

An application of the automatic domain updating to the Tonle Sap Lake, Cambodia

Tomohiro Tanaka¹ and Hidekazu Yoshioka²

¹Graduate School of Global Environmental Studies, Kyoto University, Japan

²Graduate School of Natural Science and Technology, Shimane University, Japan

Abstract:

Simulating flow dynamics in large-scale lakes is often time-consuming. For river flood simulation, the automatic domain updating (ADU), which can effectively control the simulation domain only in and around the flooded areas, has recently been developed. It is easily implementable without any computational errors for river flood simulation; however, its applicability to lake flow simulation with precipitation/evapotranspiration has not been investigated. This study examines the applicability of the ADU to large-scale lake flow simulation with the 2-dimensional local inertial equations (2D-LIE) taking the Tonle Sap Lake, Cambodia, as a study site. The 2D-LIE with the ADU demonstrated 2.1 times faster simulation with errors less than 5.5%. This efficiency was achieved owing to the wet/dry seasonal nature of the tropical lake and backflow from the mainstream of the Mekong River in the rainy season, suggesting that the ADU is applicable to large-scale lake flow simulation.

KEYWORDS 2-D lake flow simulation; automatic domain updating; computational cost; Tonle Sap Lake

INTRODUCTION

Nowadays, assessing hydrodynamics of a freshwater body using mathematical models plays a fundamental role in designing lake conservation plans. In particular, large-scale (> 100 km²) lakes found in semi-arid areas, like the Tonle Sap Lake in Cambodia, are exposed to high risk of flood and/or contamination due to poor wastewater treatment (Ung *et al.*, 2019; Hunsberger *et al.*, 2018) in common with major megacities in Southeast Asia (Luo *et al.*, 2019). The shallow water equations (SWEs) (Castro-Orgaz and Hager, 2019) is one of the most widely-used mathematical models for various large-scale water flow simulations such as river flooding (Luo *et al.*, 2018) and/or coastal flooding (Ikeuchi *et al.*, 2015). To reduce the computational burden of the SWEs, extensive collections of simplifications/approximations have been explored (Hunter *et al.*, 2007; Teng *et al.*, 2017; Mignot *et al.*, 2019). One of the most effective approximations is the local inertial equations

(LIEs) first proposed by Bates *et al.* (2010), which is the SWEs without the advective acceleration terms (Seyoum *et al.*, 2012; Martins *et al.*, 2017; Mateo *et al.*, 2017). This simplified model can overcome the severe numerical stability issue of the conventional diffusive wave equations, keeping a simpler form of the momentum equation than the SWEs (de Almeida and Bates, 2013; Yamazaki *et al.*, 2013).

Another approach includes domain control techniques to reduce unnecessary processes in simulation. Yamaguchi *et al.* (2007) proposed the dynamic domain defining method. This method divides a simulation domain into several blocks and only computes the blocks containing wet cells to improve computational efficiency. A far simpler and more straightforward alternative, referred to as automatic domain updating (ADU), has recently been proposed by Tanaka *et al.* (2019). The ADU controls the simulation area at a cell level, i.e. it efficiently detects wet cells and their neighboring ones that may become wet in the next time step (shown as the blue and yellow cells in Figure 1, respectively). These domain control techniques can be easily implemented, reducing the computational costs by omitting the calculation of governing equations (and even omitting the conventional simple dry check process, i.e. checking if water depth is positive or not (Medeiros and Hagen, 2013) in dry areas: the green cells in Figure 1.

Since the ADU controls the simulation domain at a cell level in and around flood areas by tracing their boundaries at each time step, it implicitly assumes that the expansion of the flood areas starts only from specific (and thus known) cells. This assumption is reasonable when simulating river flooding events where the flood water always comes from river channels whose locations are known a priori. This scheme, in contrast, may not be transferable to simulating pluvial flooding events (floods caused by intensive rainfall exceeding drainage capacity) because the rainfall would be supplied to cells far away from the flood area as an external driver, and the flooding there cannot be straightforwardly handled by the ADU. Both pluvial and fluvial floods could happen in the same simulation domain. One example is large-scale lake flow simulation in a semi-arid region where most of the heavy rainfall occurs in the rainy season during which the lake area widely expands (Kummu *et al.*, 2014). In such a case, the impacts of omitting rainfall supply on dry cells might be limited because a

Correspondence to: Tomohiro Tanaka, Graduate School of Global Environmental Studies, Kyoto University, Room 584, C1-4, Kyotodaigakukatsura, Nishikyo-ku, Kyoto, Kyoto 615-8540, Japan. E-mail: tanaka.tomohiro.7c@kyoto-u.ac.jp

