
DOI link
https://doi.org/10.1016/j.wse.2017.12.004

ePrints link
http://eprint.ncl.ac.uk/pub_details2.aspx?pub_id=245160

Date deposited
18/01/2018

Copyright
© 2017 Hohai University. Production and hosting by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

Licence
This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International licence
An efficient dynamic uniform Cartesian grid system for inundation modeling

Jing-ming Hou a, Run Wang a, Hai-xiao Jing a,*, Xia Zhang a, Qiu-hua Liang b, Yan-yan Di c

a School of Hydraulic and Hydropower, Xi’an University of Technology, Xi’an 710048, China
b School of Civil Engineering and Geosciences, Newcastle University, Newcastle Upon Tyne NE1 7RU, UK
c Hydrology Bureau, Yellow River Conservancy Commission, Zhengzhou 450004, China

Received 23 April 2017; accepted 30 September 2017
Available online ▪▪▪

Abstract

A dynamic uniform Cartesian grid system was developed in order to reduce the computational time in inundation simulation using a Godunov-type finite volume scheme. The reduction is achieved by excluding redundant dry cells, which cannot be effectively avoided with a conventional Cartesian uniform grid system, as the wet area is unknown before computation. The new grid system expands dynamically with wetting, through addition of new cells according to moving wet-dry fronts. The new grid system is straightforward in implementation. Its application in a field-scale flood simulation shows that the new grid system is able to produce the same results as the conventional grid, but the computational efficiency is fairly improved.

© 2017 Hohai University. Production and hosting by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

Keywords: Uniform Cartesian grid system; Flood simulation; Wetting and drying; Shallow water equation; Godunov-type finite volume scheme

1. Introduction

Flooding is a type of natural disaster that raises an enormous threat to lives and property. This threat is likely to escalate as a result of global warming and climate change. For example, southern England has seen the wettest winter for more than 200 years from 2013 to 2014, leading to severe and long-lasting flooding in Somerset. According to data provided by the Flood Protection Association (FPA) in the UK, of the 28 million homes in the UK, over 5 million are currently at risk, as well as over 300,000 business premises and many more public and utility services buildings. Moreover, flooding is likely to cause other environmental problems. For instance, it is linked closely to water quality (Hrdinka et al., 2012). Flood prediction, which can be achieved using numerical models, can help guide people toward protection measures for the upcoming floods so that the damage can be significantly alleviated. In recent years, Godunov-type finite volume schemes have gained more attention in flood simulation, for example in Delis et al. (2008), Liang (2010), Jeong et al. (2012), Hou et al. (2013b), Ata et al. (2013), and Guan et al. (2014), as they are capable of capturing shocks, and preserving accuracy and robustness.

Simulation results have also been proven to be very sensitive to grid resolution (Wilson and Atkinson, 2003; Ozdemir et al., 2013). With finer grids, more realistic terrain features can be reflected. However, high-resolution modeling of real-world flooding may sometimes require millions of computational nodes or cells to accurately represent the domain topography of a floodplain, making such simulations computationally prohibitive on most existing computers. This signals
a need to make computations faster. There are several approaches to improving the computational efficiency of a flood model. Liu and Pender (2013) used a cellular automata-based rapid flood spreading model to generate an estimated inundation map. Despite the fast computation, this approach fails to compute flow velocities. Another approach solves the simplified governing equations derived by neglecting the dynamic terms in the two-dimensional (2D) shallow water equations (SWEs) (Bates et al., 2010; Wang et al., 2011a). However, the neglected dynamic terms can affect the accuracy of the evaluated velocities, especially for flows with transient flow features. Some researchers have adopted adaptive grids to improve the computation efficiency, for instance George and Leveque (2008), Popinet (2011, 2012), and Liang (2012). As mentioned in Liang and Borthwick (2009) and Popinet (2011), when estimating the new values in newly created cells on adaptive grids, the equations of mass conservation and water surface continuity, only one of the two can be satisfied. Both of these equations are important in flood simulation, as a violation of the former will lead to erroneous inundation areas and unsatisfactory values of the latter will bring about disturbance of the conservation property (C-property) (Bermudez and Vazquez, 1994). Due to this deficiency of the adaptive grids, uniform grid-based shallow water flow models are preferred and attract more attention (Hervouet, 2000; Sanders et al., 2010; Pu et al., 2012; Smith and Liang, 2013; Xia et al., 2017). As a uniform grid is easy to generate from the produced terrain DEM, the numerical methods can be straightforwardly implemented, and the complex terrain features can be simply reflected by using uniform grids.

