

An unstructured finite volume model for dam-break floods with wet/dry fronts over complex topography

Lixiang Song¹, Jianzhong Zhou^{1,*}, Qingqing Li¹, Xiaoling Yang²
and Yongchuan Zhang¹

¹*School of Hydropower and Information Engineering, Huazhong University of Science and Technology, Wuhan 430074, People's Republic of China*

²*School of Science, Hubei University of Technology, Wuhan 430068, People's Republic of China*

SUMMARY

A robust, well-balanced, unstructured, Godunov-type finite volume model has been developed in order to simulate two-dimensional dam-break floods over complex topography with wetting and drying. The model is based on the nonlinear shallow water equations in hyperbolic conservation form. The inviscid fluxes are calculated using the HLLC approximate Riemann solver and a second-order spatial accuracy is achieved by implementing the MUSCL reconstruction technique. To prevent numerical oscillations near shocks, slope-limiting techniques are used for controlling the total variation of the reconstructed field. The model utilizes an explicit two-stage Runge–Kutta method for time stepping, whereas implicit treatments for friction source terms. The novelties of the model include the flux correction terms and the water depth reconstruction method both for partially and fully submerged cells, and the wet/dry front treatments. The proposed flux correction terms combined with the water depth reconstruction method are necessary to balance the bed slope terms and flux gradient in the hydrostatical steady flow condition. Especially, this well-balanced property is also preserved in partially submerged cells. It is found that the developed wet/dry front treatments and implicit scheme for friction source terms are stable. The model is tested against benchmark problems, laboratory experimental data, and realistic application related to dam-break flood wave propagation over arbitrary topography. Numerical results show that the model performs satisfactorily with respect to its effectiveness and robustness and thus has bright application prospects. Copyright © 2010 John Wiley & Sons, Ltd.

Received 7 April 2010; Revised 22 June 2010; Accepted 29 June 2010

KEY WORDS: shallow water equations; finite volume; unstructured; well-balanced scheme; wetting and drying; source terms; complex topography; dam-break floods

1. INTRODUCTION

Dam-break flood resulting from a sudden release of water due to the catastrophic collapse of a dam is a serious threatening environment hazard, which causes loss of human life and substantial socio-economic damage. Prediction of dam-break flood is of interest and is required to identify the inundated area, the travel time of the flood waves, and the time series of water depth on a place. The shallow water equations (SWEs) give a surprisingly accurate representation of dam-break flood due to its long wave hydrodynamics and the neglectable vertical acceleration of water particles. For this purpose, many numerical approaches [1–13] have been developed to simulate the routing of the ensuing flood from the failure of a dam based on SWEs during the previous decades.

*Correspondence to: Jianzhong Zhou, School of Hydropower and Information Engineering, Huazhong University of Science and Technology, Wuhan 430074, People's Republic of China.

†E-mail: jz.zhou@hust.edu.cn

Although numerical models of dam-break flood simulation have been studied for many years, three persistent problems dog the schemes in practice. First, the discontinuities, such as shocks, contact discontinuities and shear waves, can easily develop spontaneously even from smooth initial data due to the hyperbolic character of SWEs, and so they must be dealt with both mathematically and computationally [14]. The use of a naive method can lead to spurious oscillations near a shock wave, computing shocks with the wrong strength, or propagating shocks with the wrong speed [15]. Second, when modeling realistic dam-break floods over highly irregular bottom topography, the numerical treatments of source terms of SWEs play an important role in the final accuracy of the obtainable results [10], while most available numerical methods face the problem of imbalance between the bed slope source terms and flux gradient, particularly when unstructured grids are used. Third, inundation modeling always involves flooding and recession process, and correspondingly requires proper strategies to deal with wetting and drying. In finite volume schemes where the depth-averaged velocity components are normally determined by dividing the discharge per unit width by the local water depth, unphysical high velocities can be predicted at wet/dry fronts where the water depth is extremely small. These can lead in turn to predictions of negative water depth and numerical instability.

Godunov-type methods, which compute mass and momentum fluxes by solving a local Riemann problem at the interface between neighboring cells, have recently been developed in the framework of finite volume discretization for solving the SWEs on structured or unstructured meshes. These methods provide efficient approaches to accommodate sub-, super- and trans-critical flows as well as flows with discontinuities. The high-order reconstruction methods and TVD limiter techniques [14] give the general formulation basis of a 'high-order accuracy, high-resolution method' that tackles the first problem described above. At the same time, considerable progress has been made in devising numerical models of dam-break flood with regard to the treatments of source terms and wet/dry fronts. To balance the flux gradient and bed slope source terms, several treatments combined with Godunov finite volume schemes are presented in the literature (see e.g. [1–5, 13, 16, 17]). Liang and Borthwick [1] presented a Godunov-type shallow flow solver on dynamically adaptive quadtree grids, and solved the well-balanced SWEs with wet/dry fronts by a local bed slope modification method. Begnudelli and Sanders [2, 4] tracked fluid volume and the free surface elevation in partially submerged cells for wetting and drying, and developed a high-resolution, unstructured finite volume model for shallow water flows over arbitrary topography. Zia and Banihashemi [3] introduced an explicit Godunov-type solution algorithm for the SWEs, which can deal with wet/dry fronts and uneven beds. The proposed method of bed slope source terms discretization is based on the assumption of constant water level. Zhou *et al.* [16] proposed surface gradient method (SGM), which is suitable for both steady and unsteady shallow water flows and is easy to implement. The SGM is efficient, but requires a variable reconstruction step. Kuiry *et al.* [5] used the first-order Euler implicit scheme for time discretization with a Godunov-type upwind finite volume formulation. Generally, implicit schemes allow for large time step and unconditional stability but this is not the case when using Riemann solver. The proposed pressure term treatments ensure the still water condition in cells which are fully submerged. Yoon and Kang [6] employed the fully implicit treatments of friction source terms to alleviate the problems associated with numerical instabilities due to small water depths near the wet/dry fronts. Valiani *et al.* [10] treated the bottom slope in an explicit way while the bed friction in a semi-implicit way. Nikolos and Delis [11] extended the treatment of bed slope term to second-order accuracy within the node-centered finite volume formulation. Xia *et al.* [12] considered the effects of sediment concentrations and bed evolution on the flood wave propagation and used a matrix method to jointly solve the hydrodynamic, sediment transport, and morphological processes. George [17] proposed an augmented Riemann solver in which bottom was taken into account. The solver can preserve all steady flows in one dimension but this is not the case in two dimensions.

The present work reports on the developing and testing of a numerical model for dam-break floods which embodies the ability to accurately capture the hydraulic jump without unphysical numerical oscillation, handle arbitrary topography and boundary of problem domains, converge to and exactly preserve the hydrostatic steady states, simulate sub-, super-, and trans-critical

flows, and prevent the numerical instability problem from small water depth at wet/dry fronts. The present numerical model is second-order accurate both in space and time and is based on a Godunov-type finite volume scheme implemented on unstructured triangular grids. In addition, HLLC approximate Riemann solver is used for the evaluation of the numerical fluxes at the cell interfaces, assuming a local Riemann problem in the direction normal to the interface. The volume/free-surface relationships (VFRs) [2] are incorporated into the present scheme to make a distinction between the free surface elevation at centroid and the cell-averaged depth, whereas the bed elevation is assigned to the vertices of cells. Flux correction is added for both partially and fully submerged cells and is necessary to ensure the hydrostatic steady flow condition. The possible stiff problem due to the explicit two-stage Runge–Kutta treatment of friction terms is demonstrated, and an implicit treatment that does not invert the direction of velocity is proposed to enhance the model stability and efficiency. Numerical tests are carried out and the results show that the present model performs satisfactorily and thus has bright application prospects.

2. GOVERNING EQUATIONS

The two-dimensional depth-averaged SWEs, which can be derived from integrating the Navier–Stokes equations over the flow depth by assuming a hydrostatic pressure distribution, as well as a sufficiently small channel slope and constant fluid density with free surface and bottom boundary [15], are widely used for governing the dam-break flows, and are verified by researchers with experimental and field data [10, 18]. These equations are a simplified model of approximating the depth and horizontal momentum of the flows, which is governed by the following time-dependent system of non-linear partial differential equations in conservative formulation with source terms:

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{E}}{\partial x} + \frac{\partial \mathbf{G}}{\partial y} = \mathbf{S} \quad (1)$$

where t denotes time; x and y are Cartesian coordinates; \mathbf{U} is the vector of conserved physical variables; \mathbf{E} and \mathbf{G} are flux vectors in the x - and y -directions, respectively; \mathbf{S} represents the source vector, in which

$$\mathbf{U} = \begin{bmatrix} h \\ hu \\ hv \end{bmatrix}, \quad \mathbf{E} = \begin{bmatrix} hu \\ hu^2 + \frac{1}{2}gh^2 \\ huv \end{bmatrix}, \quad \mathbf{G} = \begin{bmatrix} hv \\ huv \\ hv^2 + \frac{1}{2}gh^2 \end{bmatrix},$$

$$\mathbf{S} = \mathbf{S}_0 + \mathbf{S}_f = \begin{bmatrix} 0 \\ ghS_{0x} \\ ghS_{0y} \end{bmatrix} - \begin{bmatrix} 0 \\ ghS_{fx} \\ ghS_{fy} \end{bmatrix} \quad (2)$$

where h is the water depth measured from bed; u and v are the depth-averaged flow velocity components along the x - and y -directions, respectively; g is the acceleration due to gravity; (S_{0x}, S_{0y}) are the bed slopes in the x - and y -directions, and (S_{fx}, S_{fy}) is the bed friction slopes in the x - and y -directions, respectively. The source terms can also include viscosity, turbulence, wind effects, and the Coriolis term, but are neglected in this paper. Assuming a fixed bottom elevation $b(x, y)$, $S_{0x} = -\partial b / \partial x$ and $S_{0y} = -\partial b / \partial y$. In this study, the bed friction slopes are assumed to be given by using the Manning-resistance formula

$$S_{fx} = \frac{n^2 u \sqrt{u^2 + v^2}}{h^{4/3}}, \quad S_{fy} = \frac{n^2 v \sqrt{u^2 + v^2}}{h^{4/3}} \quad (3)$$

where n is the empirical Manning roughness coefficient.