Received 28 January, 2020
Accepted 16 March, 2020
Published online 7 April, 2020

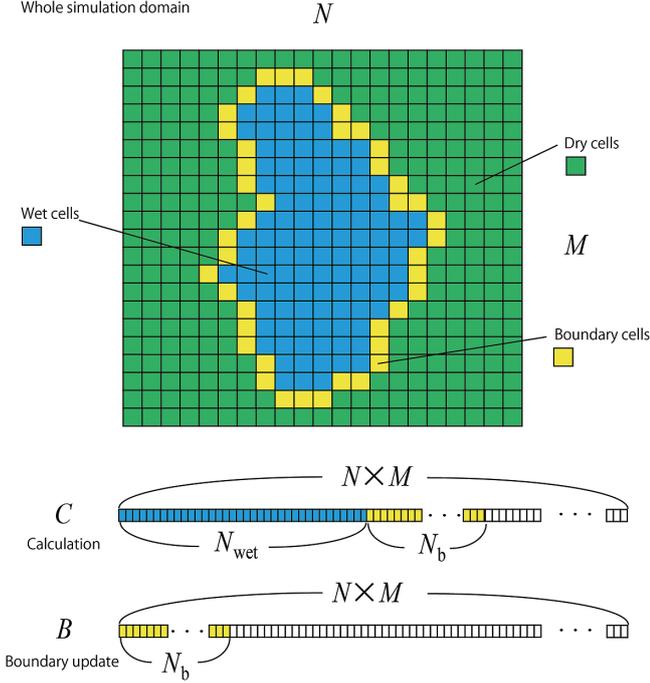


Figure 1. Definitions of the key terms and variables used in the ADU. The blue cells show the wet cells having water depths larger than a prescribed threshold value (0.01 (m) in this study); the yellow cells show the boundary cells; the green cells correspond to the dry cells. The numerical simulation takes place in total N_{wet} wet (blue) cells and N_b boundary (yellow) cells. If a yellow cell gets wet at a certain time step, it is updated as a wet cell (blue), and its surrounding dry cells are updated as boundary (yellow) cells. The momentum and mass balance equations are solved only at blue and yellow cells registered to an array C ; the model checks dryness only at cells registered to an array B . We can accelerate the computation through limiting a process to check if a cell is dry or not only in the yellow cells

large part of the target area is already wet (and included in the simulation area) in rainy seasons, while almost no rainfall is supplied in the dry seasons during which the lake shrinks.

This study examines the applicability of the ADU to a recently developed 2D lake flow model based on the LIEs on the Tonle Sap Lake (TSL) in the Mekong River basin, Cambodia, as a case study (Tanaka *et al.*, 2018b). Computational results by the model with the ADU are compared with the one without the ADU focusing on the flood area and the observed data of the water stage. The impacts of ignorance of rainfall on dry cells, which is a key technique potentially contributing to efficient numerical computation in the ADU, are investigated as well. Finally, computational efficiency of the ADU is discussed for both parallel/serial simulation settings assuming the different availability of computational resources.

2D LAKE FLOW MODEL

Governing equations

This study uses the lake flow model based on the 2D-LIE as the governing equations, developed by Tanaka *et al.* (2018b). The 2D-LIE consists of the continuity equation

$$\frac{\partial h}{\partial t} + \frac{\partial p}{\partial x} + \frac{\partial q}{\partial y} = r + Q_{\text{tr}} - e \quad (1)$$

and the momentum equations

$$\frac{\partial p}{\partial t} + gh \left[\frac{\partial(h+z)}{\partial x} + \frac{n^2 p |p|}{h^{10/3}} \right] = 0, \quad (2)$$

$$\frac{\partial q}{\partial t} + gh \left[\frac{\partial(h+z)}{\partial y} + \frac{n^2 q |q|}{h^{10/3}} \right] = 0, \quad (3)$$

where t is the time; z is the bed elevation, p is the line discharge in the x -direction, q is the line discharge in the y -direction, h is the water depth, r is the source term due to rainfall, e is the sink term due to evapotranspiration, Q_{tr} is the inflow from tributary rivers, and g is the gravitational acceleration coefficient, and n is the Manning's roughness coefficient. The equations can be solved subject to appropriate initial and boundary conditions. Usually, the initial state of water depth and fluxes in the x - and y -directions are set as explained later; the water stage is prescribed at out-flow boundaries, while the line discharges are specified at inflow boundaries.

Discretization

The 2D-LIE is discretized with a semi-implicit finite difference scheme (Bates *et al.*, 2010), which is based on a spatio-temporally staggered discretization and an efficient treatment of the friction slope terms in the momentum Equations (2)–(3). The whole domain of water flows is uniformly discretized into an orthogonal structured computational mesh having rectangular cells. The time increment for the temporal integration is denoted as Δt . The spatial increments in the x and y directions are denoted as Δx and Δy , respectively. The quantity S evaluated at the time $t = k\Delta t$ at the location $(x, y) = (i\Delta x, j\Delta y)$ is denoted as $S_{i,j}^k$, where i, j , and k are integers. The quantities with half-integer sub- and/or super-scripts are defined in the same manner.

