Efficient surface water flow simulation on static Cartesian grid with local refinement according to key topographic features

Jingming Hou[1], Run Wang[1], Qiuhua Liang[1], Zhanbin Li[1], Mian Song Huang[2] and Reihnard Hinkelmann[3]

School of Water Resources and Hydroelectric Engineering, Xi'an University of Technology, Xi'an, 710048, China

Institute of Water Resources & Electric Power, Qinghai University, Xining, 810016,China.

Chair of Water Resources Management and Modeling of Hydrosystems, Department of Civil Engineering, Technische Universität Berlin. TIB 1-B14, Gustav-Meyer-Allee 25, 13355 Berlin, Germany

Abstract

Aiming at improving the computational efficiency without accuracy losses for surface water flow simulation, this paper presents a structured but non-uniform grid system incorporated into a Godunov-type finite volume scheme. The proposed grid system can detect the key topographic features in the computational domain where high-resolution mesh is in need for reliably solving the shallow water equations. The mesh refinement is automatically carried out in these areas while the mesh in the rest of the domain remains coarse. The criterion determining the refinement is suggested by a dimensionless number with a fixed value of 0.2 after sensitivity analysis. Three laboratory and field-scale test cases are employed to demonstrate the performance of the model for flow simulations on the new non-uniform grids. In all of the tests, the grid system is shown to successfully generate high-resolution mesh only in those areas with abruptly changing topographic features that dominate the flooding processes. To produce numerical solutions of similar accuracy, the non-uniform grid based model is able to accelerate by about two times comparing with the fine uniform grid based counterpart.

Keywords: Shallow water equations, Non-uniform grid, Finite volume method, Godunov-type scheme, Flood

1. Introduction

Numerical models play an important role in predicting surface water flow and the relevant processes e.g. sediment and pollutants transport[16, 11, 29]. Those models based on the finite volume Godunov-type schemes have received much attention in the recent years, due to their great capability in providing accurate and shock-capturing solutions, for example in [22, 17, 12]. Associated with these shallow water flow models, two types of grids are normally used to discretize the computational domains, i.e.

structured and unstructured grids. Structured grids have been adopted by numerous shallow water flow models, for instance, [22, 37, 5, 38], owning to their simple grid structure and easy numerical implementation. On the other hand, unstructured grids generally require complicated grid generation algorithms and careful numerical considerations when implementing higher order numerical schemes, although they provide great flexibility in comforting to complex domain geometries. A few unstructured shallow flow models have been reported for simulating hydrodynamic process on irregular domains [6, 17]. In practice, raster-based digital elevation models (DEMs) are generally used to represent the topography and thus facilitate the use of structured grids. Moreover, as mentioned in [23] complex geometry could be also relatively well-captured by structured non-uniform Cartesian grids. Therefore, structured grids could be a good choice for practical shallow water flow simulation.

As the simplest type of structured grids, uniform Cartesian grids are simple and do not require any data structure to store grid information. However, on uniform grids, high-resolution simulations in a large domain will inevitably involve large number of computational nodes and substantially increase computational cost. This essentially restricts their application in realistic surface water flow simulations that usually involve large domains. There are several ways to improve the computational efficiency of a shallow water flow model. Bates et al. reported a simplified partial inertial model, which is derived by neglecting the dynamic terms in the full 2D shallow water equations (SWEs) and hence is computationally more efficient than the full models. This approach was applied in [36] to handle the hydrodynamic component of the proposed numerical model. However, without dynamic terms, the model may not be able to reliably represent the physical processes and predict velocity for rapidly-varying flows, e.g. dam breaks. Parallel computing also provides an effective way to release the computational burden of a shallow water flow model e.g. Hinkelmann and Zielke [15], Sanders et al. [35], Zhang et al. [46], Hinkelmann [14]. In recent years, the graphic processing units (GPUs) have also been actively explored to accelerate shallow flow models [41, 9, 45].

Another popular approach to improve computational efficiency is through optimizing grid resolution during a simulation by using adaptive mesh refinement (AMR) [25, 7, 26]. Nesting methods, patching algorithms and dynamically adaptive gridding techniques have been widely used to provide static or dynamic AMR for shallow water flow simulations. A nested grid may be generated by superimposing a fine grid over a coarse one [31] and was employed by Heggelund and Berntsen [13] for shallow water modeling. However, the complex interaction between the fine and coarse grids during flow calculation can reduce solution accuracy at the fine mesh boundaries and give rise to numerical instabilities at the interface between the grids [31, 13]. Due to these unavoidable inherent disadvantages, nested grids have received less attentions in the field of computational fluid dynamics in recent years. Patching (multiblock) algorithms are another type of approach to achieve a locally refined grid [34]. This technique decomposes the computational domain into a coarse grid that is connected to sets of subgrids with various resolutions. Hu et al. [20] solved the SWEs incorporating this gridding technique. Although the patching algorithms can be generally implemented in a straightforward way, it is however necessary to perform flow computation on individual patches and link the solutions together, which may also introduce extra numerical inaccuracy to the flow solution. Dynamically adaptive grids can also be used flexibly for local grid refinement, by means of adjusting cell density according to the domain features and/or flow conditions [33, 7]. However, complicated grid generation techniques and intensive storage requirement for grid information may be commonly needed, which may restrict their wider applications in certain circumstances. In addition, when estimating flow information in those newly created cells, between mass and water surface, only one of them can be preserved [32, 21]. This means a dynamically adaptive grid based (e.g. quadtree grids) may not be able to maintain mass conservation and the so-called conservation property (C-property [4]) at the same time. It is well-acknowledged that mass conservation is a key factor should be satisfied in a shallow water flow model for practical applications. From the numerical point of view, C-property affects the numerical stability of a shallow flow model and thereby is of importance to preserve its robustness.

This work focuses on grid optimization using static structured non-uniform grids for shallow water flow simulations. In shallow water flow modeling, high-resolution computation is only required in those areas of the domain where the flow patterns are to be affected by complex topographic features, e.g. walls or flow paths, that stop or direct the water (Figure 1). Therefore, using high-resolution grids only in those areas of major interest and importance for flow hydrodynamics will contribute to improve the computational efficiency but meanwhile maintain solution accuracy. Liang [23] proposed a new structured but non-uniform Cartesian grid system for solving the SWEs. Apart from being able to easily achieve local grid refinement, the resulting grids effectively avoid the troublesome dynamical links between different numerical solutions as required for a nested grid and do not need to separate the computation domain into different individual mesh components as required by a patched grid. Furthermore, the grid system also boasts of substantially simplified grid generation procedure in comparison with a quadtree grid. However, manual work is still required to define those areas that need local refinement.

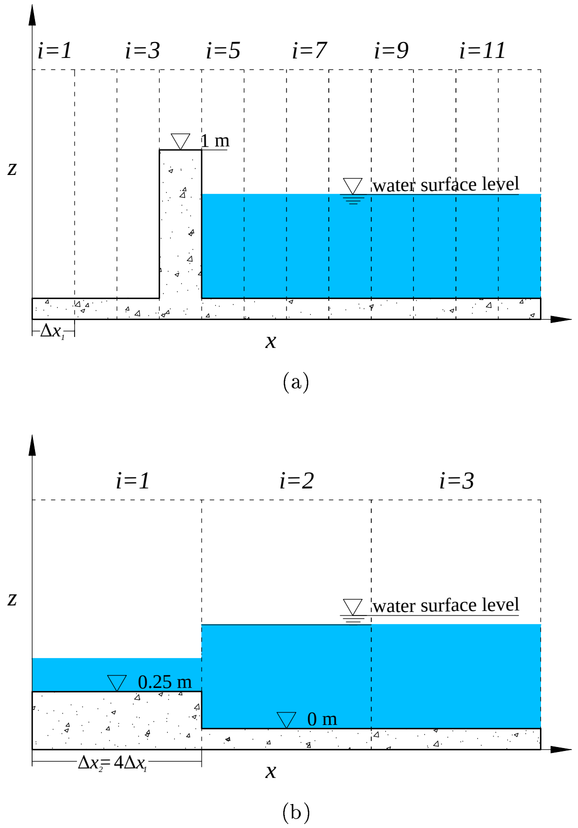


Figure 1: Spatial discretization for a floodplain with levee: (a) using fine grid, (b) using coarse grid.

This work is devoted to devise the grid system as proposed by Liang [23] to automatically identify and refine the crucial areas of the domain where high-resolution mesh is necessary for accurate shallow water flow simulation, aiming to avoid the manual intervention as requested by the original grid generation procedure and to improve the computational efficiency. The remainder of the paper is arranged as following: the governing shallow water equations for shallow water flow modeling are introduced in Section 2; Section 3 describes the improved structured non-uniform Cartesian grid system; Section 4 introduces a 2D finite volume Godunov-type scheme that is compatible to the new grid system; Section 5 demonstrates the merits of the improved grid system and the associated shallow water flow model using three test cases; and brief conclusions are finally drawn in Section 6.

2. Non-uniform structured Cartesian grid

A computational grid of good quality may have a great and positive impact on model efficiency and accuracy. This work aims to introduce a grid system that is able to make a good tradeoff between the

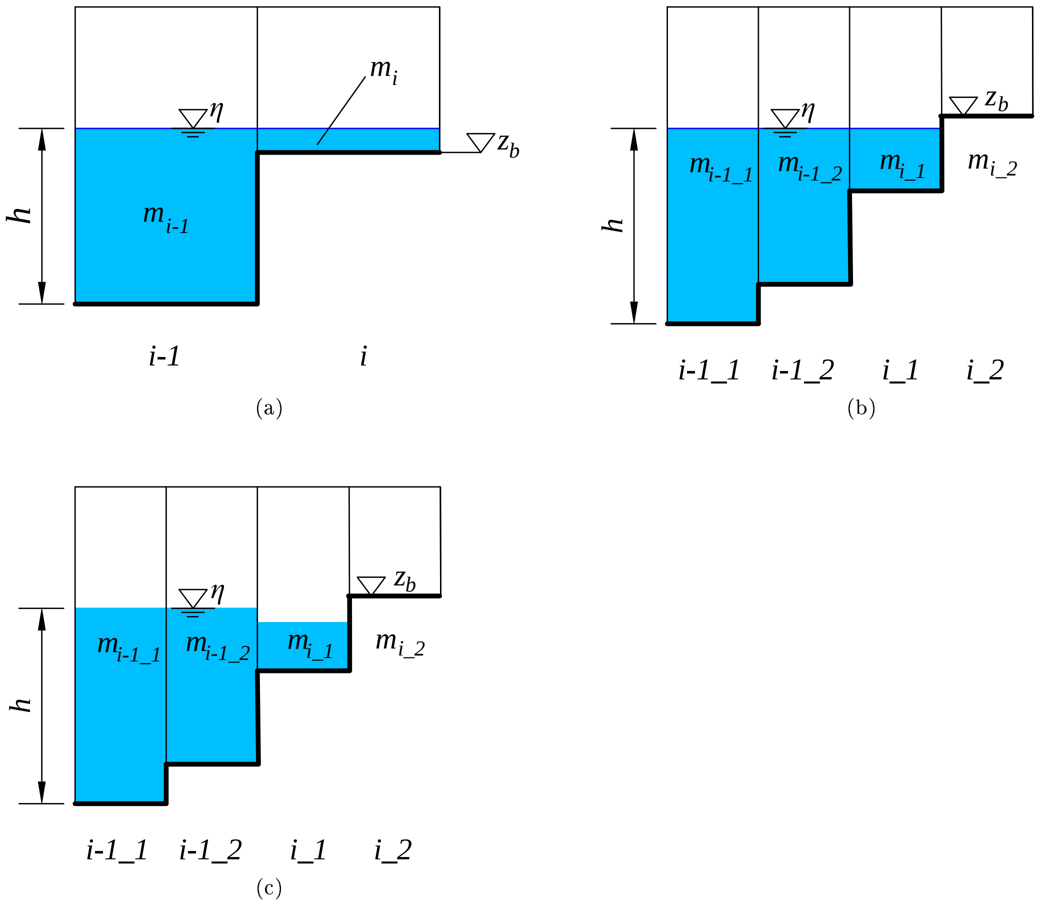


Figure 2: Contradictions between the mass conservation and C-property for dynamically adaptive grids: (a) flow variables in parent cells, (b) new values of flow variables in refined cells by satisfying the C-property, (c) new values of flow variables in refined cells by satisfying the mass conservation.

computational efficiency and simulation reliability. The new grid system is devised to offer refined cells at certain parts of the domain where high-resolution representation of topographic features are essential for shallow water flow simulation, while leaving the rest of the domain discretized by coarse cells. The generation of such a computational grid is carried out in two steps. Firstly, the parts demanding high-resolution representation are marked in the whole domain. Secondly, the grid is refined locally in the marked areas and coarsened in the rest of the domain. The process is similar to the dynamically grid adaptation as reported in [24, 32, 33], but static non-uniform grids are herein created. The reason why the static grids are preferred in this work is that the dynamically adaptive grids can not satisfy the mass conservation and the C-property at the same time and intricate adaptation may actually slow down the computation, although they improve the model efficiency comparing to the overall refined grids. It is straightforward to see the additional computational effort of the adaptation but not easy to understand why dynamically adaptive grids can only preserve either mass conservation or the C-property instead of both. As demonstrated in Figure 2, if a scheme is able to preserve the C-property under the static flow condition given for coarse grid shown in Figure 2(a), the mass conservation in a fully wet cell (cell $i-1$) can be satisfied after being refined into two sub-cells $i-1\_1$ and $i-1\_2$ by using linear interpolation for water level and bed elevation (Figure 2 (b)), i.e. $m_{i-1} = m_{i-1\_1} + m_{i-1\_2}$ with m standing for the mass of a cell. However, the mass conservation in the partially wet cells (cell i in Figure 2(a), this kind of cells are always adjacent to dry cells and have small averaged water depths) can not be guaranteed because water depth in the parent cell i might not equal to the average water depth of the two child cells, when keeping uniform water surface as plotted in Figure 2(b)). On the other hand, if the mass conservation needs to be preserved at the partially wet cell, i.e. $h_i = 0.5\left(h_{i\_1} + h_{i\_2}\right)$, the same water level as that of a fully wet cell can not be ensured (Figure 2 (c)). The difference of water level will drive the static water into spurious motion. It is well-acknowledged that both the mass conservation and the C-property are of importance for an accurate and robust shallow water flow model. The mass conservation affects the water volume and in turn the water level. The C-property usually reflects the model ability to preserve the so-called well-balanced condition which means the fluxes and the slope source terms are balanced between each other and thus no spurious momentum is presented [10, 17, 19]. Avoiding dynamical adaptation, a static non-uniform grid with local refinement can actually overcome this issue.