3. THE NUMERICAL METHOD

3.1. Computational grid and data structure

In this study, the governing equations are solved on unstructured Delaunay triangular meshes by finite volume method, and the computation domain is divided into triangulated grids. All computational meshes in this paper were generated by constrained Delaunay triangulation using software developed by Shewchuk [19], which can be readily obtained online. A grid is characterized by three vertices, three edges, and three or less immediate neighbor grids. Vertices are specified by the Cartesian coordinates, and edges are defined by the two vertices that they connect. The description of the topology is as follows:

- (1) The vertices of each cell have to be numbered in the counter-clockwise direction, as vertices $v_1 - v_2 - v_3$ shown in Figure 1(a). This can be achieved by reordering the vertices based on the sign of the corresponding computed triangle area, i.e. if we compute the area of cell C_i by:

$$\Omega_i = \frac{1}{2}[(x_1 - x_2)(y_1 + y_2) + (x_2 - x_3)(y_2 + y_3) + (x_3 - x_1)(y_3 + y_1)] \quad (4)$$

where Ω_i denotes the computed area of cell C_i . The vertices $v_1 - v_2 - v_3$ are in the anti-clockwise direction if $\Omega_i > 0$, otherwise, we should exchange the index of vertices v_2 and v_3 , and set the cell area $-\Omega_i$.

- (2) Each edge has a start node, an end node, and a unit outward normal vector. As edges shown in Figure 1(b), the start nodes of edges e_1 , e_2 and e_3 are v_2 , v_3 , and v_1 , respectively, and the end nodes of edges e_1 , e_2 , and e_3 are v_3 , v_1 , and v_2 , respectively. Then the unit outward normal vector \mathbf{n} can be computed by:

$$\mathbf{n} = \begin{bmatrix} n_x \\ n_y \end{bmatrix} = \frac{1}{\sqrt{(x_e - x_s)^2 + (y_e - y_s)^2}} \begin{bmatrix} y_e - y_s \\ x_s - x_e \end{bmatrix} \quad (5)$$

where (x_s, y_s) and (x_e, y_e) denote the coordinates of start node and end node of edge, respectively.

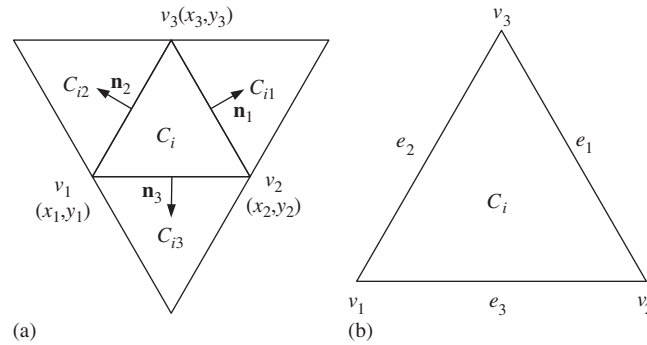
- (3) The neighbor cells of each cell should be numbered the same as edges. If the edge is not shared by another cell, then this edge is termed as boundary edge. One could assign an integral boundary-type mark to boundary edge, for example, 1 for inflow boundary and 2 for outflow boundary.
- (4) The bed elevation is assigned to vertices, and the fluid variables, such as water depth and flow velocity are specified at the centroid of each cell. The terrain within each cell is assumed to vary linearly, i.e. $b_i(x, y) = \alpha_i x + \beta_i y + \gamma_i$. Then the gradient of bottom in cell C_i is

$$\alpha_i = \frac{1}{2\Omega_i}[(y_2 - y_3)b_{i,1} + (y_3 - y_1)b_{i,2} + (y_1 - y_2)b_{i,3}] \quad (6)$$

$$\beta_i = \frac{1}{2\Omega_i}[(x_3 - x_2)b_{i,1} + (x_1 - x_3)b_{i,2} + (x_2 - x_1)b_{i,3}] \quad (7)$$

where $b_{i,k}$ denotes the bed elevation at vertex v_k ($k = 1, 2, 3$) in cell C_i . This corresponds to represent the terrain with second-order spatial accuracy [2]. Because the variables described by (4)–(7) are only dependent upon grid coordinates, they are computed in a preprocessing step for each cell to reduce the computational effort.

Based on the topology described above, the Java object-oriented language is selected to be used in this study. Figure 2 shows the classes of node, edge, triangle and cell used in the programming and some properties of the classes. For convenience, we use $C.p$ to represent the property p of class C . Then $\text{Node}.b$ denotes the bed elevation. $\text{Node}.cells$ denotes the set of cells which are around node. $\text{Edge}.nx$ and $\text{Edge}.ny$ are the components of the outward normal vector. $\text{Edge}.inNtriId$ is the index of edge in the neighbor triangle and will be used in the flux computation. In Figure 1(a), for example, the index of edge

Figure 1. Diagram of the triangular cell C_i with three neighbor cells.

Node	Edge	Triangle	Cell
id : int x : double y : double b : double cells : ArrayList<Cell>	nx : double ny : double length : double inNtriId : int startNode : Node endNode : Node boundType : int outNorFlux: double[3]	id : int area : double alpha : double beta : double edges : Edge[3] nodes : Node[3] neighTri : Triangle[3]	id : int h : double u : double v : double manning : double triangle : Triangle centroidX : double centroidY : double

Figure 2. Classes and properties designed for node, edge, triangle and cell.

specified by v_1 and v_3 in cell C_i equals to 2, and then the property $inNtriId$ of edge, which is specified by v_1 and v_3 in cell C_{i2} , equals to 2. Edge.outNorFlux is the outward normal flux vector. Triangle.alpha and Triangle.beta are the gradients of bed elevation in the x - and y -directions, respectively. Triangle.neighTri denotes the immediate neighbor cells. Cell.manning is the manning roughness coefficient.

- (5) The conserved variable Cell.h is a cell average of water depth. Thus V_i , which denotes the volume of water in C_i , is equal to $h_i \Omega_i$. Cell.h is a nice approximation of water depth at the cell centroid if the cell is wet, i.e. the water depths at three vertices of the cell are positive. But when at least one and no more than two vertices of a cell have a zero water depth, the cell is defined as a partially wet cell and Cell.h could not approximate the water depth at the cell centroid well [2]. This wilders the variable reconstruction process that needs the corresponding value at the cell centroid. See Section 3.4 for more details.

3.2. Finite volume method

Integrating (1) over arbitrary triangle cell C_i , the integral form of the SWEs obtained using the divergence theorem is given by

$$\frac{\partial}{\partial t} \int_{C_i} \mathbf{U} d\Omega = - \oint_{\partial C_i} \mathbf{F} \cdot \mathbf{n} dl + \int_{C_i} \mathbf{S} d\Omega \quad (8)$$

in which \mathbf{n} is the unit outward vector normal to boundary ∂C_i ; $d\Omega$ and dl are the area and arc elements, respectively. The integrand $\mathbf{F} \cdot \mathbf{n}$ is the outward normal flux vector in which $\mathbf{F} = [\mathbf{E}, \mathbf{G}]^T$. The computation of the flux vector in the normal direction allows the treatment of the two-dimensional problem as a local one-dimensional problem.

The finite volume method is based on (8), and the discretization of (8) ensures that the mass and momentum will be conserved across a discontinuity [14]. Discretizing (8), the equation becomes

$$\Omega_i \frac{d\mathbf{U}_i}{dt} = - \sum_{k=1}^3 \mathbf{F}_{i,k}(\mathbf{U}_l, \mathbf{U}_r) \cdot \mathbf{n}_{i,k} L_{i,k} + \mathbf{S}_i \quad (9)$$

where $\mathbf{U}_i = (1/\Omega_i) \int_{C_i} \mathbf{U} d\Omega$ is the cell-averaged vector of conserved variables, Ω_i is the area of cell C_i ; i and k are the index of cell and edge of cell, respectively; $L_{i,k}$ is the arc length of edge k of cell i ; \mathbf{S}_i are the sources associated with cell i ; and $\mathbf{F}_{i,k} \cdot \mathbf{n}_{i,k}$ is the estimate of outward normal flux across edge k of cell i . Consider the rotation matrix and its inverse matrix

$$\mathbf{T}_n = \begin{bmatrix} 1 & 0 & 0 \\ 0 & n_x & n_y \\ 0 & -n_y & n_x \end{bmatrix}, \quad \mathbf{T}_n^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & n_x & -n_y \\ 0 & n_y & n_x \end{bmatrix} \quad (10)$$

where $\mathbf{n} = (n_x, n_y)^T$. Using the rotational invariance property of the two-dimensional SWEs [15], then

$$\mathbf{F}_{i,k}(\mathbf{U}_l, \mathbf{U}_r) \cdot \mathbf{n}_{i,k} = \mathbf{T}_{n_{i,k}}^{-1} \mathbf{E}(\mathbf{T}_{n_{i,k}} \mathbf{U}_l, \mathbf{T}_{n_{i,k}} \mathbf{U}_r) \quad (11)$$

Let

$$\hat{\mathbf{U}} = \mathbf{T}_n \mathbf{U} = \begin{bmatrix} h \\ h(un_x + vn_y) \\ h(-un_y + vn_x) \end{bmatrix} = \begin{bmatrix} h \\ hu_{\perp} \\ hu_{\parallel} \end{bmatrix} \quad (12)$$

where $u_{\perp} = un_x + vn_y$ and $u_{\parallel} = -un_y + vn_x$ are velocity perpendicular and tangential to the cell face, respectively. Then

$$\mathbf{F}(\mathbf{U}) \cdot \mathbf{n} = \mathbf{T}_n^{-1} \mathbf{E}(\hat{\mathbf{U}}) = \begin{bmatrix} hu_{\perp} \\ hu u_{\perp} + \frac{1}{2} gh^2 n_x \\ h v u_{\perp} + \frac{1}{2} gh^2 n_y \end{bmatrix} \quad (13)$$

Equation (11) means that the two-dimensional SWEs can be solved by transforming \mathbf{U} into $\hat{\mathbf{U}}$ and solving the corresponding local one-dimensional Riemann problems.

