

## Dynamically adaptive tree grid modeling for simulation and visualization of rainwater overland flow

Anurak Busaman<sup>1,2,\*†</sup>, Khamron Mekchay<sup>1</sup>, Suchada Siripant<sup>1</sup> and Somporn Chuai-Aree<sup>2</sup>

<sup>1</sup>*Department of Mathematics and Computer Science, Faculty of Science, Chulalongkorn University, Bangkok, Thailand*

<sup>2</sup>*Department of Mathematics and Computer Science, Faculty of Science and Technology, Prince of Songkla University, Muang, Pattani, Thailand*

### SUMMARY

Modeling and visualization of a rainwater overland flow is an important tool for a risk assessment, preparation, evacuation planning, and real-time forecasting of flood warning. The objective of this research is to develop a numerical software to visualize the rainwater overland flow based on a finite volume method for shallow water equations and in combination with the dynamically adaptive tree grid technique and the dynamic domain-defining method. The obtained simulations for several experiments were tested and compared with results in literature, both theoretical and experimental results. The comparisons with non-adaptive grids show that the developed algorithm for simulation is very efficient and has a potential for practical usages, in terms of computational times and accuracy. Copyright © 2015 John Wiley & Sons, Ltd.

Received 16 October 2014; Revised 1 June 2015; Accepted 6 June 2015

**KEY WORDS:** hydrostatic reconstruction; rainwater overland flow; well-balanced method; adaptive grids; dynamic domain-defining method

### 1. INTRODUCTION

Water flooding has very often occurred in the world. They caused many related problems such as flooding, landslide, disease, and other problems. The situations have mostly occurred by continuously heavy rain, and it damaged humans and economies. The problems lead to numerical algorithm to visualize a rainwater overland flow, which is an important tool for a risk assessment, preparation, evacuation planning, and real-time forecasting of flood warning. Moreover, the simulation could also be used to analyze water management, because it can reproduce information on water depth, water speed, and flow directions in any regions.

In general, simulation and visualization of a rainwater overland flow model involve numerical methods with high computational demand for solving shallow water equations. The shallow water model has been first introduced in one-dimensional by Saint-Venant [1], and Zhang and Cundy [2] are the first to use shallow water equations for rainwater overland flow equations. Fiedler and Ramirez [3] are the first to use finite volume methods (FVMs), and Delestre and team [4, 5] are the first to use well-balanced schemes for rainwater overland flow modeling. The other related applications for hyperbolic problems are dam break [6, 7], tsunami [8, 9], river flow [10], and tides [11].

The well-balanced property, where the flux gradients are balanced with the source terms for steady-state solutions, is very important for numerical schemes to avoid oscillated steady-state

\*Correspondence to: Anurak Busaman, Department of Mathematics and Computer Science, Faculty of Science, Chulalongkorn University, Bangkok 10330, Thailand.

†E-mail: anurak-b@bunga.pn.psu.ac.th

solution. Bermudez and Vazquez [12] are the first to mention the problem of balance between the pressure term and the topography source term, and Greenberg and LeRoux [13] are the first that developed well-balanced schemes to avoid this problem.

Because topography has an important influence on a rainwater overland flow model, using of FVM requires a large amount of digital terrain elevation data grid cells to improve accuracy of a simulation, and in practice, the computational cost is very expensive. This needs attention for designing algorithms and schemes for reducing the number of computed cells without losing much accuracy in the results. One solution is to use adaptive grid methods where the methods continuously adjust grid resolutions to follow features in the flow. By adaptive techniques, Liang [7] solved two-dimensional (2D) shallow water equations with dynamically adaptive quad tree grids using FVM for dam and dyke break simulations, Popinet [9] developed a quad tree adaptive solver for tsunami modeling with 2D shallow water equations using a generalized well-balanced positivity-preserving scheme of Audusse *et al.* [14], and An and Yu [15] used hydrostatic reconstruction and quad tree grid to model urban flooding. George [6] solved 2D shallow water equations with well-balanced high-resolution FVMs and block-structured dynamic adaptive mesh refinement for modeling floods in rugged terrain and applied to the Malpasset dam-break flood (France, 1959). Moreover, Schreiber, Bungartz, and Bader [16] presented a method for solving hyperbolic partial differential equations based on dynamically adaptive triangular grid.

An alternative technique is the dynamic domain-defining method (dynamic DDM) described by Yamaguchi *et al.* [17] for the development of GIS-based flood simulation software for flood-risk assessment, where during the simulation, the calculation area is adjusted (expanded or shrunk) to exclude dry grid cells. However, computational times for the dynamic DDM can still continuously increase, if the calculation area is continuously expanded. This needs attention to utilize an adaptive grid method with the dynamic DDM; an adaptive grid method can reduce the number of grid cells by merging the grid cells in some calculation area, while the dynamic DDM can automatically define the calculation area to exclude unnecessary cells.

In this research, we develop a numerical algorithm and software to simulate and visualize rainwater overland flow models on natural topography based on FVMs for shallow water equations by modifying the adaptive grid technique, developed in [7, 9, 15] for rectangular quad tree grids, and combining with the dynamic DDM. Here, the idea of quad tree grids is extended for general tree grids to handle natural topography data in general.

The rest of the paper is organized as follows. Section 2 describes the designing of the data structure for general rectangular tree grids, while Section 3 presents the rainwater overland flow model. The numerical algorithm of FVM based on the adaptive grids and dynamic DDM are explained in Section 4, and the numerical simulations and results are shown in Section 5.

## 2. TREE GRIDS

The data structure for tree grids is designed for adaptivity based on rectangular quad tree grids appeared in [7, 9, 15], where each parent cell has four children and the levels of adjacent cells cannot differ more than one. In order to handle general natural topography data, we design in the algorithm that parent cells can have any number of children cells with square dimension and the levels of adjacent cells can differ more than one. In addition, all parent cells in the same tree level must have the same number of children. For example, Figure 1 shows a regularized grid with four subdivision levels having one big cell in level 1, and a number of children for parent cells in levels 1, 2, and 3 are  $3 \times 3$ ,  $2 \times 2$ , and  $3 \times 3$ , respectively, while level 4 cells have no child. The corresponding hierarchical tree structure is illustrated in Figure 2.

## 3. EQUATIONS

A process of a rainwater overland flow in general is very complex. The process occurs when the rainfall intensity exceeds soil infiltration, and water begins to accumulate on the surface and then flows downslope under gravity as overland flow. In order to reduce the complexity process of simulation, we consider only a model that consists of rainfall intensity, where the infiltration is very

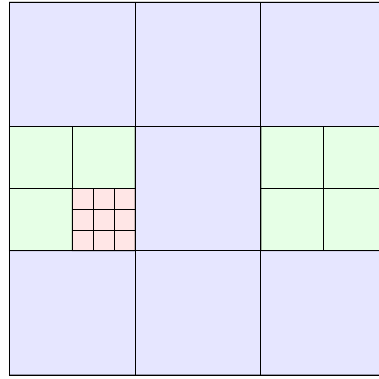


Figure 1. Regularized tree grids.

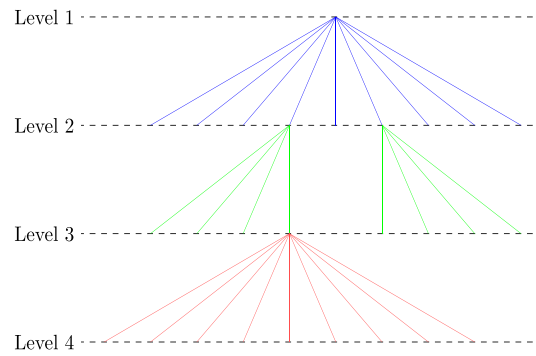


Figure 2. Hierarchical data structure.

little compared with the volume of the flow and can be ignored in addition to other processes such as saturation excess, erosion, evaporation, and subsurface flow. The 2D model is developed based on shallow water equations for determining the behavior of rainwater flow above bottom elevation without infiltration. The model system is presented in vector form as follows:

$$\frac{\partial \vec{w}}{\partial t} + \frac{\partial \vec{f}(\vec{w})}{\partial x} + \frac{\partial \vec{g}(\vec{w})}{\partial y} = \vec{z}(\vec{w}) + \vec{s}(\vec{w}), \quad (1)$$

where the vector  $\vec{w} = [h, uh, vh]^T$  of dependent variables consisting of the water depth  $h$ , the discharges per unit width  $uh$  and  $vh$  with velocity components  $u$  and  $v$  in the  $x$  and  $y$  directions, while

$$\vec{f}(\vec{w}) = \begin{bmatrix} uh \\ u^2h + g\frac{h^2}{2} \\ uvh \end{bmatrix} \text{ and } \vec{g}(\vec{w}) = \begin{bmatrix} vh \\ uvh \\ v^2h + g\frac{h^2}{2} \end{bmatrix} \quad (2)$$

are flux vectors of the system in the direction of the  $x$ -axis and  $y$ -axis, respectively. The right-hand side of the system (1) represents the gravity force vector

$$\vec{z}(\vec{w}) = \begin{bmatrix} 0 \\ -ghS_{0x} \\ -ghS_{0y} \end{bmatrix}, \quad (3)$$

comprises the bottom slopes  $S_{0x} = \frac{\partial z}{\partial x}$  and  $S_{0y} = \frac{\partial z}{\partial y}$ , where  $z$  is height of topography, and the vector of source term

$$\vec{s}(\vec{w}) = \begin{bmatrix} q \\ -S_{fx} \\ -S_{fy} \end{bmatrix}, \quad (4)$$

where  $q$  is the rainfall rate that is vertically added to control volume, and

$$S_{fx} = \frac{gn^2\sqrt{u^2+v^2}uh}{h^{4/3}} \quad \text{and} \quad S_{fy} = \frac{gn^2\sqrt{u^2+v^2}vh}{h^{4/3}} \quad (5)$$

are the friction slopes representing the effect of bed roughness and  $n$  is the Manning's resistance coefficient. In these equations,  $g$  is the gravity constant, and  $t$  is time.