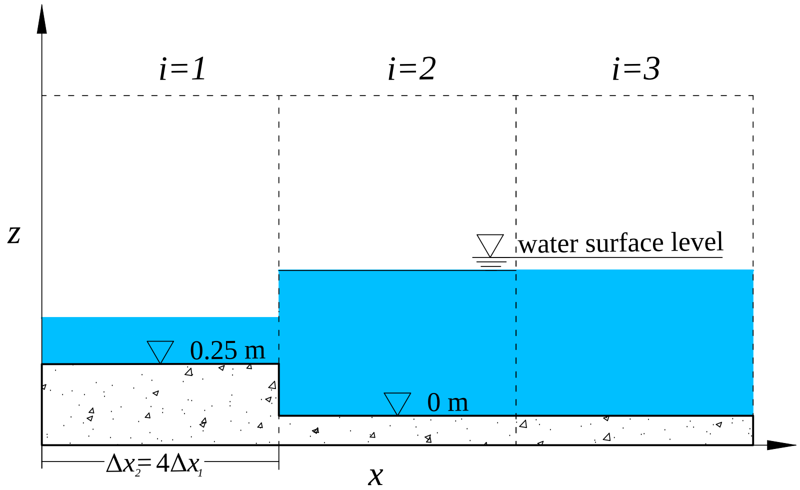


Figure 3: Spatial discretization for a floodplain with levee using a local refined grid.

The key task of generating a high-quality static non-uniform grid is to properly locate the regions of the domain where the flow dynamics is sensitive to grid resolution. Generally speaking, both the transient flow features and the domain topography may exert a demand for high-resolution solution. For example, shock-like wave fronts induced by a dam break should be captured by a high-resolution numerical scheme or mesh. This work adopts a second-order finite volume Godunov-type scheme that is able to resolve localized shock-like flow discontinuities satisfactorily on relatively coarse grids [19]. The floodplain features may be generalized into bed slope and bed friction that depends on roughness height. Friction can retard the flow but not affect the flood extent [3, 30]. While the bed elevation can not only expedite or retard the flow, but also plays a crucial role in directing the flow path and determining the final flood extent. Therefore, terrain features are indeed the most important factor affecting flow development because they can create or block flow pathways, leading to dominant effects for the whole event [44, 30]. For example, a dike could block flow and a ditch may divert the flow to other parts of the floodplain. This may have a more profound effect on the flooding process than the flow hydrodynamics itself. Therefore, if the key terrain features cannot be accurately represented, the simulation may not be reliable or even lead to misleading predictions. Figure 1 demonstrates such a case with a thin levee holding back flood, if the grid is fine enough to precisely resolve the levee, flooding will not occur. But if a coarse mesh is used, the height of the levee is likely to be underestimated and overtopping will take place and lead to flooding. Therefore, high-resolution mesh is essential to reflect the genuine bed topography of the floodplain under investigation. But refining the whole domain is not desired as it will substantially increase the number of cells used for flow calculation and, in most of the cases, coarse meshes are adequate for describing bed topography without abrupt changes, as shown in Figure 3. Therefore, mesh resolution should be adapted to accurately represent the domain features during a simulation in order to optimize the performance of the model.

To create an optimized non-uniform grid, an appropriate parameter should be proposed to indicate the bed variation. The dimensionless gradient of bed elevation is adopted in this work and is defined at an arbitrary cell ic by the following expression

$$\Theta_{ic} = \frac{G_{ic}}{P_q(\mathbf{G})}, \tag{1}$$

where G_{ic} is the averaged gradient of the bed elevation at cell ic and

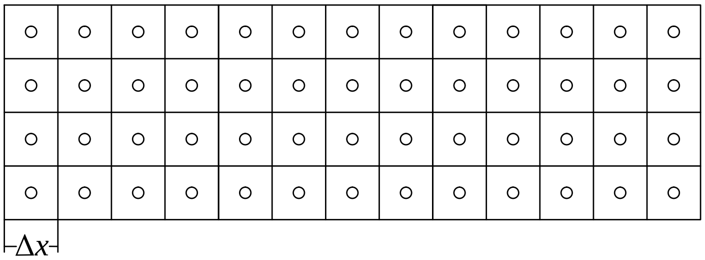


Figure 4: Initial fine grid generated from given topographic data.

$$G_{ic} = \sqrt{\left(\frac{\partial z_b}{\partial x}\right)^2_{ic} + \left(\frac{\partial z_b}{\partial y}\right)^2_{ic}}, \tag{2}$$

$\mathbf{G}$ denotes a vector containing the bed elevation gradients for all cells on a grid in descending order. $P_q(\mathbf{G})$ returns the qth percentile of $\mathbf{G}$ and $q = 1 - s_a$ with s_a indicating the sensitivity of cell refinement. The higher the value of s_a, the more cells will be refined.

In shallow water flow simulations, high-resolution topographic data (DEMs) describing the land surface of a floodplain are usually available with a relatively cheap price. To generate an optimized non-uniform Cartesian grid that accurately reflects the topographic features of the domain, the below six steps will be followed:

Step 1: An initial uniform fine grid is specified to reflect highest resolution of the topographic data available for the domain under consideration, as can be seen in Figure 4;

Step 2: Set up a dummy background grid from the lower left corner of the initial fine grid with individual cell size defined to be integer times larger than the original fine cell, as demonstrated in Figure 5. Therefore, the maximum coarsening level $levm$ is calculated by

$$levm = log_2\left(\frac{\Delta x_b}{\Delta x}\right), \tag{3}$$

where, Δx and Δx_b represent the cell size of the initial fine and the background coarse grids, respectively. If the initial grid is not divided evenly by the background grid, the remaining cells, e.g. the rightmost column of cells as sketched in Figure 5, will not take part in the local grid coarsening;

Step 3: At a background cell, the dimensionless gradients for all fine cells inside it are computed

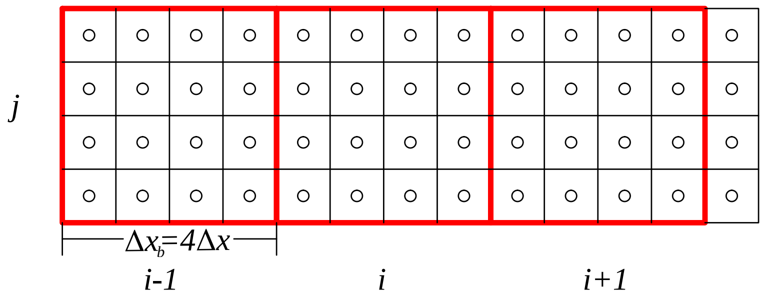


Figure 5: Defined background coarse grid with $levm = 2$.

from equation (1). Since the gradients are actually evaluated on uniform grids, the computation of the bed gradients in x- and y-directions for a cell with index of $(i,\ j)$ are undertaken as

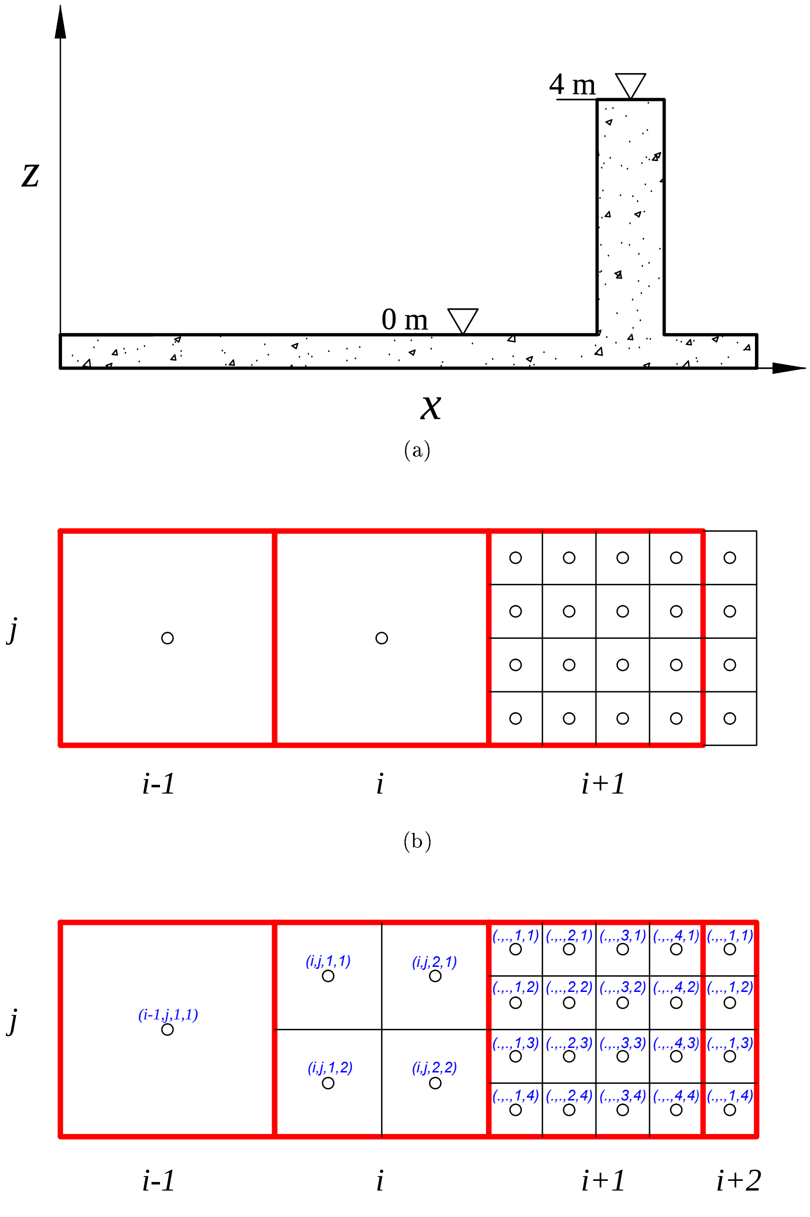
$$\frac{\partial z_b}{\partial x}_{i,j} = \max\left(\left|\frac{z_{b(i,j)} - z_{b(i-1,j)}}{\Delta x}\right|, \left|\frac{z_{b(i+1,j)} - z_{b(i,j)}}{\Delta x}\right|\right), \tag{4}$$

$$\frac{\partial z_b}{\partial y}_{i,j} = \max\left(\left|\frac{z_{b(i,j)} - z_{b(i,j-1)}}{\Delta y}\right|, \left|\frac{z_{b(i,j+1)} - z_{b(i,j)}}{\Delta y}\right|\right). \tag{5}$$

The maximum values used herein help precisely capture the possible abrupt bed;

Step 4: If all Θ_{ic} of the fine cells in a background cell are lower than 1, the fine cells are coarsened to be the same level as the background cell. Otherwise, the cells do not need to be coarsened. For example, the fine cells in the background cells $(i-1,\ j)$ and $(i,\ j)$ are coarsened to background cell level due to the small value of Θ_{ic} computed from smooth bed (Figure 6(a) and (b)). The bed elevation at the centers of the newly coarsen cells can be computed by averaging that of child cells. Therefore, the exchange data among different grids just occurs in the coarsening process through averaging the variables from child cells. Such interpolation could perfectly preserve the conservation and very straightforwardly to be implemented. However, the interpolation may slightly damp the distribution of the variable and lead to numerical diffusion which is a common numerical problem on coarse grid. Fortunately, as the area with key terrain feature which may cause complex flow pattern can be well-delimited and refined by the new grid system, together with the second order numerical scheme used, such drawback may not affect the results so much. In spite of this, storing the values at the nodes instead of cell centers may improve the computational accuracy and will be investigated in the future research;

Step 5: As in [23], a 2 : 1 rule is introduced to ensure smooth transition of cells, i.e. no cell is



(c)

Figure 6: Grid coarsening: (a) bed elevation under consideration, (b) coarsened cells, (c) grid is regulated to meet 2:1 rule.

allowed to have a neighbor that is more than two times bigger or smaller. As plotted in Figure 6(b), the cell $(i,\ j)$ does not meet the $2:1$ rule since its eastern neighbor is four times smaller. So cell $(i,\ j)$ should be regularized to be in a lower coarsening level (Figure 6(c));

Step 6: The new non-uniform Cartesian grid is then created but the grid connection information must also be specified. This current grid system does not involve a data structure to store information, which is normally used by other types of non-uniform grids such as unstructured grids and quadtree grids. Each cell on the new grid can be simply identified by four indexes, $(i,\ j,\ i_s,\ j_s)$, where i_s and j_s are integers from 1 to M_s [23, 24]. M_s denotes the number of sub-cells in x- or y-direction in background cell $(i,\ j)$ and $M_s = 2^{levm-lev}$ with lev indicating the coarsening level of $(i,\ j)$. Taking the new grid in Figure 6(c) as an example, the coarsening levels of $(i-1,\ j)$, $(i,\ j)$ and $(i+1,\ j)$ are 2, 1 and 0, respectively. On such a non-uniform grid, neighbors of an arbitrary cell are entirely determined by simple algebraic relationships. For a cell $(i,\ j,\ i_s,\ j_s)$ whose neighbors are within the same background cell, their indexes, denoted by $(i_N,\ j_N,\ i_{sN},\ j_{sN})$ with the subscript N representing 'neighbor', can be obtained easily in Table 1. If its neighbors are located in the adjacent background cells, the integer k may be '1' or '2' for an interface due to likely different coarsening levels. Their indexes $(i_{N_k},\ j_{N_k},\ i_{sN_k},\ j_{sN_k})$ for each interface can be determined by:

Table 1: Cell $(i,\ j,\ is,\ js)$'s neighbors within the same background cell $(i,\ j)$

neighbors	north	east	south	west
i_N	i	i	i	i
j_N	j	j	j	j
i_{sN}	is	$is+1$	is	$is-1$
j_{sN}	$js+1$	js	$js+1$	js

Eastern neighbors

$$\begin{aligned} i_{N_k} &= i+1, \\ j_{N_k} &= j, \\ i_{sN_k} &= 1, \\ j_{sN_k} &= Ceiling\left(j_s \times 2^{lev-lev_N}\right) - k + 1, \end{aligned} \tag{6}$$

where k is from 1 to $\max(1,\ lev - lev_N + 1)$.

Western neighbors

$$
\begin{aligned}
i_{N_k} &= i - 1, \\
j_{N_k} &= j, \\
i_{sN_k} &= Ms_N, \\
j_{sN_k} &= Ceiling\left(j_s \times 2^{lev - lev_N}\right) - k + 1.
\end{aligned} \tag{7}
$$

Northern neighbors

$$
\begin{aligned}
i_{N_k} &= i, \\
j_{N_k} &= j + 1, \\
i_{sN_k} &= Ceiling\left(i_s \times 2^{lev - lev_N}\right) - k + 1, \\
j_{sN_k} &= 1.
\end{aligned} \tag{8}
$$

Southern neighbors

$$
\begin{aligned}
i_{N_k} &= i, \\
j_{N_k} &= j - 1, \\
i_{sN_k} &= Ceiling\left(i_s \times 2^{lev - lev_N}\right) - k + 1, \\
j_{sN_k} &= Ms_N.
\end{aligned} \tag{9}
$$

After following the above six steps, the non-uniform Cartesian grid with local high-resolution cells is generated and the neighboring information for all cells is determined. Apparently, the storage requirement for the new grid is minimized by avoiding any data structure to store neighbor information, by means of using simple algebraic formulas. The grid coarsening process greatly reduces the total number of cells and thus the computational time is remarkably reduced. Since the regions of interest remain to be high-resolution, the reliability of the prediction will not deteriorate although the number of cells has been greatly reduced. This can certainly enhance model efficiency and capability for practical shallow water flow simulations in large domains.