Assume that $h_{i,j}^k$, $p_{i+1/2,j}^{k+1/2}$, and $q_{i,j+1/2}^{k+1/2}$ have already been computed at each cell. We then discretize the 2D-LIE to temporally update these quantities in a staggered manner. Firstly, the continuity Equation (1) is discretized as

$$\frac{h_{i,j}^{k+1} - h_{i,j}^k}{\Delta t} + \frac{p_{i+1/2,j}^{k+1/2} - p_{i-1/2,j}^{k+1/2}}{\Delta x} + \frac{q_{i,j+1/2}^{k+1/2} - q_{i,j-1/2}^{k+1/2}}{\Delta y} = r_{i,j}^k + Q_{\text{tr},i,j}^k - e_{i,j}^k \quad (4)$$

Secondly, the momentum Equations (2)–(3) are discretized as

$$\frac{p_{i+1/2,j}^{k+1/2} - p_{i+1/2,j}^{k-1/2}}{\Delta t} + g\hat{h}_{i+1/2,j}^k \left[\frac{(h_{i+1,j}^k + z_{i+1,j}) - (h_{i,j}^k + z_{i,j})}{\Delta x} + \frac{n^2 p_{i+1/2,j}^{k+1/2} |p_{i+1/2,j}^{k-1/2}|}{(\hat{h}_{i+1/2,j}^k)^{10/3}} \right] \quad (5)$$

= 0

and

$$\frac{q_{i,j+1/2}^{k+1/2} - q_{i,j+1/2}^{k-1/2}}{\Delta t} + g\hat{h}_{i,j+1/2}^k \left[\frac{(h_{i,j+1}^k + z_{i,j+1}) - (h_{i,j}^k + z_{i,j})}{\Delta y} + \frac{n^2 q_{i,j+1/2}^{k+1/2} |q_{i,j+1/2}^{k-1/2}|}{(\hat{h}_{i,j+1/2}^k)^{10/3}} \right] \quad (6)$$

= 0.

Here, the non-negativity preserving numerical water depths $\hat{h}_{i+1/2,j}^k$ and $\hat{h}_{i,j+1/2}^k$ are evaluated as

$$\hat{h}_{i+1/2,j}^k = \max\{h_{i,j}^k + z_{i,j}, h_{i+1,j}^k + z_{i+1,j}\} - \max\{z_{i,j}, z_{i+1,j}\} \quad (7)$$

and

$$\hat{h}_{i,j+1/2}^k = \max\{h_{i,j}^k + z_{i,j}, h_{i,j+1}^k + z_{i,j+1}\} - \max\{z_{i,j}, z_{i,j+1}\}, \quad (8)$$

respectively (de Almeida and Bates, 2013). Although the friction term (the last term on the left hand side of Equations (5) and (6)) is discretized in a semi-implicit manner in terms of fluxes $p_{i+1/2,j}^{k+1/2}$ and $q_{i,j+1/2}^{k+1/2}$, they are expressed only by variables at the previous time step; therefore, it does not require any iterative procedures that conventional implicit schemes require. As suggested in the discretized equations, the water depth or the line discharge at each cell is calculated using the information coming from this and neighbor cells. Thereby, the ADU algorithm is easily implemented as detailed below.

AUTOMATIC DOMAIN UPDATING

Structure of the ADU

The ADU is a computational technique recently developed by Tanaka *et al.* (2019) for efficient river flood simulation. Theoretically, this method is applicable to any type of numerical simulation dealing with wave propagation phenomena expressed by hyperbolic partial differential equations with finite difference schemes. The core concept of the algorithm in the ADU is tracing the flood area and its surrounding cells by updating the boundary of the flood area at each time step (see also Figure 1).

The implementation procedure of the ADU is explained as follows:

- (1) To prepare the two vectors $C = (c_1, \dots, c_{KL})$ and $B = (b_1, \dots, b_{KL})$, representing the list of cells for calculation and for boundary updating, respectively. Here, K and L are the total number of cells in the x and y directions, respectively
- (2) To register N_{wet} wet cells at the initial time to C and their surrounding N_b cells to C and B (in the same way from initial time), where ‘‘wet’’ is defined as the status in which water depth is larger than a criterion (0.01 m in this study). This minimum water depth is set to avoid numerical instability caused by a too small

water depth in Equations (8) and (9). The length of B is N_b and that of C is $(N_{\text{wet}} + N_b)$

- (3) To solve Equations (2) and (3) at all the cells belonging to C
- (4) To check all the cells in B ; if b_i becomes wet, to exclude it from the boundary list B . Then, add its surrounding cells to C and B (i.e. simulation domain expands and the boundary is updated accordingly). If it and all its surrounding cells become dry, the cell is excluded from B , and the surrounding ones not included in B are added to C and B (i.e. simulation domain shrinks), otherwise to include it in B again
- (5) To update the total number of boundary cells N_b and the total number of cells for simulation $(N_{\text{wet}} + N_b)$
- (6) To repeat Steps 3 and 4 until the terminal time of computation

As indicated in the above procedures, this algorithm does not require any specific libraries. It is therefore transferable to models in any computer language. This study used a C++ language.

Implementation of ADU on lake flow simulation

The implementation of ADU to river flooding is largely different from that to lake flow simulations due to rainfall and evapotranspiration (right hand side of Equation (1)), while the latter is usually solely by the river discharge from the upstream. Because rainfall and/or evapotranspiration could happen at any cells over the domain, one may have to simulate the whole domain in case of not using the ADU. However, when it comes to floodplains having heavy rainfall and major floods in the same season, such as a semi-arid area having dry and wet seasons, there is still room for applying the ADU to improve computational efficiency, by allowing rainfall and evapotranspiration only in wet cells as demonstrated later in this paper. These effects will be quantitatively evaluated in the following application.

