^2)_{\mathrm{R}}]+\frac{1}{2}g(\phi_{\mathrm{L}}-\phi_{\mathrm{R}})(h_{\mathrm{LR}}^{(0)})^2 +g\phi_{\mathrm{LR}}^{(0)}(z_{\mathrm{b_R}}-z_{\mathrm{b_L}})\left(z_{\mathrm{LR}}^{(0)}-\frac{z_{\mathrm{b_R}}+z_{\mathrm{b_L}}}{2}\right) \tag{42}$$

where

$$\phi_{\mathrm{LR}}^{(0)}=\min(\phi_{\mathrm{L}};\phi_{\mathrm{R}}) \tag{43a}$$

$$z_{\mathrm{LR}}^{(0)}=\begin{cases} z_{\mathrm{L}} & \text{if } z_{\mathrm{b_L}}<z_{\mathrm{b_R}} \\ z_{\mathrm{R}} & \text{if } z_{\mathrm{b_L}}>z_{\mathrm{b_R}} \end{cases} \tag{43b}$$

$$h_{\mathrm{LR}}^{(0)}=\begin{cases} h_{\mathrm{L}} & \text{if } \phi_{\mathrm{L}}>\phi_{\mathrm{R}} \\ h_{\mathrm{R}} & \text{if } \phi_{\mathrm{L}}<\phi_{\mathrm{R}} \end{cases} \tag{43c}$$

Combining (35) and (42) yields the following equation:

$$\frac{1}{2}g(\phi_{\mathrm{L}}-\phi_{\mathrm{R}})[(h_{\mathrm{LR}}^{(0)})^2-(h_{\mathrm{LR}}^{(1)})^2]+g(z_{\mathrm{b_L}}-z_{\mathrm{b_R}})\left[(\phi h)_{\mathrm{LR}}^{(1)}-\phi_{\mathrm{LR}}^{(0)}\left(z_{\mathrm{LR}}^{(0)}-\frac{z_{\mathrm{b_R}}+z_{\mathrm{b_L}}}{2}\right)\right]=0 \tag{44}$$

Equation (44) involves the unknown variables $(\phi h)_{\mathrm{LR}}^{(1)}$ and $h_{\mathrm{LR}}^{(1)}$. However, the equation is valid whatever the values of ϕ_{R}, ϕ_{L}, $z_{\mathrm{b_L}}$ and $z_{\mathrm{b_R}}$ may be. Considering the particular case $\phi_{\mathrm{L}}=\phi_{\mathrm{R}}$ in (44) leads to

$$(\phi h)_{\mathrm{LR}}^{(1)}=\phi_{\mathrm{LR}}^{(0)}\left(z_{\mathrm{LR}}^{(0)}-\frac{z_{\mathrm{b_R}}+z_{\mathrm{b_L}}}{2}\right) \tag{45}$$

whereas the case $z_{\mathrm{b_R}}=z_{\mathrm{b_L}}$ implies $(h_{\mathrm{LR}}^{(1)})^2=(h_{\mathrm{LR}}^{(0)})^2$. Since $h_{\mathrm{LR}}^{(1)}$ corresponds to the water depth at the interface, this value should be positive and thus:

$$h_{\mathrm{LR}}^{(1)}=h_{\mathrm{LR}}^{(0)} \tag{46}$$

2.3.2. Estimation of $(\phi h)_{\mathrm{LR}}^{(S)}$ *and* $h_{\mathrm{LR}}^{(S)}$. A momentum balance on two cells on both sides of a porosity and bottom elevation discontinuity is used. The integral form of the source term can be written as

$$(\Delta S_0)_{\mathrm{LR}}^{n+1/2}=-g(\phi h)_{\mathrm{LR}}^{(S)}(z_{\mathrm{b_R}}-z_{\mathrm{b_L}})+\tfrac{1}{2}g(h_{\mathrm{LR}}^{(S)})^2(\phi_{\mathrm{R}}-\phi_{\mathrm{L}}) \tag{47}$$

and the momentum balance can be written as:

$$(\phi hu^2+\tfrac{1}{2}g\phi h^2)_{\mathrm{L}}^{n+1/2}-(\phi hu^2+\tfrac{1}{2}g\phi h^2)_{\mathrm{R}}^{n+1/2}+(\Delta S_0)_{\mathrm{LR}}^{n+1/2}=0 \tag{48}$$

Combining (47) with (48) leads to:

$$q(u_{\mathrm{L}}-u_{\mathrm{R}})+\tfrac{1}{2}g[(\phi h^2)_{\mathrm{L}}-(\phi h^2)_{\mathrm{R}}]+g(\phi h)_{\mathrm{LR}}^{(S)}(z_{\mathrm{b_L}}-z_{\mathrm{b_R}})-\tfrac{1}{2}g(h_{\mathrm{LR}}^{(S)})^2(\phi_{\mathrm{L}}-\phi_{\mathrm{R}})=0 \tag{49}$$

DOI: 10.1002/fld

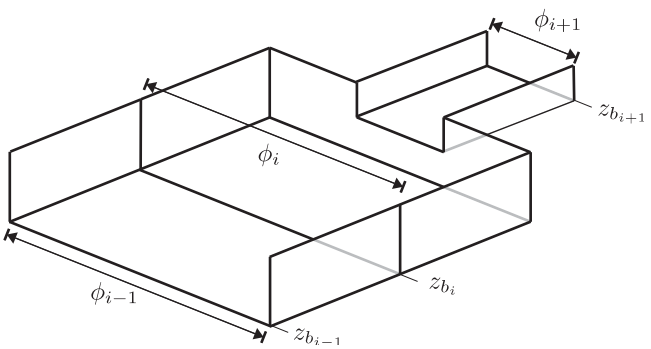

Figure 4. Considered sketch to establish the momentum equation for the estimation of $(\phi h)^{(2)}_{\mathrm{LR}}$ and $h^{(2)}_{\mathrm{LR}}$.

Substituting $q(u_{\mathrm{L}}-u_{\mathrm{R}})$ in (49) by the expression (42) yields:

$$g(z_{\mathrm{b_L}}-z_{\mathrm{b_R}})\left[(\phi h)^{(S)}_{\mathrm{LR}}-\phi^{(0)}_{\mathrm{LR}}\left(z^{(0)}_{\mathrm{LR}}-\frac{z_{\mathrm{b_R}}+z_{\mathrm{b_L}}}{2}\right)\right]+\frac{1}{2}g(\phi_{\mathrm{L}}-\phi_{\mathrm{R}})[(h^{(0)}_{\mathrm{LR}})^2-(h^{(S)}_{\mathrm{LR}})^2]=0 \tag{50}$$

As (50) is equivalent to (44), it is possible to conclude that:

$$(\phi h)^{(S)}_{\mathrm{LR}}=(\phi h)^{(1)}_{\mathrm{LR}} \tag{51}$$

$$(h^{(S)}_{\mathrm{LR}})^2=(h^{(1)}_{\mathrm{LR}})^2\Rightarrow h^{(S)}_{\mathrm{LR}}=h^{(1)}_{\mathrm{LR}} \tag{52}$$

2.3.3. Estimation of $(\phi h)^{(2)}_{\mathrm{LR}}$ *and* $h^{(2)}_{\mathrm{LR}}$. A steady-state momentum balance is carried out on three cells ($i-1$, i and $i+1$) with a porosity and bottom elevation discontinuity between the cells i and $i+1$ (see the sketch on Figure 4).

The momentum balance over the cell i can be written as:

$$(\phi hu)^{n+1}_i-(\phi hu)^n_{i+1}=\frac{\Delta t}{\Delta x_i}\left(F^*_2+\frac{\lambda_3}{\lambda_3-\lambda_1}\Delta S_0\right)^{n+1/2}_{i-1/2}-\frac{\Delta t}{\Delta x_i}\left(F^*_2+\frac{\lambda_1}{\lambda_3-\lambda_1}\Delta S_0\right)^{n+1/2}_{i+1/2}=0 \tag{53}$$

Combining (32b) with (53) leads, for the interface between the cells $i-1$ and i, to:

$$\begin{aligned}
&\left(F^*_2+\frac{\lambda_3}{\lambda_3-\lambda_1}\Delta S_0\right)^{n+1/2}_{i-1/2}\\
&\quad=\frac{1}{\lambda_3-\lambda_1}\left[\lambda_3\left(\phi hu^2+\frac{1}{2}g\phi h^2\right)_{i-1}-\lambda_1\left(\phi hu^2+\frac{1}{2}g\phi h^2\right)_i-\lambda_3\lambda_1((\phi hu)_{i-1}-(\phi hu)_i)\right]\\
&\qquad+\frac{\lambda_3+\lambda_1}{\lambda_3-\lambda_1}\left[\frac{1}{2}g(h^{(2)}_{i-1/2})^2(\phi_{i-1}-\phi_i)+g(z_{\mathrm{b}_{i-1}}-z_{\mathrm{b}_i})(\phi h)^{(2)}_{i-1/2}\right]\\
&\qquad+\frac{\lambda_3}{\lambda_3-\lambda_1}\left[g\phi^{(0)}_{\mathrm{LR}}(z_{\mathrm{b}_{i-1}}-z_{\mathrm{b}_i})\left(z^{(0)}_{\mathrm{LR}}-\frac{z_{\mathrm{b}_{i-1}}+z_{\mathrm{b}_i}}{2}\right)-\frac{1}{2}g(\phi_{i-1}-\phi_i)(h^{(0)}_{\mathrm{LR}})^2\right]
\end{aligned} \tag{54}$$

DOI: 10.1002/fld

Since there is no discontinuity between cells $i-1$ and i:

$$\begin{aligned}(\phi hu^2+\tfrac{1}{2}g\phi h^2)_{i-1} &= (\phi hu^2+\tfrac{1}{2}g\phi h^2)_i \neq (\phi hu^2+\tfrac{1}{2}g\phi h^2)_{i+1}\\ (\phi hu)_{i-1} &= (\phi hu)_i = (\phi hu)_{i+1} = q\\ \phi_{i-1} &= \phi_i \neq \phi_{i+1}\\ z_{\mathrm{b}_{i-1}} &= z_{\mathrm{b}_i} \neq z_{\mathrm{b}_{i+1}}\end{aligned} \tag{55}$$

Introducing (55) into (54) leads to:

$$\begin{aligned}\left(F_2^*+\frac{\lambda_1}{\lambda_3-\lambda_1}\Delta S_0\right)_{i-1/2}^{n+1/2} &= \frac{1}{\lambda_3-\lambda_1}\left[\lambda_3\left(\phi hu^2+\frac{1}{2}g\phi h^2\right)_i-\lambda_1\left(\phi hu^2+\frac{1}{2}g\phi h^2\right)_i\right]\\ &= \left(\phi hu^2+\frac{1}{2}g\phi h^2\right)_i\end{aligned} \tag{56}$$