3. Governing equations and numerical methods

3.1. Shallow water equations

In this work, a hydrodynamic inundation model is developed to numerically solving the 2D shallow water equations (SWEs) within the framework of a well-balanced finite volume Godunov-type scheme. The 2D pre-balanced SWEs derived by [25] are chosen herein as governing equations. In the pre-balanced form of the SWEs, the SWEs are reformulated and the water level η is applied as a main variable instead of the water depth, so as to easier to achieve well-balanced state.

$$\frac{\partial \mathbf{q}}{\partial t}+\frac{\partial \mathbf{f}}{\partial x}+\frac{\partial \mathbf{g}}{\partial y}=\mathbf{S}, \tag{10}$$

$$\mathbf{q}=\begin{bmatrix}\eta\\ q_x\\ q_y\end{bmatrix},\ \mathbf{f}=\begin{bmatrix}uh\\ u^2h+g\left(\eta^2-2\eta z_b\right)/2\\ uvh\end{bmatrix},\ \mathbf{g}=\begin{bmatrix}vh\\ vuh\\ v^2h+g\left(\eta^2-2\eta z_b\right)/2\end{bmatrix}, \tag{11}$$

$$\mathbf{S}=\mathbf{S}_b+\mathbf{S}_f=\begin{bmatrix}0\\ -g\eta\partial z_b/\partial x\\ -g\eta\partial z_b/\partial y\end{bmatrix}+\begin{bmatrix}0\\ -C_f u\sqrt{u^2+v^2}\\ -C_f v\sqrt{u^2+v^2}\end{bmatrix}, \tag{12}$$

where t denotes the time; x and y represent the Cartesian coordinates; $\mathbf{q}$ is the vector of conserved flow variables containing η, q_x and q_y which are the water level, and unit-width discharges in the x- and y-directions, respectively; $q_x = uh$ and $q_y = vh$ with h, u and v being the water depth and depth-averaged velocities in the x- and y-directions, respectively; z_b is the bed elevation and $\eta = z_b + h$; $\mathbf{f}$ and $\mathbf{g}$ are the flux vectors in the x- and y-directions, respectively; $\mathbf{S}$ represents the source vector that can be split into slope source terms $\mathbf{S}_b$ and friction source terms $\mathbf{S}_f$; C_f is the bed roughness coefficient evaluated herein by $gn^2/h^{1/3}$ with n being the Manning coefficient.

3.2. Numerical scheme

On the proposed new grid system, the SWEs are solved numerically by adopting a two-step MUSCL-Hancock scheme originally developed in [42], within the framework of a cell-centered finite volume Godunov-type scheme as in [23, 24, 38]. It consists of a predictor step and a corrector step. In the predictor step, intermediate flow variables are predicted over half of a time interval. These intermediate variables are then utilized in the corrector step to update the results to a new time level. The friction source terms $\mathbf{S}_f$ are not evaluated within the MUSCL-Hancock scheme but independently evaluated by a splitting point-implicit method proposed in [27]. Since the numerical scheme is overall explicit, the solution stability is essentially controlled by the Courant-Friedrichs-Lewy (CFL) condition, which is used to estimate time step for next iteration during a simulation. Open and closed boundaries are implemented, as in [25]. This section is devoted to present MUSCL-Hancock scheme for solving the SWEs on the proposed non-uniform Cartesian grids. A more effective way to reconstruct Riemann states and a simpler approach to maintain the C-property are also introduced. It should be noted that the new grid system proposed in this work is independent of the numerical solver and therefore valid for other types of numerical schemes.

3.2.1. Predictor step

In this step, the intermediate flow variables at the center of an arbitrary cell ic are computed over half of a time interval using the following formula

$$\begin{aligned}\mathbf{q}_{ic}^{pre} &= \mathbf{q}_{ic}^{n+1/2} = \mathbf{q}_{ic}^{n} - \frac{\Delta t}{2\Delta x_{ic}}\left[\mathbf{f}\left(\mathbf{q}_E^-\right) - \mathbf{f}\left(\mathbf{q}_W^+\right)\right] \\ &- \frac{\Delta t}{2\Delta y_{ic}}\left[\mathbf{g}\left(\mathbf{q}_N^-\right) - \mathbf{g}\left(\mathbf{q}_S^+\right)\right] + \frac{\Delta t}{2}\mathbf{S}_{ic}^{k}, \end{aligned} \tag{13}$$

where the superscript pre denotes predictor step and n stands for the time level; $+$ and $-$ represent the right and left hand side of a cell interface, respectively; the subscripts E, W, N and S indicate the east, west, north and south interfaces of cell ic; $\mathbf{q}_E$, $\mathbf{q}_W$, $\mathbf{q}_N$ and $\mathbf{q}_S$ represent the values of flow variables extrapolated at the midpoints of the corresponding interface which are on the inner side of the cell under consideration and can be computed using the MUSCL linear reconstruction to give second-order accuracy in space. For example, at eastern interface of cell ic, the face values $\mathbf{q}_E^-$ are given by

$$\mathbf{q}_E^- = \mathbf{q}_{ic}^n + \frac{\Delta x_{ic}}{2}\nabla\bar{\mathbf{q}}_{x,ic}^n, \tag{14}$$

where $\nabla\bar{\mathbf{q}}_{ic,x}^n$ denotes the vector of the limited gradients of flow variables in the x-direction. In this work, the Minmod limiter which is well-known for its stability is applied to compute the limited gradients from the unlimited upwind and central gradients, see [22]. To better preserve the conservation property (C-property) as proposed by Bermudez and Vazquez [4], the water depth at interface is also extrapolated from equation (14) and the bed elevation is accordingly evaluated by

$$z_{bE}^- = \eta_E^- - h_E^-. \tag{15}$$

The face values obtained from equation (14) are used by equation (11) to compute the fluxes. It should be noted that for dry cells with wet neighbors and wet cells with dry neighbors, piecewise constant values are adopted, i.e. first order scheme is used and thus $\mathbf{q}_E^- = \mathbf{q}_{ic}^n$, with the aim of avoiding numerical instability. All the dry cells that do not neighbor to a wet cell are not necessary to be included in the actual calculation. In this work, a cell is classified as dry when it has a water depth smaller than a wet-dry tolerance $\varepsilon_d = 10^{-6}$m.

The slope source terms are computed as in [27], e.g. in the x-direction

$$\mathbf{S}_{bx,ic}= -g\eta_{ic}\frac{z_{bE}^- - z_{bW}^+}{\Delta x_{ic}}, \tag{16}$$

where $\eta_{ic} = \left(\eta_E^- + \eta_W^+\right)/2$. In a similar way, the slope source term in the y-direction can be evaluated.

3.2.2. Corrector step

In the corrector step, based on the predicted intermediate values, the flow variables are updated to a new time level using a fully conservative scheme:

$$\mathbf{q}_{ic}^{n+1} = \mathbf{q}_{ic}^n - \frac{\Delta t}{\Delta x_{ic}}\left[\mathbf{f}_E^{cor} - \mathbf{f}_W^{cor}\right] - \frac{\Delta t}{\Delta y_{ic}}\left[\mathbf{g}_N^{cor} - \mathbf{g}_S^{cor}\right] + \Delta t\mathbf{S}_{ic}^{cor}, \tag{17}$$

where the superscript *cor* represents the corrector step.

The Harten, Lax and Leer approximate Riemann solver with contact wave restored, i.e. HLLC approximate Riemann solver as proposed by Toro et al. [40], is applied in this work to compute the interface fluxes. Taking the fluxes at eastern cell interface as an example, $\mathbf{f}_E^{cor}$ can be evaluated by seeking the solution to a local Riemann problem R defined by the left and right Riemann states of the flow variables:

$$\mathbf{f}_E^{cor} = R(\bar{\mathbf{q}}_E^-, \bar{\mathbf{q}}_E^+), \tag{18}$$

in which, the Riemann states $\bar{\mathbf{q}}_E^-$ and $\bar{\mathbf{q}}_E^+$ are obtained from the following extrapolation and reconstruction steps. Non-uniform grids make the extrapolation become troublesome due to the possible difference in cell dimensions, for instance, cell ic as shown in Figure 7 has two eastern neighbors that are half of its own size. Under this circumstance, the fluxes into the smaller cells are first computed and the final fluxes as required can be simply obtained by

$$\mathbf{f}_E^{cor} = 0.5\left(\mathbf{f}_{E_1}^{cor} + \mathbf{f}_{E_2}^{cor}\right), \tag{19}$$

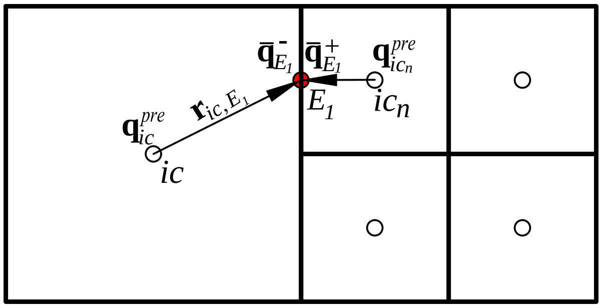

$\bar{\mathbf{q}}^-_{E_1}$
$\bar{\mathbf{q}}^+_{E_1}$
$\mathbf{q}^{pre}_{ic_n}$
$\mathbf{r}_{ic,E_1}$
E_1
ic_n
$\mathbf{q}^{pre}_{ic}$
ic

Figure 7: Extrapolation for the flow variables at an interface on non-uniform grids.

where $\mathbf{f}^{cor}_{E_1} = R(\bar{\mathbf{q}}^-_{E_1}, \bar{\mathbf{q}}^+_{E_1})$ and $\mathbf{f}^{cor}_{E_2} = R(\bar{\mathbf{q}}^-_{E_2}, \bar{\mathbf{q}}^+_{E_2})$. The extrapolation and reconstruction steps are demonstrated in detail as how to obtain $\bar{\mathbf{q}}^-_{E_1}$ and $\bar{\mathbf{q}}^+_{E_1}$.

In Figure 7, the values beside point E_1 can be computed along the interface from the extrapolated values at the midpoint of faces on the inner sides of cells [23, 24]. But different dimensions of the related faces demand a complex algorithm. In this work, the interface values are extrapolated directly from the cell centers by using an approach developed for unstructured grids (Hou et al. [17, 19, 18]). This approach requires only the limited gradients for each cell and the coordinates for the points. Since the limited gradients of flow variables in the cell have been calculated from the predictor step, it is convenient to use these values of gradients directly in the corrector step. For example, the values at the left and the right hand side of a cell interface as shown in Figure 7 can be obtained by

$$\bar{\mathbf{q}}^-_{E_1} = \mathbf{q}^{pre}_{ic} + \nabla\bar{\mathbf{q}}_{ic} \cdot \mathbf{r}_{ic,E_1},\, \bar{\mathbf{q}}^+_{E_1} = \mathbf{q}^{pre}_{ic_n} + \nabla\bar{\mathbf{q}}_{ic_n} \cdot \mathbf{r}_{ic_n,E_1}, \tag{20}$$

where, $\mathbf{q}^{pre}_{ic}$ and $\mathbf{q}^{pre}_{ic_n}$ are the intermediate flow variables predicted over half of a time interval at the centers of two adjacent cells ic and ic_n, respectively; $\nabla\bar{\mathbf{q}}_{ic}$ denotes the vector of limited gradients of cell ic and $\nabla\bar{\mathbf{q}}_{ic} = [\nabla\bar{\mathbf{q}}_{ic,x}, \nabla\bar{\mathbf{q}}_{ic,y}]^T$ with $\nabla\bar{\mathbf{q}}_{ic,x}$ and $\nabla\bar{\mathbf{q}}_{ic,y}$ being the limited gradients in the x- and y-directions, respectively and are obtained in the predictor step; $\mathbf{r}_{ic,E_1}$ is the position vector from the cell center to the point under consideration, i.e. E_1 at the interface. As in the predictor step, the face values of water depth $\bar{h}^-_{E_1}$ and $\bar{h}^+_{E_1}$ are extrapolated in a similar way while the face values of bed elevation are computed by subtracting water depth from the extrapolated water surface level, i.e.

$$\bar{z}^-_{bE_1} = \bar{\eta}^-_{E_1} - \bar{h}^-_{E_1},\, \bar{z}^+_{bE_1} = \bar{\eta}^+_{E_1} - \bar{h}^+_{E_1}. \tag{21}$$

Once the face values of discharges and the water depths are obtained, the corresponding velocity components can be calculated from

$$\begin{aligned} &\bar{u}^-_{E_1} = \bar{q}^-_{xE_1}/\bar{h}^-_{E_1},\ \bar{v}^-_{E_1} = \bar{q}^-_{yE_1}/\bar{h}^-_{E_1}, \\ &\bar{u}^+_{E_1} = \bar{q}^+_{xE_1}/\bar{h}^+_{E_1},\ \bar{v}^+_{E_1} = \bar{q}^+_{yE_1}/\bar{h}^+_{E_1}. \end{aligned} \tag{22}$$

To maintain numerical stability, if the extrapolated water depth is lower than the wet-dry tolerance ε_d, face values of velocity are directly imposed to be zero.

In the next step, the flow variables computed from equations (20) and (21) should be modified by a so-called non-negative water depth reconstruction proposed by Audusse and Bristeau [1]. Apart from ensuring non-negative water depth, the reconstruction also easily preserves the C-property when combining with the governing shallow water equations adopted in this work. The reconstruction can be carried out as

$$\begin{aligned} &\bar{h}^-_{E_1} = \max\left(0,\ \bar{\eta}^-_{E_1} - \bar{z}_{bE_1}\right), \\ &\bar{h}^+_{E_1} = \max\left(0,\ \bar{\eta}^+_{E_1} - \bar{z}_{bE_1}\right), \end{aligned} \tag{23}$$

where

$$\bar{z}_{bE_1} = \max\left(\bar{z}^-_{bE_1},\ \bar{z}^+_{bE_1}\right). \tag{24}$$

A modified approach as proposed in (Hou et al., 2013) is adopted herein to provide an easier way for satisfying the C-property at wet-dry interfaces than that used in [1, 22, 23, 43, 24],

$$\bar{z}_{bE_1} \leftarrow \min(\bar{z}_{bE_1},\ \bar{\eta}^-_{E_1}). \tag{25}$$

Accordingly, the face values of water level and discharge are reconstructed by

$$\begin{aligned} &\bar{\eta}^-_{E_1} = \bar{\eta}^-_{E_1},\ \ \bar{q}^-_{xE_1} = \bar{h}^-_{E_1}\bar{u}^-_{E_1},\ \ \bar{q}^-_{yE_1} = \bar{h}^-_{E_1}\bar{v}^-_{E_1}, \\ &\bar{\eta}^+_{E_1} = \bar{h}^+_{E_1} + \bar{z}_{bE_1},\ \ \bar{q}^+_{xE_1} = \bar{h}^+_{E_1}\bar{u}^+_{E_1},\ \ \bar{q}^+_{yE_1} = \bar{h}^+_{E_1}\bar{v}^+_{E_1}. \end{aligned} \tag{26}$$

The reconstructed water levels and discharges are finally employed by the HLLC approximate Riemann solver (equation (18)) to evaluate fluxes.