To simulate flooding efficiently and accurately, this study used a robust full shallow water model on a dynamic uniform Cartesian grid system, which might optimize the computational grid according to flow features. The governing equations and the numerical scheme for the inundation model are briefly introduced in Section 2; the new grid system is addressed in Section 3 and its performance is demonstrated and analyzed through discussion of a field-scale test case involving flood routing in Section 4 and Section 5; and Section 6 describes the findings of this study.

2. Governing equations and numerical methods

The numerical inundation model was developed by solving the 2D SWEs numerically, within a framework of a well-balanced cell-center Godunov-type finite volume method. The 2D pre-balanced SWEs proposed in Liang and Borthwick (2009) were chosen as governing equations:

$$\frac{\partial q}{\partial t} + \frac{\partial f}{\partial x} + \frac{\partial g}{\partial y} = S$$

(1)

where

$$q = \begin{bmatrix} \eta \\ q_x \\ q_y \end{bmatrix}$$

(2)

$$f = \begin{bmatrix} u \frac{uh}{uvh} \\ v \frac{v}{vuh} \\ \frac{uh}{v} \end{bmatrix}$$

(3)

$$g = \begin{bmatrix} v \frac{v}{v} \\ u \frac{u}{uvh} \\ \frac{v}{v} \end{bmatrix}$$

(4)

$$S = \begin{bmatrix} m_w \\ -gq_x - C_f u \sqrt{u^2 + v^2} \\ -gq_y - C_f v \sqrt{u^2 + v^2} \end{bmatrix}$$

(5)

where \(t\) is time; \(x\) and \(y\) are the Cartesian coordinates; \(q\) is the vector of conserved flow variables containing \(\eta\), \(q_x\), and \(q_y\), which are the free surface water level and the unit-width discharges in the \(x\)- and \(y\)- directions, respectively; \(q_x = uh\) and \(q_y = vh\); \(h\), \(u\), and \(v\) are the water depth and the depth-averaged velocities in the \(x\)- and \(y\)- directions, respectively; \(\eta\) is the bed elevation and \(\eta = h - b\); \(f\) and \(g\) are the flux vectors in the \(x\)- and \(y\)- directions, respectively; \(S\) is the source vector; \(m_w\) is the source or sink of mass caused by rainfall or infiltration; and \(C_f\) is the bed roughness coefficient, determined herein as \(gn^2/h^{1/3}\), with \(n\) being the Manning coefficient.

The SWEs are solved numerically with the two-step MUSCL-Hancock scheme developed in Leer (1984) and adjusted for the SWE simulation in Zhou et al. (2002), within the framework of the Godunov-type cell-centered finite volume method for Cartesian grids in Liang (2010, 2011, 2012), and Smith and Liang (2013). It consists of a predictor step and a corrector step. In the first step, the intermediate flow variables are computed over half of a time interval. Those predicted variables are then utilized in the corrector step to update the results to a new time level. The friction source terms are not evaluated within the MUSCL-Hancock scheme but independently evaluated by a splitting point-implicit method proposed in Liang and Marche (2009). Since the MUSCL-Hancock is an overall explicit scheme, the Courant-Friedrichs-Lewy (CFL) condition must be satisfied to ensure solution stability. The CFL condition proposed in Liang (2012) is employed to estimate the time step. Open and closed boundaries are treated as in Liang and Borthwick (2009). In this work, the aforementioned numerical methods are not documented in detail and readers can look them up in the references listed above.