APPLICATION

Study site and simulation setting

The TSL is the largest lake in Southeast Asia, and is one of the largest aquatic environments for ecosystem exposed to social and climate changes (Salmivaara *et al.*, 2016; Chan *et al.*, 2019; Uk *et al.*, 2018). The map of the study area is shown in Figure 2. The total area of the figure is 46,350 km², in which the area surrounded by major highways with higher elevation (20,156 km²) was set as the total simulation domain (where elevation data is shown in Figure 2). The actual simulation domain changes during the simulation by the ADU algorithm as explained above. Eleven tributary rivers flow into the TSL and lake water outflows into the Mekong River through the Tonle Sap River. The lake flow dynamics are strongly affected by the seasonal trend of the flow regimes in the Mekong River. In the rainy season from June to November, backflow occurs from the Mekong River to the TSL due to high water level, while the flooded water recedes to the Mekong River in the dry season from December to May.

The target period covers from July 1998 to December 2003 including a recent devastating flood event in 2000. Rainfall and temperature data are provided from gauge

points (see Figure 2). Evapotranspiration is estimated from monthly temperature using the Thornthwaite method (Singh, 2016). Tributary river discharge data from the previous simulation result (Tanaka *et al.*, 2018b) by the distributed hydrological model Geomorphology-Based Hydrological Model (Yang *et al.*, 2002) is provided as the lateral inflow. The 1-D hydraulic model MIKE11 was used to generate the boundary water stage at the Prek Kdam station (Figure 2) (Fujii *et al.*, 2003). Assuming the mild change in the water depth with a small flow velocity at the initial time of the simulation, the fluxes are set to be zero at the initial time. The initial water depth was set such that water stage becomes 1 m at grids with elevation lower than 1 m (corresponding to the main water body of the lake in the end of the dry season). Different and more realistic initial conditions can be set if such a condition is available based on some observation data, which was not available in our case. The 2D-LIE was numerically discretized with a spatial resolution of 500 m with an orthogonal structured computational mesh and a Manning’s roughness coefficient of $0.03 \text{ m}^{-1/3}$ s, which was used in Tanaka *et al.* (2018b) and confirmed that it reasonably reproduced the observed water stage at the Kampong Luong station and outflow discharge at the Prek Kdam station.

Accuracy

The simulated water levels at the Kampong Luong station with/without the ADU are compared in Figure 3 (a).

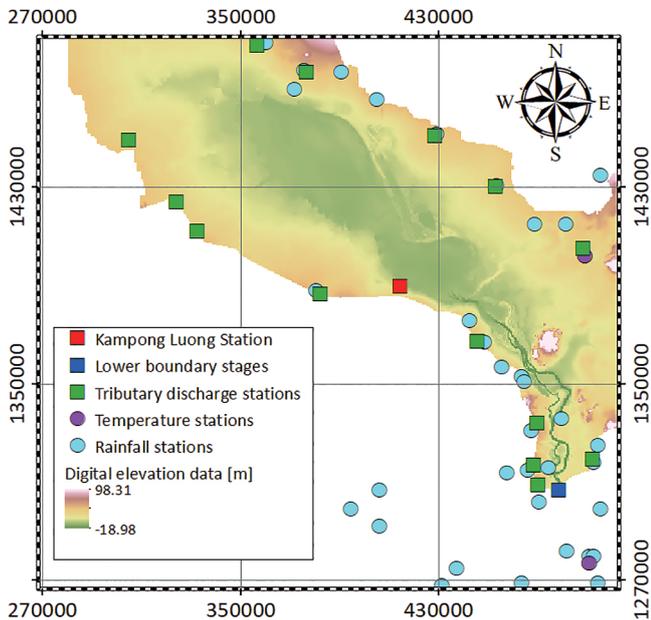


Figure 2. The simulation domain by the 2D-LIE on the projected coordinate VN-2000/UTM zone 48 N (the unit of scale is meter). Rainfall and temperature are observed at the points indicated by the light-blue and purple circles, respectively. Upstream tributary discharges (the green boxes) and downstream boundary water stage at the Prek Kdam station (the blue box) are given as the upstream and downstream boundary conditions, respectively. The data at the Kampong Luong station (the red box) was used for the validation of the 2D-LIE with the ADU

They are so close to each other that the result with the ADU (red) is not visible during most of the simulation period (except for the dry season in 2001 and 2003), showing that the result with the ADU is only slightly smaller (0.15% in the dry season and 0.87% in the rainy season) than that without the ADU. Figure 3 (b) shows the time series of flood area in both simulations with/without the ADU (orange) and only in the simulation without the ADU (blue). A larger difference in the dry season appeared in remote areas far from the main lake where the water is supplied by local rainfall though its area is much smaller than the total flood area.