Similarly, $\bar{\mathbf{q}}_{E_2}^-$ and $\bar{\mathbf{q}}_{E_2}^+$can be obtained and used to compute $\mathbf{f}_{E_2}^{cor}$. In the corrector step, the slope source terms are discretized in a similar way to the predictor step (equation (16)). For example, the slope source term in the x-direction is calculated by

$$\mathbf{S}_{ic,x}^{cor} = -g\bar{\eta}_{ic,x}\frac{\bar{z}_{bE} - \bar{z}_{bW}}{\Delta x_{ic}}, \tag{27}$$

where $\bar{\eta}_{ic,x} = \left(\bar{\eta}_E^- + \bar{\eta}_W^+\right)/2$, and $\bar{z}_{bE}$ and $\bar{z}_{bW}$ are calculated from equation (25). If a non-uniform grid is involved, e.g. cells ic in Figure 7, $\bar{\eta}_{ic,x}$ and $\bar{z}_{bE}$ are obtained by taking the averaged values from the two smaller cells:

$$\bar{\eta}_E^- = \frac{\bar{\eta}_{E_1}^- + \bar{\eta}_{E_2}^-}{2}, \ \ \bar{z}_{bE}^- = \frac{\bar{z}_{bE_1}^- + \bar{z}_{bE_2}^-}{2}. \tag{28}$$

The slope source term in the y-direction can be computed in a similar way.

4. Test cases

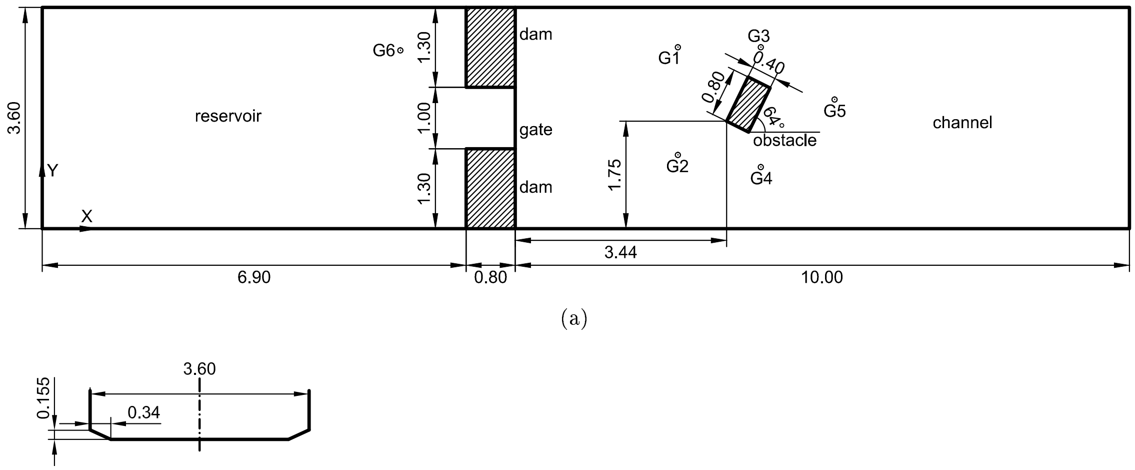
In this section, the performance of the current non-uniform grid based shallow water flow model is demonstrated by applying it to simulate three benchmark tests involving flow over uneven beds in experimental and field scales. The results are compared with those obtained from alternative models that feature with identical numerical schemes but on coarse and fine uniform grids. A Courant number of 0.5 is used in all the tests to enable adaptive time steps.

4.1. Dam-break flow interrupted by a single block

A physical model as shown in Figure 8 was set up in Université Catholique de Louvain, Belgium [39] to investigate the effects of a single building block on the propagation of dam-break induced flood waves. The presence of the building leads to an abrupt change of bed elevation, which may considerably affect the flow hydrodynamics. It should be noted that the block is high enough and therefore overtopping won't happen. This provides an ideal test case to validate the capability of the proposed non-uniform grid system in capturing local topographic features and providing reliable predictions. The grid is created by the new non-uniform grid system with $s_a = 0.2$ (Figure 9). Three levels of cells are used, with their size ranging from 0.02 m to 0.08 m, and a total of 91,122 cells are generated. Obviously, local high resolution is preserved around the block and the side walls. Simulations are also performed on uniform grids of resolution 0.02 m by 0.02 m (fine grid) and 0.04 m by 0.04 m (coarse grid) for the purpose of comparison. As suggested in [39], the Manning coefficient of 0.01 sm$^{-1/3}$ is imposed throughout the whole domain. The water level at the reservoir and the downstream floodplain are initialized as 0.4 m and 0.02 m above the horizontal bed. The simulations are run for $t = 25\,\mathrm{s}$.

Figure 10 compares the computed water depths in the domain on different grids. The flow patterns predicted on the new grid and the fine grid are observed to be very similar in both of the reservoir and floodplain. The results on the coarse grid are found to be a little more diffusive. Near the block, the disturbance to the flow caused by staircase approximation of boundary is evident. Different depth pattern in the reservoir is also predicted on the coarse grid, compared to the other two grids. During the simulations, the time histories of water level and velocity are recorded on three gauge points as sketched in Figure 8, which are located at the left and right hand sides of the block and in the reservoir, respectively. The time histories are presented in Figures 11, 12 and 13 and compared with laboratory measurements, respectively. The water levels and velocities computed on the new and fine grids agree generally well with the measurements at all gauges. To quantitatively assess the performance of the new grid system, the RMSE (Root Mean Square Error) of the water level evolutions at gauges G3 and G4 on three kinds of grids are computed. At G3, the RMSEs of fine, coarse and new grid systems are 0.0214, 0.0163 and 0.0195, respectively. Those at G4 are 0.0205, 0.0190 and 0.0198, respectively. It is verified again that the proposed new grid system has relatively high accuracy in simulating shallow flow simulations. Though the RMSE for the coarse grid at G3 is lower than other grids, the computed water level is lower than the measured ones, implying unsafe flood predictions. Slightly higher water level, which is of the most importance in shallow water flow modeling for safe flood prediction, is well-reproduced by the model on either the new non-uniform grid or the fine grids. As a result of the raised water level, the velocity becomes lower than that on coarse grid especially for that in the x direction. It is a kind of sideeffect when using 2D shock capturing schemes simulating actual 3D flows with high turbulence.

Compared with the one run on the fine grid, the non-uniform grid based model is 3.48 times as efficient due to the use of much less grid cells. Therefore, the current model offers a promising option for efficient shallow water flow simulations.



(a)

(b)

Figure 8: Dam-break flow against a single block: (a) set-up of physical model and locations of measurement gauges G1-G6, (b) cross section of the channel. (m)

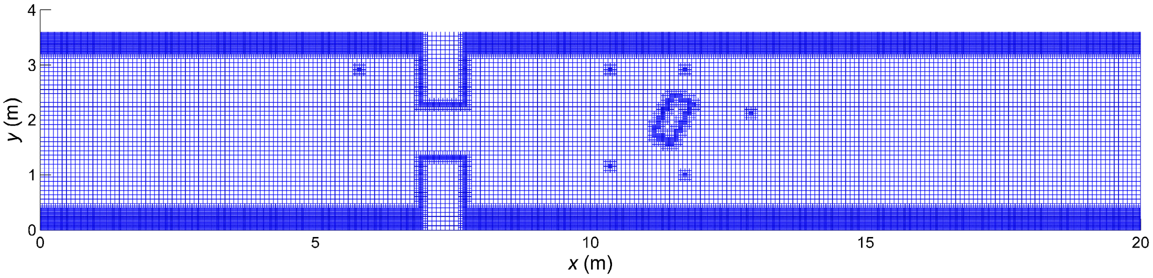
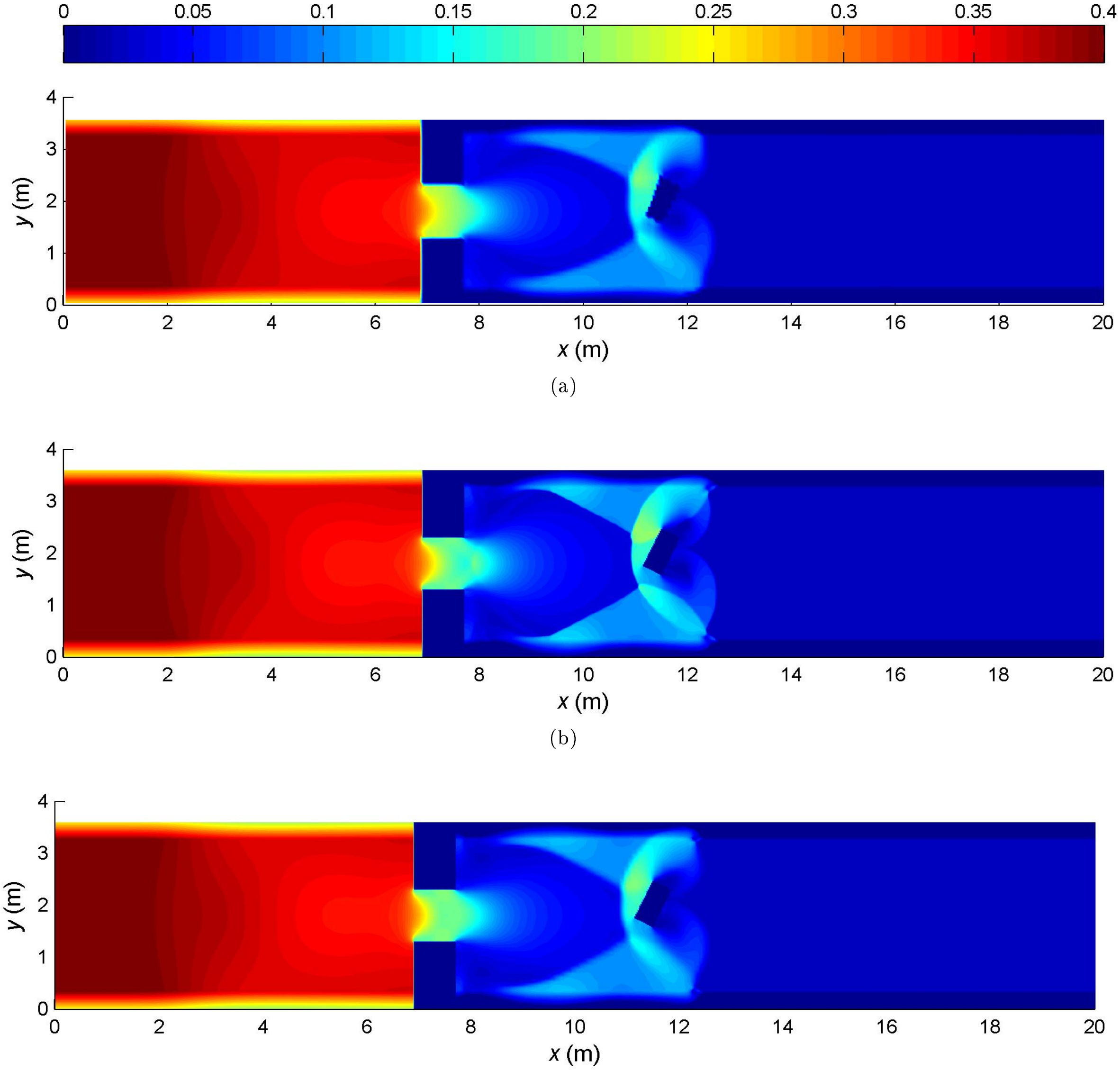


Figure 9: Dam-break flow against a single block: computational grid generated by the new grid system.

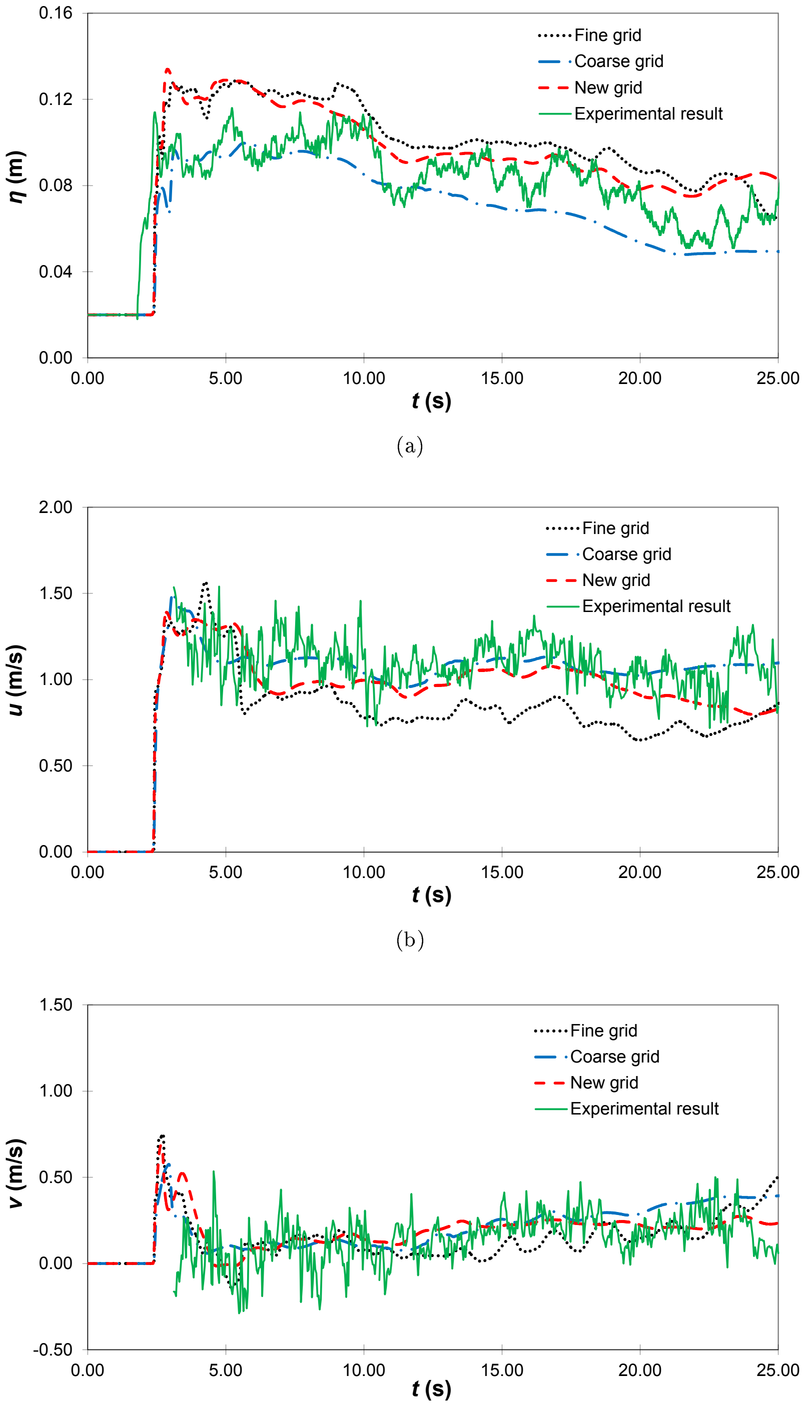


(a)

(b)

(c)

Figure 10: Dam-break flow against a single block: computed water depth at $t = 3\,\mathrm{s}$ on: (a) coarse grid, (b) fine grid, (c) non-uniform grid. (Unit: m).