#### 4. NUMERICAL SCHEME

We present in this section the developed numerical algorithm for the system (1) using a conservative FVM based on well-balanced schemes with hydrostatic reconstruction proposed by Audusse *et al.* [14] and Delestre *et al.* [4] to simulate and visualize the rainwater flows over non-flat bottom on adaptive grids. We utilize the scheme so that the system preserves the nonnegative of the water depth (negative water depths are physically incorrect and undesirable in numeric) and conserves the total water depth, including the ability to compute dry states. Furthermore, the scheme also satisfies the natural properties for still water, where  $(u, v) = 0$  and  $h + z = \text{constant}$  is the solution of the system, reflecting the fact that a lake at rest remains so, and satisfies the discrete entropy inequality to guarantee that the total energy behaves physically correct [14].

However, the hydrostatic reconstruction can fail when the topography slope becomes too big compared with the water height as demonstrated in [18], but most of the time in those cases, the shallow water equations are no longer valid. Some improvements have been performed in order to overcome those limits in [19, 20]. Incorporation of these improvements is beyond the adopted scheme of this research. The details of the numerical scheme and algorithm of this research are described as follows.

##### 4.1. Finite volume formulation

The finite volume formulation was performed on physical domain that is made of regularized cells. Each cell is considered as a main control volume whose boundary is formed by the four direct walls surrounding it. An example of a single main control volume is shown in Figure 3.

By integrating Equation (1) over an arbitrary cell domain  $\Omega$  and using the Green's theorem, we have

$$\int_{\Omega} \frac{\partial \vec{w}}{\partial t} d\Omega + \oint_{\tau} \vec{F} \cdot \vec{n} d\tau = \int_{\Omega} \vec{z}(\vec{w}) d\Omega + \int_{\Omega} \vec{s}(\vec{w}) d\Omega, \quad (6)$$

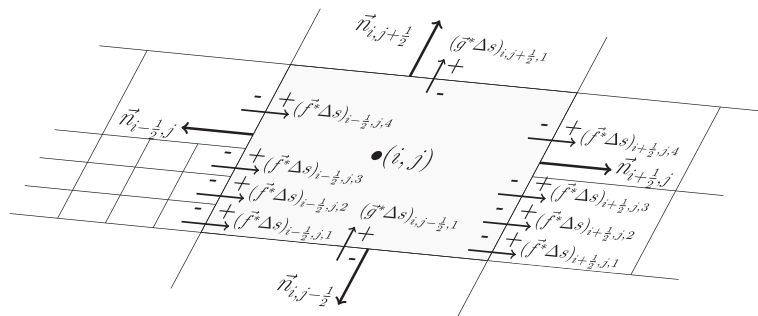


Figure 3. The main control volume and descriptions of numerical fluxes.

where  $\tau$  is the boundary of  $\Omega$ ,  $\vec{n}$  is the unit outward normal vector to the boundary, and  $\vec{F} = (\vec{f}(\vec{w}), \vec{g}(\vec{w}))$  is the vector that consists of the flux vectors in Equation (1) at each interface of the cell boundary.

Dividing Equation (6) with the cell area  $\Delta = \Delta x_{i,j} \times \Delta y_{i,j}$  and denoting  $\vec{W}_{i,j}$ ,  $\vec{Z}_{i,j}$ , and  $\vec{S}_{i,j}$  as the averages of  $\vec{w}_{i,j}$ ,  $\vec{z}(\vec{w}_{i,j})$ , and  $\vec{s}(\vec{w}_{i,j})$  over the cell, respectively, the finite volume formulation becomes

$$\frac{\partial \vec{W}_{i,j}}{\partial t} = -\frac{1}{\Delta} \oint_{\tau} \vec{F} \cdot \vec{n} d\tau + \vec{Z}_{i,j} + \vec{S}_{i,j}, \quad (7)$$

where the subscripts  $i$  and  $j$  denote the spatial indices in  $x$  and  $y$  directions.

In computation using uniform grids, the line integral in Equation (7) can be discretized by substituting  $\oint_{\tau} \vec{F} \cdot \vec{n} d\tau$  with the sum over the four walls around the control volume as follows:

$$\frac{\partial \vec{W}_{i,j}}{\partial t} = -\frac{1}{\Delta} \left( \vec{f}_{i+\frac{1}{2},j}^* \Delta y_{i,j} - \vec{f}_{i-\frac{1}{2},j}^* \Delta y_{i,j} + \vec{g}_{i,j+\frac{1}{2}}^* \Delta x_{i,j} - \vec{g}_{i,j-\frac{1}{2}}^* \Delta x_{i,j} \right) + \vec{Z}_{i,j} + \vec{S}_{i,j}, \quad (8)$$

where  $\vec{f}_{i-\frac{1}{2},j}^*$ ,  $\vec{f}_{i+\frac{1}{2},j}^*$ ,  $\vec{g}_{i,j-\frac{1}{2}}^*$ , and  $\vec{g}_{i,j+\frac{1}{2}}^*$  are the numerical flux functions, depending upon the chosen schemes. The subscripts  $(i - \frac{1}{2}, j)$ ,  $(i + \frac{1}{2}, j)$ ,  $(i, j - \frac{1}{2})$ , and  $(i, j + \frac{1}{2})$  are indices for the four walls of the boundary  $\tau$ .

However, because our computing model is performed on dynamically adaptive grids considered as non-uniform grids, the numerical flux function for each wall is computed as the sum of numerical fluxes of sub-interfaces generated by a reconstruction of the cell boundary, as illustrated in Figure 3. The reconstruction detail is described in Section 4.5. Equation (8) can be rewritten as follows:

$$\begin{aligned} \frac{\partial \vec{W}_{i,j}}{\partial t} = \frac{1}{\Delta} & \left[ \sum_{k=1}^{m_{i-\frac{1}{2},j}} (\vec{f}^* \Delta s)_{i-\frac{1}{2},j,k} - \sum_{k=1}^{m_{i+\frac{1}{2},j}} (\vec{f}^* \Delta s)_{i+\frac{1}{2},j,k} \right. \\ & \left. + \sum_{k=1}^{m_{i,j-\frac{1}{2}}} (\vec{g}^* \Delta s)_{i,j-\frac{1}{2},k} - \sum_{k=1}^{m_{i,j+\frac{1}{2}}} (\vec{g}^* \Delta s)_{i,j+\frac{1}{2},k} \right] + \vec{Z}_{i,j} + \vec{S}_{i,j}, \end{aligned} \quad (9)$$

where the subscript  $k$  indicates the sub-interfaces,  $\Delta s$  is width of the sub-interface, and  $m_{i,j}$  denotes the numbers of sub-interfaces for boundary walls around the cell.

#### 4.2. Numerical flux function

In the developed algorithm, a computational scheme based on the work of Audusse *et al.* has been employed for numerical fluxes for each sub-interface. The application of this approach to non-uniform grids domain can be expressed by

$$\vec{f}_{i\pm\frac{1}{2},j,k}^* = \vec{f}^{HLL} \left( \vec{U}_{i\pm\frac{1}{2},j,k-}, \vec{U}_{i\pm\frac{1}{2},j,k+} \right) + \vec{L}_{i\pm\frac{1}{2},j,k}^x, \quad (10)$$

$$\vec{g}_{i,j\pm\frac{1}{2},k}^* = \vec{g}^{HLL} \left( \vec{U}_{i,j\pm\frac{1}{2},k-}, \vec{U}_{i,j\pm\frac{1}{2},k+} \right) + \vec{L}_{i,j\pm\frac{1}{2},k}^y, \quad (11)$$

where

$$\vec{L}_{i\pm\frac{1}{2},j,k}^x = \left[ 0, \frac{g}{2} \left( h_{i\pm\frac{1}{2},j,k\mp}^2 - \hat{h}_{i\pm\frac{1}{2},j,k\mp}^2 \right), 0 \right]^T, \quad (12)$$

$$\vec{L}_{i,j\pm\frac{1}{2},k}^y = \left[ 0, 0, \frac{g}{2} \left( h_{i,j\pm\frac{1}{2},k\mp}^2 - \hat{h}_{i,j\pm\frac{1}{2},k\mp}^2 \right) \right]^T, \quad (13)$$

and  $\vec{f}^{HLL}$  and  $\vec{g}^{HLL}$  are approximated Riemann solvers of shallow water equations without source terms in  $x$  and  $y$  directions, respectively. They are computed with the reconstructed state,  $\vec{U} = [\hat{h}, u\hat{h}, v\hat{h}]^T$ , where  $u$  and  $v$  are the reconstructed values of the velocities in  $x$  and  $y$  directions.  $\hat{h}$  is particular value based on Audusse's scheme for preserving the lake-at-rest condition to guarantee that the water depth is nonnegative. The evaluations of  $\hat{h}$  are given by

$$\hat{h}_{i\pm\frac{1}{2},j,k\mp} = \max\left(0, h_{i\pm\frac{1}{2},j,k\mp} + z_{i\pm\frac{1}{2},j,k\mp} - z_{i\pm\frac{1}{2},j,k}\right), \quad (14)$$

$$\hat{h}_{i,j\pm\frac{1}{2},k\mp} = \max\left(0, h_{i,j\pm\frac{1}{2},k\mp} + z_{i,j\pm\frac{1}{2},k\mp} - z_{i,j\pm\frac{1}{2},k}\right), \quad (15)$$

where the heights of interface cells

$$z_{i\pm\frac{1}{2},j,k} = \max\left(z_{i\pm\frac{1}{2},j,k-}, z_{i\pm\frac{1}{2},j,k+}\right), \quad (16)$$

$$z_{i,j\pm\frac{1}{2},k} = \max\left(z_{i,j\pm\frac{1}{2},k-}, z_{i,j\pm\frac{1}{2},k+}\right), \quad (17)$$

and  $h_{i\pm\frac{1}{2},j,k+}$ ,  $h_{i\pm\frac{1}{2},j,k-}$ ,  $h_{i,j\pm\frac{1}{2},k+}$ , and  $h_{i,j\pm\frac{1}{2},k-}$  are the reconstructed values of water depths at sub-interface  $k$ .