3. Dynamic uniform Cartesian grid system

In practical application, the small-scale features (i.e., walls and ditches) can have a significant impact on flood propagation. Despite the increased accuracy created by applying a high-resolution grid, it significantly increases the computational power required to simulate flood events over large domains. For example, Bates (2012) claimed that simulations
at very high resolutions (grid resolution of 10 cm) would take 104 times longer than at 2 m. As mentioned in the introduction, there are a couple of means to promoting the efficiency of computation. It was achieved in this study by using a new uniform Cartesian grid system, which can reduce the number of cells under consideration.

In flood routing, generally speaking, just a small part of the domain is wet initially and the wet part then starts to spread as the river level escalates or the dam or defense breaks, which is actually a problem involving moving wet-dry boundaries (Bates and Hervouet, 1999). Moving wet-dry boundaries are preferred for computation on fixed grids, as shown in Liang and Marche (2009), Delis and Nikolos (2013), and Hou et al. (2013c), as fixed grids are easy to generate and can avoid the numerical difficulties induced by dynamically adaptive grids. However, fixed grids will lead to massive amounts of redundant dry cells, which take up a large amount of memory and computational effort in inundation simulation. While the schemes proposed, for example, by Wang et al. (2011b) and Hou et al. (2013a) can avoid the computation of flow variables on dry cells away from wet-dry interfaces (white cells sketched in Fig. 1), a loop to identify these cells is still required. Moreover, these dry cells may become wet later in the computation and must be stored in the memory throughout the simulation. However, certain cells remain dry at all times, and thus are not necessarily required in the simulation. As the inundation area is unknown before simulation, it is impossible to determine the location of persistently dry cells. To avoid underestimation of the flooded area, a computational domain that covers a lot of persistently dry cells is commonly used (Fig. 1), for example the rectangular domains used in Liang (2010) and Singh et al. (2011). Apparently, simulation efficiency can be improved by removing the persistently dry cells. Based on the idea described above, a new uniform Cartesian grid system was developed to discard persistently dry cells in inundation simulation (Fig. 2).

The new grid system was created according to flood features. When the flood spreads, the grid expands dynamically to cover wet cells and two layers of dry cells adjacent to wet cells (in this study, wet and dry cells were determined by checking if the water depth was higher than a criterion of $10^{-6}$ m). In contrast, the grid remains static when the flood recedes; the reason for this is addressed in the last paragraph of this section. That means new wet cells are added to the grid system in wetting, but no cells are deleted in drying. In a time step, the new uniform Cartesian grid is generated following the steps below:

Step 1: Dry cells are identified adjacent to wet ones. These dry cells are termed root cells in this paper and are actually the boundary cells of an inundation area. If it is the first time step, the root cells are defined from the initial and boundary conditions. They can be the dry cells next to water bodies, inflow boundaries, dikes, or dam breaches, from which water will burst into the area under consideration. For example, root cells with ID numbers of 1, 2, 3, and 4 were generated near an inflow boundary in this study, as illustrated in Fig. 3(a).

Step 2: A layer of dry cells is extended from the root cells identified in last step. The extension is carried out cell by cell. For instance, if root cell 1 is under consideration, four sub-steps are required to add new dry cells: (1) A loop over all four neighboring cells is taken to check whether a neighboring cell has already been added to the grid. (2) If the above condition is
not satisfied for a neighboring cell, a new cell is added to the grid. For example, as cell 1 does not have northern and western neighboring cells, new dry cells 5 and 6 are added, respectively, as plotted in Fig. 3(b).