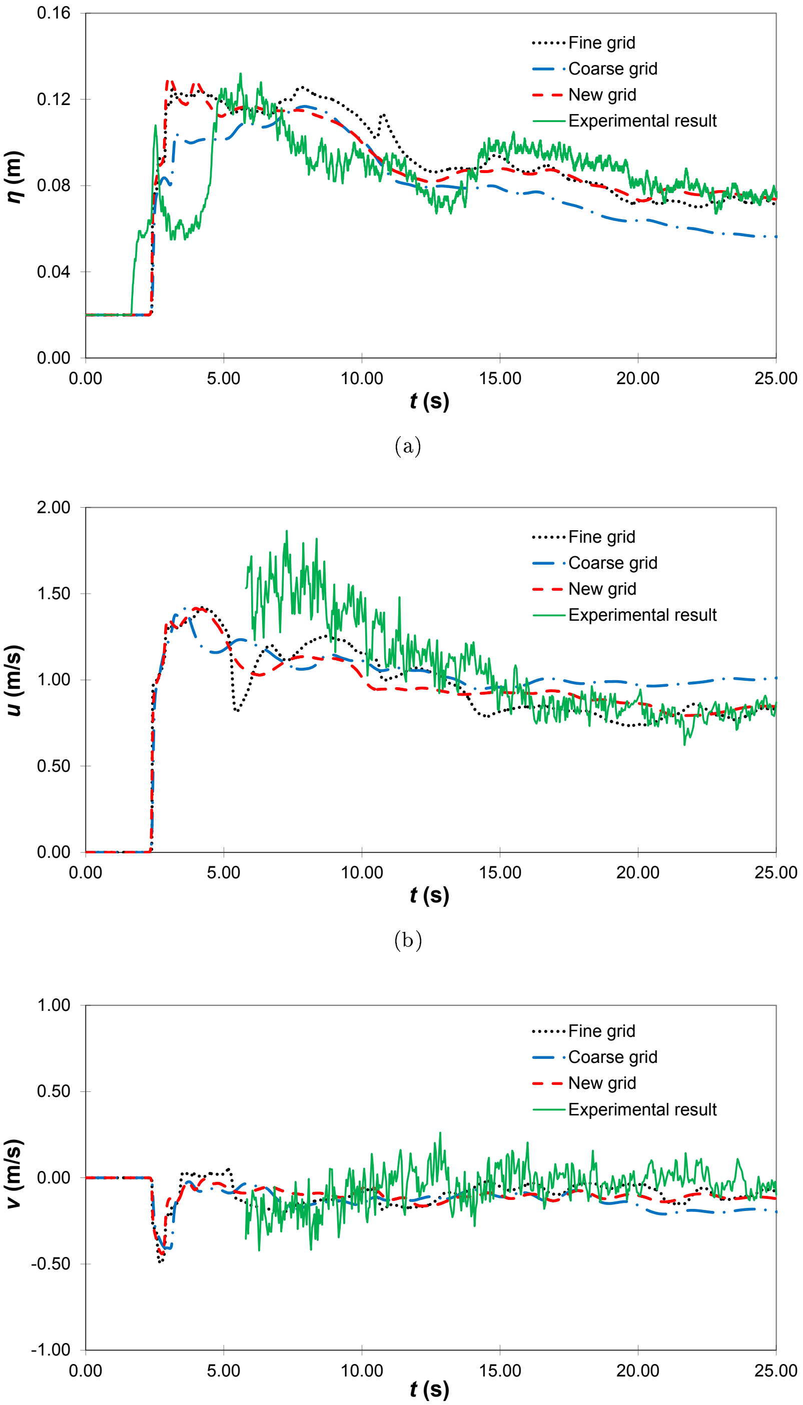


(a)

(b)

(c)

Figure 11: Dam-break flow against a single block: computed and experimental results at gauge G3: (a) water level, (b) velocity in x-direction, (c) velocity in y-direction.



(a)

(b)

(c)

Figure 12: Dam-break flow against a single block: computed and experimental results at gauge G4: (a) water level, (b) velocity in x-direction, (c) velocity in y-direction.

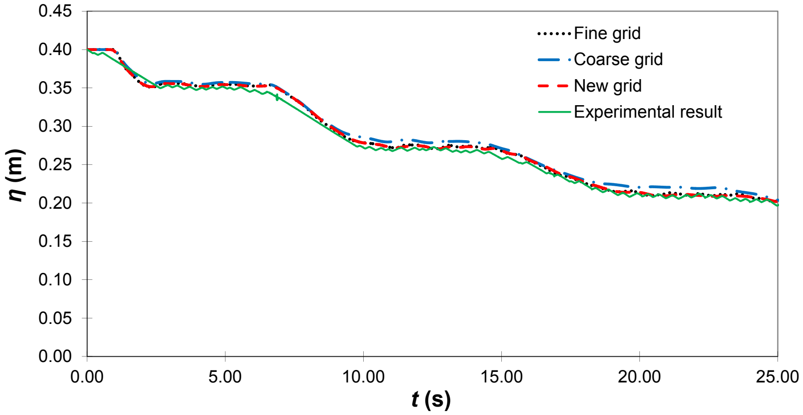


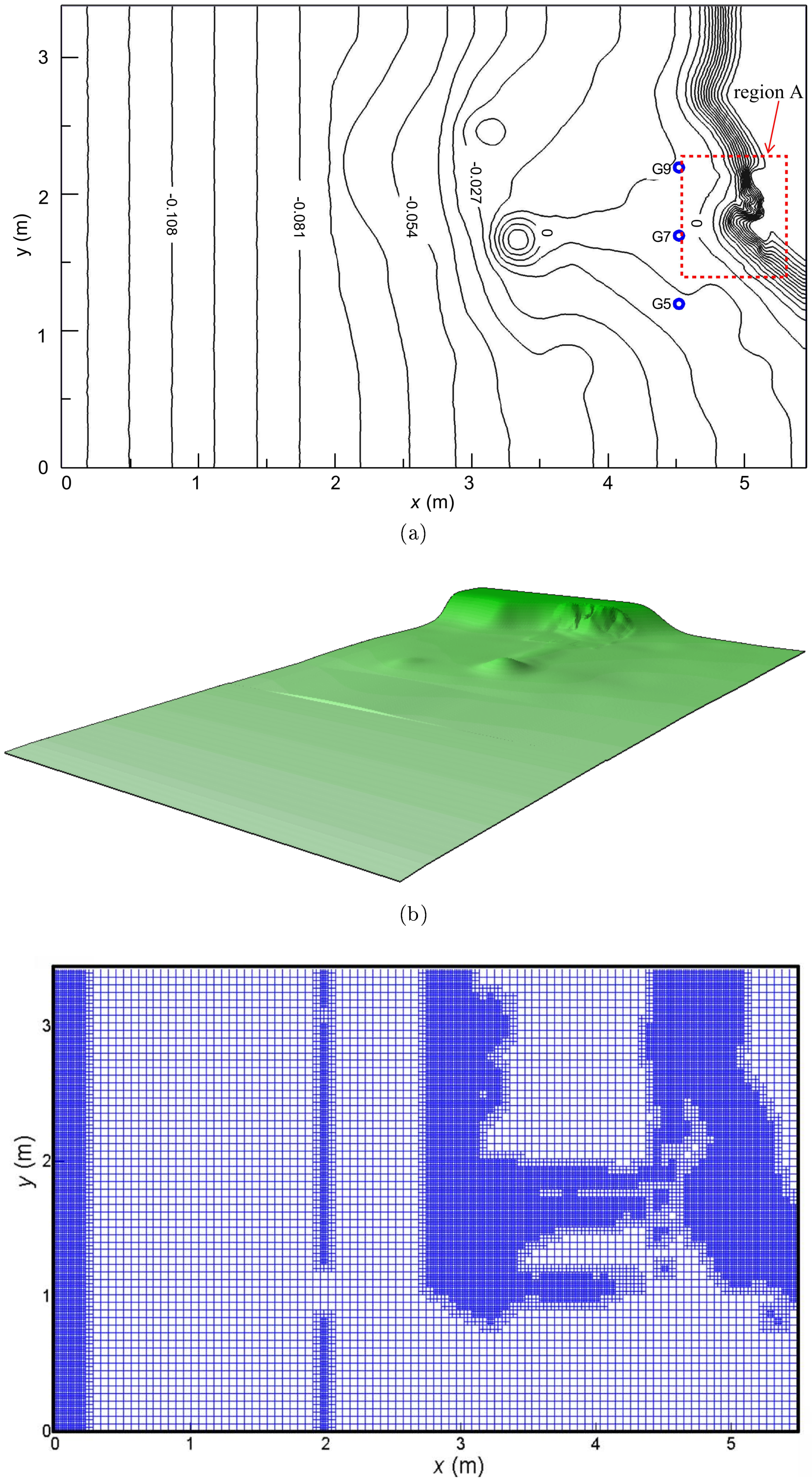
Figure 13: Dam-break flow against a single block: numerical and experimental results of water level at gauge G6.

4.2. Tsunami in Monai valley

Monai valley, located in the south-west of Okushiri island, Japan, was attacked by a severe tsunami in 1993, causing a run-up of more than 30 m. This event was reproduced in a laboratory by a 1: 400 physical model in the research institute for electric power industry (CRIEPI) in Abiko, Japan [28]. Occurred in a domain with irregular bathymetric and topographic features, this provides another ideal benchmark test case to assess the capability of the current non-uniform grid based model for efficient simulation of complex shallow flow hydrodynamics. The computational domain and the bed elevations are sketched in Figures 14 (a) and (b) in 2D and 3D view, respectively. The locally refined computational grid is generated with $s_a = 0.2$ and is plotted in Figure 14 (c). The areas with abrupt change of bed elevation are covered by fine grid cells of 0.014 m by 0.014 m while the rest of the domain is coarsened to two levels higher. A fine and a coarse uniform grids with resolution of 0.014m and 0.028 m are also adopted for comparative simulations. The experimental tsunami is numerically reproduced by imposing at the left domain boundary an incident wave as indicated in Figure 15. Uniform Manning coefficient of 0.001 $\mathrm{sm}^{-1/3}$ is used throughout the whole domain as suggested by other researchers [32, 8].

The numerically predicted wave run-up and run-down are illustrated in Figure 16 at $t = 14\,\mathrm{s}$, $16\,\mathrm{s}$ and $18\,\mathrm{s}$. The tsunami wave that batters the island is clearly predicted by the models using the fine grid and the proposed non-uniform grid, both producing very similar results. The predicted and measured maximum run-ups, occurring at about $t = 16.9\,\mathrm{s}$, are plotted in Figure 17 by means of identifying wet-dry fronts. The wet-dry fronts predicted on the fine and the current non-uniform grids agree well with each other and the experimental records. However, the simulation on coarse grid fails to capture the wet-dry fronts accurately and the flow fronts in the two small valleys are apparently different from the measurements and predictions on other grids. Moreover, the maximum run-up computed by the non-uniform and fine grids are recorded to be 30.6 m which matches closely the field survey value of 30 m. In addition, Figure 18 compares the measured and computed water levels at the selected gauges as indicated in Figure 14(a). At all of gauges, the time histories of water level predicted on the current non-uniform grid are well-aligned with those produced on the fine grid. However, those predicted on the coarse grid are observed to deviate from those resulting from other simulations. Satisfactory agreement is also achieved between the measured water level and the ones computed on the fine and the new non-uniform grids. The RMSEs of the computed water level on the fine, coarse and the new grid systems are 0.00380, 0.00416 and 0.00386, respectively at G5, 0.00362, 0.00391, 0.00367, respectively at G7, 0.00358, 0.00411 and 0.00363, respectively at G9. It is indicated quantitatively the proposed new grid system could reliably reproduce shallow water wave propagation. However, as plotted in Figure 18, the computed water level is not so sensitive to the grid resolution as those in last bam-break test case, the reason might be the flow pattern in this test case is not so discontinuous as the dam-break problem which is very sensitive to the grid resolution. While providing predictions of similar accuracy, the simulation on the new non-uniform grid is computationally 1.93 times more efficient than the one on the fine grid.

The sensitivity of the new non-uniform grid based model to the selection of s_a is also tested in this case. The predicted time histories of water level at the aforementioned gauges are plotted in Figure 19, with s_a varying from 0.1 to 0.4. No distinctive difference among the results of different s_a is observed, indicating that the model is not essentially very sensitive to s_a with a value higher than 0.1. Since higher s_a means more cells maintain to be fine and consequently higher computational effort, an appropriate s_a is required for trading off the computational accuracy and efficiency. $s_a = 0.2$ is proposed in this work for the new non-uniform grid system, following numerical experiments.



(a)

(b)

(c)

Figure 14: Tsunami in Monai valley: (a) 2D bathymetry, (b) 3D bathymetry, (c) computational grid created by the new grid system. (Unit: m).

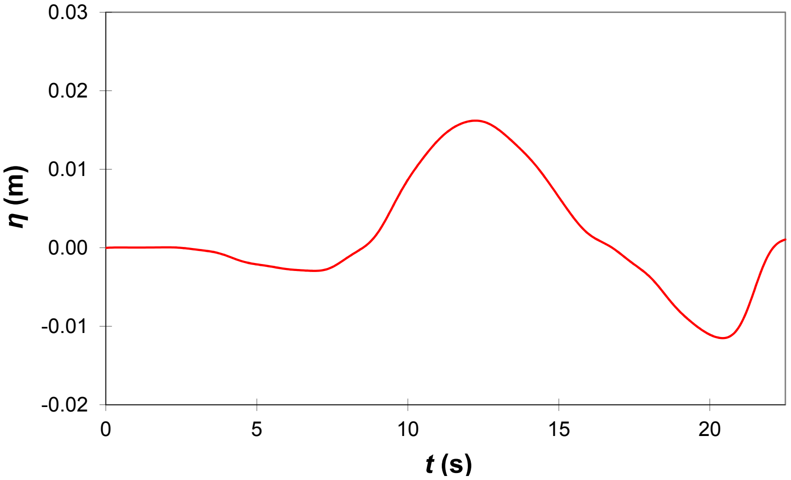


Figure 15: Tsunami in Monai valley: inflow boundary condition.