In these equations, the markers  $(-)$  and  $(+)$  of  $k$  are indices for the reconstructed values of the left and right of sub-interface at  $k$  in the  $x$  direction, while the  $y$  direction, the signs  $(-)$  and  $(+)$  indicate the bottom and top reconstructed state at each sub-interface. These are shown in Figure 3.

#### 4.3. Approximated Riemann solver

The Riemann solvers  $\vec{f}^{HLL}$  and  $\vec{g}^{HLL}$  in Equations (10) and (11) are computed using the HLL Riemann solver, which was developed by Harten, Lax, and Van Leer [21] described as follows:

$$\vec{f}^{HLL}(\vec{U}_w, \vec{U}_e) = \frac{\alpha^+ \vec{f}(\vec{U}_w) - \alpha^- \vec{f}(\vec{U}_e) + \alpha^+ \alpha^- (\vec{U}_e - \vec{U}_w)}{\alpha^+ - \alpha^-}, \quad (18)$$

$$\vec{g}^{HLL}(\vec{U}_s, \vec{U}_n) = \frac{\beta^+ \vec{g}(\vec{U}_s) - \beta^- \vec{g}(\vec{U}_n) + \beta^+ \beta^- (\vec{U}_n - \vec{U}_s)}{\beta^+ - \beta^-}, \quad (19)$$

where  $\vec{f}(\vec{U})$  and  $\vec{g}(\vec{U})$  are flux functions of  $\vec{U}$ . The wave speeds  $\alpha^+$ ,  $\alpha^-$ ,  $\beta^+$ , and  $\beta^-$  are assigned based on the work of Kurganov, Noelle, and Petrova [22] as follows:

$$\alpha^\pm = \pm \max\left(0, \pm\left(u_w \pm \sqrt{g\hat{h}_w}\right), \pm\left(u_e \pm \sqrt{g\hat{h}_e}\right)\right), \quad (20)$$

$$\beta^\pm = \pm \max\left(0, \pm\left(v_s \pm \sqrt{g\hat{h}_s}\right), \pm\left(v_n \pm \sqrt{g\hat{h}_n}\right)\right), \quad (21)$$

when  $u - \sqrt{g\hat{h}}$  and  $u + \sqrt{g\hat{h}}$  are the smallest and largest eigenvalues of Jacobian matrix  $\frac{\partial \vec{f}(\vec{U})}{\partial \vec{U}}$ , and for  $\frac{\partial \vec{g}(\vec{U})}{\partial \vec{U}}$  gives  $v - \sqrt{g\hat{h}}$  and  $v + \sqrt{g\hat{h}}$ .

#### 4.4. Gravity force

A simple well-balanced choice for the centered source term of gravity force was presented in Audusse's scheme [14], where the gravity force was calculated by the slope of the reconstructed topography height,  $z$ , with the average of the reconstructed water depth at center point of the cell.

In the application of this approach to non-uniform grids, the gravity force is computed by the average of gravity forces on all sub-plates from the reconstruction. The formulation for the gravity force is given as follows:

$$\vec{Z}_{i,j} = [0, Z_x, Z_y]^T, \quad (22)$$

where

$$Z_x = -\frac{1}{\Delta} g \sum_k^{m^x} \bar{h}_k^x \left( z_{i+\frac{1}{2},j,k-} - z_{i-\frac{1}{2},j,k+} \right) \Delta s_k^x, \quad (23)$$

$$Z_y = -\frac{1}{\Delta} g \sum_k^{m^y} \bar{h}_k^y \left( z_{i,j+\frac{1}{2},k-} - z_{i,j-\frac{1}{2},k+} \right) \Delta s_k^y, \quad (24)$$

and  $\bar{h}_k^x = (h_{i-\frac{1}{2},j,k+} + h_{i+\frac{1}{2},j,k-})/2$  and  $\bar{h}_k^y = (h_{i,j-\frac{1}{2},k+} + h_{i,j+\frac{1}{2},k-})/2$  are the averages of the reconstructed water depths at center point in the  $x$  and  $y$  directions, respectively, while  $\Delta s_k^x = \Delta s_{i-\frac{1}{2},j,k+} = \Delta s_{i+\frac{1}{2},j,k-}$  and  $\Delta s_k^y = \Delta s_{i,j-\frac{1}{2},k+} = \Delta s_{i,j+\frac{1}{2},k-}$  are widths of reconstructed interfaces in the two Cartesian directions, and  $m^x$  and  $m^y$  are number of the reconstructed interfaces in the  $x$  and  $y$  directions.

#### 4.5. Reconstruction procedure

In this section, the reconstruction procedure is presented. The common goal is to interpolate the profiles of the various quantities within each cell. This recovers the information about the behavior of the quantities inside cell that is lost and also improves the accuracy of numerical result to have a better estimation for solving the Riemann problem at the interface between two adjacent cells. To ensure stable solutions, the reconstruction is performed using a slope limited extrapolation, presented for 2D non-uniform grids in [8] and [9] as follows:

$$\vec{Q}_{x,y} = \vec{Q}_{i,j} + \sigma_{i,j}(x - x_{i,j}) + \delta_{i,j}(y - y_{i,j}), \quad (25)$$

where  $\sigma$  and  $\delta$  are minmod slope limiters, while  $\vec{Q}$  is the reconstructed variable vector,  $(x, y)$  is the interpolated position, and  $(x_{i,j}, y_{i,j})$  is the position at center point of the cell  $(i, j)$ . In this paper, the formulation can be written as follows:

$$\vec{Q}_{i\pm\frac{1}{2},j,k\mp} = \vec{Q}_{i,j} + \sigma_{i,j,k} \left( x_{i\pm\frac{1}{2},j,k} - x_{i,j} \right) + \delta_{i,j,k} \left( y_{i\pm\frac{1}{2},j,k} - y_{i,j} \right), \quad (26)$$

$$\vec{Q}_{i,j\pm\frac{1}{2},k\mp} = \vec{Q}_{i,j} + \sigma_{i,j,k} \left( x_{i,j\pm\frac{1}{2},k} - x_{i,j} \right) + \delta_{i,j,k} \left( y_{i,j\pm\frac{1}{2},k} - y_{i,j} \right), \quad (27)$$

where  $(x_{i\pm\frac{1}{2},j,k}, y_{i\pm\frac{1}{2},j,k})$  and  $(x_{i,j\pm\frac{1}{2},k}, y_{i,j\pm\frac{1}{2},k})$  are positions of walls at sub-interfaces  $k$  in the  $x$  and  $y$  directions. In these equations, the minmod slope limiters  $\sigma$  and  $\delta$  are defined by

$$\sigma_{i,j,k} = MM \left( \frac{\vec{Q}_{i+1,j,k} - \vec{Q}_{i,j}}{\Delta x_{i,j}}, \frac{\vec{Q}_{i,j} - \vec{Q}_{i-1,j,k}}{\Delta x_{i,j}} \right), \quad (28)$$

$$\delta_{i,j,k} = MM \left( \frac{\vec{Q}_{i,j+1,k} - \vec{Q}_{i,j}}{\Delta y_{i,j}}, \frac{\vec{Q}_{i,j} - \vec{Q}_{i,j-1,k}}{\Delta y_{i,j}} \right), \quad (29)$$

where  $\vec{Q} = [h, u, v, h + z]^T$  are the reconstructed variables. Indices  $(i - 1, j, k)$ ,  $(i + 1, j, k)$ ,  $(i, j - 1, k)$ , and  $(i, j + 1, k)$  refer four neighbor cells (the left, right, bottom, and top positions

of the center cell  $(i, j)$  that correspond to the position of the sub-interface  $k$ . Function  $MM$  is minmod slope limiter function, which is defined by

$$MM(a, b) = \frac{\text{sign}(a) + \text{sign}(b)}{2} \min(|a|, |b|), \quad (30)$$

where  $\text{sign}$  returns 1 or  $-1$  according to the sign of the parameters.

The reconstruction of  $z$  is obtained by doing the difference between reconstructions of  $h + z$  and  $h$ . This choice of reconstruction of  $z$  is carried out following Audusse *et al.* to preserve the steady-state and nonnegativity of water depth.

#### 4.6. Total variation diminishing

The concept of total variation diminishing (TVD) methods is to have the solution that is both second-order and higher-order accuracy in time and does not produce spurious oscillations. The second-order TVD Runge–Kutta method (also called Heun's method) is employed to solve the ODE without source term  $\tilde{S}_{i,j}$  in Equation (9). The scheme has iteration steps as follows:

$$\vec{W}^{(1)} = \vec{W}^n + \Delta t_1 \Phi(\vec{W}^n), \quad (31)$$

$$\vec{W}^{(2)} = \vec{W}^{(1)} + \Delta t_2 \Phi(\vec{W}^{(1)}), \quad (32)$$

$$\vec{W}^{n+1} = \frac{\vec{W}^n + \vec{W}^{(2)}}{2}, \quad (33)$$

where the superscripts (1) and (2) are shown as iteration steps, while  $n$  is the time step. Sizes of time steps in these iterations are given by  $\Delta t_1$  and  $\Delta t_2$ . In this scheme,  $\Phi$  is the right-hand side functions in Equation (9) without the source term  $\tilde{S}_{i,j}$ .

#### 4.7. Source term computation

The source term vector  $\vec{S}_{i,j}$  in Equation (9) consists of the rainfall rate and the friction forces. For the rainfall rate, because this one is the water depth that is vertically added to control volume per unit time, we update a new value of  $h_{i,j}^{n+1}$  in Equation (33) by adding  $q_{i,j}^n \Delta t$ , namely

$$h_{i,j}^{n+1} \leftarrow h_{i,j}^{n+1} + q_{i,j}^n \Delta t, \quad (34)$$

where  $q_{i,j}^n$  is rainfall rate average in each cell  $(i, j)$  in the range of time step from  $n$  to  $n + 1$ .

