A Numerical Algorithm and Visualization Software for Flood Simulation in Urban Area: A Case Study of West Jakarta, Indonesia

Saeful Bahtiar, Somporn Chuai-Aree., and Anurak Busaman

***Abstract*—**Flood is an overflow of a large amount of water beyond its normal limits, especially in the area of normally dry land which can cause many losses. To minimize those effects and to predict flood occurrence, simulation and mathematical models are required. The purpose of this paper is to develop a numerical algorithm for flood simulation and visualization in West Jakarta, Indonesia. The numerical algorithm was constructed based on shallow water equations (SWEs) that were solved by using finite volume method (FVM) and first order well-balanced scheme equipped with the dynamic domain defining method. The numerical algorithm was validated in the scenario of Kleefsman's dam breaking test case. The result of validation shows that experiments have good agreement with the simulation data. The numerical algorithm and investigated software were applied to terrain data of West Jakarta, Indonesia. The simulated results were compared for two cases by involving with buildings and without buildings. The comparison shows that the algorithm with buildings can reduce the number of computational grid cells since the buildings grid cells were ignored. This algorithm and the software can be applied for other regions.

***Keywords*—**Finite volume method, First order well-balanced scheme, Flood simulation, Shallow water equations.

I. Introduction

Flooding is a serious problem which occurs in many countries over the world. When flooding occurs, they often lead to environmental damages, economic damages and loss of human lives. Flood disasters cannot be terminated completely however the adverse effect of flooding can be minimized with the computer simulation. As a result of these damages, several researches have been carried out on flooding in urban area using numerical modeling [3, 6, 13, 14] which is considered as an effective and important tool to reduce risk, to prepare, to plan evacuation and also to simulate the forecast of flood warning.

The shallow water equations are some of the mathematical models commonly used to simulate flood in urban area from the river [6], the sea [4] and; dam break [11, 12, 16]. They further provided information needed on water depth, water height, and the velocity of water flow. To get the solutions by using numerical methods of shallow water equations, analytical computation is hard to obtain.

The finite volume method is the numerical method which is often adapted to solve shallow water equations as it can solve the simple and complex domain grid cells [16]. The simulation of water flow using standard numerical method may fail particularly in the area with different slope of topography. Some areas may sometimes lead to dry or wet environments and produce negative water depth. This problem can be overcome by using the well-balanced schemes in [1, 2, 8]. Audusse [1] and Busaman [3] used a first order scheme to solve the well-balanced scheme based on the hydrostatic reconstruction.

To solve the shallow water equations using the numerical methods, normally we need the high performance computing with all grids. The wet and dry grid cells will be considered during simulation time which took more time and costly to consider. The dynamic domain defining method often used to reduce computational cost is employed in this study based on the work of Busaman [3] that only considers the wet grid area and automatically flows into the dry grid cells.

The purpose of this paper is to develop a numerical algorithm by considering with buildings and without buildings for flood simulation and visualization in urban area. In this study, West Jakarta, Indonesia is considered as the worst case of flooding was recorded in this region in 2007 [10]. This region further has small slope of topography which fits to the aim of this research. The developed numerical algorithm was constructed based on the finite volume method and first order well-balanced scheme [1], and equipped with the dynamic domain defining method [3, 15] based on structure grid method for solving shallow water equations.

The paper is organized into 7 sections. Section 2 presents the model based on shallow water equations. Section 3 explains about the numerical solution obtained. In section 4, the validation model test with an experiment is discussed. Section 5 illustrates the application to city flood simulation with the involvement of buildings and without buildings. Section 6 shows the result and discussion of the simulation and

Saeful Bahtiar is with the Department of Applied Mathematics and Computer Science, Faculty of Science and Technology, Prince of Songkla University, Pattani, Thailand (phone: (+62) 082112585202; fax: -; e-mail: saefulbahtiar250889@gmail.com).

Somporn Chuai-Aree and Anurak Busaman were with the Department of Applied Mathematics and Computer Science, Faculty of Science and Technology, Prince of Songkla University, Pattani, Thailand (phone: (+66) 804261112; fax: -; e-mail: schuaiaree@gmail.com and anulove108@hotmail.com).

section 7 summarizes the conclusion of this study.