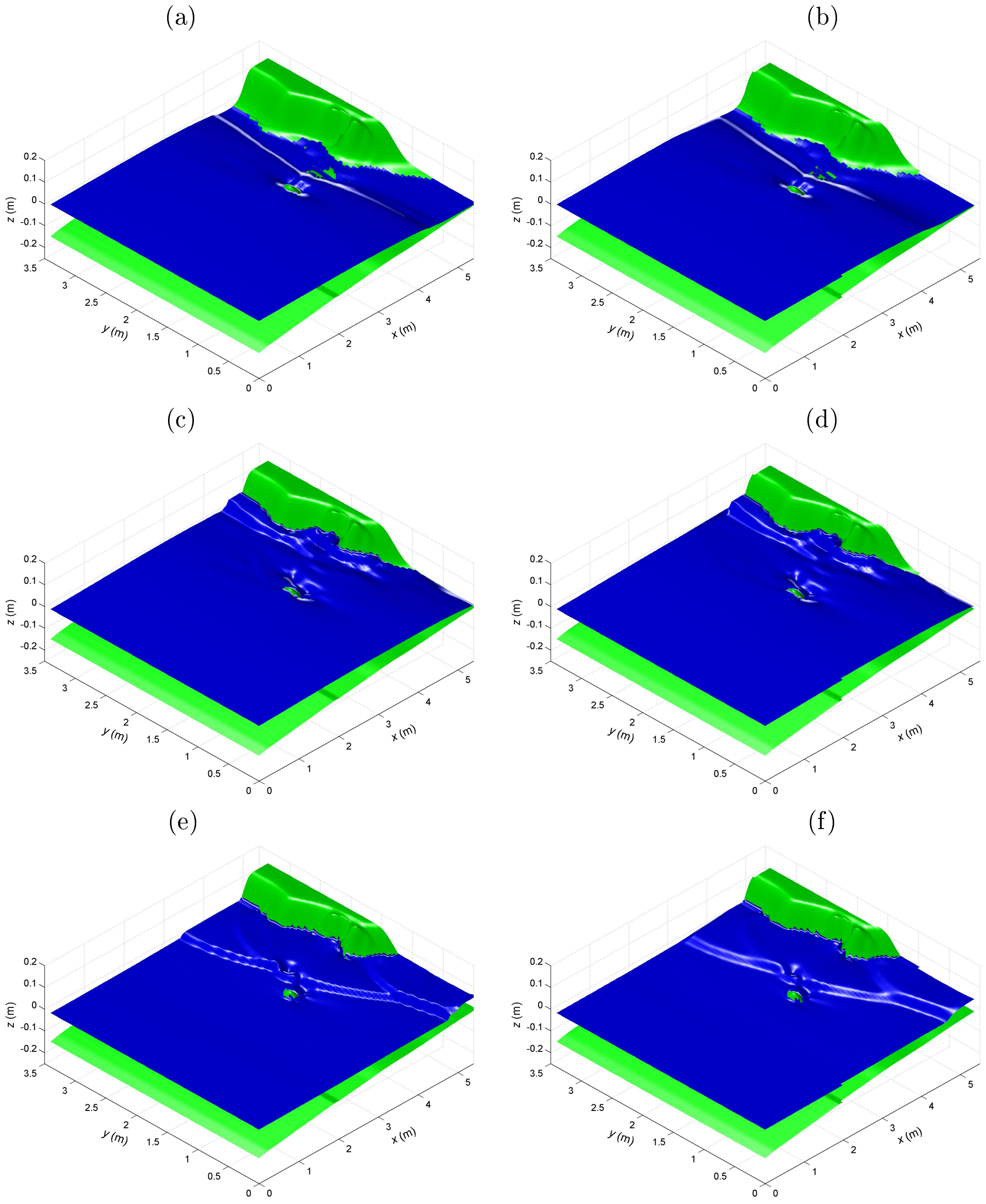
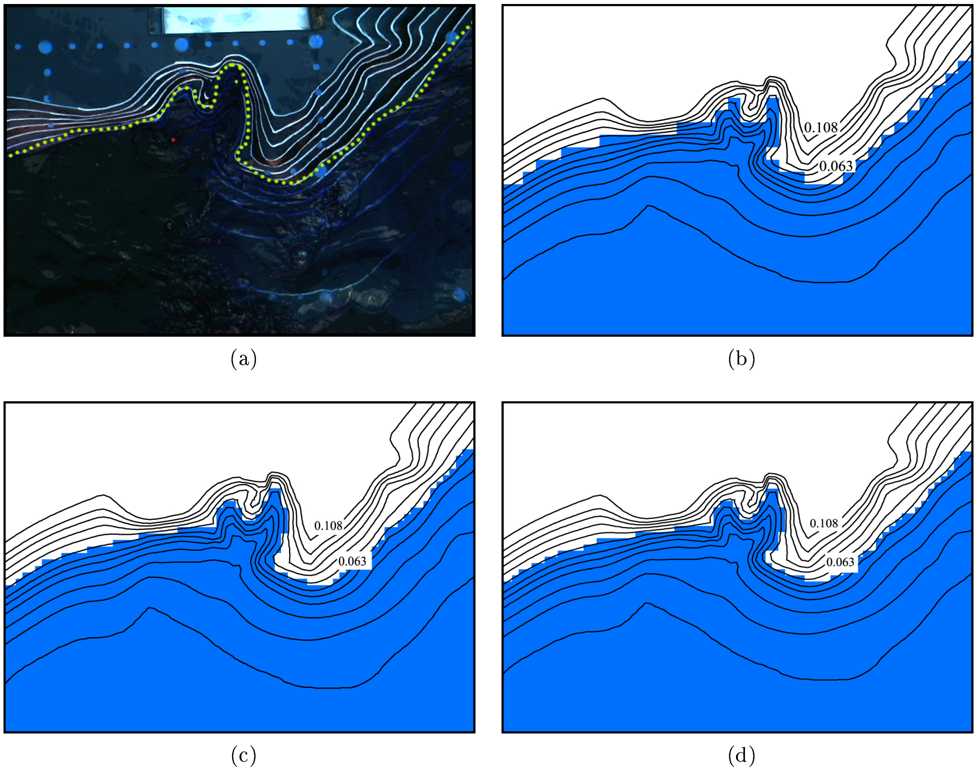


Figure 16: Tsunami in Monai valley: computed wave run-up and run-down on: (a) the fine grid at $t = 14\,\mathrm{s}$, (b) the new grid at $t = 14\,\mathrm{s}$, (c) the fine grid at $t = 16\,\mathrm{s}$, (d) the new grid at $t = 16\,\mathrm{s}$, (e) the fine grid at $t = 18\,\mathrm{s}$, (f) the new grid at $t = 18\,\mathrm{s}$.



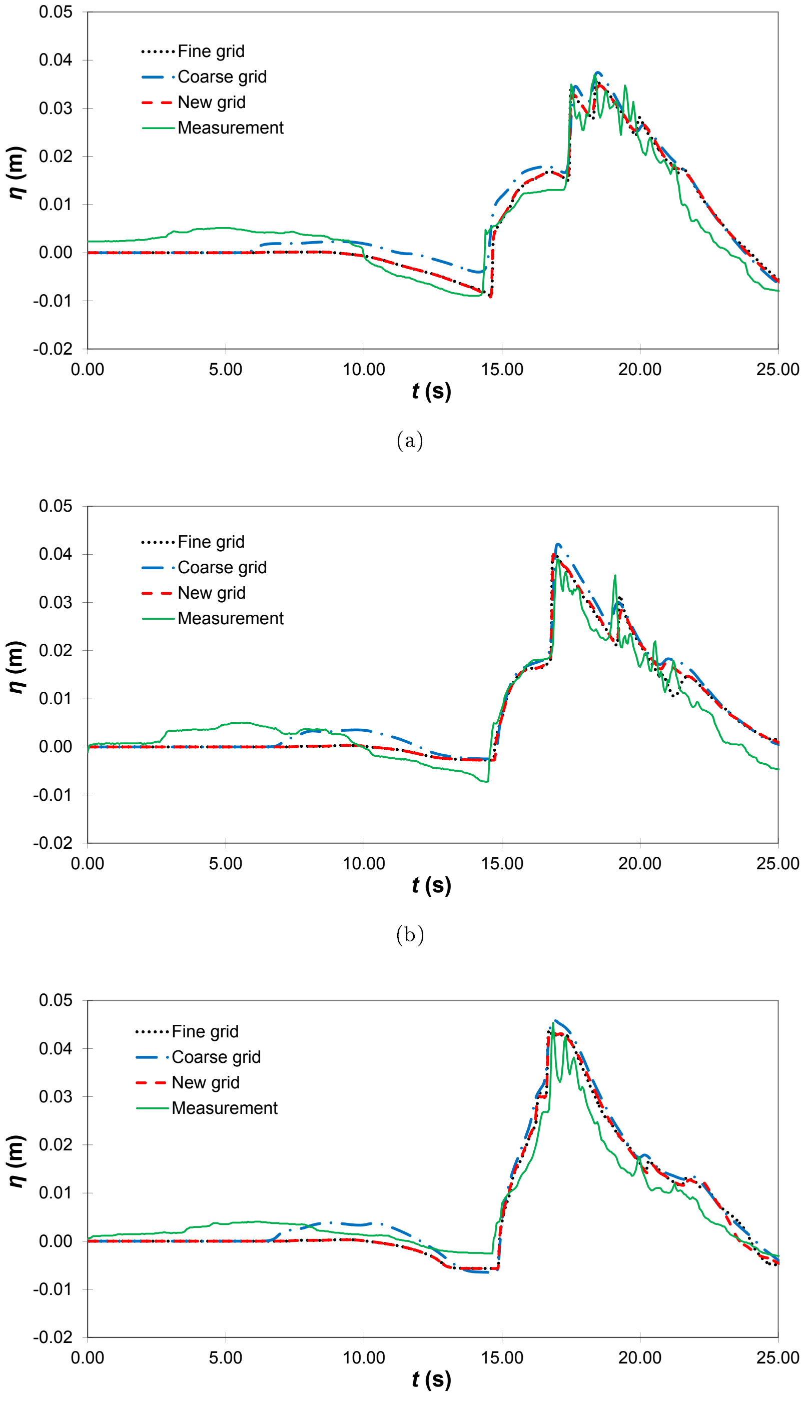
(a)

(b)

(c)

(d)

Figure 17: Tsunami in Monai valley: (a) snapshot of experimental run-up at $t = 16.9\,\mathrm{s}$; (b) computed run-up on the coarse grid at $t = 16.9\,\mathrm{s}$; (c) computed run-up on the fine grid at $t = 16.9\,\mathrm{s}$; (d) computed run-up on the new grid at $t = 16.9\,\mathrm{s}$. The area plotted is the region A shown in Figure 14(a). . (Unit: m).

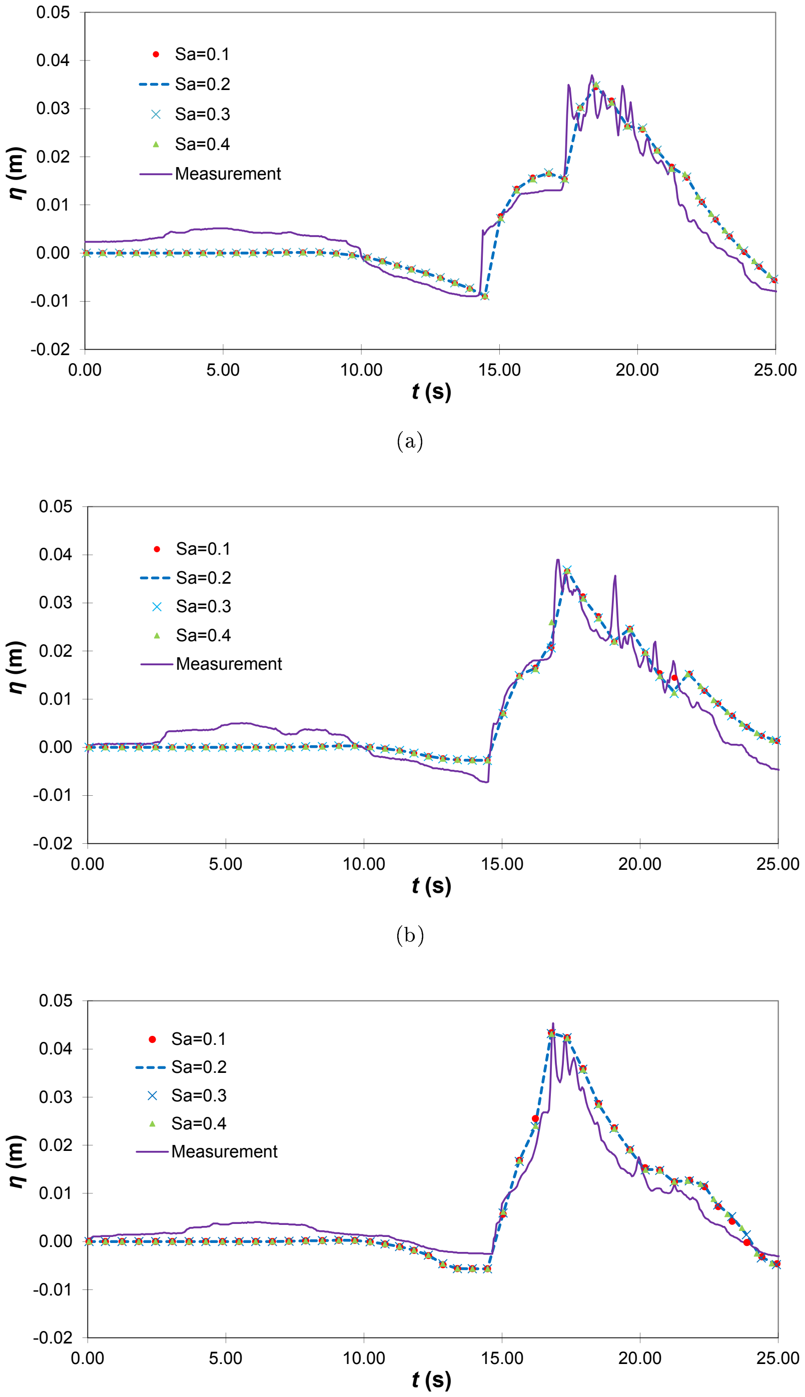


(a)

(b)

(c)

Figure 18: Tsunami in Monai valley: computed and experimental water levels at gauge: (a) G5, (b) G7, (c) G9.



(a)

(b)

(c)

Figure 19: Tsunami in Monai valley: computed water levels by using different s_a on the new grid at: (a) G5, (b) G7, (c) G9.