At the interface between the cells i and $i+1$, the momentum flux can therefore be written as:

$$\begin{aligned}&\left(F_2^*+\frac{\lambda_1}{\lambda_3-\lambda_1}\Delta S_0\right)_{i+1/2}^{n+1/2}\\ &=\frac{1}{\lambda_3-\lambda_1}\left[\lambda_3\left(\phi hu^2+\frac{1}{2}g\phi h^2\right)_i-\lambda_1\left(\phi hu^2+\frac{1}{2}g\phi h^2\right)_{i+1}-\lambda_3\lambda_1[(\phi hu)_i-(\phi hu)_{i+1}]\right]\\ &\quad+\frac{\lambda_3+\lambda_1}{\lambda_3-\lambda_1}\left[\frac{1}{2}g(h^{(2)}_{i+1/2})^2(\phi_i-\phi_{i+1})+g(z_{\mathrm{b}_i}-z_{\mathrm{b}_{i+1}})(\phi h)^{(2)}_{i+1/2}\right]\\ &\quad+\frac{\lambda_1}{\lambda_3-\lambda_1}\left[g\phi^{(0)}_{\mathrm{LR}}(z_{\mathrm{b}_i}-z_{\mathrm{b}_{i+1}})\left(z^{(0)}_{\mathrm{LR}}-\frac{z_{\mathrm{b}_i}+z_{\mathrm{b}_{i+1}}}{2}\right)-\frac{1}{2}g(\phi_i-\phi_{i+1})(h^{(0)}_{\mathrm{LR}})^2\right]\end{aligned} \tag{57}$$

Applying the continuity equation in (57) leads to:

$$\begin{aligned}&\left(F_2^*+\frac{\lambda_1}{\lambda_3-\lambda_1}\Delta S_0\right)_{i+1/2}^{n+1/2}\\ &=\frac{1}{\lambda_3-\lambda_1}\left[\lambda_3\left(\phi hu^2+\frac{1}{2}g\phi h^2\right)_i-\lambda_1\left(\phi hu^2+\frac{1}{2}g\phi h^2\right)_{i+1}\right]\\ &\quad+\frac{\lambda_3+\lambda_1}{\lambda_3-\lambda_1}\left[\frac{1}{2}g(h^{(2)}_{i+1/2})^2(\phi_i-\phi_{i+1})+g(z_{\mathrm{b}_i}-z_{\mathrm{b}_{i+1}})(\phi h)^{(2)}_{i+1/2}\right]\\ &\quad+\frac{\lambda_1}{\lambda_3-\lambda_1}\left[g\phi^{(0)}_{\mathrm{LR}}(z_{\mathrm{b}_i}-z_{\mathrm{b}_{i+1}})\left(z^{(0)}_{\mathrm{LR}}-\frac{z_{\mathrm{b}_i}+z_{\mathrm{b}_{i+1}}}{2}\right)-\frac{1}{2}g(\phi_i-\phi_{i+1})(h^{(0)}_{\mathrm{LR}})^2\right]\end{aligned} \tag{58}$$

The steady-state assumption implies that

$$(\phi hu)_i^{n+1}-(\phi hu)_i^n=0 \tag{59}$$

DOI: 10.1002/fld

and hence

$$\left(F_2^* + \frac{\lambda_3}{\lambda_3 - \lambda_1}\Delta S_0\right)_{i-1/2}^{n+1/2} = \left(F_2^* + \frac{\lambda_1}{\lambda_3 - \lambda_1}\Delta S_0\right)_{i+1/2}^{n+1/2} \tag{60}$$

Combining (60), (58) and (56) yields:

$$\begin{aligned}
&\lambda_1 \left\{ q(u_i - u_{i+1}) + \frac{1}{2} g[(\phi h^2)_i - (\phi h^2)_{i+1}] \right\} \\
&+ (\lambda_3 - \lambda_1) \left[\frac{1}{2} g (h_{i+1/2}^{(2)})^2 (\phi_i - \phi_{i+1}) + g(z_{\mathrm{b}_i} - z_{\mathrm{b}_{i+1}})(\phi h)_{i+1/2}^{(2)} \right] \\
&+ \lambda_1 \left[g \phi_{\mathrm{LR}}^{(0)} (z_{\mathrm{b}_i} - z_{\mathrm{b}_{i+1}}) \left(z_{\mathrm{LR}}^{(0)} - \frac{z_{\mathrm{b}_i} - z_{\mathrm{b}_{i+1}}}{2} \right) - \frac{1}{2} g(\phi_i - \phi_{i+1})(h_{\mathrm{LR}}^{(0)})^2 \right] = 0
\end{aligned} \tag{61}$$

Substituting $q(u_i - u_{i+1})$ in (61) by the expression (42) leads to:

$$\tfrac{1}{2} g (h_{i+1/2}^{(2)})^2 (\phi_i - \phi_{i+1}) + g(z_{\mathrm{b}_i} - z_{\mathrm{b}_{i+1}})(\phi h)_{i+1/2}^{(2)} = 0 \tag{62}$$

As (62) is valid for all existing cases depending on the respective values of $\phi_{i+1}, \phi_i, z_{\mathrm{b}_i}$ and $z_{\mathrm{b}_{i+1}}$, it is possible to conclude that $h_{i+1/2}^{(2)} = 0$ and $(\phi h)_{i+1/2}^{(2)} = 0$.

Substituting $(\phi h)_{\mathrm{LR}}^{(1)}$, $(\phi h)_{\mathrm{LR}}^{(2)}$, $(\phi h)_{\mathrm{LR}}^{(S)}$, $h_{\mathrm{LR}}^{(1)}$, $h_{\mathrm{LR}}^{(2)}$ and $h_{\mathrm{LR}}^{(S)}$ in the system of equations (32) yields the following expression for the flux through the interface:

$$\begin{aligned}
F_1^* &= \frac{1}{\lambda_3 - \lambda_1} [F_{2,\mathrm{L}} - F_{2,\mathrm{R}} - \lambda_1 F_{1,\mathrm{L}} + \lambda_3 F_{1,\mathrm{R}}] \\
&\quad + \frac{1}{\lambda_3 - \lambda_1} \left[\frac{1}{2} g(\phi_{\mathrm{R}} - \phi_{\mathrm{L}})(h_{\mathrm{LR}}^{(0)})^2 - g \phi_{\mathrm{LR}}^{(0)} (z_{\mathrm{b_R}} - z_{\mathrm{b_L}}) \left(z_{\mathrm{LR}}^{(0)} - \frac{z_{\mathrm{b_R}} + z_{\mathrm{b_L}}}{2} \right) \right]
\end{aligned} \tag{63a}$$

$$F_2^* = \frac{1}{\lambda_3 - \lambda_1} [\lambda_3 F_{2,\mathrm{L}} - \lambda_1 F_{2,\mathrm{R}} - \lambda_1 \lambda_3 (F_{1,\mathrm{L}} - F_{1,\mathrm{R}})] \tag{63b}$$

$$F_3^* = v(F_1^* - F_{1,A}) + F_{3,A} \tag{63c}$$

where $\phi_{\mathrm{LR}}^{(0)}$, $z_{\mathrm{LR}}^{(0)}$ and $h_{\mathrm{LR}}^{(0)}$ are defined by (43).

2.4. Practical implementation

The values computed for the flux using the system of equations (63) are only valid for the flux in the intermediate constant zone. For the purpose of the solver, it is necessary to determine the flux at the interface between the cells, i.e. the initial discontinuity of the Riemann problem. Five different cases can be defined depending on the wave configuration and the respective location of the intermediate zone and the interface (see Figure 5 for more detail).

It is recalled that configurations C1 and C5 correspond to a supercritical regime, configurations C2 and C4 correspond to a critical regime and configuration C3 corresponds to the subcritical regime. It is necessary to estimate the celerities λ_1^* and λ_3^* in the constant intermediate zone. Along

DOI: 10.1002/fld

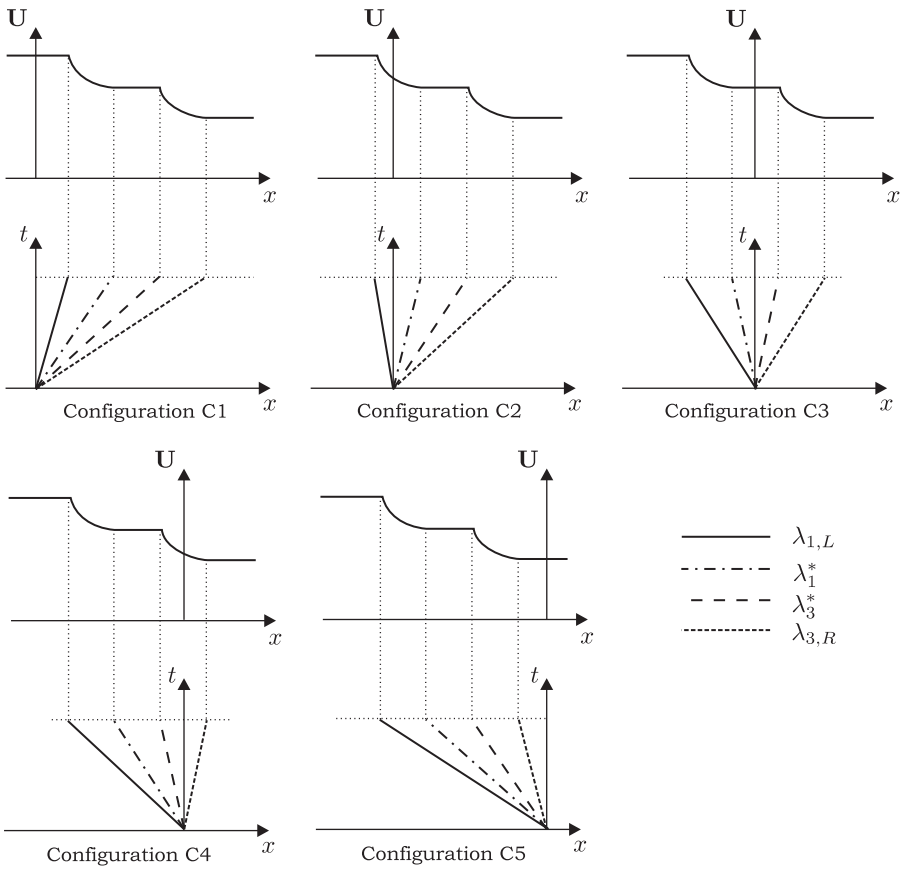

Figure 5. Relative position of the intermediate region to the interface.