II. SHALLOW WATER EQUATIONS

In this paper, the shallow water equations are used to describe flood inundation in urban area. The shallow water equations are shown as follows:

$$\frac{\partial}{\partial t}\begin{bmatrix} h \\ uh \\ vh \end{bmatrix} + \frac{\partial}{\partial x}\begin{bmatrix} uh \\ u^2h + \frac{1}{2}gh^2 \\ uvh \end{bmatrix} + \frac{\partial}{\partial y}\begin{bmatrix} vh \\ uvh \\ v^2h + \frac{1}{2}gh^2 \end{bmatrix} = \begin{bmatrix} 0 \\ -gh\frac{\partial z}{\partial x} \\ -gh\frac{\partial z}{\partial y} \end{bmatrix} + \begin{bmatrix} R \\ -S_{fx} \\ -S_{fy} \end{bmatrix}, \quad (1)$$

where h is the water depth (m); uh and vh are water discharges (m^3/s) per unit width with velocity components u and v in the x and y axis, respectively; z is the height of topography (m);. S_{fx} and S_{fy} are the bed friction forces in x- and y-directions, it can be estimated by using the Manning's resistance law by given $S_{fx} = gn^2 uh\sqrt{u^2+v^2}\,h^{-\frac{4}{3}}$ and $S_{fy} = gn^2 vh\sqrt{u^2+v^2}\,h^{-\frac{4}{3}}$, where n is Manning's resistance coefficient; R is the lateral inflow, which consists of the addition of rainfall rate into the river; g is the gravity constant (m/s²). The (1) can be written in the matrix form as:

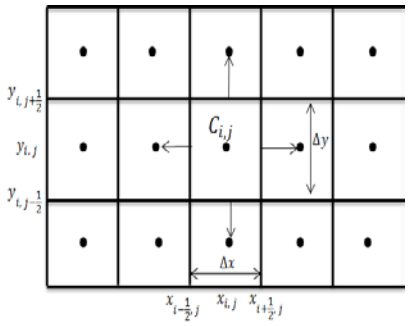
$$\frac{\partial \vec{q}}{\partial t} + \frac{\partial \vec{f}(\vec{q})}{\partial x} + \frac{\partial \vec{g}(\vec{q})}{\partial y} = \vec{z} + \vec{s}, \quad (2)$$

where $\vec{q} = [h \quad uh \quad vh]^T$ is the dependent variable; $\vec{f}(\vec{q})$ and $\vec{g}(\vec{q})$ are the flux functions in x and y directions, respectively. The right hand side of the (2) is the source term consisting of the gravity force $\vec{z}$ and the friction force $\vec{s}$.

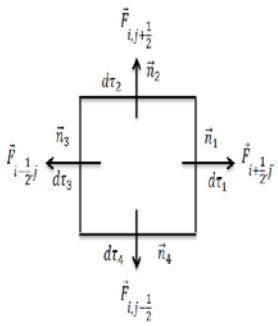
III. NUMERICAL METHOD

This section presents the numerical methods which generate the solution of the model by using finite volume method with the first order well-balanced scheme based on Audusse [1] work for solving the (2).

A. Finite Volume Method



(a) Cartesian in 2D grid.



(b) Structure grid.

Fig. 1 The notation computational grid cell.

The finite volume method performs on a discretized domain that consists of structured grid cells as shown in Fig. 1. By integrating (2) over an arbitrary cell domain Ω having the boundary τ, using Green's theorem, Audusse's [1] scheme and finally the equation is divided by the cell area $\Delta x \Delta y$, then we have

$$\frac{\partial \vec{Q}_{i,j}}{\partial t} = -\frac{1}{\Delta x \Delta y}\oint_{\tau} \vec{F}\cdot\vec{n}\,d\tau + \vec{Z}_{i,j} + \vec{S}_{i,j}, \quad (3)$$

where $\vec{Q}_{i,j}$, $\vec{Z}_{i,j}$ and $\vec{S}_{i,j}$ are the average of $\vec{q}_{i,j}$, $\vec{z}_{i,j}$ and $\vec{s}_{i,j}$ over the cell, respectively. $\vec{F} = \left[\vec{f}(\vec{q}) \quad \vec{g}(\vec{q})\right]^T$ is the flux function, $\vec{n}$ is the outward normal vector of the cell boundary; i and j denote the spatial index of the cell. By using the Audusse's [1] scheme, the gravity force $\vec{z}$ can be distributed to the numerical flux for each sub interfaces, the discrete form of the (3) is given by:

$$\frac{\partial \vec{Q}_{i,j}}{\partial t} = -\frac{1}{\Delta x \Delta y}\sum_{k=1}^{kn}\left(\vec{F}_k\cdot\vec{n}_k\right)d\tau_k + \vec{S}_{i,j}, \quad (4)$$

where k is the index of the sub-interface k, and $d\tau_k$ is size of the sub-interface between the cell and the neighbor cell k, kn is the total number of edges in the cell. Since the computing model considers structured grid cells as shown in Fig. 1(b) we can write,

$$\frac{\partial \vec{Q}_{i,j}}{\partial t} = -\frac{1}{\Delta x \Delta y}\sum_{k=1}^{4}\left(\vec{F}_k\cdot\vec{n}_k\right)d\tau_k + \vec{S}_{i,j}, \quad (5)$$

In order to keep the scheme as simple as the first order, Euler's method for the time derivative is used, the (5) can be discretized as:

$$\vec{Q}_{i,j}^{n+1} = \vec{Q}_{i,j}^{n} - \frac{\Delta t}{\Delta x}\left(\vec{F}_{i+\frac{1}{2},j} - \vec{F}_{i-\frac{1}{2},j}\right) - \frac{\Delta t}{\Delta y}\left(\vec{G}_{i,j+\frac{1}{2}} - \vec{G}_{i,j-\frac{1}{2}}\right) + \vec{S}_{i,j}^{n}, \quad (6)$$

where Δt is time step size; $\left(i\pm\frac{1}{2}, j\right)$ and $\left(i, j\pm\frac{1}{2}\right)$ are the indices for the walls of boundary τ. $\vec{Q}_{i,j}$ represents a cell average value over the grid cell at (i,j) at time t_n. $\vec{F}_{i,j}$ and $\vec{G}_{i,j}$ are the numerical fluxes, depending upon the chosen scheme

In the (6), $\vec{F}_{i\pm\frac{1}{2},j}$ and $\vec{G}_{i,j\pm\frac{1}{2}}$ can be calculated by using the formula based on the Harten, Lax and Van Leer (HLL) [5] as:

$$\vec{F}_{i\pm\frac{1}{2},j} = \frac{F_L\,\alpha^+ - F_R\,\alpha^- + \alpha^-\alpha^+\left(U_R - U_L\right)}{\left(\alpha^+ - \alpha^-\right)} + \vec{L}^{x}_{i\pm\frac{1}{2},j}, \quad (7)$$

$$\vec{G}_{i,j\pm\frac{1}{2}} = \frac{G_D\,\beta^+ - G_U\,\beta^- + \beta^+\beta^-\left(U_U - U_D\right)}{\left(\beta^+ - \beta^-\right)} + \vec{L}^{y}_{i,j\pm\frac{1}{2}}, \quad (8)$$

where F and G are flux functions of U; α^+, α^-, β^+ and

β^- are the wave speeds computed based on the work of Kurganov [9] as follows:

$$\alpha^{\pm} = \pm\max\left(\pm u_L + \sqrt{g\hat{h}_L}, \pm u_R + \sqrt{g\hat{h}_R}, 0\right), \quad (9)$$

$$\beta^{\pm} = \pm\max\left(\pm v_D + \sqrt{g\hat{h}_D}, \pm v_U + \sqrt{g\hat{h}_U}, 0\right), \quad (10)$$

where $u \pm \sqrt{g\hat{h}}$ and $v \pm \sqrt{g\hat{h}}$ are the largest and smallest eigenvalues of the Jacobian matrix of $\frac{\partial \vec{f}(\vec{U})}{\partial \vec{U}}$ and $\frac{\partial \vec{g}(\vec{U})}{\partial \vec{U}}$ respectively; $\hat{h}$ is the depth of water flow. $\vec{L}^x_{i\pm\frac{1}{2},j}$ and $\vec{L}^y_{i,j\pm\frac{1}{2}}$ are the terms to satisfy the balance of momentum flux and momentum gravity forces. To solve the well-balanced scheme, we used the first order Audusse's [1] scheme in this study which given by:

$$\vec{L}^x_{i\pm\frac{1}{2},j} = \begin{bmatrix} 0 & \frac{g}{2}\hat{h}^2_{i\pm\frac{1}{2},j^L_R} & 0 \end{bmatrix}^T,$$
$$\vec{L}^y_{i,j\pm\frac{1}{2}} = \begin{bmatrix} 0 & 0 & \frac{g}{2}\hat{h}^2_{i,j\pm\frac{1}{2}^D_U} \end{bmatrix}^T \quad (11)$$

where $\hat{h}$ can be compute based on Audusse's [1] scheme for preserving the lake at rest condition and guarantee the non-negativity water depth. It can be expressed by using hydrostatic reconstruction as:

$$\hat{h}_{i\pm\frac{1}{2},j^L_R} = \max\left(0, h_{i,j} + z_{i,j} - \max\left(z_{i,j}, z_{i\pm1,j}\right)\right), \quad (12)$$

$$\hat{h}_{i,j\pm\frac{1}{2}^D_U} = \max\left(0, h_{i,j} + z_{i,j} - \max\left(z_{i,j}, z_{i,j\pm1}\right)\right), \quad (13)$$

where h is water depth on the surface, and z is height of topography.

B. Source Terms

The source term consists of the rate of the river's water level increases by rainfall rate and two friction forces. For the increasing rainfall rate ($R^n_{i,j}$), the water depth is vertically added by rainfall rate to control volume per time unit, we update a new value of $h^{n+1}_{i,j}$ in Eq. (6) in vector form, as:

$$h^{n+1}_{i,j} = h^*_{i,j} + \Delta t R^n_{i,j}, \quad (14)$$

where $h^*_{i,j}$ is the result of calculation from (6) by using finite volume method in the part of water depth. $\Delta t R^n_{i,j}$ is the average rate of water level increasing by rainfall rate, in each (i, j) in the range of time step from n to $n+1$. For two friction forces can be computed by a semi-implicit method to ensure stability criteria and in order to preserve the steady states. It updates a new value of $uh^{n+1}_{i,j}$ as follows:

$$uh^{n+1}_{i,j} = \frac{uh^*_{i,j}}{\left(1 + \frac{\Delta t g n^2 \sqrt{\left(u^n\right)^2 + \left(v^n\right)^2}}{\left(u^{n+1}\right)^{\frac{4}{3}}}\right)}, \quad (15)$$

where $uh^*_{i,j}$ same as $h^*_{i,j}$ in the part of water discharge in x direction. Similarly, $vh^{n+1}_{i,j}$ can be computed by following (15) using $vh^{n+1}_{i,j}$ instead of $uh^{n+1}_{i,j}$.

C. Stability Condition

In the explicit scheme, we need a proper time step to maintain stability, and the Courant Friedrichs Lewy (CFL) condition is used. The stability criterion of an explicit numerical scheme condition is used to estimate the time step:

$$\Delta t \le 0.5\frac{\Delta A_{\min}}{\lambda_{\max}}, \quad (16)$$

where Δt is the time step size**.** In the above equation $\Delta A_{\min} = \min\left(\Delta x_{i,j}, \Delta y_{i,j}\right), (i,j) \in D$, where D is the computational domain, while $\lambda_{\max}$ is the maximum absolute value of all the wave speeds in the computational domain which is the maximum of $\left(\alpha^+, \alpha^-, \beta^+, \beta^-\right)$ for all the interfaces of all the cells.

D. Boundary Condition

In this section, the closed and opened boundaries applied in the algorithm are defined. The closed boundary condition was used in the experimental real data and can be expressed as:

$$h = 0,\ uh = 0,\ vh = 0,\ z = z_{large}, \quad (17)$$

where z_{large} is a large constant value for the topography height. The opened boundary condition was applied to the flood simulation, and can be given by:

$$h_{m+1} = h_m,\ uh_{m+1} = uh_m,\ vh_{m+1} = vh_m,\ z_{m+1} = z_m. \quad (18)$$

In the (17) and (18), 0 and $m+1$ are referred to as ideal cells outside the computational domain. And $\forall(x, y) \in \partial\Omega$.