For the friction force, we used the semi-implicit method (proposed in [4]) to ensure a stability criteria and in order to preserve steady states at rest. It updates a new value of  $uh_{i,j}^{n+1}$  in Equation (33) as follows:

$$uh_{i,j}^{n+1} \leftarrow \frac{uh_{i,j}^{n+1}}{1 + \Delta t \frac{g n^2 \sqrt{(u_{i,j}^n)^2 + (v_{i,j}^n)^2}}{(h_{i,j}^{n+1})^{4/3}}}. \quad (35)$$

Similarly,  $vh_{i,j}^{n+1}$  can be computed by a formula in the form of Equation (35) using  $vh_{i,j}^{n+1}$  instead of  $uh_{i,j}^{n+1}$ .



#### 4.8. Stability condition

In order to have stable scheme, the maximum of time step size was limited by the Courant–Friedrichs–Lewy (CFL) condition (was presented by Hagen [23]) as follows:

$$\Delta t_m \leq 0.5 \frac{\Delta A_{\min}}{\lambda_{\max}}, \quad m = 1, 2, \quad (36)$$

where  $\Delta t_m$  are the time step sizes at iteration step  $m = 1, 2$  for Equations (31) and (32) of the second-order TVD Runge–Kutta process, and  $\Delta t$  in Equations (34) and (35) is computed from average value of  $\Delta t_1$  and  $\Delta t_2$ . In the foregoing equation,  $\Delta A_{\min} = \min(\Delta x_{i,j}, \Delta y_{i,j})$  is the minimum dimension for all cells, while  $\lambda_{\max}$  is the maximum absolute value of all the wave speeds in the computational domain, which is the maximum of  $\max(\alpha^+, -\alpha^-, \beta^+, -\beta^-)$  for all the sub-interfaces of all the cells.

#### 4.9. Boundary condition

In this paper, we propose two types of boundary conditions: the open (transmissive) and the closed (solid wall). The open boundary conditions can be defined as follows:

$$\begin{aligned} h_0 &= h_1, & u_0 &= u_1, & v_0 &= v_1, & z_0 &= z_1, \\ h_{m+1} &= h_m, & u_{m+1} &= u_m, & v_{m+1} &= v_m, & z_{m+1} &= z_m. \end{aligned} \quad (37)$$

The closed boundary conditions are imposed as follows:

$$\begin{aligned} h_0 &= 0, & u_0 &= 0, & v_0 &= 0, & z_0 &= z_{large}, \\ h_{m+1} &= 0, & u_{m+1} &= 0, & v_{m+1} &= 0, & z_{m+1} &= z_{large}, \end{aligned} \quad (38)$$

where the subscripts 0 and  $m + 1$  are referred to as fictitious cells outside the computational domain, while  $z_{large}$  is some big constant value for the topography height, for example, we assign  $z_{large} = 9999$ .

#### 4.10. Grid adaptivity method

The accuracy of the numerical simulation of rainwater overland flow can be increased by grid refinement to have smaller cell size on high resolution of the digital terrain data grids. However, computation is very expensive for very fine grids, in terms of time and resources if it is performed for all cells in the computational domain. To overcome this, a grid adaptivity technique is applied so that cells are adjusted according to some conditions.

The refinement criterion is based on two conditions: the gradient of the free surface  $H = h + z$  and the water depth. A grid cell is marked for refinement if the following two conditions hold:

Condition 1: The cell has at least one neighbor cell that

$$|\nabla H| \Delta_l > \epsilon, \quad (39)$$

where  $\nabla H$  is the free surface gradient,  $\Delta_l$  is the distance between the checked cell and the neighbor cell, and  $\epsilon$  is an adjustable parameter. The condition was proposed by Popinet [9] for tsunami simulation, which expected that the error was covered by the discretization of sharp fronts.

Condition 2: The cell or at least one of the neighbor cells has water depth

$$h > \xi, \quad (40)$$

where  $\xi$  is an adjustable parameter. Condition 2 is important for simulation of rainwater flow over natural topography, because it is possible that the cell and its neighbors are dry or have very small depths but satisfy condition 1. In this case, the cell should not be marked for refinement, because it has no water flow.

### Refinement procedure

A marked cell is refined by creating children cells in the next level. The newly assigned dependent value for each children cell is computed based on the slope limited extrapolation. For non-uniform grids, each children dependent value is computed using the average of the minmod slopes of the values of its neighbors, namely

$$D_{k,s}^{l+1} = D_{i,j}^l + \bar{\sigma}_{k,s}(x_{k,s} - x_{i,j}) + \bar{\delta}_{k,s}(y_{k,s} - y_{i,j}), \quad (41)$$

where  $(i, j)$  and  $(k, s)$  refer to the spatial indices of the parent cell and the child cell, respectively.  $D_{k,s}^{l+1}$  is the dependent values (which are  $h$ ,  $uh$ , and  $vh$ ) of the child cell at level  $l + 1$ , while  $D_{i,j}^l$  is the values of its parent cell. The position of the child cell is  $(x_{k,s}, y_{k,s})$ , while the position of its parent cell is  $(x_{i,j}, y_{i,j})$ . Here,

$$\bar{\sigma}_{k,s} = \frac{1}{N_{k,s}} \sum_{n \in \Omega_{k,s}} \sigma_{i,j,n} \quad \text{and} \quad \bar{\delta}_{k,s} = \frac{1}{M_{k,s}} \sum_{m \in \Omega_{k,s}} \delta_{i,j,m} \quad (42)$$

are the averages minmod slope limiters from its neighbor cells in  $x$  and  $y$  directions, respectively, where  $\Omega_{k,s}$  is the region of the child cell.  $M_{k,s}$  and  $N_{k,s}$  are the number of columns and rows, formed by its neighbor cells, respectively.  $\sigma_{i,j,n}$  and  $\delta_{i,j,m}$  are the minmod slope limiters as shown in Section 4.5.

### Coarsening procedure

All children cells of the same parent cell are combined or merged into their parent cell if the computed value for the parent

$$D_{i,j}^l = \frac{1}{n_{i,j}^l \times n_{i,j}^l} \sum_{k,s \in \Omega_{i,j}} D_{k,s}^{l+1} \quad (43)$$

does not satisfy either of the refinement conditions. In this case, the newly merged cell will have the dependent values defined by (43). In Equation (43),  $n_{i,j}^l \times n_{i,j}^l$  is the number of children.

In the computation, the heights of topography on computational grid cells,  $z$ , are obtained via provided digital terrain data on each level; otherwise, in the case that the digital terrain data are not given, the values are obtained similar to the other dependent values as in (41) or (43).

### 4.11. Topography interpolation

In general modeling and simulation based on natural topography, the heights of topography on computational grid cells,  $z$ , are usually obtained using the digital terrain data grids. Since in computation using adaptive grids, it is possible that the grid cell resolution is higher than that of the data; therefore, the topography interpolation is required. In this work, we used the bilinear interpolation technique to obtain the heights of topography. The approximated value of the cell at  $(i, j)$  is obtained by

$$\begin{aligned} z_{i,j} = & z_{x,y}(x + 1 - m)(y + 1 - n) + z_{x+1,y}(m - x)(y + 1 - n) \\ & + z_{x,y+1}(x + 1 - m)(n - y) + z_{x+1,y+1}(m - x)(n - y), \end{aligned} \quad (44)$$

where  $(x, y)$  is the spatial index of the topography data grid cell with  $x = \lfloor m \rfloor$  and  $y = \lfloor n \rfloor$ ,

$$m = i \times n_d^x / n_c^x \quad \text{and} \quad n = j \times n_d^y / n_c^y \quad (45)$$

are the mapped indices of  $(i, j)$  into the data grid cells,  $n_d^x$  and  $n_d^y$  are the numbers of columns and rows of the topography data grids, and  $n_c^x$  and  $n_c^y$  are the numbers of columns and rows of the computational grids, respectively.

#### 4.12. Dynamic domain-defining method

In order to reduce computational time, the dynamic DDM is applied in this modeling and simulation. The method automatically expands or shrinks the calculation area during the simulation to exclude unnecessary cells for the computation. To apply this method to the rainwater overland flow simulation, cells are checked for the computation based on three conditions of indicators. The  $(i, j)$  cell is excluded from the domain, that is, the calculation area is shrunk, if the following three conditions hold:

1. the cell is dry, that is,  $h_{i,j}^n \leq 0$ ;
2. all neighbor cells are dry, that is,  $h_{i,j}^n \leq 0$  for all  $h_{i,j}^n \in N_{i,j}$ , the set of neighbors of the  $(i, j)$  cell; and
3. the cell has no rainfall, that is,  $q_{i,j}^n \leq 0$ .

We used this checking because when the cell and its neighbor cells are all dry with no rainfall, the fluxes of flowing are zero; therefore, the cell dependent value is zero, and we know the exact solution of the cell, so the numerical computation is unnecessary for the cell.

On the other hand, the calculation area is expanded where the cell and its neighbor cells are included to the domain whenever these cells are not in the domain. The cell is checked (that it is in the domain), before including the cell to the domain for protecting against errors and avoiding computation from repeated work.

In the algorithm, the dynamic DDM is employed to obtain a new computational domain for each iteration after the solution of (31) and (32) is obtained via TVD Runge–Kutta method.

#### 4.13. Algorithm overview

We present here an overview of the algorithm for the computational scheme presented previously. The developed algorithm consists of several steps describing the calculations procedures, as illustrated in Figure 4. The detail for each step is described as follows:

- Step 1: set the initial values, the cell sizes, the number of children in each level, and the maximum of tree grid levels.
- Step 2: set the computational grid boundaries; see Section 4.9.
- Step 3: conduct the reconstruction procedure; see Section 4.5.
- Step 4: calculate the numerical fluxes and the maximum eigenvalue.
- Step 5: calculate the solutions of  $\vec{W}^{(1)}$ ; see (31).
- Step 6: define a new domain based on the dynamic DDM; see Section 4.12.