each characteristic, the Riemann invariants are used to compute u^* and c^*:

$$\begin{aligned} u^*+2c^* &= u_{\mathrm{L}}+2c_{\mathrm{L}} \\ u^*-2c^* &= u_{\mathrm{R}}-2c_{\mathrm{R}} \end{aligned} \tag{64}$$

The source terms have to be taken into account in (64) but the experiment shows that they can be neglected at this stage. Solving the system of equations (64) leads to:

$$\begin{aligned} u^* &= \frac{u_{\mathrm{L}}+u_{\mathrm{R}}}{2}+c_{\mathrm{L}}-c_{\mathrm{R}} \\ c^* &= \frac{u_{\mathrm{L}}-u_{\mathrm{R}}}{4}+\frac{c_{\mathrm{L}}+c_{\mathrm{R}}}{2} \end{aligned} \tag{65}$$

The celerities λ_1^* and λ_3^* are computed as follows:

$$\begin{aligned} \lambda_1^* &= u^*-c^* \\ \lambda_3^* &= u^*+c^* \end{aligned} \tag{66}$$

DOI: 10.1002/fld

The comparison of λ_1^*, λ_3^*, $\lambda_{1,\mathrm{L}}$ and $\lambda_{3,\mathrm{R}}$ enables to determine the wave configuration and thus the value of the flux vector component at the initial discontinuity. For the critical regime, the fluxes are computed using the classical HLLC formulae:

$$
\begin{array}{ll}
\lambda_{1,\mathrm{L}}>0 & \begin{cases} F_1 = F_{1,\mathrm{L}} \\ F_2 = F_{2,\mathrm{L}} \end{cases} \quad \text{C1} \\
\lambda_{1,\mathrm{L}}<0 \ \& \ \lambda_1^*>0 & \begin{cases} F_1 = \dfrac{1}{\lambda_3-\lambda_1}[\lambda_3 F_{1,\mathrm{L}} - \lambda_1 F_{1,\mathrm{R}} - \lambda_1\lambda_3(z_\mathrm{L}-z_\mathrm{R})] \\ F_2 = \dfrac{1}{\lambda_3-\lambda_1}[\lambda_3 F_{2,\mathrm{L}} - \lambda_1 F_{2,\mathrm{R}} - \lambda_1\lambda_3(F_{1,\mathrm{L}}-F_{1,\mathrm{R}})] \end{cases} \quad \text{C2} \\
\lambda_1^*<0 \ \& \ \lambda_3^*>0 & \begin{cases} F_1 = F_1^* \\ F_2 = F_2^* \end{cases} \quad \text{C3} \\
\lambda_3^*<0 \ \& \ \lambda_{3,\mathrm{R}}>0 & \begin{cases} F_1 = \dfrac{1}{\lambda_3-\lambda_1}[\lambda_3 F_{1,\mathrm{L}} - \lambda_1 F_{1,\mathrm{R}} - \lambda_1\lambda_3(z_\mathrm{L}-z_\mathrm{R})] \\ F_2 = \dfrac{1}{\lambda_3-\lambda_1}[\lambda_3 F_{2,\mathrm{L}} - \lambda_1 F_{2,\mathrm{R}} - \lambda_1\lambda_3(F_{1,\mathrm{L}}-F_{1,\mathrm{R}})] \end{cases} \quad \text{C4} \\
\lambda_{3,\mathrm{R}}<0 & \begin{cases} F_1 = F_{1,\mathrm{R}} \\ F_2 = F_{2,\mathrm{R}} \end{cases} \quad \text{C5}
\end{array}
\tag{67}
$$

For any waves configuration, the fluxes at the interface can be computed and the values of the conserved variables can therefore be calculated at each time.

3. COMPUTATIONAL EXAMPLES

Eight different geometries have been used to determine the efficiency of the PorAS solver under different flow conditions (subcritical, supercritical, transcritical and hydraulic jump) leading to 13 tests cases (named $T_{i,(\mathrm{sub,\,sup,\,trans,\,jump})}$). The results are compared with the analytical (or semi-analytical) solution and with results calculated using the classical HLLC solver presented in [5]. Each geometry is described in the corresponding sub part. Geometry 1 has been developed to evaluate the PorAS solver performance under steady flow condition ($T_{1,(\mathrm{sub,\,sup,\,trans,\,jump})}$). Further comparisons have also been conducted on the test cases proposed in [6]: $T_{2,(\mathrm{sub,\,trans})}$ and in [5]: T_3, T_4 and T_5. A convergence study is performed both for the PorAS and the HLLC solver: $T_{6,(1,2,3)}$. The efficiency of the PorAS solver is also assessed by comparing the model results with experimental data: T_7. For all the following figures, the reader may have some difficulties to identify the different curves that are merged most of the time.

3.1. One-dimensional flow in case of variable porosity ($T_{1,(\mathrm{sub,\,sup,\,trans,\,jump})}$)

The first geometry consists of permanent flow in a channel with nil bottom slope and variable porosity. The four-associated test cases highlight theperformance of the proposed solver in the

DOI: 10.1002/fld

presence of a spatially varying porosity. A parabolic profile has been chosen for the porosity

$$\phi(x)=\begin{cases} ax^2+bx+c & \text{for } x\in[X_\mathrm{m}-L_0;\,X_\mathrm{m}+L_0] \\ \phi_0 & \text{elsewhere} \end{cases}$$
$$a=\frac{\phi_0-\phi_\mathrm{m}}{L_0^2},\quad b=-2aX_\mathrm{m},\quad c=\phi_0+a(X_\mathrm{m}^2-L_0^2) \tag{68}$$

where $2L_0$ is the length of the porosity variation, X_m is the position of the middle of the porosity variation, ϕ_0 is the porosity outside of the porosity variation area and ϕ_m is the porosity in the middle of the porosity variation. The computational parameters are given in Table I. In steady flow, the momentum equation can be written as:

$$\frac{\mathrm{d}}{\mathrm{d}x}\left(\frac{q^2}{\phi h}+\frac{1}{2}g\phi h^2\right)=\frac{1}{2}gh^2\frac{\mathrm{d}\phi}{\mathrm{d}x} \tag{69}$$

Combining the continuity equation in steady flow with the momentum Equation (69) leads to

$$\phi\frac{\mathrm{d}h}{\mathrm{d}x}(1-Fr^2)=hFr^2\frac{\mathrm{d}\phi}{\mathrm{d}x} \tag{70}$$

where Fr represents the Froude number. Depending on the flow regime, the discretization has to be done from upstream to downstream (supercritical flow):

$$h_\mathrm{d}=h_\mathrm{u}-h_\mathrm{u}\frac{u_\mathrm{u}^2}{\phi(gh_\mathrm{u}-u_\mathrm{u}^2)}(\phi_\mathrm{u}-\phi_\mathrm{d}) \tag{71}$$

The reference for the test case $T_{1,\mathrm{sup}}$ is established by prescribing the upstream boundary condition $h=0.2\,\mathrm{m}$ with a unit discharge $q=1\,\mathrm{m^2\,s^{-1}}$. For the test case $T_{1,\mathrm{sub}}$, the unit discharge is also $q=1\,\mathrm{m^2\,s^{-1}}$ and the downstream boundary condition is $h=1.1\,\mathrm{m}$. For the transcritical test case $T_{1,\mathrm{trans}}$, it is easy to show that the critical point where $Fr=1$ is necessarily obtained for a nil

Table I. Parameters for the first geometry: steady flow test cases (one-dimensional flow in narrowing channel—$T_{1,(\mathrm{sub,\,sup,\,trans,\,jump})}$).

Symbol	Meaning	Value
g	Gravitational acceleration	$9.81\,\mathrm{m\,s^{-2}}$
ϕ_0	Porosity outside of the porosity variation	1
	Minimum of the porosity:	
ϕ_m	for test cases $T_{1,\mathrm{sup}}$ and $T_{1,\mathrm{jump}}$	0.6
	for test cases $T_{1,\mathrm{sub}}$ and $T_{1,\mathrm{trans}}$	0.5
L_0	Length of the porosity variation	25 m
x_m	Position of the minimal porosity ϕ_m in the domain	50 m
L	Domain length	$100\,\mathrm{m}\times 1\,\mathrm{m}$
Δx	Cell size	$0.1\,\mathrm{m}\times 1\,\mathrm{m}$
q	Prescribed unit discharge at the upstream boundary	$1\,\mathrm{m^2\,s^{-1}}$
	Prescribed water depth at the downstream boundary:	
h_{av}	for test case $T_{1,\mathrm{sup}}$	0.2 m
	for test case $T_{1,\mathrm{sub}}$	1.1 m

DOI: 10.1002/fld

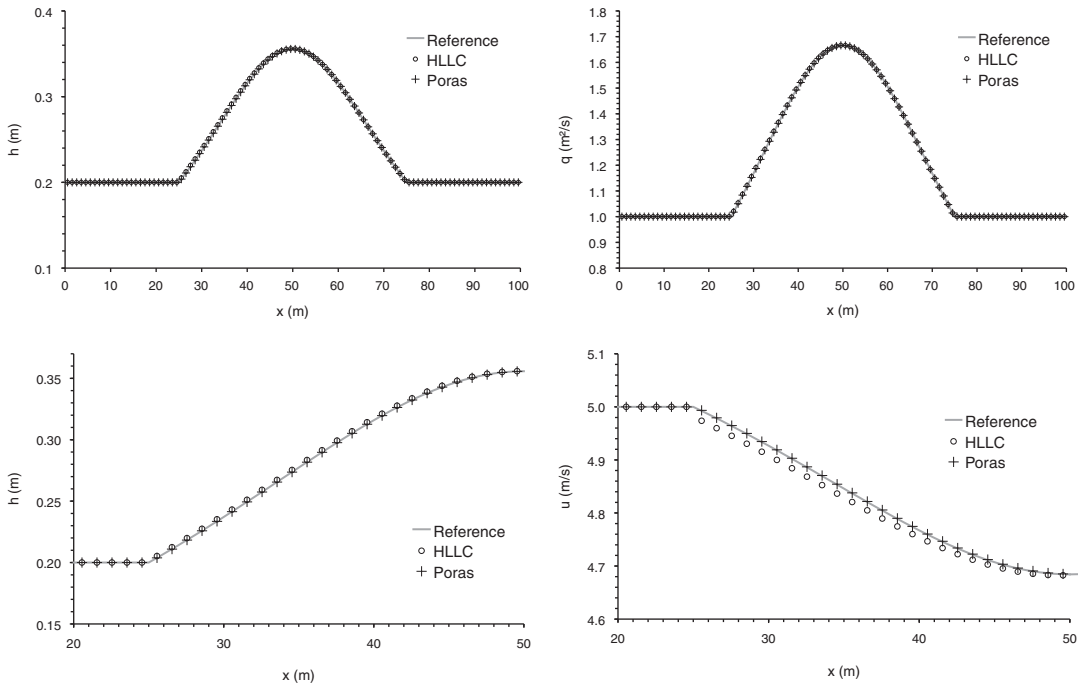

Figure 6. Supercritical one-dimensional flow in narrowing channel ($T_{1,\mathrm{sup}}$). Reference, HLLC and PorAS solutions; Top left, simulated depth; Top right, simulated unit discharge; Bottom left, detail of the water depth profile in the narrowing part; Bottom right, simulated velocity in the narrowing part.