3.3. HLLC approximate Riemann solver

Based on [15], the numerical flux of HLLC is computed as follows:

$$\mathbf{F}(\mathbf{U}_l, \mathbf{U}_r) \cdot \mathbf{n} = \begin{cases} \mathbf{F}_l & \text{if } s_1 \geq 0 \\ \mathbf{F}_{*l} & \text{if } s_1 < 0 \leq s_2 \\ \mathbf{F}_{*r} & \text{if } s_2 < 0 < s_3 \\ \mathbf{F}_r & \text{if } s_3 \leq 0 \end{cases} \quad (14)$$

where \mathbf{U}_l and \mathbf{U}_r are the left and right Riemann states of a local cell interface, respectively; $\mathbf{F}_l = \mathbf{F}(\mathbf{U}_l) \cdot \mathbf{n}$ and $\mathbf{F}_r = \mathbf{F}(\mathbf{U}_r) \cdot \mathbf{n}$ are both computed by (13). \mathbf{F}_{*l} and \mathbf{F}_{*r} are the numerical fluxes in the left and right sides of the middle region of the Riemann solution which is divided by a contact wave; and s_1, s_2, s_3 are the speeds of the left, contact and right waves, respectively. The flux vectors \mathbf{F}_{*l} and \mathbf{F}_{*r} are given by

$$\mathbf{F}_{*l} = \begin{bmatrix} (\mathbf{E}_{\text{HLL}})^1 \\ (\mathbf{E}_{\text{HLL}})^2 n_x - u_{\parallel,l} (\mathbf{E}_{\text{HLL}})^1 n_y \\ (\mathbf{E}_{\text{HLL}})^2 n_y + u_{\parallel,l} (\mathbf{E}_{\text{HLL}})^1 n_x \end{bmatrix} \quad \mathbf{F}_{*r} = \begin{bmatrix} (\mathbf{E}_{\text{HLL}})^1 \\ (\mathbf{E}_{\text{HLL}})^2 n_x - u_{\parallel,r} (\mathbf{E}_{\text{HLL}})^1 n_y \\ (\mathbf{E}_{\text{HLL}})^2 n_y + u_{\parallel,r} (\mathbf{E}_{\text{HLL}})^1 n_x \end{bmatrix} \quad (15)$$

where $u_{//,l}$ and $u_{//,r}$ are the left and right initial velocities tangential to the interface of a local Riemann problem, $(\mathbf{E}_{\text{HLL}})^1$ and $(\mathbf{E}_{\text{HLL}})^2$ are the first two components of the normal flux \mathbf{E}_{HLL} which is calculated using the HLL solver given by:

$$\mathbf{E}_{\text{HLL}} = \frac{s_3 \mathbf{E}(\hat{\mathbf{U}}_l) - s_1 \mathbf{E}(\hat{\mathbf{U}}_r) + s_1 s_3 (\hat{\mathbf{U}}_r - \hat{\mathbf{U}}_l)}{s_3 - s_1} \quad (16)$$

There are different possible choices of wave speed estimates for SWEs. Zia and Banihashenmi [3] suggest the following approximations including shock wave, rarefaction wave, and dry-bed options, whereby the wave speeds are determined from

$$\begin{aligned} s_1 &= \begin{cases} u_{\perp,l} - \sqrt{g(h_* + h_l)h_*/(2h_l)} & \text{if } h_* > h_l > 0 \\ \min(u_{\perp,l} - \sqrt{gh_l}, u_{\perp,*} - \sqrt{gh_*}) & \text{if } h_* \leq h_l \\ u_{\perp,r} - 2\sqrt{gh_r} & \text{if } h_l = 0 \end{cases} \\ s_3 &= \begin{cases} u_{\perp,r} + \sqrt{g(h_* + h_r)h_*/(2h_r)} & \text{if } h_* > h_r > 0 \\ \max(u_{\perp,r} + \sqrt{gh_r}, u_{\perp,*} + \sqrt{gh_*}) & \text{if } h_* \leq h_r \\ u_{\perp,l} + 2\sqrt{gh_l} & \text{if } h_r = 0 \end{cases} \\ s_2 &= \frac{s_1 h_r (u_{\perp,r} - s_3) - s_3 h_l (u_{\perp,l} - s_1)}{h_r (u_{\perp,r} - s_3) - h_l (u_{\perp,l} - s_1)} \end{aligned} \quad (17)$$

where $u_{\perp,l}$, $u_{\perp,r}$, h_l and h_r are the components of the left and right initial Riemann states for a local Riemann problem, and the middle states $u_{\perp,*}$ and h_* are calculated from

$$h_* = \begin{cases} \frac{1}{g} \left[\frac{1}{2} (\sqrt{gh_l} + \sqrt{gh_r}) + \frac{1}{4} (u_{\perp,l} - u_{\perp,r}) \right]^2 & \text{if } u_{\perp,r} - u_{\perp,l} < 2(\sqrt{gh_l} + \sqrt{gh_r}) \\ 0 & \text{if } u_{\perp,r} - u_{\perp,l} \geq 2(\sqrt{gh_l} + \sqrt{gh_r}) \end{cases} \quad (18)$$

$$u_{\perp,*} = \begin{cases} \frac{u_{\perp,l} + u_{\perp,r}}{2} + \sqrt{gh_l} - \sqrt{gh_r} & \text{if } h_* > 0 \\ 0 & \text{if } h_* = 0 \end{cases} \quad (19)$$

This method of middle states approximation is the same as those proposed in the literature (see e.g. [1, 3, 8, 16]) except the vacuum case considered in this paper. In our numerical experiments we find that letting $u_{\perp,*}$ and h_* be zero when the vacuum is generated by two strong rarefaction waves might be more stable for wave speed estimate.

3.4. High-order reconstruction and limiter

Because the partially wet cell average of h could not approximate the water depth at the cell centroid well, a zero gradient is assigned to the cell when it is partially wet or has a partially wet neighbor cell; hence, the scheme reverts to the first-order spatial accuracy. Otherwise, the second-order spatial accuracy is achieved by means of a linear reconstruction of the variables within a cell, as

$$p_{i,k}^{\text{Rec}} = p_i + \overline{\nabla p_i} \cdot \mathbf{r}_{i,k} \quad (20)$$

where p_i is the variable at the centroid of C_i before reconstruction; $p_{i,k}^{\text{Rec}}$ is the variable at point $k(x_k, y_k)$ in C_i after reconstruction; $\mathbf{r}_{i,k}$ is the position vector of point k relative to the centroid of

the C_i ; $\overline{\nabla p_i}$ is the limited gradient at the centroid of C_i , which is given by

$$\overline{\nabla p_i} = \varphi_i \nabla p_i \quad (21)$$

where φ_i is the limit function; ∇p_i is the unlimited cell-centered gradient and is given by

$$\nabla p_i = \left(-\frac{\alpha^2}{\alpha^3}, -\frac{\alpha^1}{\alpha^3} \right) \quad (22)$$

in which $\alpha = (\alpha^1, \alpha^2, \alpha^3) = (X_1 - X_3) \times (X_2 - X_3)$; $X_k = (x_{iCk}, y_{iCk}, p_{ik})^T$ ($k=1, 2, 3$), where x_{iCk} , y_{iCk} and p_{ik} are the coordinates and variable value at the centroid of the k th neighboring cell of C_i . The boundary-cell's gradient is calculated based on three vertices surrounding the cell and interpolation is necessary to compute the value of the variables at the vertices. In this study, the linearity preserving interpolation method [6] based on the pseudo-Laplacian formula is used to calculate vertex values. The limit function is given by [20]

$$\varphi_i = \min_{k=1,2,3} (\varphi_k), \quad \varphi_k = \begin{cases} \min \left(1, \frac{p_i^{\max} - p_i}{p_i^k - p_i} \right) & \text{if } p_i^k - p_i > 0 \\ \min \left(1, \frac{p_i^{\min} - p_i}{p_i^k - p_i} \right) & \text{if } p_i^k - p_i < 0 \\ 1 & \text{if } p_i^k - p_i = 0 \end{cases} \quad p_i^k = p_i + \nabla p_i \cdot \mathbf{r}_{i,k} \quad (23)$$

where p_i is the variable value at the centroid of C_i ; p_i^k is the unlimited variable value at the vertex v_k of C_i ; $p_i^{\max} = \max(p_i, p_{i1}, p_{i2}, p_{i3})$ and $p_i^{\min} = \min(p_i, p_{i1}, p_{i2}, p_{i3})$. The use of limit function makes the second-order schemes free from nonphysical oscillation in the presence of shocks. A cell can be identified as a partially wet cell or not, depending on the limited gradient and water level reconstruction. If a cell is partially wet, the limited gradient of the cell is set to zero.

Without loss of universality, in the following equations it will be assumed that $b_{i,1} \leq b_{i,2} \leq b_{i,3}$. Then if C_i is wet, i.e. $h_i + (b_{i,1} + b_{i,2} + b_{i,3})/3 > b_{i,3}$, the reconstruction of water depth is replaced by water level reconstruction and is given by

$$h_{i,k}^{\text{Rec}} = \eta_{i,k}^{\text{Rec}} - b_{i,k}^{\text{mid}} > 0 \quad (k=1, 2, 3) \quad (24)$$

where $h_{i,k}^{\text{Rec}}$, $\eta_{i,k}^{\text{Rec}}$, $b_{i,k}^{\text{mid}}$ are the reconstructed values of water depth and water level and bed elevation at the middle point of the edge k in C_i , respectively; and the reconstructed water level is computed by (20). If the three reconstructed depths of C_i are bigger than a small positive value ($\delta=0.1\text{m}$ is used in this study), the associated face values of the velocity are then calculated by:

$$u_{i,k}^{\text{Rec}} = (hu)_{i,k}^{\text{Rec}} / h_{i,k}^{\text{Rec}}, \quad v_{i,k}^{\text{Rec}} = (hv)_{i,k}^{\text{Rec}} / h_{i,k}^{\text{Rec}} \quad (k=1, 2, 3) \quad (25)$$

Otherwise the reconstructed values are given by

$$h_{i,k}^{\text{Rec}} = h_i, \quad u_{i,k}^{\text{Rec}} = u_i, \quad v_{i,k}^{\text{Rec}} = v_i \quad (k=1, 2, 3) \quad (26)$$

If C_i is partially wet, the water level in the wetted portion of C_i is computed by VFRs [2]:

$$\eta_i = \begin{cases} \eta_i^1 = b_{i,1} + \sqrt[3]{3h_i(b_{i,2} - b_{i,1})(b_{i,3} - b_{i,1})} & \text{if } b_{i,1} < \eta_i^1 \leq b_{i,2} \\ \eta_i^2 = \frac{1}{2}(-\gamma_1 + \sqrt{\gamma_1^2 - 4\gamma_2}) & \text{if } b_{i,2} < \eta_i^2 \leq b_{i,3} \end{cases} \quad (27)$$

where

$$\gamma_1 = b_{i,3} - 3b_{i,1}, \quad \gamma_2 = 3h_i b_{i,1} - 3h_i b_{i,3} - b_{i,2} b_{i,3} + b_{i,1} b_{i,2} + b_{i,1}^2 \quad (28)$$

Then the reconstructed depth $h_{i,k}^{\text{Rec}}$ is given by

$$h_{i,k}^{\text{Rec}} = \begin{cases} 0 & \text{if } \eta_i \leq b_{i,k}^{\min} \\ \frac{(\eta_i - b_{i,k}^{\min})^{3/2}}{2(b_{i,k}^{\max} - b_{i,k}^{\min})^{1/2}} & \text{if } b_{i,k}^{\min} < \eta_i \leq b_{i,k}^{\max} \\ \eta_i - \frac{b_{i,k}^{\max} + b_{i,k}^{\min}}{2} & \text{if } \eta_i > b_{i,k}^{\max} \end{cases} \quad (29)$$

where $b_{i,k}^{\min}$ and $b_{i,k}^{\max}$ are the minimum and maximum bed elevations of the two endpoints of the edge k in C_i . The associated face values of the velocities are set to be the centroid value.