E. Topography Interpolation

Shuttle radar topography mission (SRTM) data is used in this study, in which data is generated by using digital terrain data grids. The data presented in this form contained the grid cell size lengths of 90 meters each. Since the grid size of 90 meters is too coarsely, the interpolation of the topography is needed. We use bilinear interpolation to obtain the higher resolution of topography. The approximated value of the cell at (*i, j*) is obtained by

$$z_i = z_{x',y'}\left(x' + 1 - m\right)\left(y' + 1 - n\right) + z_{x'+1,y'}\left(m - x'\right)\left(y' + 1 - n\right)$$

$$+z_{x^{'},y^{'}+1}\left(x^{'}+1-m\right)\left(n-y^{'}\right)+z_{x^{'}+1,y^{'}+1}\left(m-x^{'}\right)\left(n-y^{'}\right), \quad (19)$$

where $\left(x^{'},y^{'}\right)$ is the position of the topographic data grid cell used for approximation with $x^{'}=\lfloor m \rfloor$ and $y^{'}=\lfloor n \rfloor$

$$m=x_i\times\frac{n^{x^{'}}}{n^{x}} \text{ and } n=y_i\times\frac{n^{y^{'}}}{n^{y}}, \quad (20)$$

The m and n are the mapped indices of the computational grid cell at position $\left(x_i,y_i\right)$. The $n^{x^{'}}$ and $n^{y^{'}}$ are numbers of columns and rows of the topographic data grid, and n^{x} and n^{y} are numbers of columns and rows of the finest computational grid.

F. Dynamic Domain Defining Method

In order to reduce the computational cost to a minimum, the dynamic domain defining method is used. The dynamic domain defining method calculation commences from wet grid cells and will compute automatically into dry grid cells area. However this is dependent on the water flow. The wet grid cells area was considered during the simulation in this study.

G. Algorithm Overview

We present here an overview of the algorithm for the computational scheme presented previously. The developed algorithm consists of several steps describing the calculation procedures, as illustrated in Fig. 2. The detail for each step is described as follows:

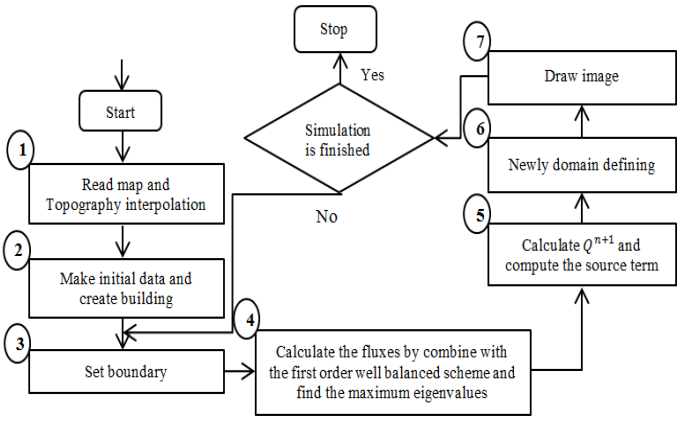


Fig. 2 Algorithm overview.

IV. Dam Break Experiment

The experiment was solved by using FVM with the first order well-balanced scheme. Fig. 3 shows the geometry of the physical experiment, where H_1, H_2, H_3, and H_4 are corresponding to the vertical probes. All dimensions are in meters.