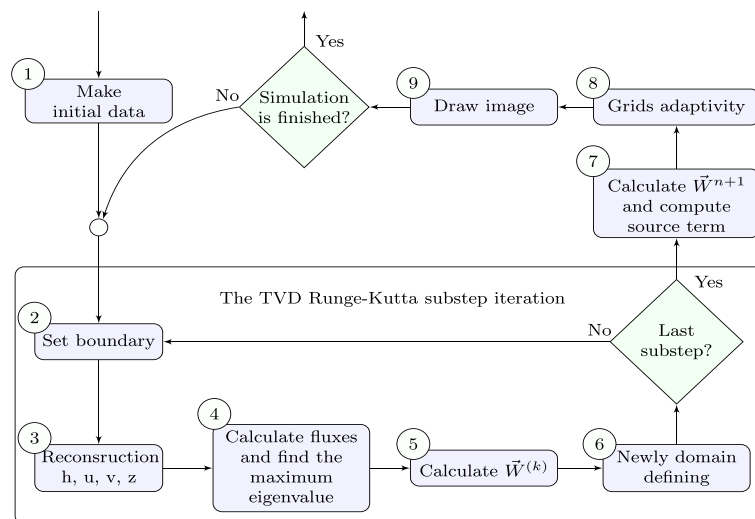


Figure 4. The algorithm overview. TVD, total variation diminishing.

- Repeat steps 2–6, for the solutions of  $\vec{W}^{(2)}$ ; see (32), and go to step 7.
- Step 7: calculate the solutions of  $\vec{W}^{n+1}$ ; see (33), and compute the source terms; see Section 4.8.
- Step 8: conduct the grid adaptivity technique; see Section 4.10.
  - I. Refinement procedure.
    - (i) For each cell whose level is not maximum level
      - mark cells that satisfy the refinement conditions 1 and 2.
    - (ii) For each marked cells
      - create new children cells, and assign the dependent values as Equation (41).
  - II. Coarsening procedure.
    - (i) For each parent cell whose children have no child
      - compute the dependent values defined in (43);
      - check the refinement conditions 1 and 2; and
      - if either of the conditions is false, then merge all children cells into a parent cell, and assign the dependent values.
- Step 9: visualize the solutions in 2D and three dimensional (3D) by OpenGL (Khronos Group, Beaverton, OR, USA).
- Repeat steps 2–9 until the simulation is finished.

Note that, in step 3, cells average values  $u$  and  $v$  are used for the reconstruction procedure, which are calculated by dividing  $uh$  and  $vh$  with  $h$ . However, because the values of  $u$  and  $v$  are not defined in case that  $h = 0$ , the function presented by Kurganov and Petrova [24] is used for finding the cells average values of  $u$  and  $v$  as follows:

$$\text{div}(a, h) = \frac{\sqrt{2}ah}{\sqrt{h^4 + \max(h^4, \eta)}}, \quad (46)$$

where  $\eta$  is a small positive constant value. In finding  $u$  and  $v$ ,  $a$  is substituted by  $uh$  and  $vh$ , respectively.

## 5. EXPERIMENTAL RESULTS

We illustrate here the numerical simulations based on adaptive grids and dynamic DDM to show the effectiveness of the developed model. The experiments consist of four tests as follows: the dam break simulation to show the adapted grid and water profile of the model; the simulation of a 2D parabolic bowl to compare with the analytical solution; the rainfall run-off simulation on sloped plane compared with real experimental data; and the simulation of rainwater overland flow on natural topography to compare the performance with non-adapted uniform grids results. The first, second, and third numerical simulations are for accuracy tests, while the fourth simulation is the result when applied for the rainwater flow model. The developed algorithm is programmed using Delphi version 7 on a 2-GHz Intel Core i7 machine with RAM 6 GB.

### 5.1. Two-dimensional dam break simulation

This experiment simulates the aspect of the water flow for a dam break behavior on  $200 \times 200$  m rectangular area with a flat horizontal frictionless bottom. The square domain is divided into two halves by infinitesimally thin dam. The breaking part of the dam is 75 m long from 100 to 175 m in the  $y$  direction; see Figure 5. The experiment is performed with the initial water depths at 10 m upstream, while the downstream is at 5 m, and with frictionless solid walls. The adaptivity of

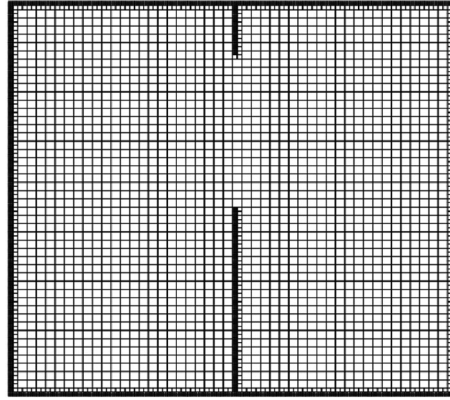


Figure 5. The initial grid cells of the rectangular dam break.

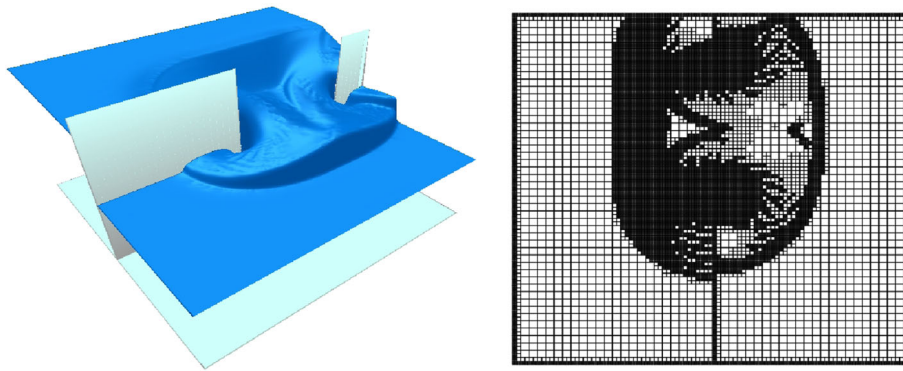


Figure 6. The three-dimensional representation (left) and the adaptive grid (right) of the dam break simulation at  $t = 5$  s.

rectangular tree grid is employed with three levels: the initial level 1 has  $50 \times 50$  grid cells; the dimensions of children for levels 1 and 2 are  $2 \times 2$  and  $3 \times 3$ , respectively; and level 3 cells have no children. In addition, the parameters used in the grid adaptivity, in (39) and (40), are given by  $\epsilon = 0.05$  and  $\xi = 0$ .

The initial tree grid is shown in Figure 5, and the simulation of the behavior of water flow together with the grid cells at time  $t = 5$  s, based on the dynamic DDM with adaptive tree grids, is shown in Figure 6. This experiment showed that the high-resolution regions, where the gradients of the free surface are steep, are obtained automatically via the adaptive technique. The results agree closely with other numerical simulation results in literature; see [7, 25–28].

### 5.2. Two-dimensional parabolic bowl simulation

To validate our developed algorithm, the numerical experiment of a water flow on 2D parabolic bed slope is performed and compared with analytic solution, presented by Wang [29]. Note that the 2D parabolic bowl test cases are a variant of Thacker's solution [30] with Manning's friction term.

For the domain  $L \text{ m} \times L \text{ m}$ , the 2D parabolic bed topography is defined by

$$z(x, y) = h_0 \left( (x - L/2)^2 + (y - L/2)^2 \right) / a^2, \quad (47)$$

where  $h_0$  and  $a$  are positive constants. It is known that the analytic solution depends on the relationship between the bed friction parameter  $\tau$  and a peak amplitude parameter  $p = \sqrt{8gh_0}/a$ .

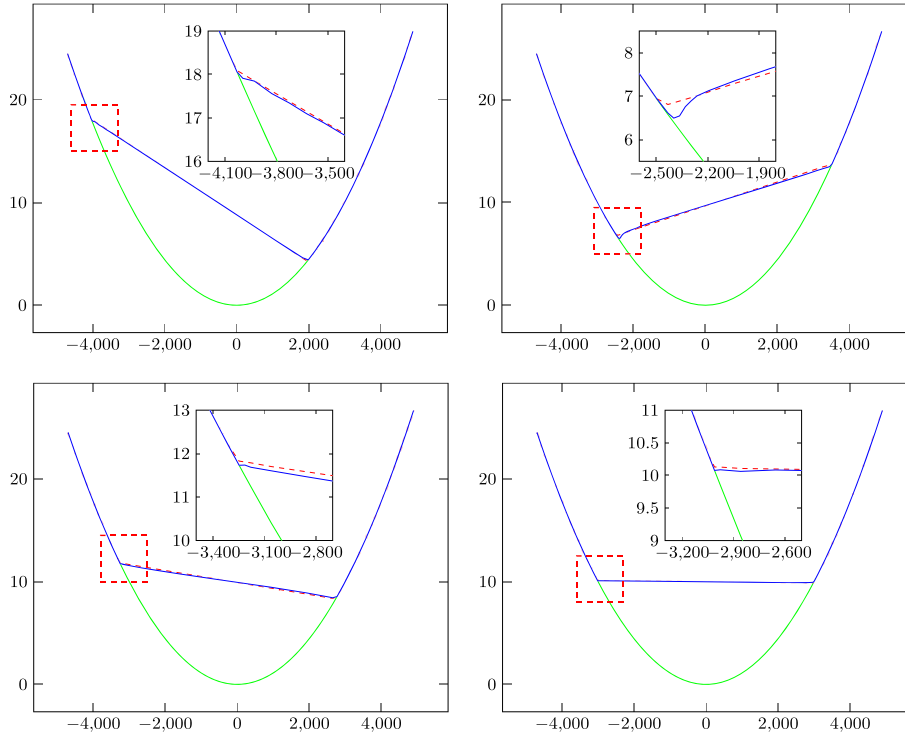


Figure 7. Water surface profiles along the centerline in the  $x$  direction of the simulations (blue) compared with the analytic solution (red) for the two-dimensional parabolic bed topography (green) at  $t = 30.3532, 671.7932, 1358.1300,$  and  $2503.5160$  s (from left to right and top to bottom). The small pictures show some zooms at the wet/dry transition for the simulations.