3.5. Bed slope treatment and flux correction

The bed slope source terms have the form $S_{i,0} = (0, S_{i,0x}, S_{i,0y})^T$ and are given by

$$S_{i,0x} = - \int_{C_i} gh\alpha_i d\Omega = -g\alpha_i \int_{C_i} h d\Omega = -g\alpha_i h_i \Omega_i \quad (30)$$

$$S_{i,0y} = - \int_{C_i} gh\beta_i d\Omega = -g\beta_i \int_{C_i} h d\Omega = -g\beta_i h_i \Omega_i \quad (31)$$

When the fluid velocity is zero, the water surface becomes horizontal, and $\eta = h + b$ is constant where $h > 0$, i.e. $\partial b / \partial x = -\partial h / \partial x$, and $\partial b / \partial y = -\partial h / \partial y$. Then we have

$$S_{i,0x} = - \int_{C_i} gh \frac{\partial b}{\partial x} d\Omega = \int_{C_i} \frac{\partial}{\partial x} \left(\frac{1}{2} gh^2 \right) d\Omega = \oint_{\partial C_i} \frac{1}{2} gh^2 n_x dl = \sum_{k=1}^3 \frac{1}{2} g \left(\int_{\varepsilon=0}^{L_{i,k}} h_\varepsilon^2 d\varepsilon \right) n_{x,k} \quad (32)$$

$$S_{i,0y} = - \int_{C_i} gh \frac{\partial b}{\partial y} d\Omega = \int_{C_i} \frac{\partial}{\partial y} \left(\frac{1}{2} gh^2 \right) d\Omega = \oint_{\partial C_i} \frac{1}{2} gh^2 n_y dl = \sum_{k=1}^3 \frac{1}{2} g \left(\int_{\varepsilon=0}^{L_{i,k}} h_\varepsilon^2 d\varepsilon \right) n_{y,k} \quad (33)$$

This indicates that, if we consider the stationary steady states, the bed slope term within the cell in a particular direction is numerically equal to the sum of the hydrostatic pressure forces on the vertical faces of a cell in the same direction [5].

For the water prism shown in Figure 3(a)–(c), and assuming $b_1 < b_2 < b_3$, then the total hydrostatic pressure force on the vertical face 1-2-2'-1' or 1-2''-1' is given by:

$$\int_{\varepsilon=0}^{L_3} h_\varepsilon^2 d\varepsilon = \begin{cases} 0, & \eta \leq b_1 \\ \frac{(\eta - b_1)^3}{3(b_2 - b_1)} L_3, & b_1 < \eta \leq b_2 \\ \frac{h_1^2 + h_1 h_2 + h_2^2}{3} L_3, & \eta > b_2 \end{cases} \quad (34)$$

where L_3 is the length of the considered edge 1-2 in the horizontal plane; η is the water surface elevation; h_1 and h_2 are the water depths at the vertices 1 and 2, respectively; b_1 , b_2 , and b_3 are the bottom elevation at the vertices 1, 2, and 3, respectively.

When the flow is in steady state with zero velocity, the numerical flux across an edge is given by:

$$\mathbf{F} = \begin{bmatrix} 0 \\ \frac{1}{2} g (h^{\text{Rec}})^2 n_x \\ \frac{1}{2} g (h^{\text{Rec}})^2 n_y \end{bmatrix} \quad (35)$$

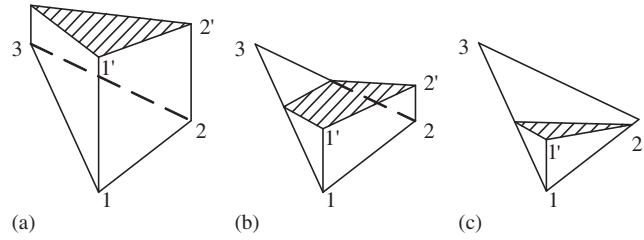


Figure 3. Water prism with sloped bed in triangular cell for: (a) $\eta > b_3$; (b) $b_2 < \eta < b_3$; and (c) $b_1 < \eta < b_2$.

where h^{Rec} is computed by (24) or (29). To balance the flux and bed slop term in the steady-state condition, flux correction terms are added to the flux computed by (14) and expressed mathematically as:

$$\mathbf{F}^{\text{Corr}} = \begin{bmatrix} 0 \\ \frac{1}{2}gH^2n_x \\ \frac{1}{2}gH^2n_y \end{bmatrix} \quad (36)$$

where H^2 is given by:

$$H^2 = \begin{cases} 0, & \eta \leq b_1 \\ \frac{(\eta - b_1)^3}{12(b_2 - b_1)}, & b_1 < \eta \leq b_2 \\ \frac{(h_2 - h_1)^2}{12}, & \eta > b_2 \end{cases} \quad (37)$$

In the presence of a stationary flow field, if a numerical scheme could preserve the conservative property, i.e. maintain the stationary state, then it satisfies the exact C-property [3]. The C-property is approved for the proposed scheme as below.

Considering a cell i which could be either wetted or partially wetted, the following relations stand in the stationary flow field situation for the k th edge ($k = 1, 2, 3$):

$$\eta_{i,k}^{\text{Rec}} = \eta = h + b = \text{Constant (if } h > 0), \quad u_{i,k}^{\text{Rec}} = v_{i,k}^{\text{Rec}} = 0 \quad (38)$$

Then we have

$$\begin{aligned} \sum_{k=1}^3 (\mathbf{F}_{i,k} \cdot \mathbf{n}_{i,k} + \mathbf{F}_{i,k}^{\text{Corr}}) L_{i,k} - \mathbf{S}_{i,0} &= \frac{1}{2}g \sum_{k=1}^3 \begin{bmatrix} 0 \\ [(h_{i,k}^{\text{Rec}})^2 + (H_{i,k})^2] L_{i,k} n_{x,k} \\ [(h_{i,k}^{\text{Rec}})^2 + (H_{i,k})^2] L_{i,k} n_{y,k} \end{bmatrix} \\ &\quad - \frac{1}{2}g \sum_{k=1}^3 \begin{bmatrix} 0 \\ \int_{\varepsilon=0}^{L_{i,k}} h_{\varepsilon}^2 d\varepsilon n_{x,k} \\ \int_{\varepsilon=0}^{L_{i,k}} h_{\varepsilon}^2 d\varepsilon n_{y,k} \end{bmatrix} \end{aligned} \quad (39)$$

where $h_{i,k}^{\text{Rec}}$ and $H_{i,k}$ is computed from (29) and (37), respectively. The integral in (39) is given by (34). It is easy to achieve that $[(h_{i,k}^{\text{Rec}})^2 + (H_{i,k})^2] L_{i,k} = \int_{\varepsilon=0}^{L_{i,k}} h_{\varepsilon}^2 d\varepsilon$ for all cases including $\eta \leq b_{i,k}^{\min}$, $\eta > b_{i,k}^{\max}$, and $b_{i,k}^{\min} < \eta \leq b_{i,k}^{\max}$. Thus we have

$$-\Omega_i \frac{d\mathbf{U}_i}{dt} = \sum_{k=1}^3 (\mathbf{F}_{i,k} \cdot \mathbf{n}_{i,k} + \mathbf{F}_{i,k}^{\text{Corr}}) L_{i,k} - \mathbf{S}_{i,0} = \mathbf{0} \quad (40)$$

Hence, (40) approves the conservative property, satisfying the exact C-property for both partially and fully submerged cells.

3.6. Friction source terms discretization

As the friction source terms lack spatial derivatives, it is easy to implement the corresponding treatment. One way is the combination of splitting method and cell centroid approximation. However, when considering a case with complex topography and involving wetting and drying, the friction term actually dominates the stability of a numerical model. This is demonstrated as follows:

Using splitting method for friction source terms implies that:

$$\frac{d\mathbf{U}_i}{dt} = \mathbf{S}_f(\mathbf{U}_i) \Rightarrow \frac{d}{dt} \begin{bmatrix} h \\ hu \\ hv \end{bmatrix} = \begin{bmatrix} 0 \\ -gn^2 hu \sqrt{u^2 + v^2} / h^{4/3} \\ -gn^2 hv \sqrt{u^2 + v^2} / h^{4/3} \end{bmatrix} \quad (41)$$

It should be noted that $dh/dt=0$, then from (41) we have:

$$\frac{d}{dt} \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} -gn^2 u \sqrt{u^2 + v^2} / h^{4/3} \\ -gn^2 v \sqrt{u^2 + v^2} / h^{4/3} \end{bmatrix} = -gn^2 h^{-4/3} \begin{bmatrix} u \sqrt{u^2 + v^2} \\ v \sqrt{u^2 + v^2} \end{bmatrix} \quad (42)$$

Let $\mathbf{u} = [u, v]^T$, $\mathbf{R}(\mathbf{u}) = -gn^2 h^{-4/3} [u \sqrt{u^2 + v^2}, v \sqrt{u^2 + v^2}]^T$. The Jacobi matrix of $\mathbf{R}(\mathbf{u})$ is

$$\mathbf{J} = \frac{\partial \mathbf{R}(\mathbf{u})}{\partial \mathbf{u}} = -gn^2 h^{-4/3} \begin{bmatrix} \sqrt{u^2 + v^2} + u^2 / \sqrt{u^2 + v^2} & uv / \sqrt{u^2 + v^2} \\ uv / \sqrt{u^2 + v^2} & \sqrt{u^2 + v^2} + v^2 / \sqrt{u^2 + v^2} \end{bmatrix} \quad (43)$$

And the eigenvalue of \mathbf{J} is:

$$\lambda_1 = -gn^2 h^{-4/3} \sqrt{u^2 + v^2}, \quad \lambda_2 = -2gn^2 h^{-4/3} \sqrt{u^2 + v^2} \quad (44)$$

If the two-stage TVD Runge–Kutta method is adopted to solve (42), we get the absolute stability condition by analyzing the model equation $y' = \lambda y$ [21]:

$$|E(\lambda \Delta t)| = |1 + \lambda \Delta t + \frac{1}{2}(\lambda \Delta t)^2| < 1 \quad (45)$$

It should be noted that: $-2 < \lambda \Delta t < 0$. The steep bottom surface may lead to extremely small water depths and predictions of high velocities at wet/dry fronts, and further makes λ_1 and λ_2 highly, i.e. the Lipschitz constant of (42) is very big, hence the stiff problem occurs. Under this circumstance, the time step will be very small due to the stability condition, and the model efficiency will be obviously decreased. To deal with this problem, an implicit scheme is adopted combined with the splitting method for stability and computational efficiency.