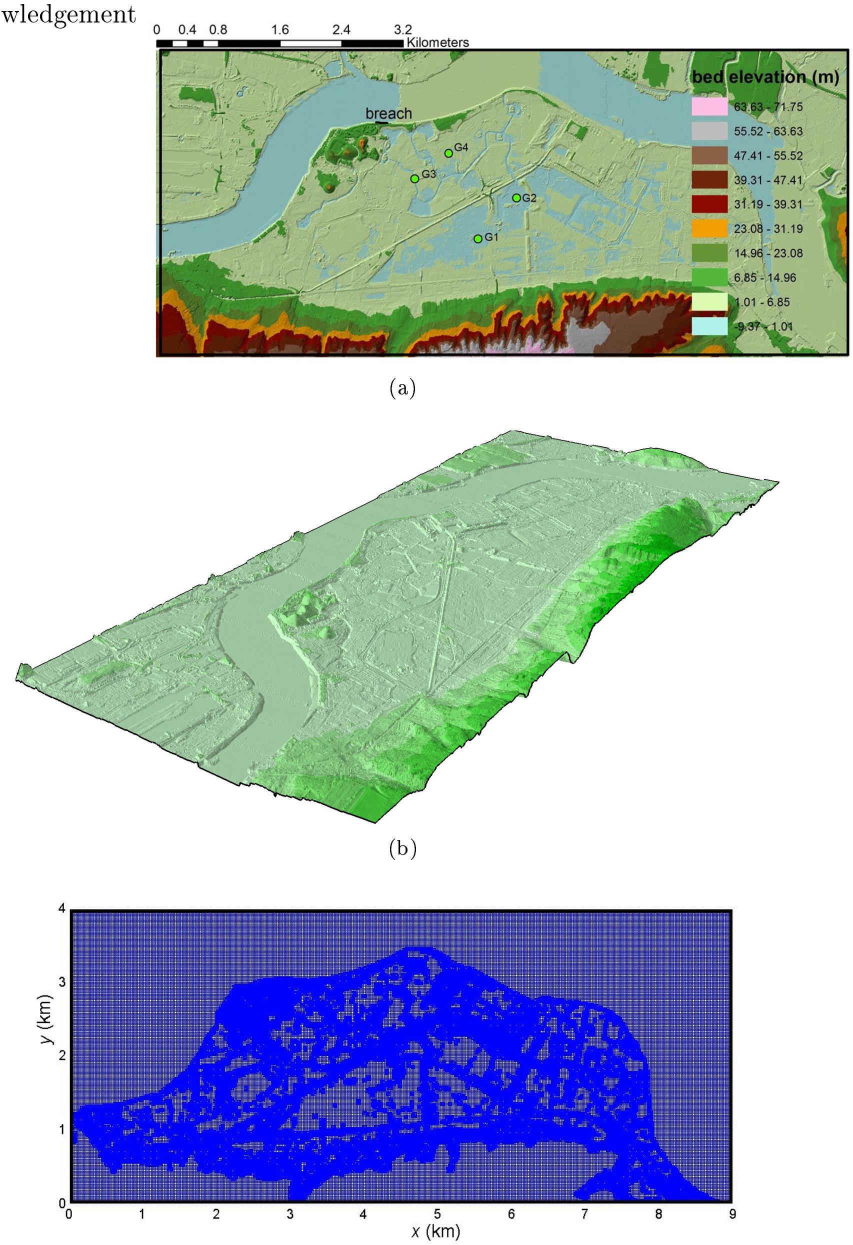
4.3. Hypothetic flood propagation and inundation in Thamesmead

The present non-uniform grid based shallow flow model is applied to simulate a hypothetic flood event caused by failure of flood defense in a 9000 m × 4000 m floodplain in Thamesmead, located on the south bank of the River Thames, United Kingdom. Processed 5m and 10m bare-earth Digital Terrain Models (DTM) as shown in Figures 20(a) and (b) are used to represent the topography for the fine and coarse uniform grids. While the grid with the highest resolution of 5m by 5m among three coarsening levels is created by the new grid system with $s_a = 0.2$ on the basis of the 5m DTM, as shown in Figure 20(c). Parts of the topography and non-uniform grid are plotted in Figure 21. It is clear that the grid remains to be high-resolution in those regions with relatively steep bed slope. The breach of the river defense is shown in Figures 20(a) and 21(a). The corresponding hydrograph leading to flood inundation into the floodplain is plotted in Figure 22 as in [22]. A constant Manning coefficient of 0.035 $\mathrm{sm}^{-1/3}$ is applied all over the computational domain.

Figure 23 compares the time histories of water level predicted by the model on the coarse, fine and new non-uniform grids at four different gauge points as indicated in Figure 20. The water levels computed on the non-uniform grid match well with those obtained on the fine uniform grid. However, this is not the case for the prediction on the coarse grid (10 m resolution), which diverge substantially from other sets of results. This is also confirmed by means of flood inundation maps at $t = 10\,\mathrm{h}$, as shown in Figure 24 for different grids. The inundation extents predicted on the fine and non-uniform grids are very close to each other, whereas that produced on the coarse grid is shown to be significantly different, especially on the right bottom part of the inundated area. The predictions on the fine and non-uniform grids giving rise to a larger flooded area and the different water level histories with the coarse grids are mainly because the high-resolution meshes can represent better small scale topographic features and flow passages.

Figure 25 illustrates the three-dimensional view of region-A as shown in Figure 21(a). A channel cutting through the embankment plays an important role to convey flood water to the other side of the floodplain. The dimension of the channel obviously determines the discharge rate and hence the flood extent on the other side of the embankment. Numerically, the shape and dimension of channel are greatly affected by the resolution of the topographic data. As sketched in Figure 25, the width of the channel is much larger on the 5 m-resolution DTM than on the 10 m-resolution DTM. High-resolution (5 m) representation of the channel is guaranteed on the non-uniform grid, see Figure 21, and so more flood water is able to be conveyed to the other side of the embankment, leading to a larger flood extent that is identical to the prediction on the fine uniform grid. In terms of performance, the non-uniform grid based model only requires 6.06 h of computational time on a desktop computer with an Intel i5-3470 CPU (quadcore) and 8G RAM, which is 38% of that needed by the fine grid model to produce similar results. The benefit of using the proposed non-uniform grid based model for large-scale flood simulations is evident.

Acknowledgement

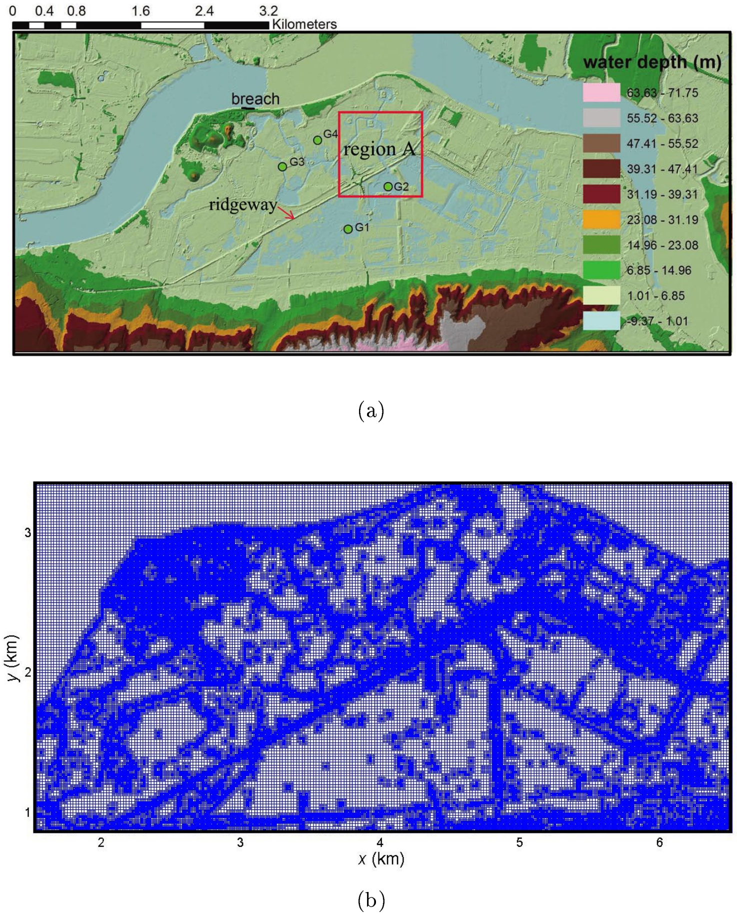


(a)

(b)

(c)

Figure 20: Hypothetic flood inundation in Thamesmead: (a) topography in 2D view, (b) topography in 3D view, (c) grid generated by the new grid system.



(a)

(b)

Figure 21: Hypothetic flood inundation in Thamesmead: (a) topography, (b) grid generated by the new grid system.

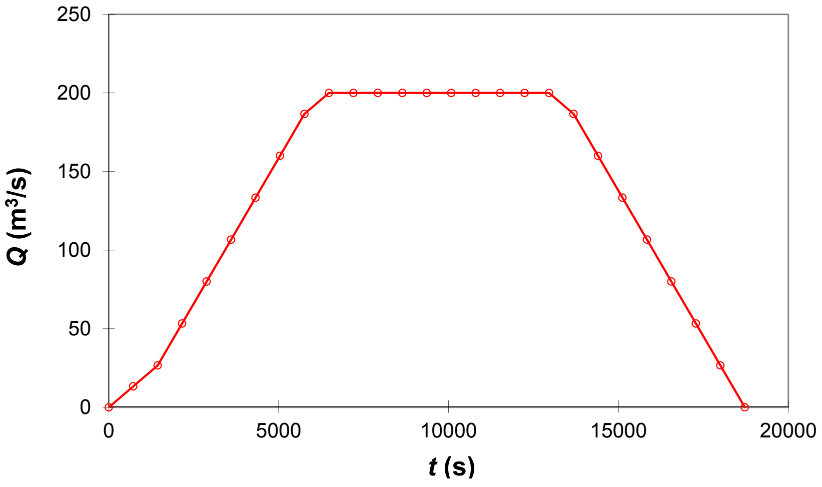
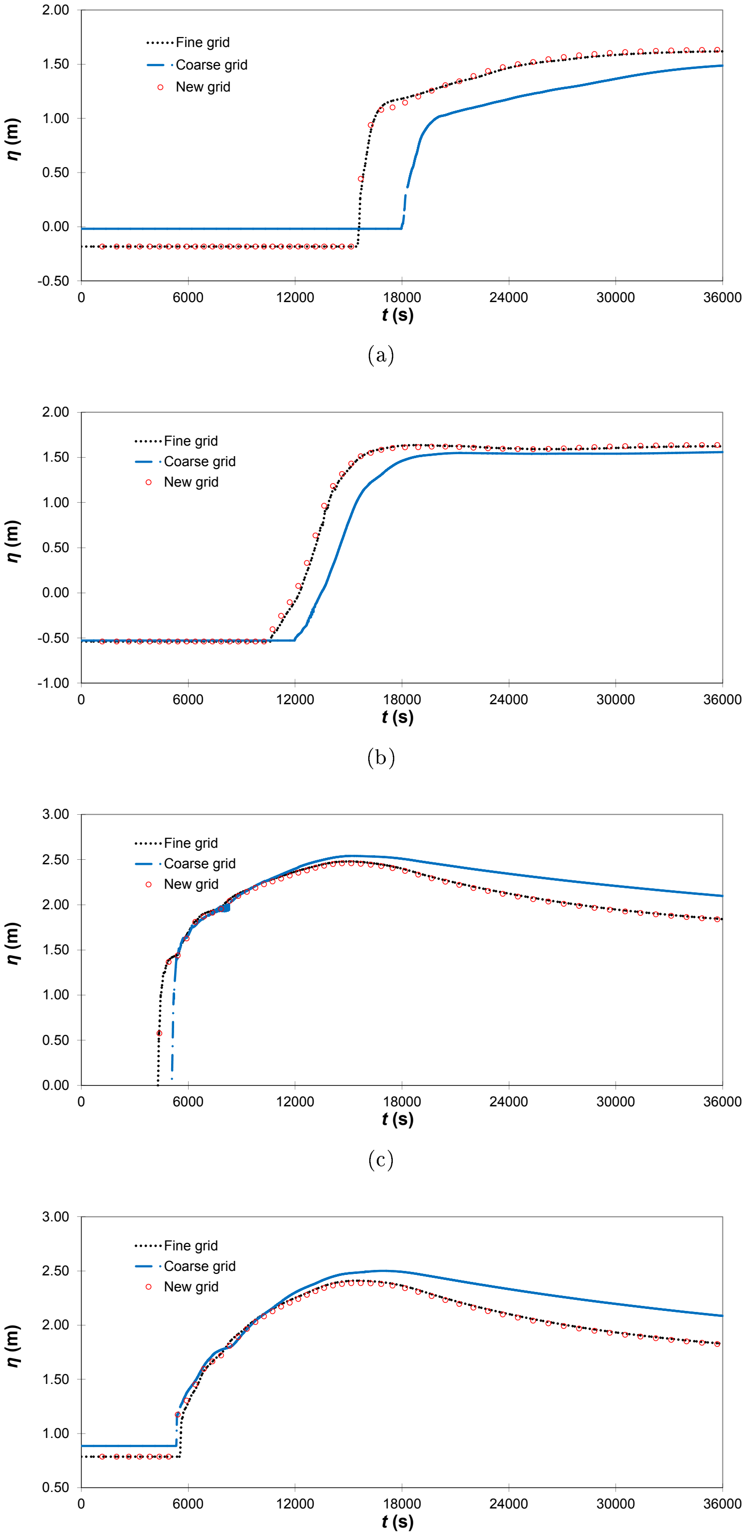


Figure 22: Hypothetic flood inundation in Thamesmead: discharge caused by defense break.



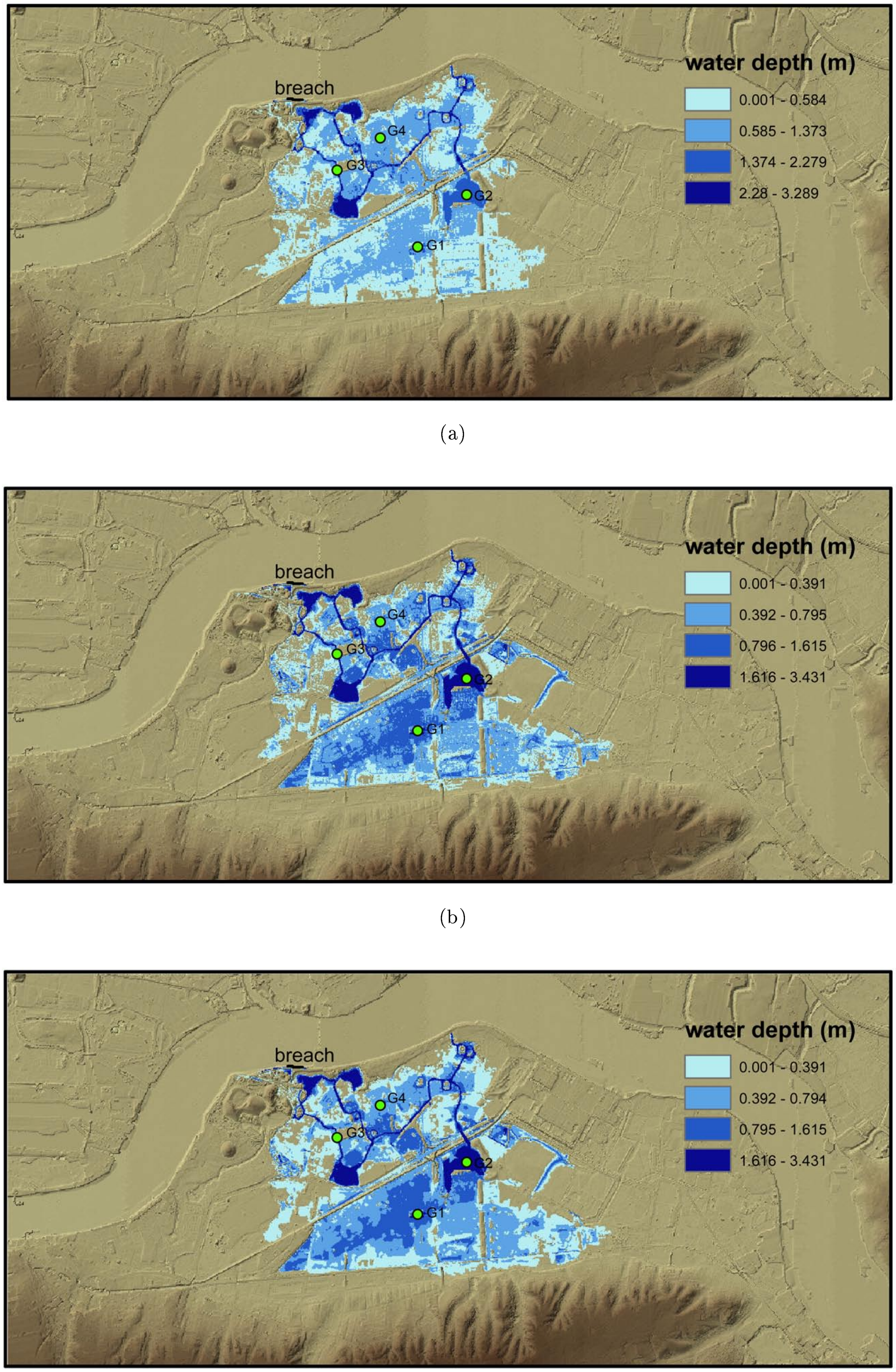
(a)

(b)

(c)

(d)

Figure 23: Hypothetic flood inundation in Thamesmead: computed water levels at gauge: (a) G1, (b) G2, (c) G3, (d) G4.