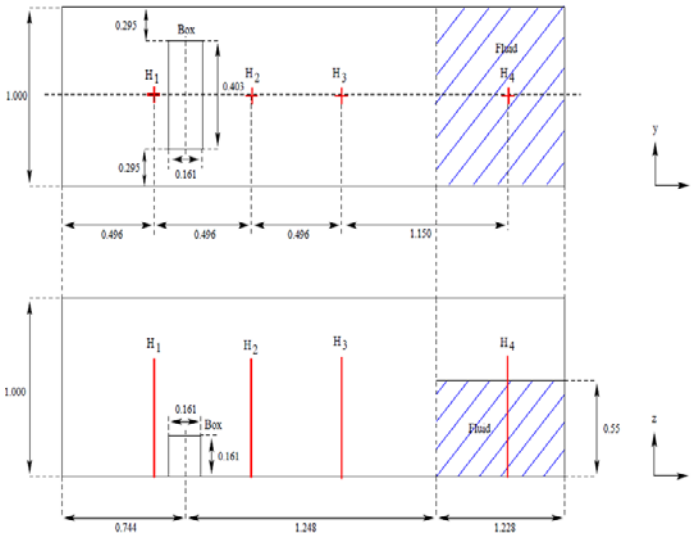
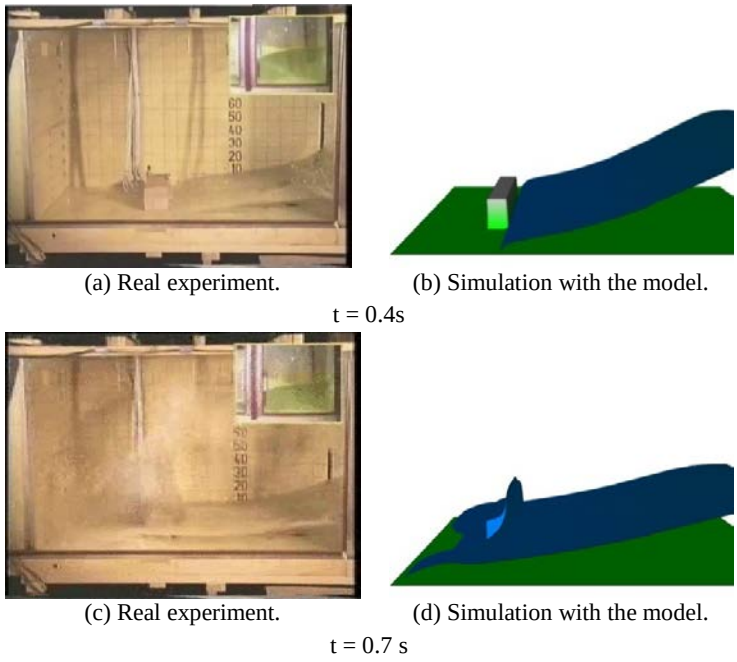


Fig. 3 The geometry of the physical experiment of Kleefsman's (2006).

In this simulation, we used 250x100 grid cells, the duration of this simulation is 6s, carried out with Manning's coefficient of 0.032 and CFL conditions set to 0.5. The simulation results of the behavior of water flow at time *t=0.4* and *0.7*s are shown in Fig. 4. The 3D representation of the result was obtained from the Kleefsman's experiment (2006)[7] shows a good agreement for our numerical solution.



(a) Real experiment. (b) Simulation with the model.

t = 0.4s

(c) Real experiment. (d) Simulation with the model.

t = 0.7 s

Fig. 4 The 3D (a and c) represents the dam-break of Kleefsman's experiment (2006) and (a and c) experiment using shallow water equations.

Fig. 5 shows the results of the comparison of the water depth at each the vertical probes in the first order scheme and the Kleefsman's [7] experiment. The results obtained show that it is close to the experimental data.

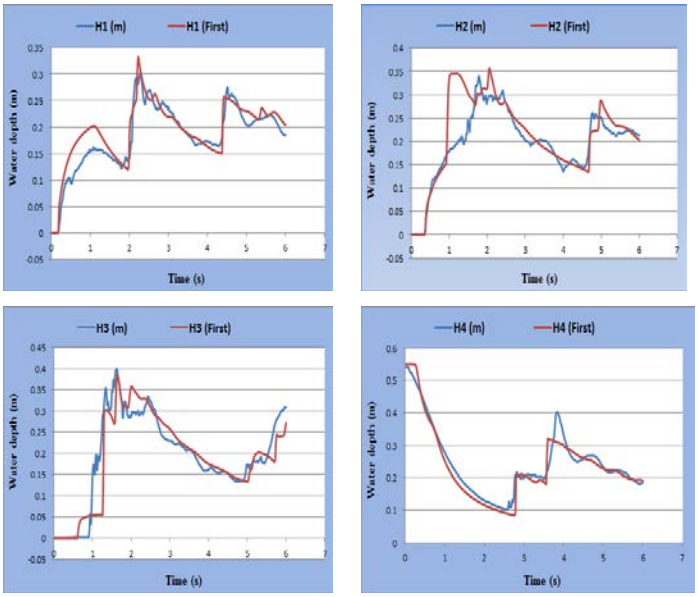
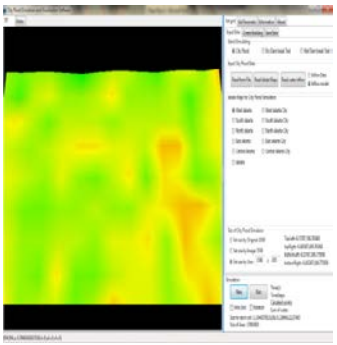
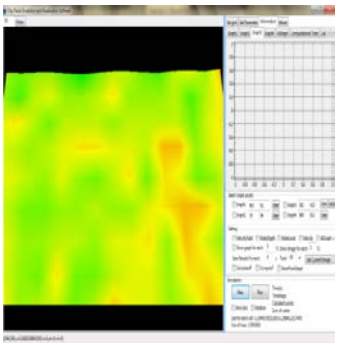


Fig. 5 The comparison of the water depth at the reservoir H1 (bottom-right), H2 (bottom-left), H3 (top-right) and H4 (top-left). The first order scheme (red) and experiment results (blue).