The average daily rainfall over the whole simulation domain used to drive the 2D-LIE with/without the ADU is

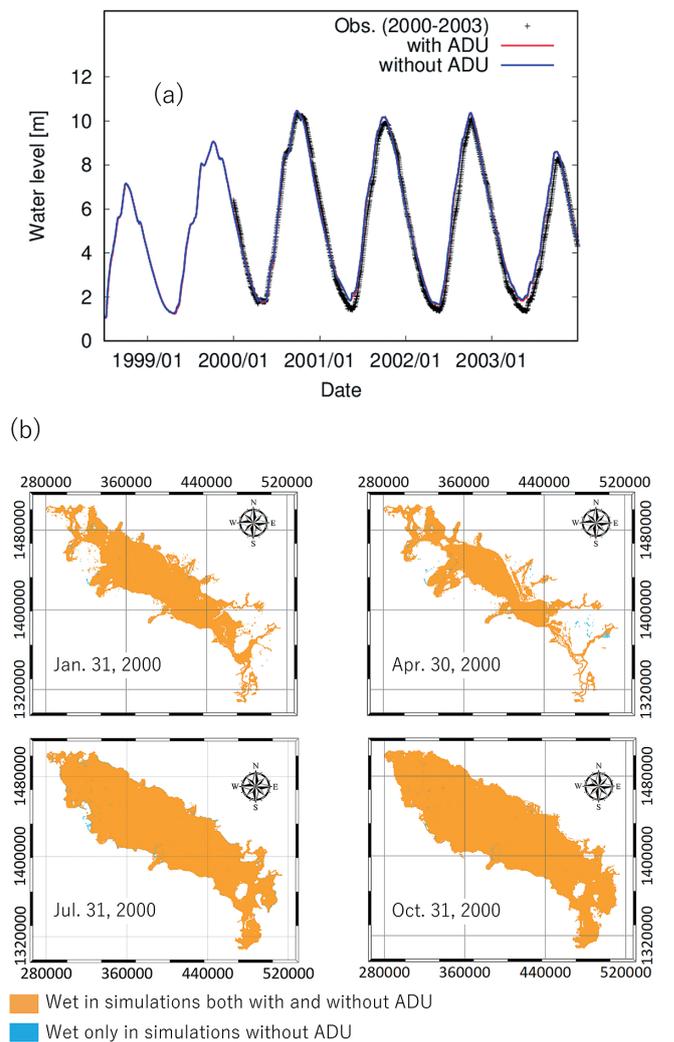


Figure 3. Simulation results with and without the ADU: (a) water stage at the Kampong Luong station (black: observation, red: computation with the ADU, blue: computation without the ADU). The red line (with ADU) is not visible during most of the simulation period except for the dry season in 2001 and 2003. (b) flood areas (blue: only in computation without the ADU; orange: common between simulations with/without the ADU). Wet cells in simulations with/without the ADU overlapped except in the eastern part of the domain in January and April 2000

plotted in Figure 4 (recall that the ADU requires the rainfall data only in the wet cells), showing a clear difference over the simulation period (41.2% on average). The monthly mean of the missing ratio γ of the daily rainfall given to the 2D-LIE with the ADU (γ_{ADU}) to the one without it (γ_{Total}) is defined as follows (Figure 5):

$$\gamma = \frac{\gamma_{Total} - \gamma_{ADU}}{\gamma_{Total}} \quad (9)$$

As shown in Figure 5, the missing ratio goes up to 70% in the later dry season because of a larger amount of rainfall missing on the dry cells during the dry season when the lake, i.e. the simulation domain of the ADU, shrinks. On the other hand, their difference decreases in the later rainy season (October to November) to around 30%. The flood area expands the most in this season; thereby, the simulation domain also expands and the missing ratio becomes smaller. As the lake flow more dynamically changes during the rainy season, the above mechanism is considered to

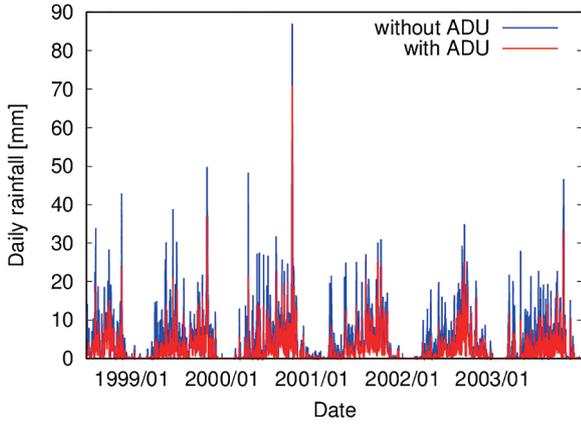


Figure 4. The time series of daily rainfall employed to externally drive the model with (red) and without (blue) the ADU. The ADU (red) missed some part of total rainfall given in the simulation without the ADU (blue)