porosity gradient. It is therefore possible to compute the water level, using Equation (70) upstream to the nil porosity gradient and Equation (71) downstream. For the test case with a hydraulic jump, the water profile upstream to the nil porosity gradient has been computed in a same way than for the transcritical test case. At the downstream end of the domain, a subcritical depth is prescribed $h=0.811\,\mathrm{m}$. The position of the hydraulic jump is determined by computing the water profile both from the nil porosity gradient to the downstream end and from the downstream end to the nil porosity gradient; the discretization being used in accordance to the flow regime. The impulse corresponding to both profiles is also computed; the hydraulic jump leading at the point where both impulses are equals. For all the test cases, the spatial discretization step for the reference solution is $\Delta x=0.1\,\mathrm{m}$. The position of the hydraulic jump being very sensitive to the downstream water depth, a second porosity variation has been used downstream to the first one for the numerical test case to ensure that the hydraulic jump stay at the correct location. The middle of the second porosity variation is located at $x=150\,\mathrm{m}$, the porosity in the middle is $\phi_{\mathrm{m}}=0.7$ and the total length is 50 m. The imposed downstream water depth for the reference corresponds exactly to the water depth upstream to the second porosity variation in the numerical simulation.

Simulations are carried out from arbitrary initial conditions over a grid of uniform cell width $\Delta x=0.1\,\mathrm{m}$. The simulation is carried out until steady flow conditions are reached over the computational domain. Figures 6–9 present the profiles obtained for both the HLLC and the PorAS solver. For the sake of clarity, only every 10th cell is represented in the Figures 6–10. In the subcritical, the supercritical and the hydraulic jump test cases, both the water depth and the unit discharge are

DOI: 10.1002/fld

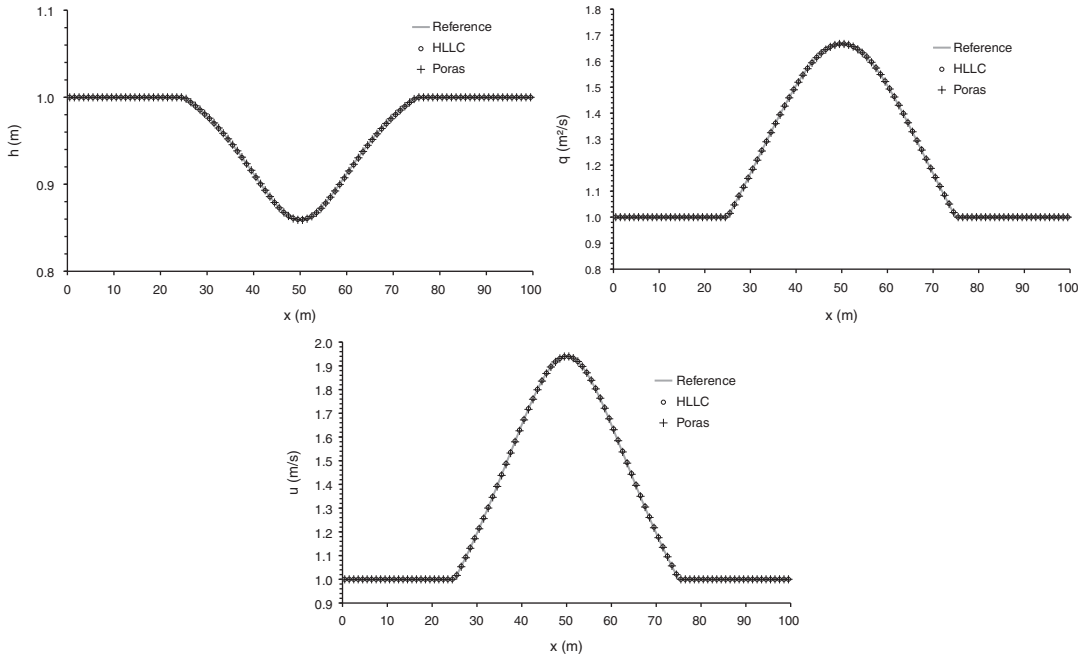

Figure 7. Subcritical one-dimensional flow in narrowing channel ($T_{1,\mathrm{sub}}$). Reference, HLLC and PorAS solutions; Top left, simulated depth; Top right, simulated unit discharge; Bottom, simulated velocity.

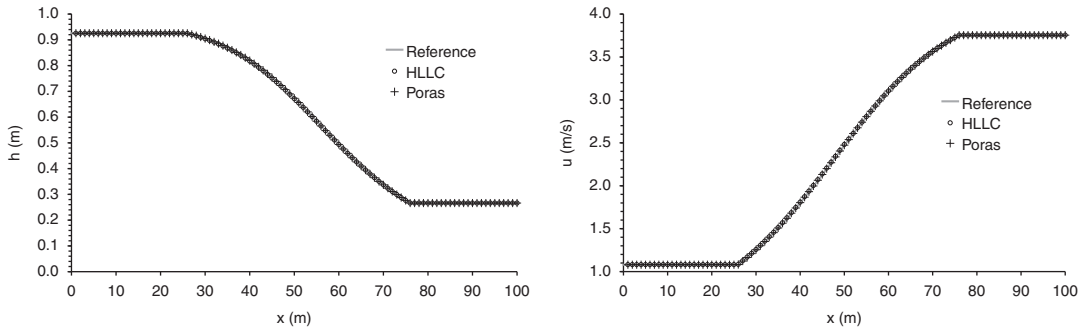

Figure 8. Transcritical one-dimensional flow in narrowing channel ($T_{1,\mathrm{trans}}$). Reference, HLLC and PorAS solutions; Left, simulated depth; Right, simulated velocity.

correctly computed by the two solvers. Nevertheless, the water depth in the narrowing part is better estimated using the PorAS solver in supercritical flow (Figure 6, bottom left). The HLLC solver underestimates the velocity in this region. For the transcritical test case, both water depth and velocity profiles are well computed by the HLLC and the PorAS solvers. Figure 10 presents the discharge computed with the HLLC and the PorAS solver. Since the test cases $T_{1,(\mathrm{sub,\,sup,\,trans,\,jump})}$ are steady-state simulations, the discharge has to be the same over the computational domain. For these simulations, the HLLC solver overestimates the discharge, whereas the PorAS solver keeps

DOI: 10.1002/fld

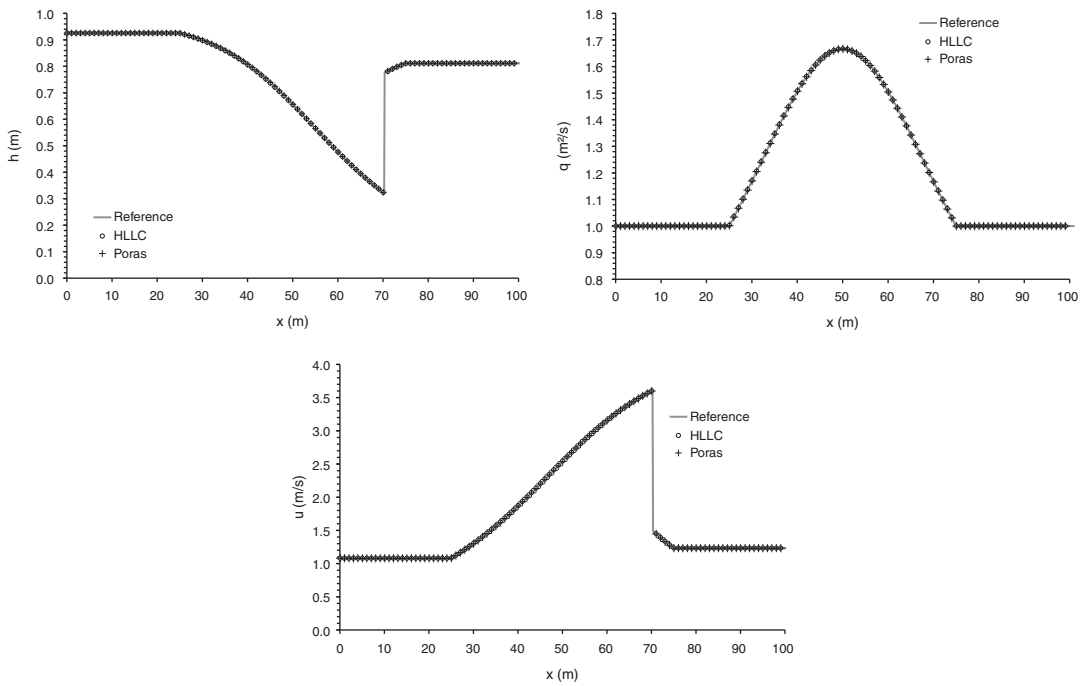

Figure 9. Hydraulic jump in narrowing channel ($T_{1,\text{jump}}$). Reference, HLLC and PorAS solutions; Top left, simulated depth; Top right, simulated unit discharge; Bottom, simulated velocity.

it constant. This phenomena may explain the velocity underestimation for the supercritical test case in the narrowing region. The inefficiency of the HLLC solver for such test cases has been investigated in [6], and as the authors suggested, it is impossible for the HLLC solver to compute correct steady-state conditions.

3.2. One-dimensional flow over a bump ($T_{2,(\text{sub},\text{trans})}$)

The previous test cases have highlighted the performance of the proposed solver in the presence of a spatially varying porosity (computed in the source term). The following test cases aim to test the PorAS solver when the varying parameter is the bottom slope. The specifications proposed in [6] have been reused (Tables II and III).

Figure 11 compares the water depth and the unit discharge computed using both solvers with the semi-analytical solution obtained by solving the momentum conservation equation. Both solvers compute correctly the water level but the HLLC solver introduces an unexpected variation of the unit discharge. The same reason as for the previous test cases can be invoked. Further analysis of this problem can be found in [6].

3.3. Dambreak problem

To evaluate the performance of the PorAS solver in transient configuration, three dambreak problems have been used.