Let $\tau = -gn^2 \sqrt{u^2 + v^2} h^{-4/3}$, (42) will be as $du/dt = \tau u$. By implementing implicit scheme [2, 22], we have:

$$u^{n+1} = \frac{1}{1 - \Delta t \hat{\tau}^n} \hat{u}^n \quad (46)$$

$$v^{n+1} = \frac{1}{1 - \Delta t \hat{\tau}^n} \hat{v}^n \quad (47)$$

where $\hat{\tau}^n = -gn^2 \sqrt{(\hat{u}^n)^2 + (\hat{v}^n)^2} (\hat{h}^n)^{-4/3}$; \hat{h}^n , \hat{u}^n , and \hat{v}^n are the initial states for the friction source terms treatment. This scheme does not invert the direction of velocity, i.e. $u^{n+1} \cdot \hat{u}^n \geq 0$, and $v^{n+1} \cdot \hat{v}^n \geq 0$, thus enhances the model stability.

3.7. Time-integration procedure

In this work, the two-stage explicit Runge–Kutta method is employed to achieve second-order accuracy in time. The time discretization formula of (9) may be written as:

$$\mathbf{U}_i^{n+1} = \mathbf{U}_i^n + \frac{1}{2}\Delta t [\mathbf{L}_i(\mathbf{U}^n) + \mathbf{L}_i(\mathbf{U}^{n+1/2})] \quad (48)$$

where \mathbf{L}_i is defined by

$$\mathbf{L}_i(\mathbf{U}) = -\frac{1}{\Omega_i} \left(\sum_{k=1}^3 (\mathbf{F}_{i,k} \cdot \mathbf{n}_{i,k} + \mathbf{F}_{i,k}^{\text{Corr}}) L_{i,k} - \mathbf{S}_{i,0} \right) + \tilde{\mathbf{S}}_{if} \quad (49)$$

and the intermediate conservative vector $\mathbf{U}_i^{n+1/2}$ is given by

$$\mathbf{U}_i^{n+1/2} = \mathbf{U}_i^n + \Delta t \mathbf{L}_i(\mathbf{U}^n) \quad (50)$$

Based on the schemes described above, the process of time integration is:

Step1: Compute the fluxes, flux correction terms and the bed slope approximation, and then \mathbf{U}^n is updated by:

$$\hat{\mathbf{U}}_i^n = \mathbf{U}_i^n - \frac{\Delta t}{\Omega_i} \left(\sum_{k=1}^3 (\mathbf{F}_{i,k} \cdot \mathbf{n}_{i,k} + \mathbf{F}_{i,k}^{\text{Corr}}) L_{i,k} - \mathbf{S}_{i,0} \right) \quad (51)$$

Step2: Make the result of (51) be the initial value, the intermediate conservative vector $\mathbf{U}_i^{n+1/2}$ is computed by (46) and (47). This step corresponds to determining the influence of $\tilde{\mathbf{S}}_{if}$ in (49).

Step3: Repeat step1 and step2 by employing $\mathbf{U}_i^{n+1/2}$ as the initial state, and finally the solution is updated by (48). After updating the solution, cells with a water depth smaller than 10^{-6} m are considered dry, and the water depth and velocities of dry cells are set to be zero.

3.8. Boundary condition

Boundary conditions are imposed on the boundary-edge of the cell and the values of the conserved variables on the edge are extrapolated from the cell-center. According to the theory of characteristics, parameters at each boundary can be computed using Riemann invariants when the contribution of the source terms are neglected [6]:

$$u_{\perp,l} + 2\sqrt{gh_l} = u_{\perp,b} + 2\sqrt{gh_b} \quad (52)$$

where the subscripts l and b denote the variables at the side of the inner cell and the boundary, respectively. Equation (52) is combined with the boundary condition to compute the normal flux at the boundary given by (13). Note that the two boundary conditions, one at the upstream end and the other at the downstream end, are needed when the flow regime is subcritical. For supercritical flow, these two boundary conditions are imposed at the upstream end in the form of water depth, unit discharge, or velocity, i.e. the h_b , $u_{\perp,b}$, and $u_{//,b}$ are directly given and the normal flux is readily obtained.

At solid slip boundaries, i.e. the normal velocity component $u_{\perp,b}$ at the face is zero, the water depth h_b is computed by (52), and the normal flux is $[0, 0.5gh_b^2n_x, 0.5gh_b^2n_y]^T$.

When a free outgoing condition, under which waves pass the boundary without any reflection, is employed at the outflow boundary, all flow variables, h_b , u_b , and v_b at the boundary face are the same as the variables at the inner cell centroid, and then the normal flux at a boundary can be computed by (13).

3.9. Stability criterion

The stability is governed by the Courant–Friedrichs–Lewy (CFL) condition:

$$\Delta t = C_r \cdot \min_{i,k} \left[\left(\frac{\Omega}{(|u_{\perp}| + \sqrt{gh})_k L_k} \right)_i \right], \quad i = 1, 2, \dots, N, \quad k = 1, 2, 3 \quad (53)$$

where Δt is the size of time step for time integration; C_r is the CFL number specified in the range $0 < C_r \leq 1$, and $C_r = 0.8$ is adopted in this study; u_\perp and h are Roe's average [14] of the reconstructed value; N is the number of cells.

4. NUMERICAL RESULTS AND DISCUSSION

4.1. Dam-break flooding over three humps in a closed channel

To evaluate the performance of the model in problems involving wetting–drying process and mass conservation, in this test we consider the case of a dam-break wave propagation over an initially dry floodplain with three humps, which was originally proposed by Kawahara and Umetsu [23] and has been reconsidered by other researchers (see e.g. [1, 2, 4, 11, 24, 25]). The dam-break occurs in a 75-m long and 30-m wide closed channel, with the bed topography defined by:

$$b(x, y) = \max[0, 1 - \frac{1}{8}\sqrt{(x-30)^2 + (y-6)^2}, 1 - \frac{1}{8}\sqrt{(x-30)^2 + (y-24)^2}, 3 - \frac{3}{10}\sqrt{(x-47.5)^2 + (y-15)^2}] \quad (54)$$

The dam is located at $x = 16$ m that initially retains still water with surface elevation 1.875 m, whereas the rest of the channel is considered dry. The Manning coefficient set to $n = 0.018$. Solid slip conditions are applied at the lateral walls of the channel. Figure 4 shows the bed topography within the computational domain. At $t = 0$, the dam collapses instantaneously. The flow domain was triangulated with 4500 elements.

Figure 5 illustrates the upstream still water retained by the dam initially and the inundation of the floodplain at different times after the dam-break has occurred. It can be seen to have the expected physical behavior. A flood wave led by a wet/dry front starts to inundate the floodplain immediately after the dam collapse. By time $t = 2$ s, the front has reached the pair of small humps and begins to rise over them. Curved bore-like reflection waves are created at the small humps and start to propagate upstream, whereas part of the wet/dry front continues downstream between the small humps. At time $t = 6$ s, the small humps are entirely submerged, and the wet/dry front has reached the large hump where part of the front runs up the large hump more than halfway, whereas water cascades around the side of the hump. The curved reflection bores continue to move upstream and begin to interact with each other and the side walls of the channel. At time $t = 12$ s, the water that is passing either side of the large hump starts to flood the lee side of the large hump. The flood wave has reflected against the large hump, releasing an upstream directed bore, whereas the previously upstream directed shock waves from the small humps have coalesced into a nearly straight bore. The maximum water level at the large hump reduces as the reflection bore moves away. After a continuous wave–wave, wave–topography and wave–wall interactions, and energy dissipation effect due to bottom friction, the flow starts to reach steady state. By $t = 300$ s,

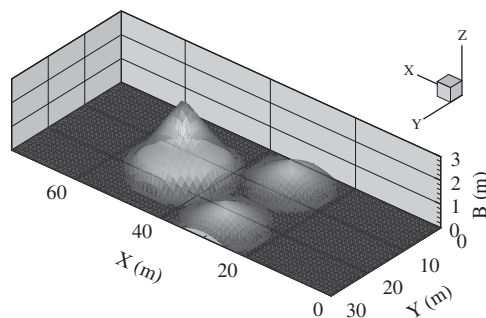


Figure 4. Topography used for dam-break flooding with three humps on grid of triangular cells.

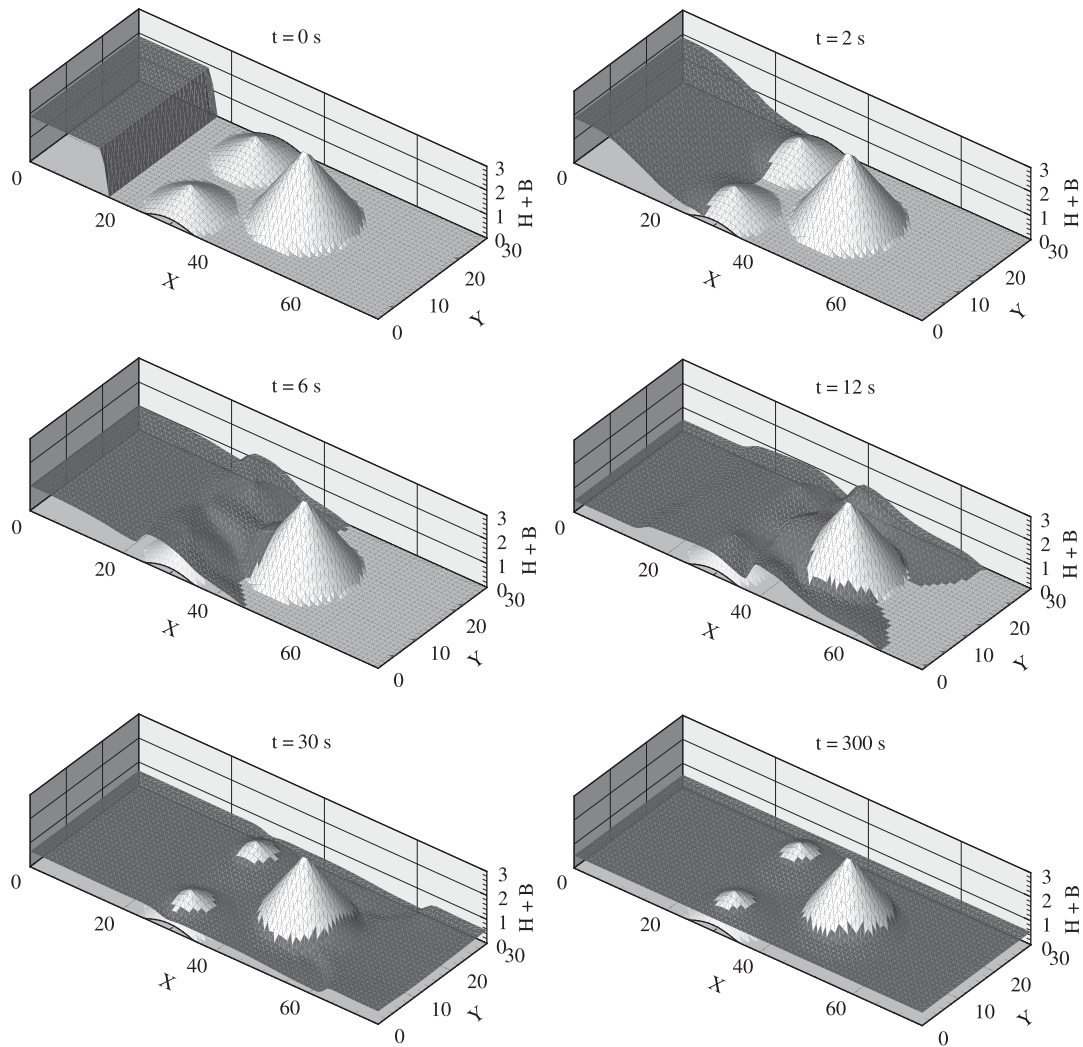


Figure 5. Dam-break flooding over three humps in a closed channel: 3D water surface evolution at times $t=0, 2, 6, 12, 30, 300$ s.