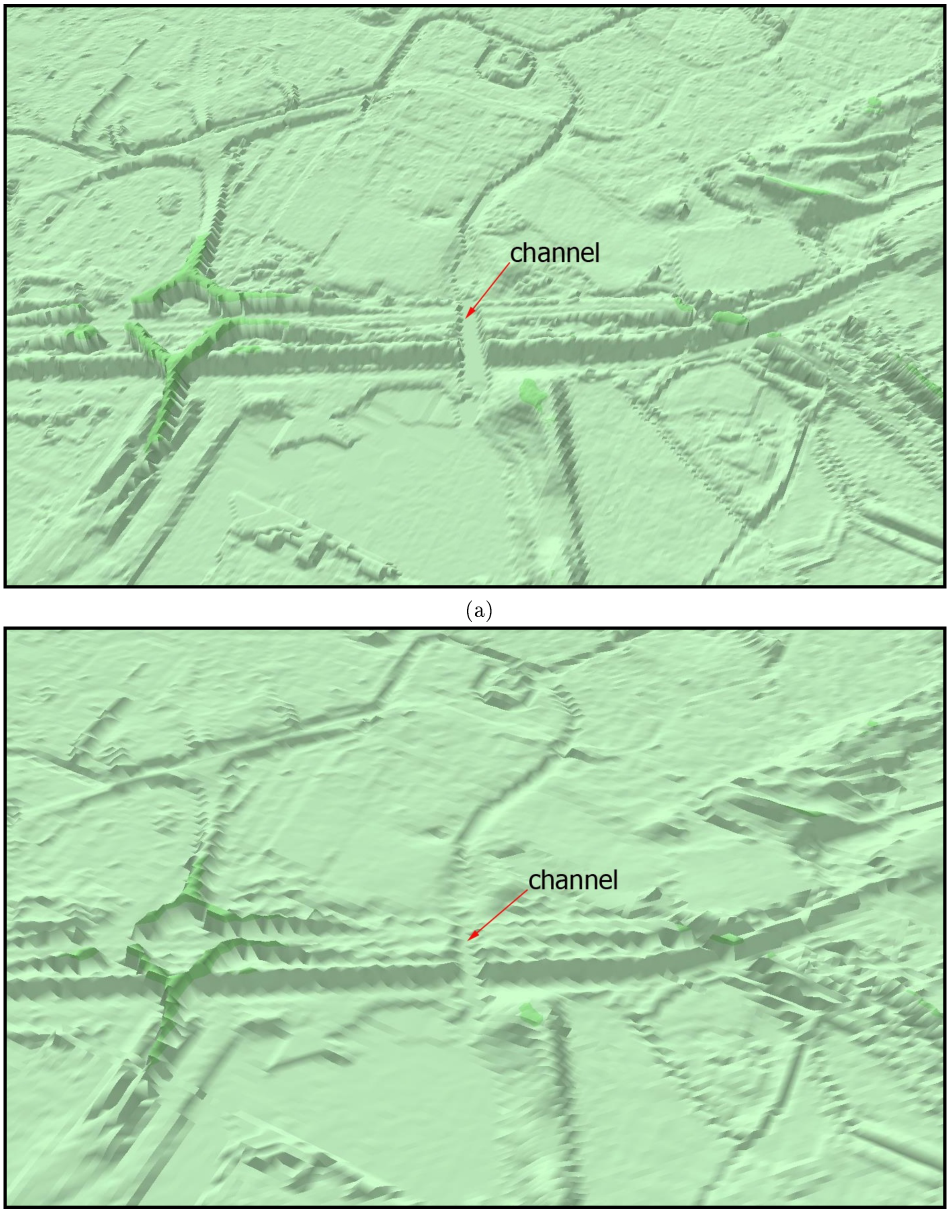


(a)

(b)

(c)

Figure 24: Hypothetic flood inundation in Thamesmead: computed flood map by using: (a) coarse grid, (b) fine grid, (c) new grid.



(a)

(b)

Figure 25: Hypothetic flood inundation in Thamesmead: 3D topography in region A: (a) with 10 m resolution, (b) with 5 m resolution.

5. Conclusions

In order to improve the computational efficiency and meanwhile preserve solution accuracy of a numerical model for shallow water flow simulations, a new structured non-uniform grid system is developed to automatically detect the topographic features where high-resolution representation is needed in a computational domain; in these regions, fine computational mesh will be generated while coarse mesh is used in the rest of the domain without manual work. The new grid system is incorporated into a robust 2D finite volume Godunov-type scheme.

The capability of the new non-uniform grid based model to accelerate shallow water flow simulations and at the same time maintain solution accuracy is verified against two laboratory and a field-scale test cases. The proposed structured but non-uniform grids are demonstrated to be able to effectively represent those key topographic features using high-resolution mesh, and thus ensure reliable simulations. Numerical results are compared with those predicted on coarse and fine uniform grids. While producing numerical results of similar accuracy, the proposed non-uniform grid based model saves about two times of computational time compared with the fine grid model. Furthermore, the new grid system does not need any redundant manual work in preprocess for delimiting the part requiring high-resolution mesh and refine it locally. Through the sensitivity analysis tests, the grid generation parameter$s_a = 0.2$ is confirmed be to a proper tradeoff for solution accuracy and simulation efficiency.

As a summary, the shallow flow model improved by the new grid system is able to accurately and efficiently reproduce the shallow water flows and therefore has potential for a wide range of practical applications. Further acceleration in computational speed is currently being explored using high-performance parallel and GPU computing.

Acknowledgements

This work is partly supported by the National Natural Science Foundation of China (19672016); State Key Program of National Natural Science Foundation of China (Grant No. 41330858) ; Natural Science Foundation of Qinghai Province (Grant No. 2015-ZJ-936Q) and the UK Natural Environment Research Council (NERC) (Grant No. NE/K008781/1).

References

[1] Audusse, E., Bristeau, M. O., 2005. A well-balanced positivity preserving second-order scheme for shallow water flows on unstructured meshes. Journal of Computational Physics 206, 311–333.

[2] Bates, P. D., Horritt, M. S., Fewtrell, T. J., 2010. A simple inertial formulation of the shallow water equations for efficient two-dimensional flood inundation modelling. Journal of Hydrology 387 (1-2), 33–45.

[3] Begnudelli, L., Sanders, B. F., 2007. Simulation of the St. Francis dam-break flood. Journal of Engineering Mechanics 133 (11), 1200–1212.

[4] Bermudez, A., Vazquez, M. E., 1994. Upwind methods for hyperbolic conservation laws with source terms. Computers & Fluids 23, 1049–1071.

[5] Dawson, R. J., Peppe, R., Wang, M., 2011. An agent-based model for risk-based flood incident management. Natural Hazards 59 (1), 167–189.

[6] Delis, A. I., Nikolos, I. K., Kazolea, M., 2011. Performance and comparison of cell-centered and node-centered unstructured finite volume discretizations for shallow water free surface flows. Archives of Computational Methods in Engineering 18, 57–118.

[7] Donat, R., Carmen Marti, M., Martinez-Gavara, A., Mulet, P., 2014. Well-balanced adaptive mesh refinement for shallow water flows. Journal Of Computational Physics 257, 937–953.

[8] Funke, S., Pain, C., Kramer, S., Piggott, M., 2011. A wetting and drying algorithm with a combined pressure/free-surface formulation for non-hydrostatic models. Advances in Water Resources 34, 1483–1495.

[9] Gandham, R., Medina, D., Warburton, T., 2015. GPU accelerated discontinuous galerkin methods for shallow water equations. Communications In Computational Physics 18, 37–64.

[10] Greenberg, J. M., Leroux, A. Y., 1996. A well-balanced scheme for the numerical processing of source terms in hyperbolic equations. SIAM Journal on Numerical Analysis 33, 1–16.

[11] Guan, M., Liang, Q., 2017. A two-dimensional hydro-morphological model for river hydraulics and morphology with vegetation 88, 10–21.

[12] Guan, M., Wright, N. G., Wright, N. G., 2014. 2D process-based morphodynamic model for flooding by noncohesive dyke breach. Journal of Hydraulic Engineering, 44–51.

[13] Heggelund, Y., Berntsen, J., 2002. A method for analysing nesting techniques for the linearized shallow water equations. International Journal for Numerical Methods in Fluids 38 (2), 163–185.

[14] Hinkelmann, R., 2005. Efficient numerical methods and information-processing techniques for modeling hydro-and environmental systems. Vol. 21. Springer, Berlin, Heidelberg.

[15] Hinkelmann, R., Zielke, W., mar 2000. Parallelization of a lagrange–Euler-model for three-dimensional free surface flow and transport processes. Computers & Fluids 29 (3), 301–325.

[16] Hou, J., Liang, Q., Li, Z., Wang, S., Hinkelmann, R., 2015. Numerical error control for second-order explicit tvd scheme with limiters in advection simulation 70 (9), 2197–2209.

[17] Hou, J., Liang, Q., Simons, F., Hinkelmann, R., 2013. A 2D well-balanced shallow flow model for unstructured grids with novel slope source term treatment. Advances in Water Resources 52, 107–131.

[18] Hou, J., Liang, Q., Simons, F., Hinkelmann, R., 2013. A stable 2D unstructured shallow flow model for simulations of wetting and drying over rough terrains. Computers & Fluids 82, 132–147.

[19] Hou, J., Simons, F., Mahgoub, M., Hinkelmann, R., 2013. A robust well-balanced model on unstructured grids for shallow water flows with wetting and drying over complex topography. Computer Methods in Applied Mechanics and Engineering 257, 126–149.

[20] Hu, K., Mingham, C. G., Causon, D. M., 2006. A mesh patching method for finite volume modelling of shallow water flow. International Journal for Numerical Methods in Fluids 50 (12), 1381–1404.

[21] Huang, W., Cao, Z., Pender, G., Liu, Q., Carling, P., 2015. Coupled flood and sediment transport modelling with adaptive mesh refinement. Science China-technological Sciences 58, 1425–1438.

[22] Liang, Q., 2010. Flood simulation using a well-balanced shallow flow model. Journal of Hydraulic Engineering 136, 669–675.

[23] Liang, Q., 2011. A structured but non-uniform cartesian grid-based model for the shallow water equations. International Journal for Numerical Methods in Fluids 66 (5), 537–554.

[24] Liang, Q., 2012. A simplified adaptive cartesian grid system for solving the 2D shallow water equations. International Journal for Numerical Methods in Fluids 69 (2), 442–458.

[25] Liang, Q., Borthwick, A. G., 2009. Adaptive quadtree simulation of shallow flows with wet dry fronts over complex topography. Computers & Fluids 38, 221–234.

[26] Liang, Q., Hou, J., Amouzgar, R., 2015. Simulation of tsunami propagation using adaptive cartesian grids. Coastal engineering journal 57 (1550016).

[27] Liang, Q., Marche, F., 2009. Numerical resolution of well-balanced shallow water equations with complex source terms. Advances in Water Resources 32, 873–884.

[28] Liu, P. L.-F., Yeh, H., Synolakis, C. (Eds.), 2008. Advanced numerical models for simulating tsunami waves and runup. World Scientific Pub. Co., Inc.

[29] Monfared, H. S. A., Darmian, D. M., 2017. Evaluation of appropriate advective transport function for one-dimensional pollutant simulation in rivers 10, 77–84.

[30] Ozdemir, H., Sampson, C. C., de Almeida, G. A. M., Bates, P. D., 2013. Evaluating scale and roughness effects in urban flood modelling using terrestrial LIDAR data. Hydrology and Earth System Sciences 17 (10), 4015–4030.

[31] Phillips, N. A., Shukla, J., 1973. On the strategy of combining coarse and fine grid meshes in numerical weather prediction. Journal of Applied Meteorology 12 (5), 763–770.

[32] Popinet, S., 2011. Quadtree-adaptive tsunami modelling. Ocean Dynamics 61, 1261–1285.

[33] Popinet, S., 2012. Adaptive modelling of long-distance wave propagation and fine-scale flooding during the tohoku tsunami. Natural Hazards and Earth System Science 12, 1213–1227.

[34] Rai, M. M., 1986. A conservative treatment of zonal boundaries for euler equation calculations. Journal of Computational Physics 62 (2), 472–503.

[35] Sanders, B. F., Schubert, J. E., Detwiler, R. L., 2010. ParBreZo: A parallel, unstructured grid, godunov-type, shallow-water code for high-resolution flood inundation modeling at the regional scale. Advances in Water Resources 33 (12), 1456–1467.

[36] Schumann, G. J.-P., Neal, J. C., Voisin, N., Andreadis, K. M., Pappenberger, F., Phanthuwongpakdee, N., Hall, A. C., Bates, P. D., 2013. A first large-scale flood inundation forecasting model. Water Resources Research 49 (10), 6248–6257.

[37] Singh, J., Altinakar, M. S., Ding, Y., 2011. Two-dimensional numerical modeling of dam-break flows over natural terrain using a central explicit scheme. Advances in Water Resources 34 (10), 1366–1375.

[38] Smith, L. S., Liang, Q., 2013. Towards a generalised GPU/CPU shallow-flow modelling tool. Computers & Fluids 88, 334–343.

[39] Soares-FrazÃ£o, S., Zech, Y., 2007. Experimental study of dam-break flow against an isolated obstacle. Journal of Hydraulic Research 45 Extra Issue, 27–36.

[40] Toro, E., Spruce, M., Speares, W., 1994. Restoration of the contact surface in the HLL-Riemann solver. Shock Waves (1994) 4, 25–34.

[41] Vacondio, R., Dal Palu, A., Mignosa, P., 2014. GPU-enhanced finite volume shallow water solver for fast flood simulations. Environmental Modelling & Software 57, 60–75.

[42] Van Leer, B., 1984. On the relation between the upwind-differencing schemes of godunov, engquist–osher and roe. SIAM Journal on Scientific and Statistical Computing 5, 1–20.

[43] Wang, Y., Liang, Q., Kesserwani, G., Hall, J. W., 2011. A 2D shallow flow model for practical dam-break simulations. Journal of Hydraulic Research 49, 307–316.

[44] Wilson, M., Atkinson, P., 2003. Sensitivity analysis of a flood inundation model to spatially-distributed friction coefficients obtained using land cover classification of Landsat TM imagery. In: IEEE International Geoscience and Remote Sensing Symposium (IGARSS), Toulouse, France. Institute of Electrical & Electronics Engineers (IEEE), pp. 1579–1581.

[45] Xia, X., Liang, Q., 2016. A GPU-accelerated smoothed particle hydrodynamics (SPH) model for the shallow water equations. Environmental Modelling & Software 75, 28–43.

[46] Zhang, S., Xia, Z., Yuan, R., Jiang, X., 2014. Parallel computation of a dam-break flow model using openmp on a multi-core computer. Journal Of Hydrology 512, 126–133.