DOI: 10.1002/fld

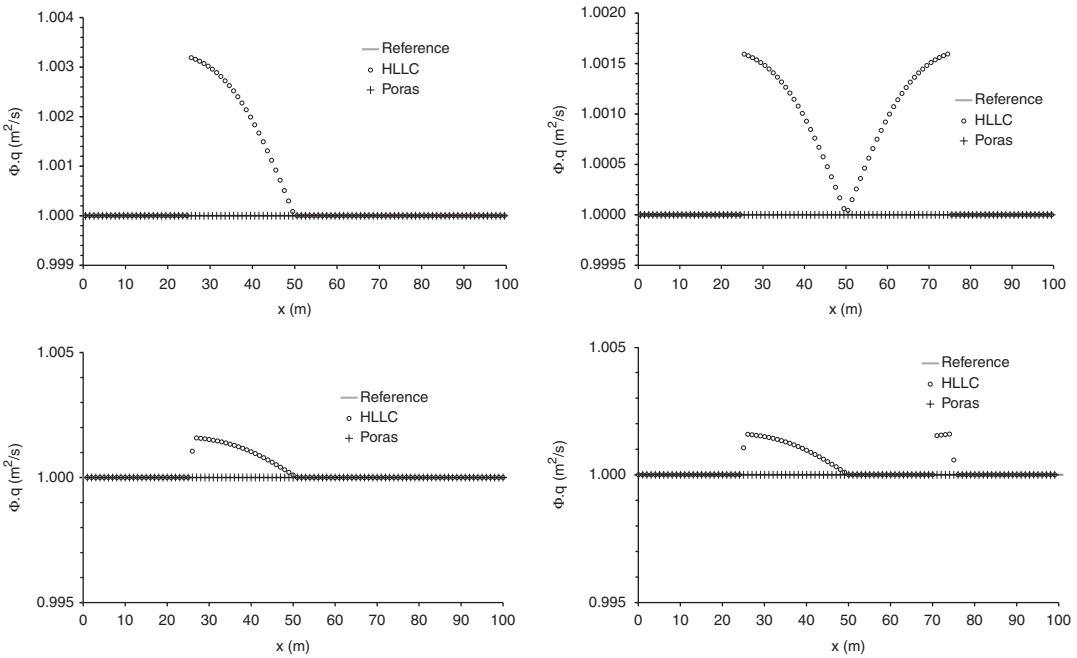

Figure 10. One-dimensional flow in narrowing channel ($T_{1,(\mathrm{sub,\,sup,\,trans,\,jump})}$). Reference, HLLC and PorAS solutions; Top left, simulated discharge for supercritical test case; Top right, simulated discharge for subcritical test case; Bottom left, simulated discharge for transcritical test case; Bottom right, simulated discharge for the hydraulic jump test case.

Table II. Parameters for the second geometry—one-dimensional subcritical flow over a bump ($T_{2,\mathrm{sub}}$).

Symbol	Meaning	Value
g	Gravitational acceleration	$9.81\,\mathrm{m\,s^{-2}}$
h_0	Bump height	0.2 m
L_0	Bump length	4 m
q	Prescribed unit discharge at the upstream boundary	$4.42\,\mathrm{m^2\,s^{-1}}$
z_d	Prescribed water depth at the downstream boundary	2 m
z_0	Initial water level	2 m
Δx	Cell size	1 m

3.3.1. One-dimensional dambreak with variable porosity (T_3). The first transient test case (T_3) consists of a dambreak over a flat bottom with a porosity varying linearly from 0 to 1. Computational parameters are given in Table IV. As explained in [5], this one-dimensional dambreak with variable porosity is equivalent to a classical circular dambreak with uniform porosity. The water level and the unit discharge for the reference are calculated by computing the classical dambreak over a grid with uniform cells width $\Delta x = 10^{-4}$ m. Figure 12 compares the results obtained using the HLLC and the PorAS solver with the reference. With both solvers, the water depth and the unit discharge are correctly estimated. The proposed solver introduces progressively two small peaks on both

DOI: 10.1002/fld

Table III. Parameters for the second geometry—one-dimensional transcritical flow over a bump ($T_{2,\mathrm{trans}}$).

Symbol	Meaning	Value
g	Gravitational acceleration	$9.81\,\mathrm{m\,s^{-2}}$
h_0	Bump height	0.2 m
L_0	Bump length	4 m
q	Prescribed unit discharge at the upstream boundary	$1.53\,\mathrm{m^2\,s^{-1}}$
z_0	Initial water level	0.1 m
Δx	Cell size	1 m

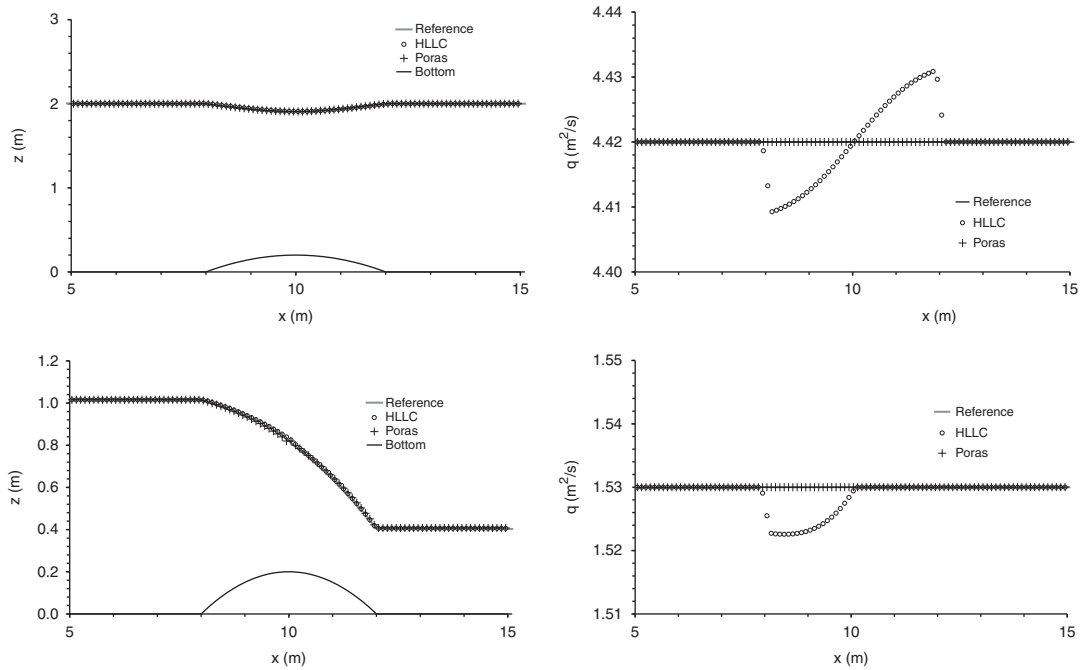

Figure 11. One-dimensional flow over a bump ($T_{2,(\mathrm{sub,\,trans})}$). Reference, HLLC and PorAS solutions; Left, simulated water level; Right, simulated unit discharge; Top, subcritical test case; Bottom, transcritical test case.

sides of the cell containing the initial discontinuity, corresponding to the critical point. Since the peaks only appear on the water depth profile, it is possible to state that the momentum transfer is correctly estimated whereas the mass transfer is not. Another difference can be observed for $x = 0$ m. The water depth profile is better estimated using the HLLC solver, whereas the PorAS solver better estimates the unit discharge profile. The water depth and the unit discharge profiles at the shock computed by the PorAS solver are also steeper.

3.3.2. One-dimensional dambreak with porosity discontinuity (T_4). The second transient test case is a one-dimensional dambreak through a porosity discontinuity, which is located at the same place

DOI: 10.1002/fld

Table IV. Parameters for the one-dimensional dambreak with variable porosity (T_3).

Symbol	Meaning	Value
g	Gravitational acceleration	$9.81\,\mathrm{m\,s^{-2}}$
$h_{0,L}$	Initial water depth on the left-hand side of the dam	10 m
$h_{0,R}$	Initial water depth on the right-hand side of the dam	1 m
L	Domain length	100 m
x_0	Dam location	50 m
Δx	Cell size	1 m
$\frac{\partial \phi}{\partial x}$	Derivative of the porosity	$0.01\,\mathrm{m^{-1}}$

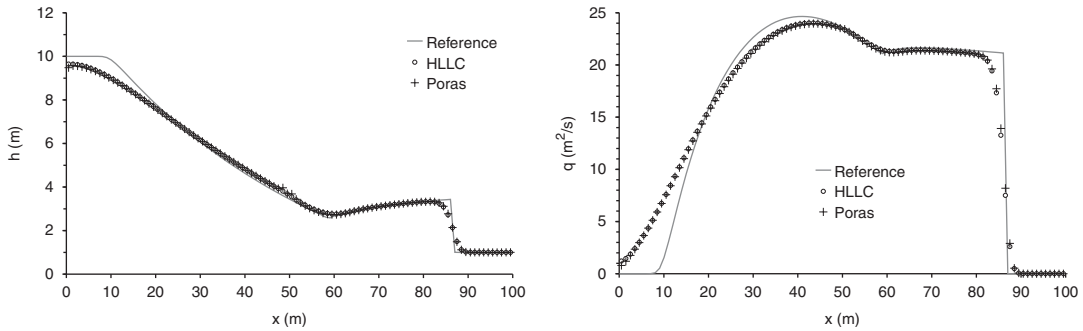

Figure 12. One-dimensional dambreak with variable porosity (T_3). Reference, HLLC and PorAS solutions; left, simulated depth, right simulated unit discharge.

Table V. Parameters for the one-dimensional dambreak with porosity discontinuity (T_4).

Symbol	Meaning	Value
g	Gravitational acceleration	$9.81\,\mathrm{m\,s^{-2}}$
$h_{0,\mathrm{L}}$	Initial water depth on the left-hand side of the dam	10 m
ϕ_{L}	Porosity on the left-hand side of the dam	1
$h_{0,\mathrm{R}}$	Initial water depth on the right-hand side of the dam	1 m
ϕ_{R}	Porosity on the right-hand side of the dam	0.1
L	Domain length	100 m
x_0	Dam location	50 m
Δx	Cell size	1 m

that the initial water discontinuity. Computational parameters are given in Table V. The method used to determine the reference profile is detailed in [5]. Figure 13 compares the profiles obtained with both solvers to the expected ones.