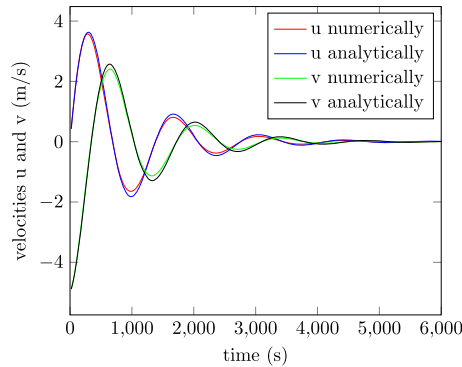


Figure 8. The evolution of the velocities of the model compared with analytic solutions at  $x = 6000$  m.

For  $\tau < p$ , the analytic solution for the water depth is

$$h(x, y, t) = h_0 - \frac{1}{2g} B^2 e^{-\tau t} - \frac{1}{g} B e^{-\tau t/2} \left( \frac{\tau}{2} \sin st + s \cos st \right) \left( x - \frac{L}{2} \right) - \frac{1}{g} B e^{-\tau t/2} \left( \frac{\tau}{2} \cos st - s \sin st \right) \left( y - \frac{L}{2} \right), \quad (48)$$

and the velocities are

$$u(t) = B e^{-\tau t/2} \sin st \quad (49)$$

$$v(t) = -B e^{-\tau t/2} \cos st, \quad (50)$$

where  $s = \sqrt{p^2 - \tau^2}/2$  and  $B$  is a constant.

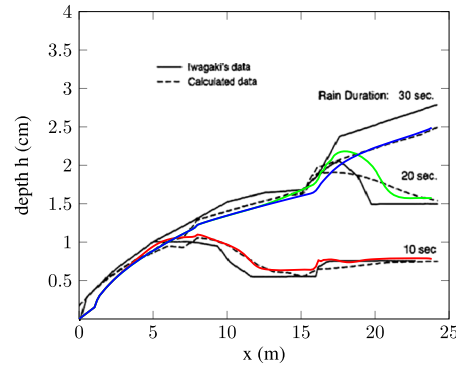


Figure 9. The water profiles along the centerline in the  $x$  direction of the simulations at 10 s (red), 20 s (green), and 30 s (blue) compared with Iwagaki's experimental and calculated data.

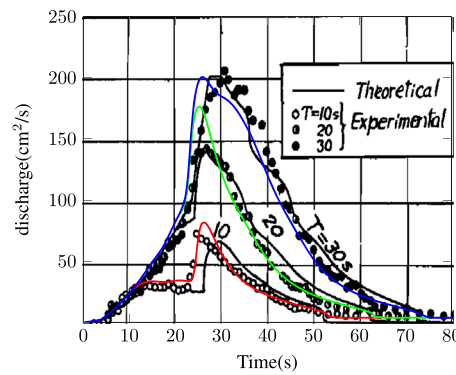


Figure 10. The evolutions of the water discharges at the downstream with durations of rainfall 10 s (red), 20 s (green), and 30 s (blue), compared with Iwagaki's experimental and calculated (theoretical) data.

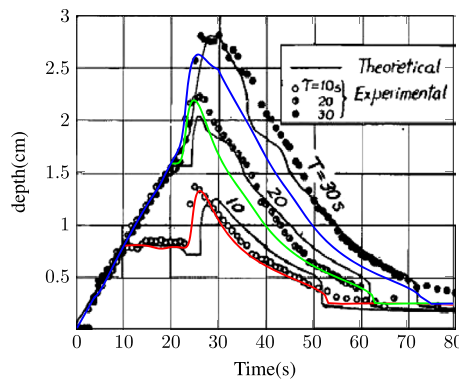


Figure 11. The evolutions of the water depth at the downstream with durations of rainfall 10 s (red), 20 s (green), and 30 s (blue), compared with Iwagaki's experimental and calculated (theoretical) data.

In this experiment, we assume that  $L = 10,000$  m,  $h_0 = 10$  m,  $a = 3000$  m, and  $B = 5$  m/s. The bed friction parameter,  $\tau = 0.002 \text{ s}^{-1}$ , is related to the Manning's coefficient  $n^2 = \frac{\tau h^{4/3}}{g \sqrt{u^2 + v^2}}$ . The initial grid level 1 has  $50 \times 50$  cells, and the maximum tree grid level is 3 with  $2 \times 2$  and  $3 \times 3$  children for levels 1 and 2, respectively, while level 3 cells have no children. The initial water depth, velocities, and bed topography are defined using the analytic solution at  $t = 0$ . The duration of the simulation is 6000 s and performed with parameters of the grid adaptivity  $\epsilon = 0.3$  and  $\xi = 0.3$  and with opened boundaries.

The water surface profiles along the centerline in the  $x$  direction obtained by the simulations are compared with the analytic solutions at different time steps. As shown in Figure 7, the profile of the water surfaces and the wet–dry fronts agree closely with the analytical solution. The evolutions of the velocities  $u$  and  $v$  compared with analytic results are shown in Figure 8. The effect of friction on the flow is clear, because the amplitudes of the velocities oscillate and decrease throughout the simulation. This experiment showed that the developed numerical scheme is capable of simulating a water flow on non-uniform topography with moving wet–dry fronts and bed roughness.

### 5.3. Rainfall simulation on sloped plane

In this test, the simulation of rainfall runoff on sloped plane was performed and compared with experimental data of Iwagaki [31]. In the physical experiment by Iwagaki, the flow was performed on 24 m long and 0.196 m wide, cascaded by three (8 m long) aluminum planes with slopes of 0.02,

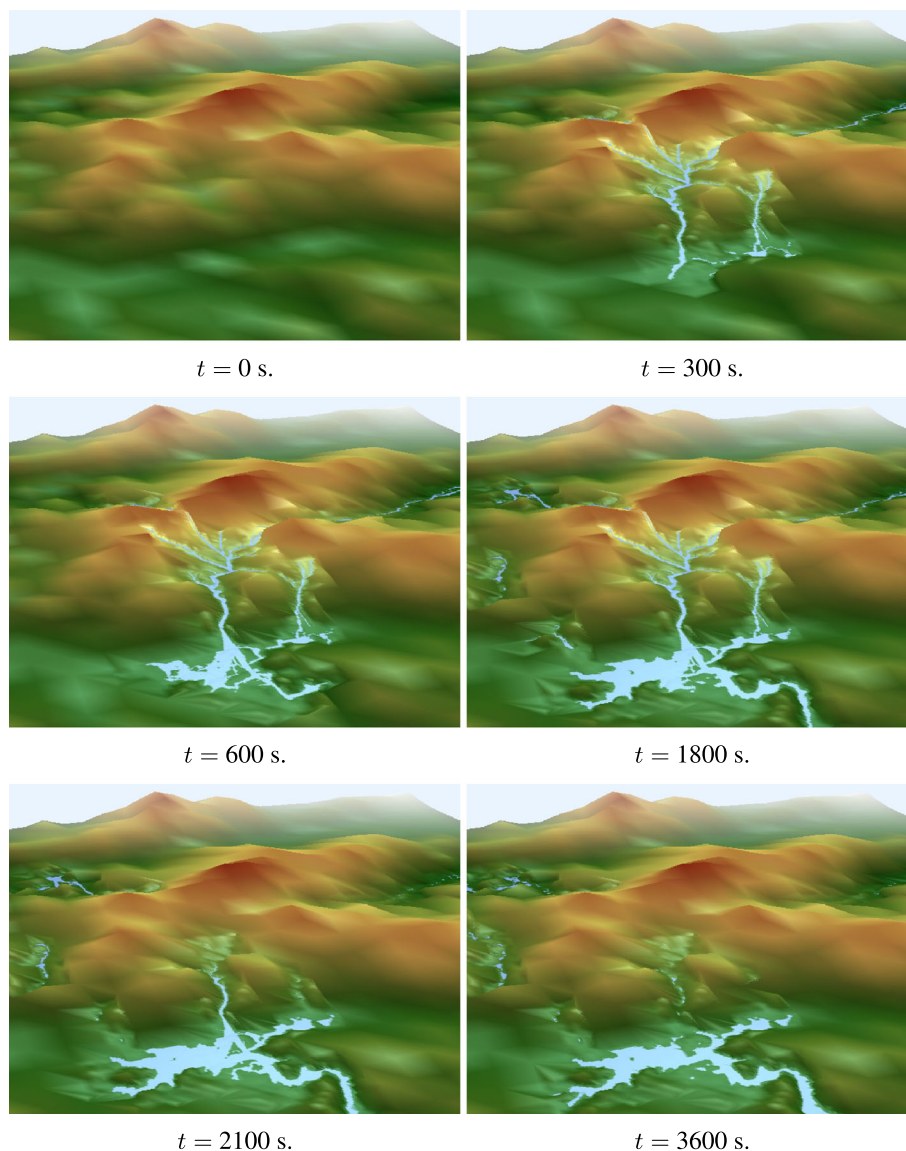


Figure 12. The three-dimensional simulations of the behavior of the rainwater overland flow at different times  $t$ .



0.015, and 0.01 in the downstream direction, which received constant rainfall intensities of 389, 230, and 288  $\text{cm h}^{-1}$ , respectively. The experiment was performed in three cases of different durations of rainfall: 10 s (case 1), 20 s (case 2), and 30 s (case 3).