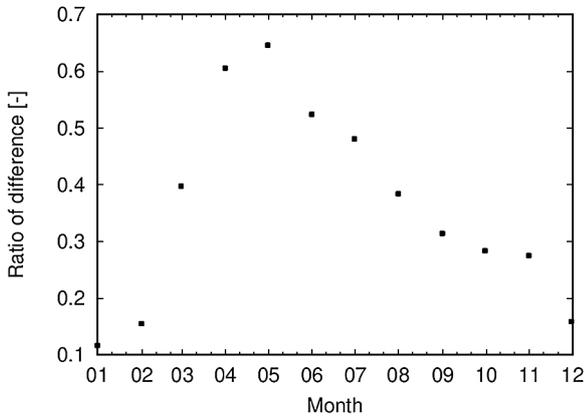


Figure 5. The monthly mean of the ratio of difference in daily rainfall between the 2D-LIE with the ADU to the one without the ADU. Missing ratio of rainfall in ADU is around 0.4 to 0.7 in the dry season and 0.1 to 0.3 in the rainy season

contribute to the smaller impact of missing rainfall on simulation results. Furthermore, this impact on the water stage and/or flood area is particularly small in the TSL, owing to the dominant backflow from the Mekong River during the rainy season.

Computational cost

The 2D-LIE with/without the ADU was compiled with Intel@ C++ compiler 14 and carried out on a Linux machine with an Intel@ Xeon@ processor (CPU E5-2680 2.8 GHz). The computational time for the 5.5-year simulation from July 1998 to December 2003 was 43.0 and 91.7 hours with and without the ADU on a single core, respectively. The ADU thus accelerates the computation by a factor of 2.1. Tanaka *et al.* (2019) demonstrated 5 to 20 times faster simulation in a case study of river flooding in Japan, meaning that the computational efficiency of the ADU for the lake flow is less than those for river flooding. This large difference is simply explained by the ratio of flood area to the whole simulation domain. The flood area in simulating river flooding, particularly in the river confluence, has a variety of spatial distribution under difference scenarios. The flood area in such a case is thus significantly affected by the spatial distribution of rainfall and river defenses. On the other hand, the problem of the lake flow seems not to encounter this issue. Therefore, the simulation domain can be detected faster using the simple dry check. The improvement of the efficiency by the ADU then becomes much smaller. Computational time of each day during the simulation period is plotted with daily flood area in Figure 6. Their daily change is consistent with each other simply because flood area corresponds to the number of cells for computation. The effect of the ADU is clear in the dry season when flood area is smaller. The 2D-LIE with the ADU is 1.8 times and 1.4 times faster than without the ADU on 8 and 48 cores, respectively. Therefore, the ADU more significantly improves computational efficiency of the 2D-LIE as the computational environment becomes less massive. Our computational results suggest that the ADU is still an effective practical method for flood simulation involving

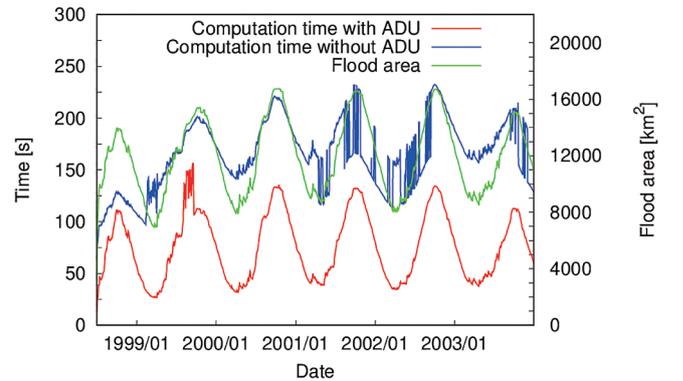


Figure 6. Time series of computation time at each time step in the 2D-LIE with (red) and without (blue) the ADU and that of flood area (green). Computation time of each time step is consistent with flood area (green) for both simulations while its time is shorter in computation with the ADU (red)

lake flow dynamics especially under limited computational resources. In summary, we find that the ADU algorithm enhances the numerical modelling of lake flow dynamics by significantly reducing the computational burden in exchange for the small reduction in accuracy in case of the large-scale lake flow simulation in semi-arid areas compared with conventional models.

CONCLUSIONS

The 2D-LIE enhanced with the ADU was applied to simulating surface water flows in and around the TSL in Cambodia. A theoretical estimate of the improvement of computational efficiency by the ADU was derived. The obtained computational results demonstrated that the total amount of rainfall used to drive the flow dynamics becomes much smaller with the ADU, mainly in the latter dry season and early rainy season during which the amount of rainfall is relatively small. Furthermore, the flow dynamics of the TSL in the rainy season is strongly affected by backflow from the Mekong River, which is qualitatively the same as river flooding. As a result, despite the significant ignorance of the rainfall in the model with the ADU, the 2D-LIE with/without the ADU are in good agreement in terms of the water stage and flood area. Note that this result is strongly affected by unique hydroclimatic features in large-scale lakes in semi-arid areas, and applicability of the ADU to areas driven by different mechanisms should be examined in future.