Both solvers compute correctly the water depth and the unit discharge. Nevertheless, they introduce diffusion through the rarefaction waves and the PorAS solver is, in that case, the most diffusive one. With the HLLC solver, the position of the shock is better estimated: at $t=4\,\mathrm{s}$, the shock should theoretically be at the position $x=94.08\,\mathrm{m}$; the position found by the HLLC solver

DOI: 10.1002/fld

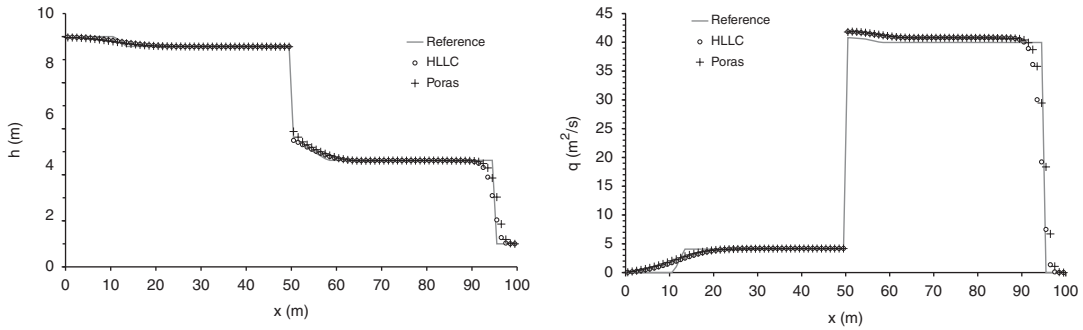

Figure 13. One-dimensional dambreak with porosity discontinuity (T_4). Reference, HLLC and PorAS solutions; left, simulated depth, right simulated unit discharge.

Table VI. Parameters for the two-dimensional dambreak with variable porosity (T_5).

Symbol	Meaning	Value
g	Gravitational acceleration	$9.81\,\mathrm{m\,s^{-2}}$
$h_{0,\mathrm{L}}$	Initial water depth on the left-hand side of the dam	10 m
$h_{0,\mathrm{R}}$	Initial water depth on the right-hand side of the dam	1 m
L	Domain length	$200\,\mathrm{m} \times 200\,\mathrm{m}$
r_0	Radius of the dam	50 m
Δx	Cell size	1 m

is $x = 95$ m and $x = 96$ m for the PorAS solver. It can be noticed that the shock remains steeper with the proposed solver.

3.3.3. Two-dimensional dambreak with variable porosity (T_5). The third transient test case (T_5) is a two-dimensional dambreak of a circular dam over a variable porosity. As shown in [5], the circular dambreak problem with a porosity proportional to the inverse of the distance to the centre of the dam has the same analytical solution as a classical one-dimensional dambreak with a constant porosity. Specifications for this test case are given in Table VI.

The calculated profiles and the analytical solution are compared in Figure 14. Both HLLC and PorAS solvers compute correct water depth and unit discharge profiles. Some differences can be observed depending on the considered direction of the profile. The differences are significant only for the proposed solver: at the critical point, the PorAS solver underestimates the water depth in the x and y directions. Nevertheless, there is always less diffusion and the water depth upstream to the shock is better estimated using the PorAS solver.

3.4. Convergence study (T_6)

The convergence of the numerical solution has been carried out on grids with decreasing spacing and time step. The test case is a steady flow in a culvert with variable porosity and bottom

DOI: 10.1002/fld

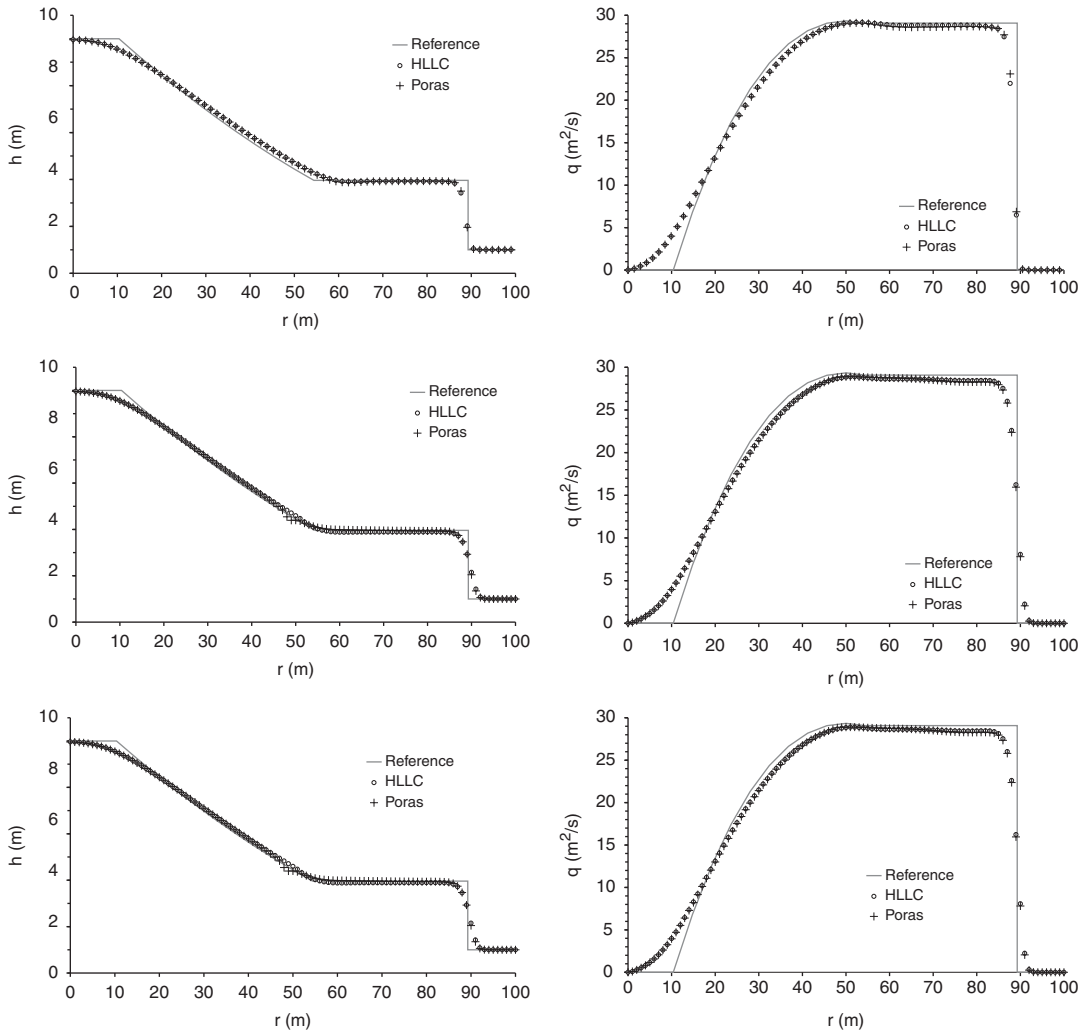

Figure 14. Two-dimensional dambreak with variable porosity (T_5). Reference, HLLC and PorAS solutions; Top: Left, simulated depth along the first bisector; Right, simulated unit discharge along the first bisector; Middle: Left, simulated depth along the X-axis; Right, simulated unit discharge along the X-axis; Bottom: Left, simulated depth along the Y-axis; Right, simulated unit discharge along the Y-axis.

elevation. The evolutions of both porosity and bottom elevation along the culvert are shown in Figure 15.

The semi-analytical solution has been calculated by discretizing the momentum equation. Combining the momentum and continuity equations leads to:

$$\phi(1-Fr^2)\frac{\mathrm{d}h}{\mathrm{d}x} = hFr^2\frac{\mathrm{d}\phi}{\mathrm{d}x} - \phi\frac{\mathrm{d}z_\mathrm{b}}{\mathrm{d}x} \tag{72}$$

DOI: 10.1002/fld

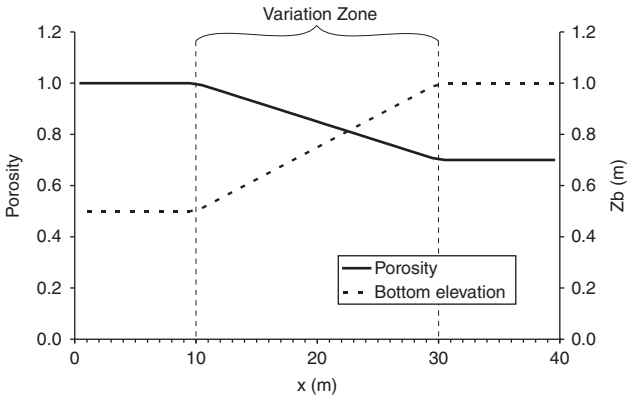

Figure 15. Longitudinal evolution of the porosity and the bottom elevation for the convergence test case.

Since the flow regime has been chosen to be subcritical, Equation (72) should be discretized from downstream to upstream:

$$h_{\mathrm{u}} = h_{\mathrm{d}} + \frac{1}{\phi_{\mathrm{d}}(1 - Fr_{\mathrm{d}}^2)}[h_{\mathrm{d}} Fr_{\mathrm{d}}^2(\phi_{\mathrm{u}} - \phi_{\mathrm{d}}) - \phi_{\mathrm{d}}(z_{\mathrm{b_u}} - z_{\mathrm{b_d}})] \tag{73}$$

To test convergence, computational time step and spatial step should be decreased in the same way. As the equations are explicitly solved, the Courant–Lax–Friedrich conditions has to be insured:

$$Cr = \frac{\lambda \Delta t}{\Delta x} \leqslant \frac{1}{2} \tag{74}$$

The time step is directly computed by the model ensuring that the Courant stability constraint is verified. The time step is therefore automatically reduced in a same proportion that the spatial step. Computational parameters are presented in Table VII. A first coarse grid ($\Delta x = 1$ m) is used to calculate the numerical solution using both the HLLC and the PorAS solvers ($T_{6,1}$). The convergence is checked by decreasing the cell width by 10 for the second convergence test case ($T_{6,2}$) and by 100 for the third convergence test case ($T_{6,3}$). For the sake of clarity, only one point every 10 cells (respectively, 100 cells) are represented in the Figure 17 (respectively, 18). The calculated profiles and the analytical solution are compared in Figures 16–18. Both HLLC and PorAS solvers compute correct water depth and unit discharge profiles for all the test cases. For the coarse grid (Figure 16), the free surface is overestimated by the PorAS solver and underestimated by the HLLC solver; both solvers overestimate the unit discharge. The computed discharge is correctly estimated by the PorAS solver and overestimated by the HLLC solver in the variation zone. As the spatial step decreases, the numerical solutions for the water elevation and the unit discharge computed with both the HLLC and the PorAS solvers get closer from the analytical solution. The difference between the discharge computed by the HLLC solver and the reference solution decreases proportionally to the spatial step decreases. The computational effort required to simulate 1000 s for each test case of the convergence study using both solvers is presented in Table VIII. For all the simulations, the PorAS solver saves 4.5% of the computational duration.

DOI: 10.1002/fld

Table VII. Parameters for the convergence test cases ($T_{6,1}$, $T_{6,2}$ and $T_{6,3}$).