steady state is achieved, with the peaks of the small humps no longer submerged. The numerical model properly simulates the complicated wetting–drying processes and fine details of the flow, and produces results that are very similar to others presented in the literature [1, 11, 24, 25].

To verify whether the present model preserves stationary solutions involving a partially wetted domain, the model was tested for a case by specifying the initial conditions of $\eta=1.875$ m, $u=v=0$. Figure 6 shows the maximum value of the computed resultant velocity at some time. As shown in Figure 6, the maximum value of the computed resultant velocity increases with time, and the magnitude of resultant velocity was computed to be roughly 10^{-12} m/s after 10^4 s.

4.2. Dam-break flow in a converging–diverging open channel with initially dry bed

Bellos *et al.* [18] performed dam-break experiments in a converging–diverging channel on various flow conditions, and therefore comparison with a real two-dimensional flow is possible. To validate the numerical model for real floods with dry bed and irregular flow domain, the initially dry bed case of Bellos *et al.* [18] is considered here. This particular case has been employed to test other numerical models [6, 26, 27]. The channel has a rectangular cross-section. The total length is 21.2 m and the bottom slope is 0.002. As shown in Figure 7, the width varies from 0.6 to 1.4 m,

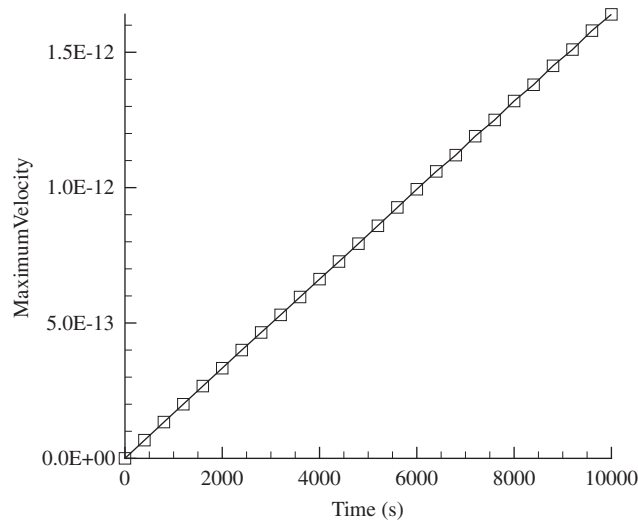


Figure 6. Stationary state preserving: the maximum value of the computed resultant velocity at different time.

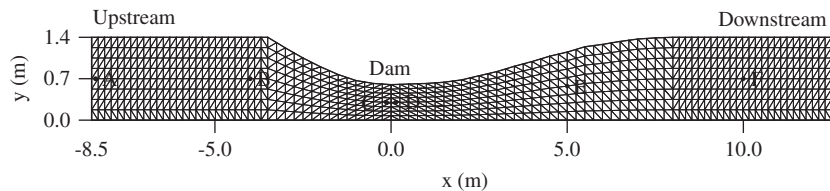


Figure 7. Computational grid and plan view channel layout for dam-break experiment of Bellos *et al.* [18]. Probe A is at $x = -8.5$ m, Probe B is at $x = -4.0$ m, Probe C is at $x = -0.0$ m, Probe D is at $x = +0.0$ m, Probe E is at $x = +5.0$ m, and Probe F is at $x = +10.0$ m.

and the channel contains a curved constriction, beginning at 5.0 m from the upstream end and terminating at 4.7 m from the downstream end. A solid slip condition is imposed at the upstream end and at side the walls. At the downstream end, a free outgoing condition is applied. A movable gate located 8.5 m from the upstream end initially separated the upstream and downstream regions and was then instantaneously removed at time $t = 0$ to simulate dam break. At the gate location, the width is 0.6 m and the bed elevation is 0.15 m. The upstream water surface elevation was 0.3 m and the downstream water depth was 0.0 m. The Manning coefficient is assumed as 0.012 and the wall frictions are neglected. The flow domain was triangulated with 1584 elements and 900 nodes. Water depth hydrographs during simulation were recorded at six spots along the centerline of the channel, i.e. $x = -8.5, -4.0, -0.0, +0.0, 5.0, 10.0$ m. The total simulation time is 70 s after the dam break.

Figure 8(a–f) shows the measured and simulated water depths at the six positions shown in Figure 7. In Figure 8, simulated depth values above the dam match the measured values well. In the downstream of the dam, the arrival time of the flood front is accurately predicted. Simulated values below the dam do not match the measured values as well as above the dam because the important vertical velocity information in the transition from sub-critical to super-critical flow is not reproduced by the present model as a result of depth-averaging. From a practical view, the simulated values and the measured values agree well and it can be concluded that the present model can be reliably applied on two-dimensional complicated dam-break problems.

4.3. Thacker's long wave resonance in a parabolic basin

Thacker [28] provides an analytical solution corresponding to free surface oscillation in a parabolic basin. The motion is oscillatory with small enough amplitude, imposed by the long wave

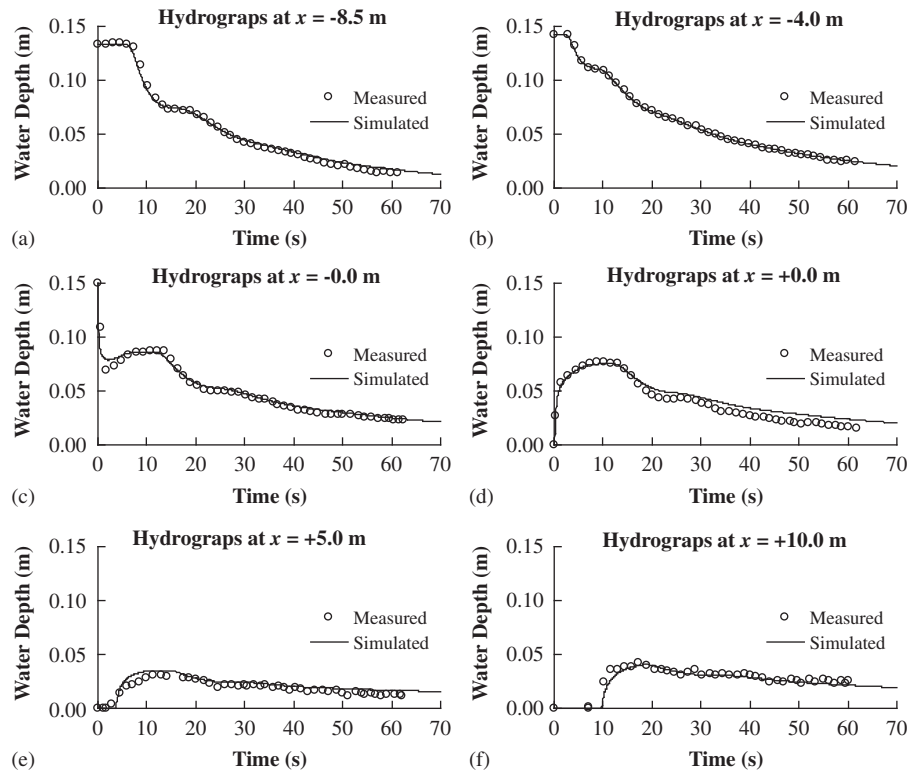


Figure 8. Comparisons of measured and simulated hydrographs: (a) measured and simulated hydrographs at $x = -8.5$ m; (b) measured and simulated hydrographs at $x = -4.0$ m; (c) measured and simulated hydrographs at $x = -0.0$ m; (d) measured and simulated hydrographs at $x = +0.0$ m; (e) measured and simulated hydrographs at $x = +5.0$ m; and (f) measured and simulated hydrographs at $x = +10.0$ m.

assumption, and there is no energy dissipation. For this case, the bed topography is defined as

$$b(x, y) = h_0 \left(\frac{x^2 + y^2}{a^2} - 1 \right) \quad (55)$$

on the computational domain $[-2\text{ m}, 2\text{ m}] \times [-2\text{ m}, 2\text{ m}]$, where h_0 is the depth of water at the centre point for a zero elevation and a is the radial distance from the centre point to the zero elevation of the shoreline. The analytical solution provided in [28] is given by

$$\eta(x, y, t) = \max \left[b(x, y), h_0 \left(\frac{\sqrt{1 - A^2}}{1 - A \cos(\omega t)} - \frac{x^2 + y^2}{a^2} \left(\frac{1 - A^2}{(1 - A \cos(\omega t))^2} - 1 \right) - 1 \right) \right] \quad (56)$$

$$u = \frac{1}{2} \frac{\omega x A \sin(\omega t)}{1 - A \cos(\omega t)} \quad v = \frac{1}{2} \frac{\omega y A \sin(\omega t)}{1 - A \cos(\omega t)} \quad (57)$$

where the frequency ω is given by $\omega = 2\pi/T = \sqrt{8gh_0}/a$, and $A = (a^4 - r_0^4)/(a^4 + r_0^4)$, r_0 is the radial distance from the centre point to the point where the shoreline is initially located. This is a valuable analytical solution for the verification of two-dimensional shallow water models with wetting and drying.