In this numerical simulation, the initial grid level 1 has  $24 \times 7$  cells, and the maximum tree grid level is 3 with  $3 \times 3$  and  $2 \times 2$  children for levels 1 and 2, respectively, while level 3 cells have no children. The initial water depth and velocities are defined as zero. The boundary conditions are defined as closed on both lateral walls and opened at the upstream and downstream. The simulation is performed with the Manning's coefficient 0.009 for the duration of 80 s and with parameters of the grid adaptivity  $\epsilon = 0.002$  and  $\xi = 0.003$ .

The numerical simulations of water profiles compared with Iwagaki's experimental and calculated data are shown in Figure 9. The comparisons of the evolutions of the discharge and water depth downstream are shown in Figures 10 and 11. The results agree closely with the rainfall run-off experiment data.

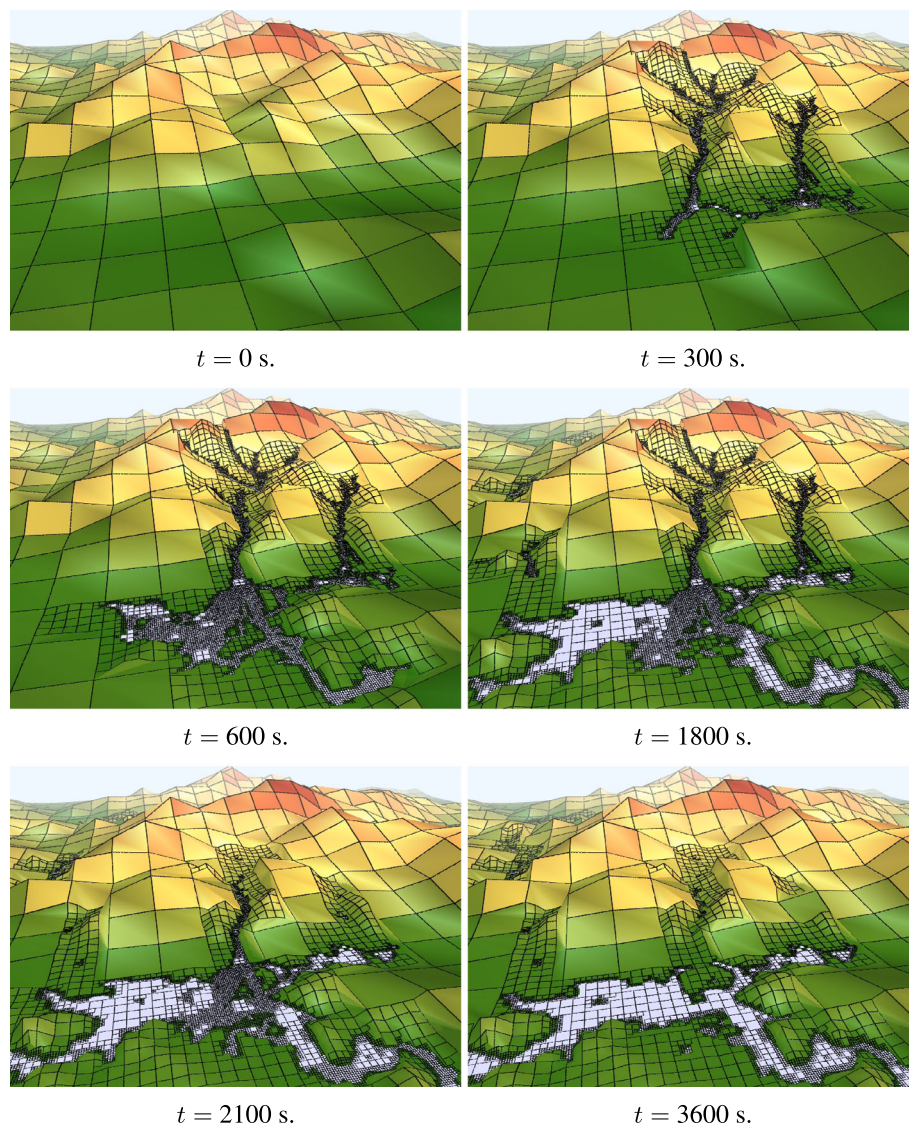


Figure 13. The three-dimensional representations of the adapted grids of the rainwater overland flow simulations at different times  $t$ .

#### 5.4. Rainwater overland flow

In this experiment, the developed dynamic DDM with adaptive tree grid numerical algorithm is employed to simulate the rainwater flow on natural topography. The accuracy and efficiency of the dynamically adaptive scheme are tested by comparing with the simulations on the uniform grids. The simulations are performed on the domain of digital terrain data  $36,000 \times 36,000$  m, generated from Shuttle Radar Topography Mission data source. The maximum tree grid level is 4 with resolutions  $900 \times 900$  m,  $180 \times 180$  m,  $90 \times 90$  m, and  $30 \times 30$  m, for levels 1, 2, 3, and 4, respectively; therefore, the initial grid level 1 has  $40 \times 40$  cells. The topography values are provided for levels 1–3, except level 4 that is obtained automatically via the bilinear interpolation. The tree grid cells for children are designed as  $5 \times 5$ ,  $2 \times 2$ , and  $3 \times 3$  for levels 1, 2, and 3, respectively, while level 4 cells have no children. The numerical experiment is simulated for 3600 s with the Manning's coefficient 0.001 and with parameters of the grid adaptivity  $\epsilon = 1.5$  and  $\xi = 0.1$ . The rainfall rate is set at 0.001 m/s within the rectangular region where  $15,750 \leq x \leq 20,250$  and  $13,500 \leq y \leq 18,000$  for the duration of 1800 s. Moreover, the boundary conditions are defined as opened boundaries.

The 3D simulations of rainwater overland flow for this experiment are presented at different times in Figure 12, and the corresponding automatically adapted grids are shown in Figure 13. The results show that the high-resolution areas are automatically obtained by the adaptivity technique in the corresponding regions where the free surface gradients are steep and have water depths more than 0.1 m, based on adaptivity parameters.

To check the accuracy and efficiency of the dynamically adaptive scheme, we performed the following test. We compared the dynamically adaptive simulation with non-adaptive simulations

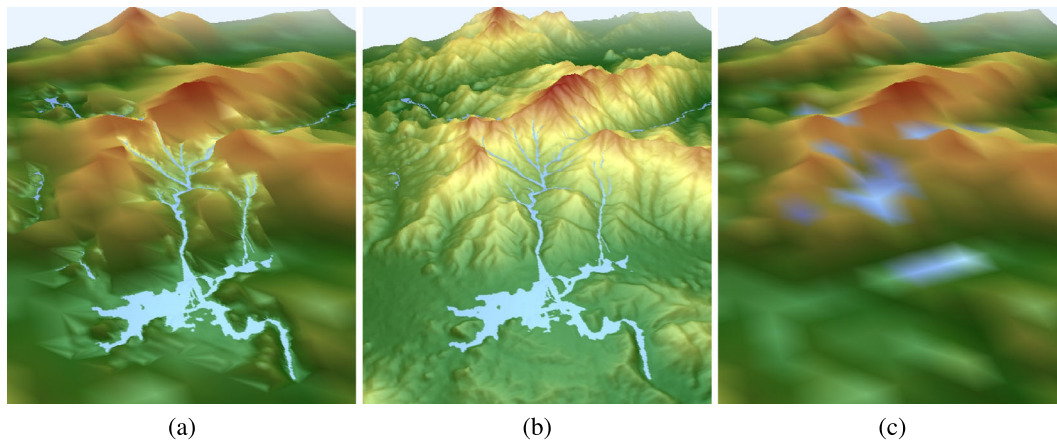


Figure 14. The three-dimensional simulations of the rainwater overland flow based on the (a) adaptive grid, (b) uniform level 4 grid (small cells), and (c) uniform level 1 grid (large cells) (c) at  $t = 1800$  s.

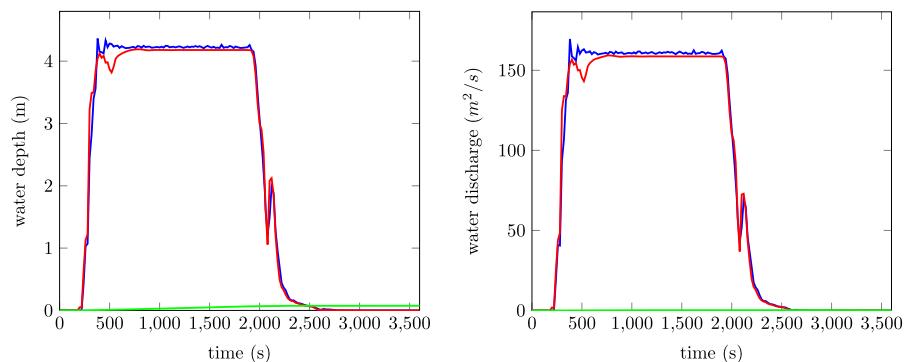


Figure 15. The comparisons of water depths (left) and discharges (right) at 16,710 and 21,060 m for the adaptive grid (blue); uniform level 4 grid, small cells (red); and uniform level 1 grid, large cells (green).

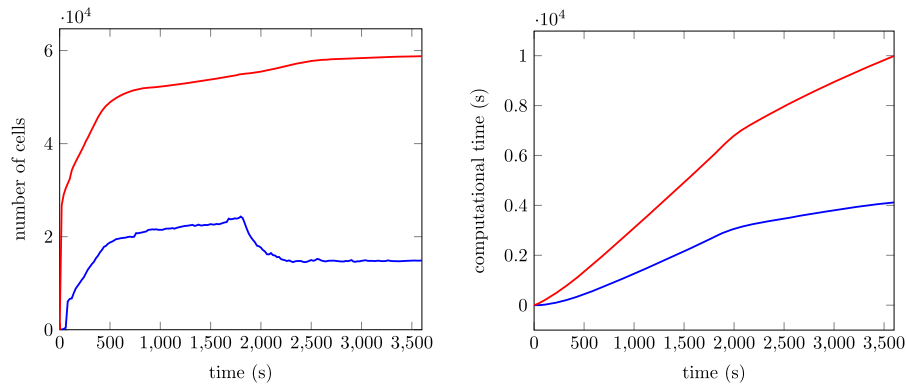


Figure 16. The comparisons of number of computational cells (left) and times (right) for adaptive grid (blue) and uniform level 4 grid, small cells (red).