Symbol	Meaning	Value
g	Gravitational acceleration	$9.81\,\mathrm{m\,s^{-2}}$
	Porosity upstream to the variation zone	0.7
	Porosity downstream to the variation zone	1
	Bottom elevation upstream to the variation zone	0.5 m
	Bottom elevation downstream to the variation zone	1.0 m
l_0	Length of the variation zone	20 m
X_0	Position of the beginning of the variation zone	10 m
L	Domain length	$100\,\mathrm{m} \times 1\,\mathrm{m}$
	Cell size	
Δx	for test case $T_{6,1}$	$1\,\mathrm{m} \times 1\,\mathrm{m}$
	for test case $T_{6,2}$	$0.1\,\mathrm{m} \times 1\,\mathrm{m}$
	for test case $T_{6,3}$	$0.01\,\mathrm{m} \times 1\,\mathrm{m}$
q	Prescribed unit discharge at the upstream boundary	$1\,\mathrm{m^2\,s^{-1}}$
z_d	Prescribed water level at the downstream boundary	2 m
z_i	Initial water depth over the domain	1 m

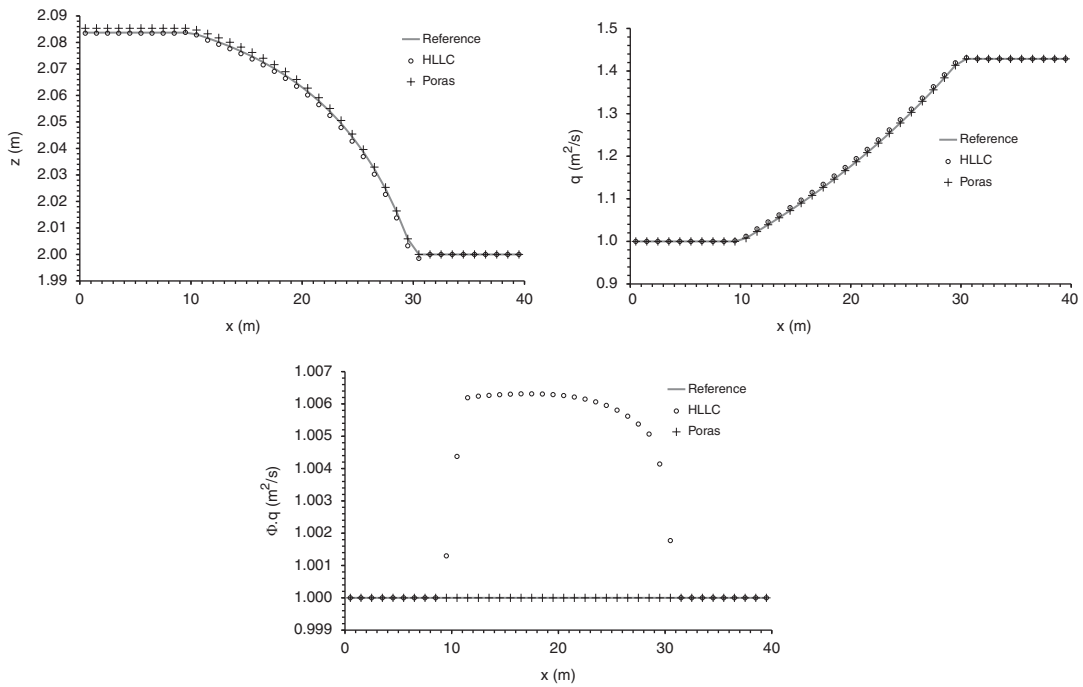

Figure 16. First convergence test case ($T_{6,1}$). Reference, HLLC and PorAS solutions; Top left, simulated water elevation; Top right, simulated unit discharge; Bottom, simulated discharge.

DOI: 10.1002/fld

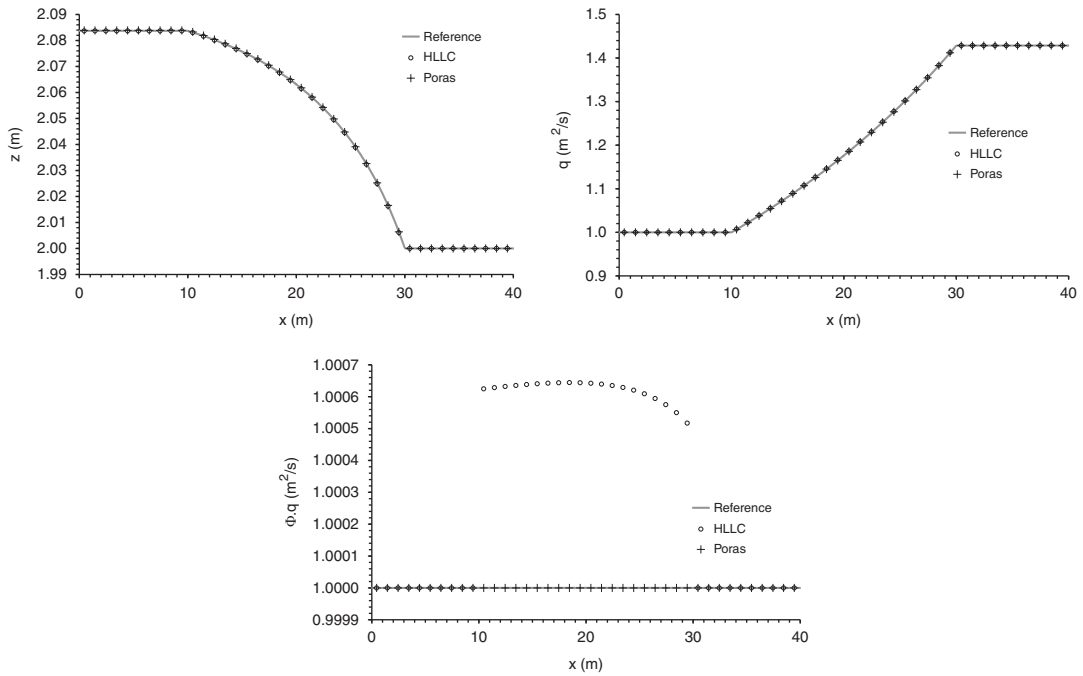

Figure 17. Second convergence test case ($T_{6,2}$). Reference, HLLC and PorAS solutions; Top left, simulated water elevation; Top right, simulated unit discharge; Bottom, simulated discharge.

3.5. Experimental test case (T_7)

The usefulness and the validity of the porosity approach for large scale modelling of floods in the presence of urbanized areas have been demonstrated in [5]. In the previous section, the validity of the PorAS solver is demonstrated using (semi-) analytical test cases. In order to check the validity of the PorAS solver under real configuration, comparisons of the model results for both HLLC and PorAS solvers are made with experimental data. The experiments were performed using a scale model of the Italian Toce valley at CESI (Italy) during the IMPACT European project. Square blocks are implemented in a 7 m long and 3.5–5 m wide model to represent urban area (Figure 19).

The measured hydrograph is used as upstream boundary conditions. The computed water surface elevations using both the PorAS and the HLLC solvers at $t = 20$ s are compared with the measured values (Figure 20). There is no significant differences between the computed profiles using the PorAS and the HLLC solver. Moreover, the computed water levels match the measurements; the explanations for the differences presented in [5] remain valid.

4. CONCLUSIONS

An approximate-state Riemann solver has been developed for the classical shallow water equations with source terms arising from non-uniform porosity and bottom elevation. Using the flux components instead of the conserved variable in the expression of the Riemann invariants allows

DOI: 10.1002/fld

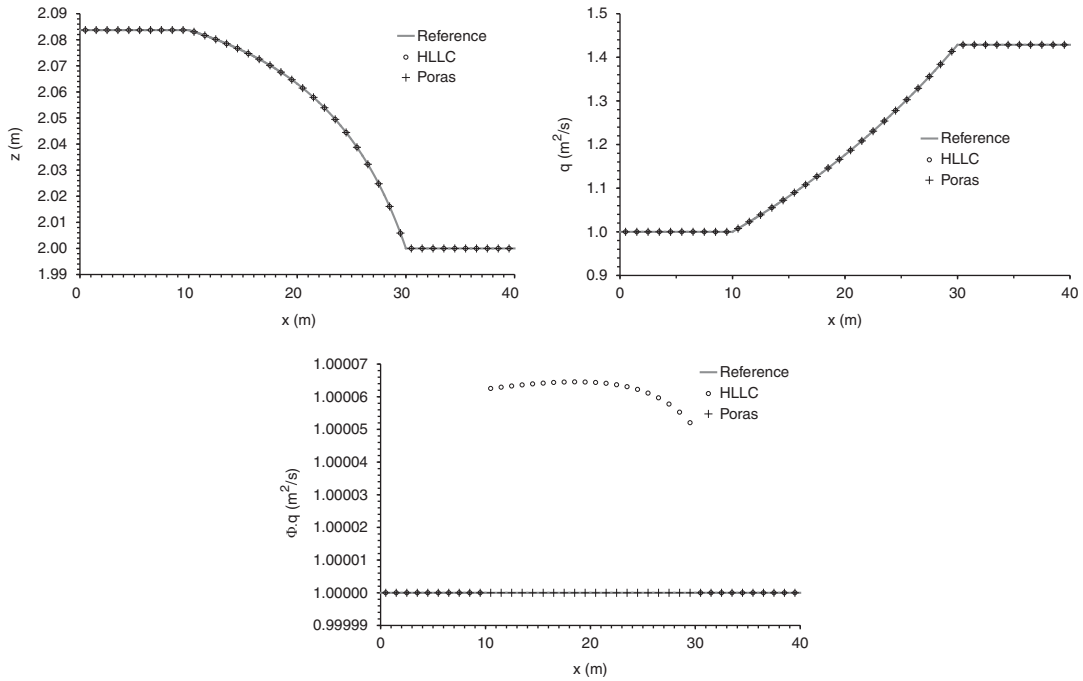

Figure 18. Third convergence test case ($T_{6,3}$). Reference, HLLC and PorAS solutions; Top left, simulated water elevation; Top right, simulated unit discharge; Bottom, simulated discharge.

Table VIII. Computational effort for the convergence test cases ($T_{6,(1,2,3)}$).