The flow domain was triangulated with 40 000 elements and 20 201 nodes. The analytical solution at time $t = 0$ is supplied as the initial condition. The parameters used for this numerical test are $a = 1$, $r_0 = 0.8$ m, and $h_0 = 0.1$ m. Figure 9 shows the comparisons between numerical and analytical results at four different times. The computed water surface elevation profiles are in excellent agreement with the analytical solution at $t = T/6$, $T/3$, $T/2$, whereas the computed

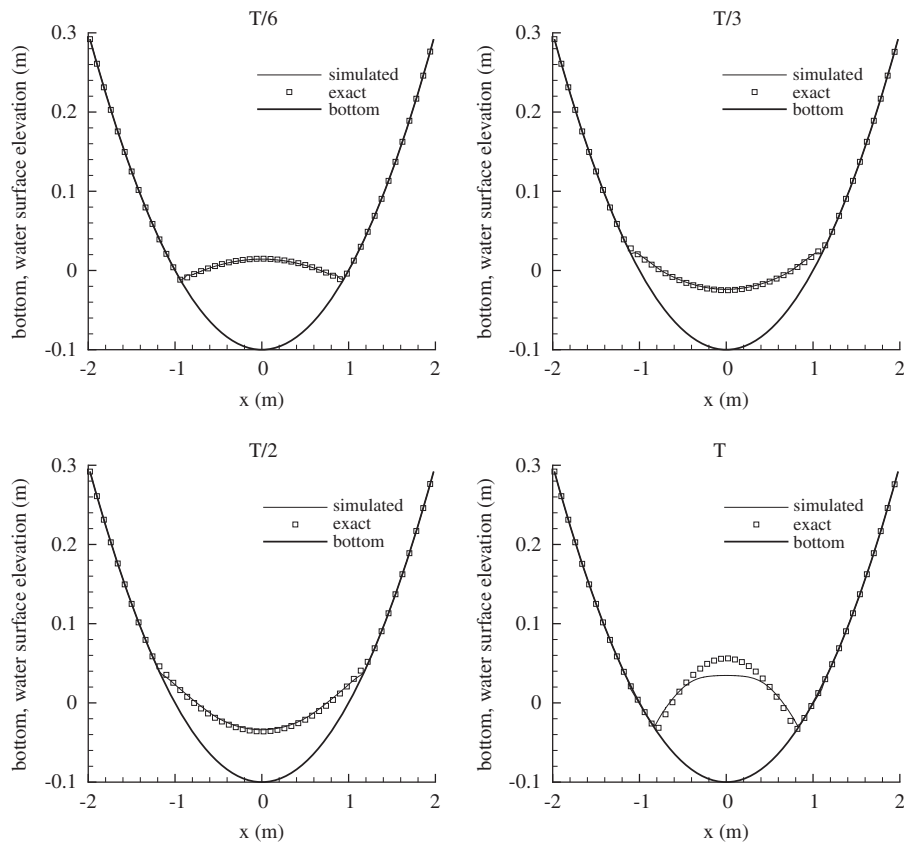


Figure 9. Thackers long wave resonance problem: comparisons of simulated and analytical results at times $t = T/6$, $T/3$, $T/2$ and T .

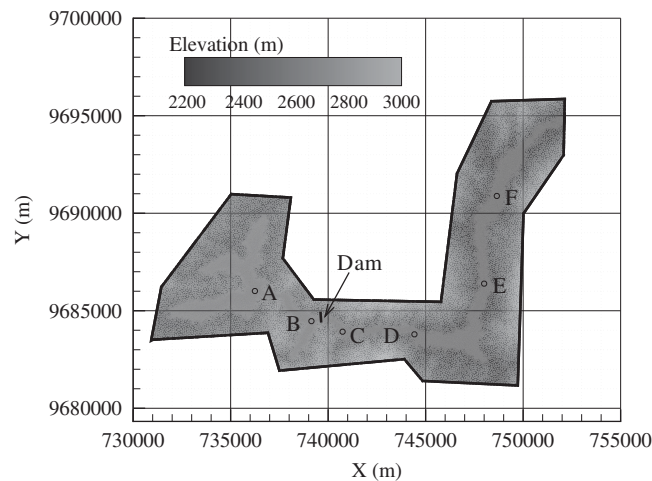


Figure 10. Topography and the locations of surveyed points.

amplitude is slightly less than the analytical solution at time $t = T$. We also found that the computed solution deviates from the analytical solution as time increases. This is possibly due to numerical damping in the scheme and the step-like representation of the bed geometry.

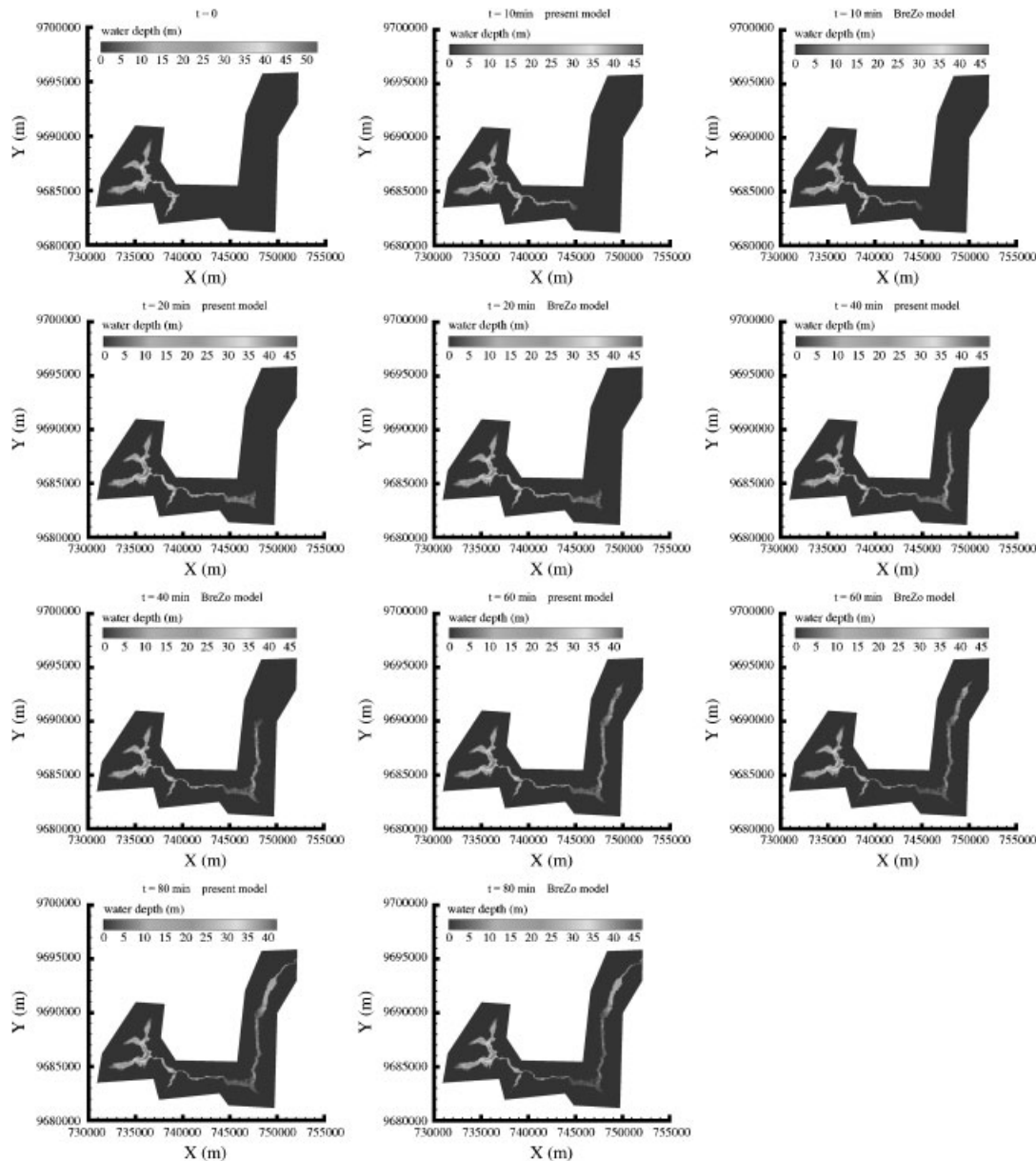


Figure 11. Filled contours plot of initial and computed water depth given by the present model and Brezo [29] at time $t = 10, 20, 40, 60$ and 80 min.

4.4. Application to dam-break flood over complex topography

The present model is applied to simulate a more practical dam-break flooding. In March 1993, the failure of a mountainside created a dam which blocked the flow of the Paute River in southern highland Ecuador. The dam subsequently breached causing a dam-break-type flood, and at least 35 casualties were reported officially. Topography and mesh data about the computation domain are available from BreZo [29]. Triangles are more densely organized along the river. The problem domain was triangulated with 74 224 elements and 37 406 nodes. As shown in Figure 10, the bottom elevations of the valley range from 2155 to 2972 m above base level which is equal to zero. The dam is considered as a straight line between the points of (x, y) coordinates being (739 602 m, 9 684 690 m) and (739 616 m, 9 684 530 m), and initially separated upstream and downstream regions. An instantaneous and complete dam failure at time $t = 0$ is assumed. The initial