(3) The bed elevation and the flow conditions of zero water depth and velocities are imposed on the newly created dry cells. (4) The neighboring information of the newly added dry cells is determined. If there are no neighboring cells, the neighboring cell ID is specified as 0. Meanwhile, the neighboring information of root cells should be updated for future use. For example, the neighboring cell IDs for root cell 1 are 5, 2, −1, and 6 to the north, east, south, and west, respectively (Fig. 3(b)). A negative ID signifies an inflow boundary.

Step 3: After a layer of dry cells is added, the aforementioned Godunov-type finite volume scheme is employed to compute the values of flow variables at the new time level for all cells excluding the newly added dry cells. As the newly added dry cells are next to dry cells, it must be dry cells at the new time level when using the explicit method (it is out of the question to obtain net fluxes of mass). For example, we only computed the fluxes and source terms for cells 1 through 4 in Fig. 3(b). In fact, the newly added dry cells are just used to compute fluxes and source terms of the roots cells. The updated flow variables indicate that cells 1 through 4 turn out to be wet at the new time level. Therefore, as illustrated in Fig. 3(c), the cells 1 through 4 are wet grids depicted in blue. The red and white grids both are dry cells, but the white grids, i.e., the newly created dry cells, are out of the calculation in this time level.

The procedure from Step 1 to 3 is repeated to generate the grid and compute the flood spreading until the desired time. For instance, in the second cycle, the newly created grid is demonstrated in Fig. 3(c).

Noticeably, the new grid is generated through addition of cells, and no subtraction is taken into account, because deleting a cell from a grid will lead to the reconstruction of all cell IDs and in turn the neighboring cell information. Therefore, the improved efficiency through reduction of cell number is likely to be offset by the reconstruction for the cell ID and neighboring information. In addition, flood recession may give rise to discontinuous wet patches, which make the grid too complex. That is why the grid will not shrink in the drying process, i.e., when the flood starts to subside. It is unhelpful to have some redundant dry cells, which may affect the efficiency in the drying process. However, in comparison to the conventional uniform Cartesian grid system, the redundant dry cells are much fewer in the new grid system and it remains preferable. Unlike the dynamic adaptive grid, the new grid system employs uniform square cells in the dynamic process, no refining and coarsening are demanded in simulation, and the system does not involve the problem of preserving the C-property and mass conservation at the same time. It should be noted that the new grid system will not increase the model's efficiency for flash flood simulation, because the heavy rainfall will wet the entire domain and no redundant dry cells will exist. In this case, the new grid system becomes a regular fixed uniform grid.

4. Simulation of defense-break flood in Thamesmead

The performance of the Godunov-type finite volume scheme has been verified widely, for example in Wang et al. (2011b) and Liang (2010, 2011, 2012). In order to accentuate the advantage of the new grid system in terms of efficiency, a field-scale flood was modeled with the Godunov-type finite volume scheme, including flood spreading and subsidence processes, on the new grid as well as on a conventional uniform Cartesian grid. A fictional flood was assumed to be caused by a river bursting its bank in Thamesmead, England. Grid resolutions of 5 m and 10 m were chosen for both grids and the dimensions of the computational domain for the conventional uniform Cartesian grid were set to be 5000 m × 4000 m, as shown in Fig. 4. Fig. 4 also indicates a 140 m long breach from which the flood flows into the floodplain. The inflow hydrograph is shown in Fig. 5. A constant infiltration rate of $2.5 \times 10^{-5}$ m/s and Manning coefficient of 0.035 m$^{1/3}$/s were specified across the flood plain. The simulations on both grids were carried out on a desktop computer with an Intel i5-3470 CPU (four cores with a clock
speed of 3.2 GHz) and 8G RAM, for 54000 s with a Courant number of 0.5.