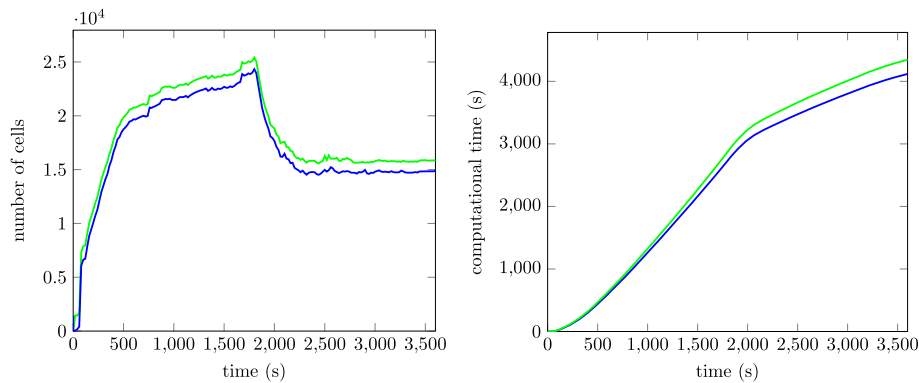


Figure 17. The comparisons of number of computational cells (left) and times (right) using adaptivity with dynamic domain-defining method (DDM) (blue) and without dynamic DDM (green).

computed using only the dynamic DDM on two uniform grids, level 1 grid (largest cells with  $40 \times 40$  cells) and level 4 grid (smallest cells with  $1200 \times 1200$  cells). The 3D representations of the rainwater overland flow simulations at 1800 s for this test are shown in Figure 14, and the profiles of water depths and discharges at the location (16,710 and 21,060 m) are shown in Figure 15. The results show that the simulation and the flow profiles obtained by the dynamically adaptive scheme are nearly the same as that obtained from the finest grid (level 4 grid), while the result from the largest cells (level 1 grid) is not acceptable. This shows the accuracy of the adaptive scheme. The efficiency of the adaptive scheme is presented by comparing the number of computational cells and the computational times for the simulations as shown in Figure 16. The results show that the adaptive scheme is more efficient because it helps reducing the computational cells and therefore reduces the computational time while keeping the same accuracy as that of the finest grid.

Moreover, to compare the effect of the dynamic DDM, the computational times and number of computational cells are compared for the adaptive schemes with or without the dynamic DDM, as shown in Figure 17. The results show that the dynamic DDM can help improving the simulation but not as much as that of adaptivity.

## 6. CONCLUSION

In this research, we developed the dynamically adaptive tree grid algorithm for simulation and visualization of rainwater overland flow based on adaptive FVMs for shallow water equations and in combination with the dynamic DDM. The numerical simulations show that the developed algorithm

is suitable for simulating and visualizing the flow of rainwater on natural topography, which agree closely with other results obtained in literature, for both experimental and theoretical results. Moreover, the adaptivity and dynamic DDM help making the computation efficient while keeping the same accuracy as that of the computation on the finest grid. Therefore, it has a potential for practical usages and can be applied to other flooding simulations on natural topography such as dam break, river flood, and flood inundation in city, including utilization for the water management system.

#### ACKNOWLEDGEMENTS

We would like to thank the editor and reviewers for their valuable comments, which have significantly improved this paper. This research was funded by Science Achievement Scholarship Thailand (SAST).

#### REFERENCES

1. de Saint Venant AJC. Théorie du mouvement non-permanent des eaux, avec application aux crues des rivières et à l'introduction des marées dans leur lit. *Comptes Rendus de l'Académie des Sciences* 1871; **73**:147–154.
2. Zhang W, Cundy TW. Modeling of two-dimensional overland flow. *Water Resources Research* 1989; **25**(9): 2019–2035.
3. Fiedler FR, Ramirez JA. A numerical method for simulating discontinuous shallow flow over an infiltrating surface. *International Journal for Numerical Methods in Fluids* 2000; **32**:219–240.
4. Delestre O, Cordier S, James F, Darboux F. Simulation of rain-water overland-flow. In *Proceedings of the 12th International Conference on Hyperbolic Problems*, University of Maryland, College Park (USA), 2008, Tadmor E, Liu J-G, Tzavaras A (eds), Proceedings of Symposia in Applied Mathematics, American Mathematical Society, Vol. 67, 2009; 537–546.
5. Delestre O. Simulation du ruissellement d'eau de pluie sur des surfaces agricoles/rain water overland flow on agricultural fields simulation. *PhD Thesis*, Université d'Orléans, French, July 2010. Available from TEL: tel.archives-ouvertes.fr/INSMI/tel-00531377/fr.
6. George DL. Adaptive finite volume methods with well-balanced Riemann solvers for modeling floods in rugged terrain: application to the Malpasset dam-break flood (France, 1959). *International Journal for Numerical Methods in Fluids* 2010.
7. Liang Q, Borthwick AGL, Stelling G. Simulation of dam- and dyke-break hydrodynamics on dynamically adaptive quad-tree grids. *International Journal for Numerical Methods in Fluids* 2004; **46**:127–162.
8. George DL. Finite volume methods and adaptive refinement for tsunami propagation and inundation. *Doctor's Dissertation*, Applied Mathematics, Washington University, St. Louis, Missouri, United States, 2006.
9. Popinet S. Quad-tree adaptive tsunami modelling. *Ocean Dynamics* 2011; **61**(9):1261–1285.
10. Goutal N, Maurel F. A finite volume solver for 1D shallow-water equations applied to an actual river. *International Journal for Numerical Methods in Fluids* 2002; **38**:1–19.
11. Sampson J, Easton A, Singh M. Modelling the effect of proposed channel deepening on the tides in Port Phillip Bay. *ANZIAM Journal* 2005; **46**:C888–C901.
12. Bermudez A, Vazquez ME. Upwind methods for hyperbolic conservation laws with source terms. *Computers & Fluids* 1994; **23**(8):1049–1071.
13. Greenberg JM, LeRoux AY. A well-balanced scheme for the numerical processing of source terms in hyperbolic equation. *SIAM Journal on Numerical Analysis* 1996; **33**:1–16.
14. Audusse E, Bouchut F, Bristeau MO, Klein R, Perthame B. A fast and stable well-balanced scheme with hydrostatic reconstruction for shallow water flows. *SIAM Journal* 2004; **25**(6):2050–2065.
15. An H, Yu S. Well-balanced shallow water flow simulation on quadtree cut cell grids. *Advances in Water Resources* 2012; **39**:60–70.
16. Schreiber M, Bungartz HJ, Bader M. Shared memory parallelization of fully-adaptive simulations using a dynamic tree-split and join approach. *19th International Conference on IEEE High Performance Computing (HiPC)*, 2012; 1–10.
17. Yamaguchi S, Ikeda T, Iwamura K, Naono K, Ninomiya A, Tanaka K, Takahashi H. Development of GIS-based flood-simulation software and application to flood-risk assessment. *2nd IMA International Conference on Flood Risk Assessment*, University of Plymouth, 2007.
18. Delestre O, Cordier S, Darboux F, James F. A limitation of the hydrostatic reconstruction technique for shallow water equations. *Comptes Rendus de l'Académie des sciences* 2012; **350**:677–681.
19. Morales de Luna T, Castro Daz MJ, Pares C. Reliability of first order numerical schemes for solving shallow water system over abrupt topography. *Applied Mathematics and Computation* 2013; **219**:9012–9032.
20. Hou J, Simons F, Liang Q, Hinkelmann R. An improved hydrostatic reconstruction method for shallow water model. *Journal of Hydraulic Research* 2014; **52**(3):432–439.
21. Harten A, Lax PD, Leer BV. On upstream differencing and Godunov-type schemes for hyperbolic conservation laws. *SIAM Review* 1983; **25**(1):35–61.

22. Kurganov A, Noelle S, Petrova G. Semi-discrete central-upwind schemes for hyperbolic conservation laws and Hamilton-Jacobi equations. *SIAM Journal* 2001; **23**(3):707–740.
23. Hagen TR, Hjelmervik JM, Lie KA, Natvig JR, Henriksen MO. Visual simulation of shallow water waves. *Simulation Modelling Practice and Theory: Special Issue on Programmable Graphics Hardware* 2005; **13**:716–726.
24. Kurganov A, Petrova G. Central-upwind schemes for two-layer shallow water equations. *SIAM Journal* 2009; **31**(3):1742–1773.
25. Richardson SR. Numerical simulation of impulsive wave over-topping events resulting from landslides. *Doctor's Thesis*, Computing and Mathematics, Manchester Metropolitan University, Manchester, United Kingdom, 2010.
26. Roshandel A, Hedayat N, Kiamanesh H. Simulation of dam break using finite volume method. *World Academy of Science, Engineering and Technology* 2010; **4**(11):103–106.
27. Suwannasri P. Parallel computation of shallow water flow problems. *Master's Thesis*, Computational Science, Chulalongkorn University, Bangkok, Thailand, 2004.
28. Valiani A, Caleffi V, Zanni A. Finite volume scheme for 2D shallow water equations application to the Malpasset dam-break. *Proceedings of the 4th CADAM Meeting*, Zaragoza, Spain, 1999; 63–93.
29. Wang Y. Numerical improvements for large-scale flood simulation. *Doctor's Dissertation*, School of Civil Engineering and Geosciences, Faculty of Science, Agriculture and Engineering, Newcastle University, Newcastle, United Kingdom, 2011.
30. Thacker WC. Some exact solutions to the nonlinear shallow-water wave equations. *Journal of Fluid Mechanics Digital Archive* 1981; **107**(1):499–508.
31. Iwagaki Y. Fundamental studies on the run-off analysis by characteristics. *Disaster Prevention Research Institute Bulletin, Kyoto University, Kyoto, Japan* 1955; **10**:1–25.