Future applications are expected in the assessment of the lake flow dynamics under climate change and/or anthropological impacts (e.g. dam operation in the Mekong River) scenarios, which require a computationally efficient model to complete a vast number of ensemble simulations (Liu *et al.*, 2018; Tanaka *et al.*, 2018a). The 2D-LIE with the ADU developed in this study can be used to address this topic. As the ADU itself is a technique to dynamically control the simulation domain, it is potentially applicable to various numerical models of flow simulations of importance in civil and environmental engineering. Such examples include the full shallow water equations, their non-Newtonian extension to debris flows (Ruiz-Villanueva *et al.*, 2019; Han *et al.*, 2019) and/or coupled solute transport phenomena (Liu, 2019), and multi-scale modelling (Li and Hodges; 2019, Fukui *et al.*, 2019). Future research would examine the ADU against these flow and transport phenomena to evaluate its computational performance in more detail.

ACKNOWLEDGMENTS

This research was funded by the SATREPS project “Establishment of Environmental Conservation Platform of Tonle Sap Lake”. Sincere appreciation goes to Dr. Sarann Ly for providing observed water stage data for the validation of the 2D-LIE with the ADU.

REFERENCES

- Bates PD, Horritt MS, Fewtrell TJ. 2010. A simple inertial formulation of the shallow water equations for efficient two-dimensional flood inundation modelling. *Journal of Hydrology* **387**: 33–45. DOI: 10.1016/j.jhydrol.2010.03.027.
- Castro-Orgaz O, Hager WH. 2019. *Shallow Water Hydraulics*. Springer, Basel, Switzerland.
- Chan B, Sor R, Ngor PB, Baehr C, Lek S. 2019. Modelling spatial and temporal dynamics of two small mud carp species in the Tonle Sap flood-pulse ecosystem. *Ecological Modelling* **392**: 82–91. DOI: 10.1016/j.ecolmodel.2018.11.007.
- de Almeida GAM, Bates P. 2013. Applicability of the local inertial approximation of the shallow water equations to flood modeling. *Water Resources Research* **49**: 4833–4844. DOI: 10.1002/wrcr.20366.
- Fujii H, Garsdal H, Ward P, Ishii M, Morishita K, Boivin T. 2003. Hydrological roles of the Cambodian floodplain of the Mekong River. *International Journal of River Basin Management* **1**: 253–266. DOI: 10.1080/15715124.2003.9635211.
- Fukui N, Prasetyo A, Mori N. 2019. Numerical modeling of tsunami inundation using upscaled urban roughness parameterization. *Coastal Engineering* **152**: 103534. DOI: 10.1016/j.coastaleng.2019.103534.
- Han Z, Su B, Li YG, Wang W, Wang WD, Huang JL, Chen GQ. 2019. Numerical simulation of debris-flow behavior based on the SPH method incorporating the Herschel-Bulkley-Papanastasiou rheology model. *Engineering Geology* **255**: 26–36. DOI: 10.1016/j.enggeo.2019.04.013.
- Hunsberger C, Work C, Herre R. 2018. Linking climate change strategies and land conflicts in Cambodia: Evidence from the Greater Aural region. *World Development* **108**: 309–320. DOI: 10.1016/j.worlddev.2018.02.008.
- Hunter NM, Bates PD, Horritt MS, Wilson MD. 2007. Simple spatially-distributed models for predicting flood inundation: A review. *Geomorphology* **90**: 208–225. DOI: 10.1016/j.geomorph.2006.10.021.
- Ikeuchi H, Hirabayashi Y, Yamazaki D, Kiguchi M, Koirala S, Nagano T, Kotera A, Kanae S. 2015. Modeling complex flow dynamics of fluvial floods exacerbated by sea level rise in the Ganges–Brahmaputra–Meghna Delta. *Environmental Research Letters* **10**: 124011. DOI: 10.1088/1748-9326/10/12/124011.
- Kummu M, Tes S, Yin S, Adamson P, Jozsa J, Koponen J, Richey J, Sarkkula J. 2014. Water balance analysis for the Tonle Sap Lake-floodplain system. *Hydrological Processes* **28**: 1722–1733. DOI: 10.1002/hyp.9718.
- Li Z, Hodges BR. 2019. Modeling subgrid-scale topographic effects on shallow marsh hydrodynamics and salinity transport. *Advances in Water Resources* **129**: 1–15. DOI: 10.1016/j.advwatres.2019.05.004.
- Liu WB, Lim WH, Sun FB, Mitchell D, Wang H, Chen DL, Bethke I, Shioyama H, Fischer E. 2018. Global freshwater availability below normal conditions and population impact under 1.5 and 2 °C stabilization scenarios. *Geophysical Research Letters* **45**: 9803–9813. DOI: 10.1029/2018gl078789.
- Liu X. 2019. A robust numerical model for shallow water governing solute transport with wet/dry interfaces. *Computer Methods in Applied Mechanics and Engineering* **351**: 85–108.