5. Results and discussion

Fig. 6 compares the computed flood maps in terms of water depth on the new uniform Cartesian grid and the conventional one with 5-m grid resolution at \( t = 2 \) h, 5 h, and 8 h. The results are exactly the same, as the new grids are able to cover all required wet cells and adjacent dry ones. To demonstrate this clearly, three schemes have been examined: the conventional uniform Cartesian grid with 10-m grid resolution (the coarser grid simulation), the conventional uniform Cartesian grid with 5-m grid resolution (the fine grid simulation), and the dynamic uniform Cartesian grid with 5-m grid resolution (the new grid simulation). The computed water level evolutions at three gauges (G1 (3185 m, 1545 m), G2 (3685 m, 2075 m), and G3 (2805 m, 2655 m)) are compared in Fig. 7 for both grids. The computed water levels on the new grid and the fine one coincide perfectly, which further indicates that the new grid does not affect the accuracy of the modeling. Thus, if the new grid can speed up the simulation for floods, it will be a promising alternative to the conventional uniform Cartesian grid. In addition, Fig. 7 demonstrates that the real-word

---

Please cite this article in press as: Hou, J.-m., An efficient dynamic uniform Cartesian grid system for inundation modeling, Water Science and Engineering (2017), https://doi.org/10.1016/j.wse.2017.12.004
inundation simulation is quite sensitive to the grid resolution, and thus high resolution is required.

Since the new grid was developed to avoid the dry cells that are not involved in inundation simulation, the grid should evolve in accordance with flood spreading. Fig. 8, along with Fig. 6, demonstrates the flood spreading and the grid evolution and suggests that the new grid expands along with the wet-dry front moving with the spreading flood. The time history of the cell number on two grids is plotted in Fig. 9. A clear upswing of the total cell number for the new grid is observed in flood spreading, but it remains nearly constant after the maximum inundation takes place around $t = 8$ h (Fig. 10). The maximum cell number recorded was 147469. In contrast, the conventional grid has 800000 cells throughout the simulation. A large amount of redundant cells are successfully excluded by the new grid system. However, as the dry cells away from wet cells do not demand the computation of fluxes and source

---

Please cite this article in press as: Hou, J.-m., An efficient dynamic uniform Cartesian grid system for inundation modeling, Water Science and Engineering (2017), https://doi.org/10.1016/j.wse.2017.12.004
terms, the efficiency increase is not directly proportional to the reduced cell number. From the computational time shown in Fig. 11, the new grid is far more economical than the conventional one. Fig. 11 quantitatively demonstrates the improved efficiency of the new grid system. It can save more than 34% of the computational effort (it is 52% faster) even in the flood recession process, when the grid is almost fixed and in turn some dry cells appear. Yet there are many more redundant dry cells on the conventional grid than on the new grid throughout the simulation, as plotted in Fig. 12. In a nutshell, the new grid system can greatly improve the computational efficiency by reducing the redundant dry cell number. In the meantime, the accuracy of the scheme is not affected at all.

6. Conclusions

In order to improve the efficiency of high-resolution inundation simulation for uniform Cartesian grids, this paper introduces a new uniform Cartesian grid system that is able to substantially reduce the number of redundant dry cells. The new grid is generated dynamically according to the moving wet-dry fronts in a straightforward way. Therefore, the requisite number of wet cells and the adjacent dry cells are exactly taken into account when the flood spreads. As the flood subsides, the grid does not shrink when cells become dry again in order to avoid a complex reconstruction of grid information. The performance of the new grid system was verified by a field-scale flood case with a grid resolution of 5 m. The computational results show that the new grid system works much better than the conventional one in terms of efficiency. It can expedite inundation simulation, including spreading and receding processes, by more than 50%. Moreover, the same results of the flow variables as the conventional grid are computed by the scheme on the new grid. This indicates that the new grid system will not affect the model’s accuracy and thus is a good alternative for high-resolution, large-dimensional inundation simulation. In future work, the proposed grid system is envisaged to be extended to fit a non-uniform grid so as to make it a more general method for improving the computational efficiency.

References


Please cite this article in press as: Hou, J.-m., An efficient dynamic uniform Cartesian grid system for inundation modeling, Water Science and Engineering (2017), https://doi.org/10.1016/j.wse.2017.12.004