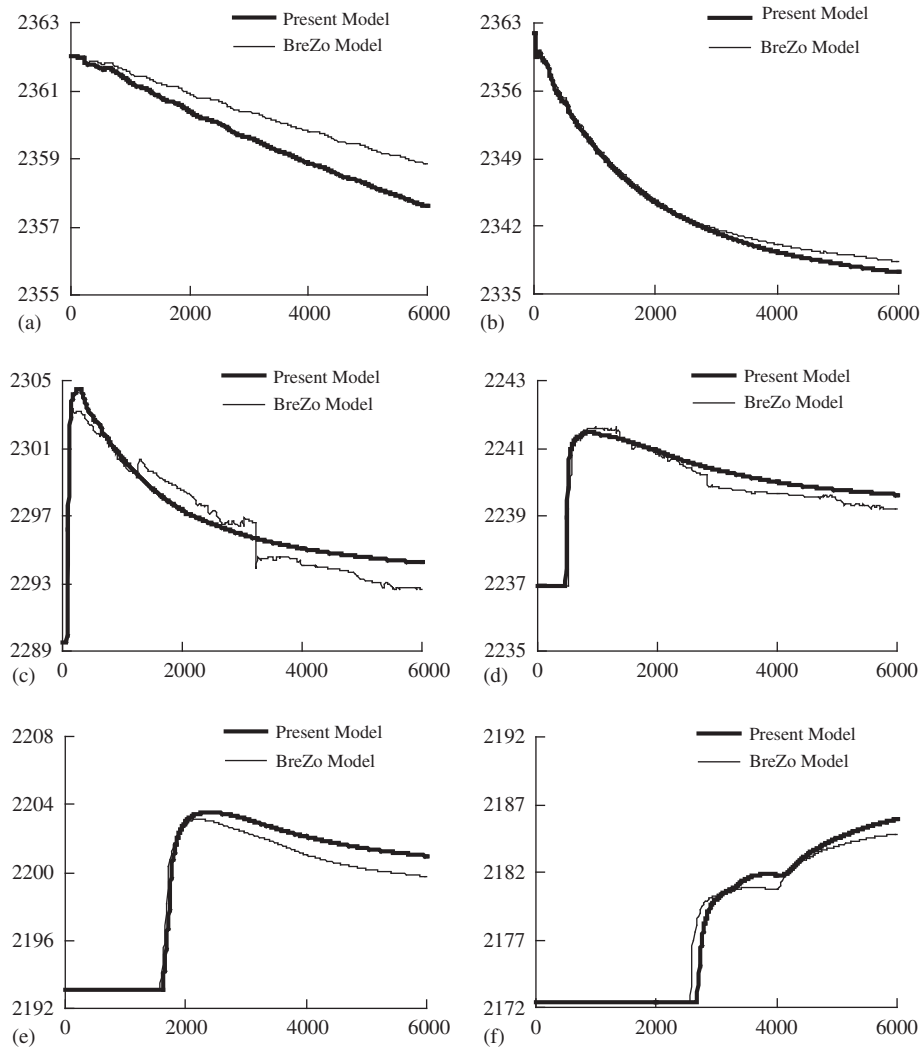


Figure 12. The simulated hydrographs comparison between the present model and BreZo at surveyed points A–F: (a) the simulated hydrographs at surveyed points A; (b) the simulated hydrographs at surveyed points B; (c) the simulated hydrographs at surveyed points C; (d) the simulated hydrographs at surveyed points D; (e) the simulated hydrographs at surveyed points E; and (f) the simulated hydrographs at surveyed points F.

upstream water level is 2,362 m above the base level, and the bed is set to dry in the downstream of the dam. A free outgoing condition is imposed along all boundaries and the uniformly distributed Manning roughness coefficient is set to 0.033.

The computation was run until the time $t = 6000$ s. The present model was run on a PC with a Core 2 2.4 GHz processor and the run time of the test case was about 7 h 40 min. The run time cost is considered not expensive given that the number of elements is over 70 000. The BreZo model was also applied to this problem for comparison. A fixed time step of 0.2 s and LTS level of 1, 2, and 3 were selected. The run time of BreZo is much shorter than the present model and it was about 36, 21, and 17 min when the LTS level was set to 1, 2, and 3, respectively. The major reason for long run time of the present model might be that the Java code is inefficient, compared with the FORTRAN code. From our experience, the running efficiency of Java code could be lower than the FORTRAN code by one order of magnitude. Besides, the present model using two-stage explicit Runge–Kutta method that must solve the Riemann problem twice in a step, while once for Hancock predictor–correction method used by BreZo. Figure 11 presents the initial and computed

water depths given by the present model and BreZo model ($LTS = 1$) at time $t = 0, 10, 20, 40, 60, 80$ min. The computed results show good consistency in flood wave propagation and inundation area. Figure 12 shows the simulated hydrographs given by the present model and BreZo ($LTS = 1$) at surveyed points A–F shown in Figure 10. The speeds of water surface decreasing at upstream surveyed points A and B are higher than BreZo results. Besides, higher peaks were produced by the present model at downstream surveyed points. The flooding arrival times are close to BreZo results. During the simulation, all velocities were monitored, and we found that unphysical high velocities (bigger than 30 m/s) were not predicted.

5. CONCLUSIONS

A numerical model based on the finite volume method has been developed and tested to simulate dam-break floods over a wet or dry bed with complicated topography. The advantages of the present model are the capabilities of handling complex geometries using unstructured grids, representing the terrain with second-order accuracy assigning the bed elevation to vertices, treating wet/dry fronts by VFRs over arbitrary topography, exactly preserving steady states by adding flux correction, and solving discontinuous solution stably by approximate Riemann solver.

The test case of dam-break flood inundation over three humps shows that well-balanced properties of the present model are preserved even with wet/dry fronts over complex topography. In the dam-break test of Bellos *et al.* [18], the computed depths show good agreement with experimental data, and the arrival time of the flood front is accurately predicted. In the test case of Thacker's long wave resonance, the computed water surface elevation profiles are in excellent agreement with the analytical solution, but deviates from the analytical solution as time increases. In the case of practical application to dam-break flooding over natural topography, the stability and suitability of the model are verified. This proves the applicability of the present model to realistic dam-break flood inundation cases.

ACKNOWLEDGEMENTS

This work was supported by a grant from the National Basic Research Program of China (Project No. 2007CB714107), a grant from the Key Projects in the National Science and Technology Pillar Program (Project No. 2008BAB29B08), and a grant from the Special Research Foundation for the Public Welfare Industry of the Ministry of Science and Technology and the Ministry of Water Resources (Project No. 200701008). The authors are grateful to Bellos *et al.* for their experimental data and Sanders and Begnudelli for their topography and mesh data about Paute River.

REFERENCES

1. Liang Q, Borthwick AGL. Adaptive quadtree simulation of shallow flows with wet–dry fronts over complex topography. *Computers and Fluids* 2009; **38**(2):221–234.
2. Begnudelli L, Sanders BF. Unstructured grid finite-volume algorithm for shallow-water flow and scalar transport with wetting and drying. *Journal of Hydraulic Engineering* (ASCE) 2006; **132**(4):371–384.
3. Zia A, Banihashemi MA. Simple efficient algorithm (SEA) for shallow flows with shock wave on dry and irregular beds. *International Journal for Numerical Methods in Fluids* 2008; **56**(11):2021–2043.
4. Begnudelli L, Sanders BF. Conservative wetting and drying methodology for quadrilateral grid finite-volume models. *Journal of Hydraulic Engineering* (ASCE) 2007; **133**(3):312–322.
5. Kuiry SN, Pramanik K, Sen D. Finite volume model for shallow water equations with improved treatment of source terms. *Journal of Hydraulic Engineering* (ASCE) 2008; **134**(2):231–242.
6. Yoon TH, Kang SK. Finite volume model for two-dimensional shallow water flows on unstructured grids. *Journal of Hydraulic Engineering* (ASCE) 2004; **130**(7):678–688.
7. Brufau P, Garcia-Navarro P. Two-dimensional dam break flow simulation. *International Journal for Numerical Methods in Fluids* 2000; **33**(1):35–57.
8. Liang Q, Borthwick AGL, Stelling G. Simulation of dam- and dyke-break hydrodynamics on dynamically adaptive quadtree grids. *International Journal for Numerical Methods in Fluids* 2004; **46**(2):127–162.
9. Abderrezzak KEK, Paquier A, Mignot E. Modelling flash flood propagation in urban areas using a two-dimensional numerical model. *Nature Hazards* 2009; **50**(3):433–460.

10. Valiani A, Caleffi V, Zanni A. Case study: Malpasset dam-break simulation using a two-dimensional finite volume method. *Journal of Hydraulic Engineering* (ASCE) 2002; **128**(5):460–472.
11. Nikolos IK, Delis AI. An unstructured node-centered finite volume scheme for shallow water flows with wet/dry fronts over complex topography. *Computer Methods in Applied Mechanics and Engineering* 2009; **198**(47–48): 3723–3750.
12. Xia J, Lin B, Falconer RA, Wang G. Modelling dam-break flows over mobile beds using a 2D coupled approach. *Advances in Water Resources* 2010; **33**(2):171–183.
13. Lai JS, Guo WD, Lin GF, Tan YC. A well-balanced upstream flux-splitting finite-volume scheme for shallow-water flow simulations with irregular bed topography. *International Journal for Numerical Methods in Fluids* 2010; **62**(8):927–944.
14. LeVeque RJ. *Finite Volume Methods for Hyperbolic Problems*. Cambridge University Press: Cambridge, 2002. ISBN: 0-521-81087-6.
15. Toro EF. *Shock-capturing Methods for Free-surface Shallow Flows*. Wiley: Chichester, 2001. ISBN: 0-471-98766-2.
16. Zhou JG, Causon DM, Mingham CG, Ingram DM. The surface gradient method for the treatment of source terms in the shallow-water equations. *Journal of Computational Physics* 2001; **168**(1):1–25.
17. George DL. Augmented Riemann solvers for the shallow water equations over variable topography with steady states and inundation. *Journal of Computational Physics* 2008; **227**(6):3089–3113.
18. Bellos CV, Soulis JV, Sakkas JG. Experimental investigation of two-dimensional dam-break induced flows. *Journal of Hydraulic Research* 1992; **30**(1):47–63.
19. Shewchuk JR. *Triangle: Engineering a 2D Quality Mesh Generator and Delaunay Triangulator*. Lecture Notes in Computer Science, vol. 1148: 1996; 203–222. Available from: <http://www.cs.cmu.edu/quake/triangle.html>.
20. Barth TJ, Jespersen DC. The design and application of upwind schemes on unstructured meshes. *AIAA Paper*, 890366, 1989.
21. Li QY, Wang NC, Yi DY. *Numerical Analysis*. Tsinghua University Press: Beijing, 2008; ISBN: 7-302-18565-4.
22. Begnudelli L, Sanders BF, Bradford SF. Adaptive Godunov-based model for flood simulation. *Journal of Hydraulic Engineering* (ASCE) 2008; **134**(6):714–725.
23. Kawahara M, Umetsu T. Finite element method for moving boundary problems in river flow. *International Journal for Numerical Methods in Fluids* 1986; **6**(6):365–386.
24. Brufau P, Vazquez-Cendon ME, Garcia-Navarro P. A numerical model for the flooding and drying of irregular domains. *International Journal for Numerical Methods in Fluids* 2002; **39**(3):247–275.
25. Delis AL, Kazolea M, Kampains NA. A robust high-resolution finite volume scheme for the simulation of long waves over complex domains. *International Journal for Numerical Methods in Fluids* 2008; **56**(4):419–452.
26. Hsu CT, Yeh KC, Yang JC. Depth-averaged two-dimensional curvilinear explicit finite analytic model for open-channel flow. *International Journal for Numerical Methods in Fluids* 2000; **33**(2):175–202.
27. Martin N, Gorelick SM. MOD_FreeSurf2D: a MATLAB surface fluid flow model for rivers and streams. *Computers and Geosciences* 2005; **31**(7):929–946.
28. Thacker WC. Some exact solutions to the nonlinear shallow-water wave equations. *Journal of Fluid Mechanics* 1981; **107**:499–508.
29. Sanders BF, Begnudelli L. BREZO: a hydrodynamic flood simulation algorithm. Available from: <http://sanders.eng.uci.edu/brezo.html>.