	PorAS solver	HLLC solver
$T_{6,1}$	1.50 s	1.52 s
$T_{6,2}$	1 min 6 s	1 min 9 s
$T_{6,3}$	1 h 38 min 53 s	1 h 43 min 20 s

the direct calculation of the fluxes at the interfaces, while accounting for the influence of the source terms in the characteristic equations. This results in improved solution of the continuity and momentum equations, in particular near steady-state configurations. The different computational examples have highlighted both advantages and defaults of the proposed solver. For the transient applications, the efficiency of the proposed solver is the same as that the efficiency of the HLLC solver. On the dambreak simulations with porosity, the over (or under) estimated water depth close to the critical point computed by the proposed solver is balanced by steeper fronts and smaller numerical diffusion. Clear improvements have been obtained with the proposed solver on steady-state simulation with varying parameters (porosity and/or bottom slope). The PorAS solver can therefore be used for both transient and steady-state simulation, substantial improvements being observed on highly variable geometry parameters. The proposed solver is used with a classical Godunov scheme and can easily be extended to higher-order scheme since no modifications to the

DOI: 10.1002/fld

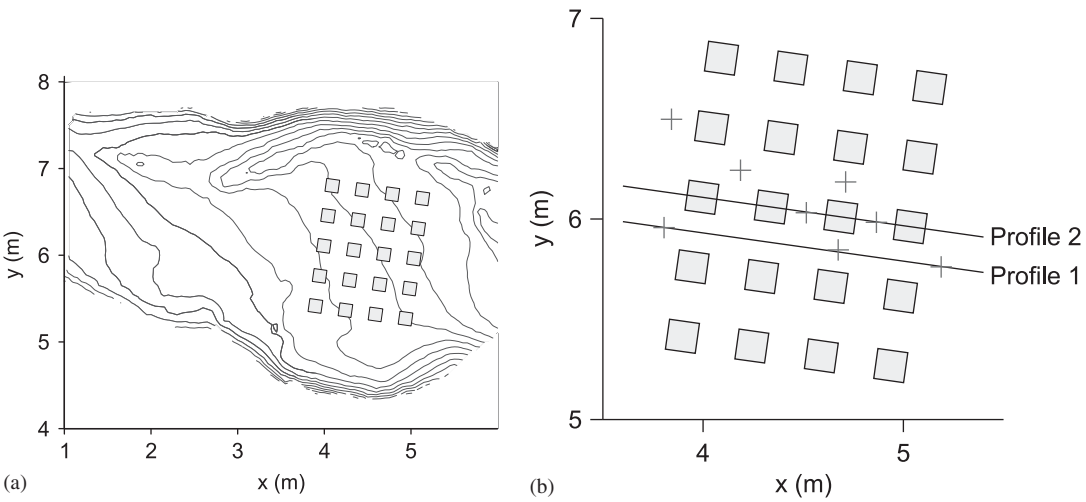

Figure 19. Topography of the valley for the Toce test case, contour line spacing 0.01 m (a). Sketch of the longitudinal profiles and measurement points (b).

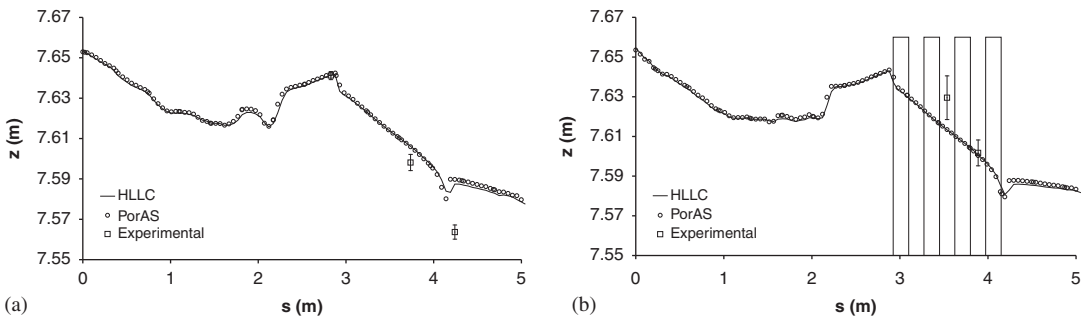

Figure 20. Water surface elevation at $t = 20$ s along profile 1 (a) and profile 2 (b).

Riemann solver are required. Further developments, especially the treatment of critical points, are under investigation.

ACKNOWLEDGEMENTS

This research has been partly carried out during a Ph.D. Thesis funded by GEI (Ginger Environnement et Infrastructures), a GINGER Group company and by the ANRT (Association Nationale de la Recherche et de la Technologie), contract no. 2006/838.

REFERENCES

1. Alcrudo F, Benkhaldoun F. Exact solutions to the Riemann problem of the shallow water equations with a bottom step. *Computers and Fluids* 2001; **30**:643–671.
2. Rochette D, Clain S. Two-dimensional computation of gas flow in a porous bed characterized by a porosity jump. *Journal of Computational Physics* 2006; **219**:104–119.

DOI: 10.1002/fld

3. Defina A, D'Alpaos L, Mattichio B. A new set of equations for very shallow water and partially dry areas suitable to 2D numerical domains. *Proceedings Specialty Conference* '*Modelling of Flood Propagation over Initially Dry Areas*', Milano, Italy, 29 June–1 July 1994.
4. Hervouet JM, Samie H, Moreau B. Modelling urban areas in dam-break flood-wave numerical simulations. *Proceedings of the International Seminar and Workshop on Rescue Actions Based on Dambreak Flow Analysis*, Seinâjoki, Finland, 1–6 October 2000.
5. Guinot V, Soares-Frazão S. Flux and source term discretization in two-dimensional shallow water models with porosity on unstructured grids. *International Journal for Numerical Methods in Fluids* 2006; **50**:309–345.
6. Lhomme J, Guinot V. A general approximate-state Riemann solver for hyperbolic systems of conservation laws with source terms. *International Journal for Numerical Methods in Fluids* 2007; **53**:1509–1540.
7. Godunov SK. A difference method for numerical calculation of discontinuous equations of hydrodynamic equations. *Matematicheskii Sbornik* 1959; **47**:271–300 (in Russian).
8. Roe PL. Approximate Riemann solvers, parameter vectors and difference scheme. *Journal of Computational Physics* 1981; **43**:357–372.
9. Roe PL, Pike J. Efficient construction and utilisation of approximate Riemann solutions. *Computing Methods in Applied Science and Engineering*. North-Holland: Amsterdam, 1984.
10. Harten A, Hyman P. Self-adjusting grid methods for one-dimensional hyperbolic conservation laws. *Journal of Computational Physics* 1983; **50**:235.
11. Roe PL. Sonic flux formulae. *SIAM Journal on Scientific and Statistical Computing* 1992; **13**(2):611–630.
12. Toro EF. *Riemann Solvers and Numerical Methods for Fluid Dynamics*. Springer: Berlin, 1997.
13. Toro EF. A linearised Riemann solver for the time-dependent Euler equations of the gas dynamics. *Proceedings of the Royal Society of London*, *Series A* 1991; **434**:683–693.
14. Toro EF, Chou CC. A linearised Riemann solver for the steady supersonic Euler equations. *International Journal for Numerical Methods in Fluids* 1993; **16**:173–186.
15. Quirk JJ. An alternative to unstructured grids for computing gas dynamic flows around arbitrarily complex two dimensional bodies. *Computers and Fluids* 1994; **23**(1):125–142.
16. Harten A, Lax P-D, Van-Leer B. On upstream differencing and Godunov-type schemes for hyperbolic conservation laws. *SIAM Review* 1983; **25**:35–61.
17. Guinot V. *Wave Propagation in Fluids*: *Models and Numerical Techniques*. ISTE Ltd., John Wiley & Sons Inc.: London, Hoboken, 2008.
18. Toro EF, Spruce M, Speares W. Restoration of the contact surface in the HLL-Riemann solver. *Shock Waves* 1994; **4**:25–34.
19. Osher S, Solomon F. Upwind difference schemes for hyperbolic conservation laws. *Mathematics of Computation* 1982; **38**:339–374.
20. Linde T. A practical, general-purpose, two-state HLL Riemann solver for hyperbolic conservation laws. *International Journal for Numerical Methods in Fluids* 2002; **40**:391–402.
21. Lax P-D. Hyperbolic systems of conservation laws. *Communications on Pure and Applied Mathematics* 1957; **10**:537–566.
22. Colella P. Glimm's method for gas dynamics. *SIAM Journal on Scientific and Statistical Computing* 1982; **3**:76–110.
23. Dukowicz JK. A general, non iterative Riemann solver for Godunov's method. *Journal of Computational Physics* 1985; **61**:119–137.
24. Guinot V. Riemann solvers for water hammer simulations by Godunov method. *International Journal for Numerical Methods in Engineering* 2000; **49**:851–870.
25. Guinot V. Numerical simulation of two-phase flow in pipes using Godunov method. *International Journal for Numerical Methods in Fluids* 2001; **50**:1169–1189.
26. Ivings MJ, Causon DM, Toro EF. On Riemann solvers for compressible fluids. *International Journal for Numerical Methods in Fluids* 1998; **28**:395–418.
27. Toro EF. Direct Riemann solvers for the time-dependent Euler equation. *Shock Waves* 1995; **5**:75–80.
28. LeVeque RJ. Balancing source terms and flux gradients in high-resolution Godunov methods: the quasi-steady wave-propagation algorithm. *Journal of Computational Physics* 1998; **146**:346–365.
29. Bermudez A, Vazquez ME. Upwind methods for hyperbolic conservation laws with source terms. *Computers and Fluids* 1994; **8**:1049–1071.
30. Greenberg JM, Le Roux AY. A well balanced scheme for the numerical processing of the source terms in hyperbolic equation. *SIAM Journal on Numerical Analysis* 1996; **33**:1–16.

DOI: 10.1002/fld

31. Gallouët T, Hérard JM, Seguin N. Some approximate Godunov schemes to compute shallow-water equations with topography. *Computers and Fluids* 2003; **32**:479–513.
32. Lhomme J. One-dimensional, two-dimensional and macroscopic approaches to urban flood modelling. *Ph.D. Thesis*, Montpellier 2 university, 2006 (in French).
33. Sanders BF, Schubert JE, Gallegos HA. Integral formulation of shallow-water equations with anisotropic porosity for urban flood modelling. *Journal of Hydrology* 2008; **362**:19–38.
34. Soares-Frazo S, Lhomme J, Guinot V, Zech Y. Two-dimensional shallow water model with porosity for urban flood modelling. *Journal of Hydraulic Research* 2008; **46**(1):45–64.
35. Garcia-Navarro P, Vazquez-Cendon ME. On numerical treatment of the source terms in the shallow water equations. *Computers and Fluids* 2000; **29**:951–979.
36. Nujíc M. Efficient implementation of non-oscillatory schemes for the computation of free-surface flows. *Journal of Hydraulic Research* 1995; **33**:101–111.
37. Jeffrey A, Tanuiti T. *Nonlinear Wave Propagation*. Academic press: New York, 1964.

DOI: 10.1002/fld