- DOI: 10.1016/j.cma.2019.03.026.
- Luo P, Kang S, Apip K, Zhou M, Lyu J, Aisyah S, Binaya M, Regmi RK, Nover D. 2019. Water quality trend assessment in Jakarta: A rapidly growing Asian megacity. *Plos One* **14**: e0219009. DOI: 10.1371/journal.pone.0219009.
- Luo P, Mu D, Xue H, Ngo-Duc T, Dang-Dinh K, Takara K, Nover D, Schladow G. 2018. Flood inundation assessment for the Hanoi Central Area, Vietnam under historical and extreme rainfall conditions. *Scientific Reports* **8**: 12623. DOI: 10.1038/s41598-018-30024-5.
- Martins R, Leandro J, Chen AS, Djordjevic S. 2017. A comparison of three dual drainage models: shallow water vs local inertial vs diffusive wave. *Journal of Hydroinformatics* **19**: 331–348. DOI: 10.2166/hydro.2017.075.
- Mateo CMR, Yamazaki D, Kim H, Champathong A, Vaze J, Oki T. 2017. Impacts of spatial resolution and representation of flow connectivity on large-scale simulation of floods. *Hydrology and Earth System Sciences* **21**: 5143–5163. DOI: 10.5194/hess-21-5143-2017.
- Medeiros SC, Hagen SC. 2013. Review of wetting and drying algorithms for numerical tidal flow models. *International Journal for Numerical Methods in Fluids* **71**: 473–487. DOI: 10.1002/fld.3668.
- Mignot E, Li XF, Dewals B. 2019. Experimental modelling of urban flooding: A review. *Journal of Hydrology* **568**: 334–342.
- Ruiz-Villanueva V, Mazzorana B, Blade E, Burkli L, Iribarren-Anacon P, Mao L, Nakamura F, Ravazzolo D, Rickenmann D, Sanz-Ramos M, Stoffel M, Wohl E. 2019. Characterization of wood-laden flows in rivers. *Earth Surface Processes and Landforms* **44**: 1694–1709. DOI: 10.1002/esp.4603.
- Salmivaara A, Kumm M, Varis O, Keskinen M. 2016. Socio-Economic changes in Cambodia's unique Tonle Sap Lake area: A spatial approach. *Applied Spatial Analysis and Policy* **9**: 413–432. DOI: 10.1007/s12061-015-9157-z.
- Seyoum SD, Vojinovic Z, Price RK, Weesakul S. 2012. Coupled 1D and noninertia 2D flood inundation model for simulation of urban flooding. *Journal of Hydraulic Engineering* **138**: 23–34. DOI: 10.1061/(ASCE)Hy.1943-7900.0000485.
- Singh VP. 2016. Handbook of Applied Hydrology, 2nd Edition. McGraw-Hill Education, New York, USA.
- Tanaka T, Tachikawa Y, Ichikawa Y, Yorozu K. 2018a. Flood risk curve development with probabilistic rainfall modelling and large ensemble climate simulation data: a case study for the Yodo River basin. *Hydrological Research Letters* **12**: 28–33. DOI: 10.3178/hrll.12.28.
- Tanaka T, Yoshioka H, Siev S, Fujii H, Fujihara Y, Hoshikawa K, Ly S, Yoshimura C. 2018b. An Integrated Hydrological-Hydraulic Model for Simulating Surface Water Flows of a Shallow Lake Surrounded by Large Floodplains. *Water* **10**: 1213. DOI: 10.3390/w10091213.
- Tanaka T, Tachikawa Y, Ichikawa Y, Yorozu K. 2019. An automatic domain updating method for fast 2-dimensional flood-inundation modelling. *Environmental Modelling & Software* **116**: 110–118. DOI: 10.1016/j.envsoft.2019.02.018.
- Teng J, Jakeman AJ, Vaze J, Croke BFW, Dutta D, Kim S. 2017. Flood inundation modelling: A review of methods, recent advances and uncertainty analysis. *Environmental Modelling & Software* **90**: 201–216. DOI: 10.1016/j.envsoft.2017.01.006.
- Uk S, Yoshimura C, Siev S, Try S, Yang H, Oeurng C, Li S, Hul S. 2018. Tonle Sap Lake: Current status and important research directions for environmental management. *Lakes & Reservoirs: Research & Management* **23**: 177–189. DOI: 10.1111/lre.12222.
- Ung P, Peng C, Yuk S, Tan R, Ann V, Miyanaga K, Tanji Y. 2019. Dynamics of bacterial community in Tonle Sap Lake, a large tropical flood-pulse system in Southeast Asia. *Science of the Total Environment* **664**: 414–423. DOI: 10.1016/j.scitotenv.2019.01.351.
- Yamaguchi S, Ikeda T, Iwamura K, Naono K, Ninomiya A, Tanaka K, Takahashi H. 2007. Development of GIS-based flood-simulation software and application to flood-risk assessment. ed. *2nd IMA International Conference on Flood Risk Assessment*, September 4–5, 2007 Plymouth, UK.
- Yamazaki D, de Almeida GAM, Bates PD. 2013. Improving computational efficiency in global river models by implementing the local inertial flow equation and a vector-based river network map. *Water Resources Research* **49**: 7221–7235. DOI: 10.1002/wrcr.20552.
- Yang D, Herath S, Musiak K. 2002. A hillslope-based hydrological model using catchment area and width functions. *Hydrological Sciences Journal* **47**: 49–65. DOI: 10.1080/02626660209492907.