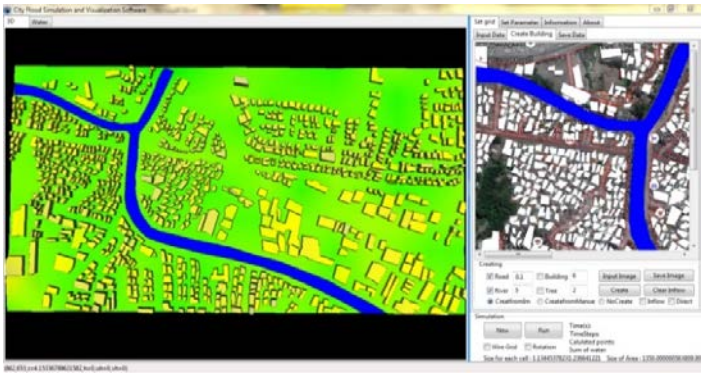
V. The Characteristics of a City Flood Simulation with Buildings and Without Buildings Helpful



(a) Set grid and parameter.



(a) Set grid and parameter.



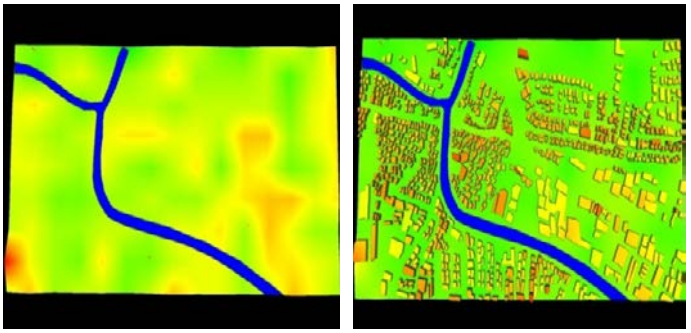
(c) Create urban data.

Fig. 6. Process of creating the urban data.

To simulate the flooding in west Jakarta, It needs to create an urban data using SRTM data contained only terrain topography data without building. For the urban data, it is needed to create the buildings, river and road data suitable to the position of each building. This can be done by considering the map in Google Earth for all objects in the simulation (building, river and road) related to the latitude and longitude of the topography. We develop the software based on Pascal language in Lazarus software to create urban data (building, river and road). Before creating the urban data, we have to set up the grid cells and information needed as shown in Figs. 6(a) and 6(b). Then we can make the urban data to draw the building, river and road in the Google Earth which has been mapped to the accurate position of latitude and longitude the data used as shown in Fig. 6(c) that has been processed (left) and input picture (right).

VI. Result and Discussion

The considered simulation area is one part of West of Jakarta in Indonesia. This area was flooded occurrence every year and also has a small slope of topography. The source of water in this area is from the river after heavy rainfall. The depth of the river is assumed to be 5 meters and the initial water depth in the river to be 4 meters. In this study, we assume that the river increases by rain by 0.0005 m^3/s with use the simulation time 1800s. The size of the grid is 1190 x 655 cells. Six points, P1, P2, P3, P4, P5 and P6 were selected to measure the results of flood information in each position. The position of each point was distributed to position of in front of the building, behind the building, between the building with many buildings and a few buildings. The selected points are shown in Fig. 7



(a) Points without buildings. (b) Points with buildings.

Fig. 7. The positions of the points for measure the parameters.

The positions of the selected points P1, P2, P3, P4, P5 and P5 are (84, 24), (198, 412), (511, 556), (854, 472), (524, 346) and (761, 312), respectively. In the simulation, the Manning's coefficient was set to 0.01 and the CFL condition is 0.5. Almost all the arrival times of the water in the simulation time without buildings are faster than in the simulation with buildings as shown in Fig. 8. This is because of the interruption of the building which causes the slow water flow.

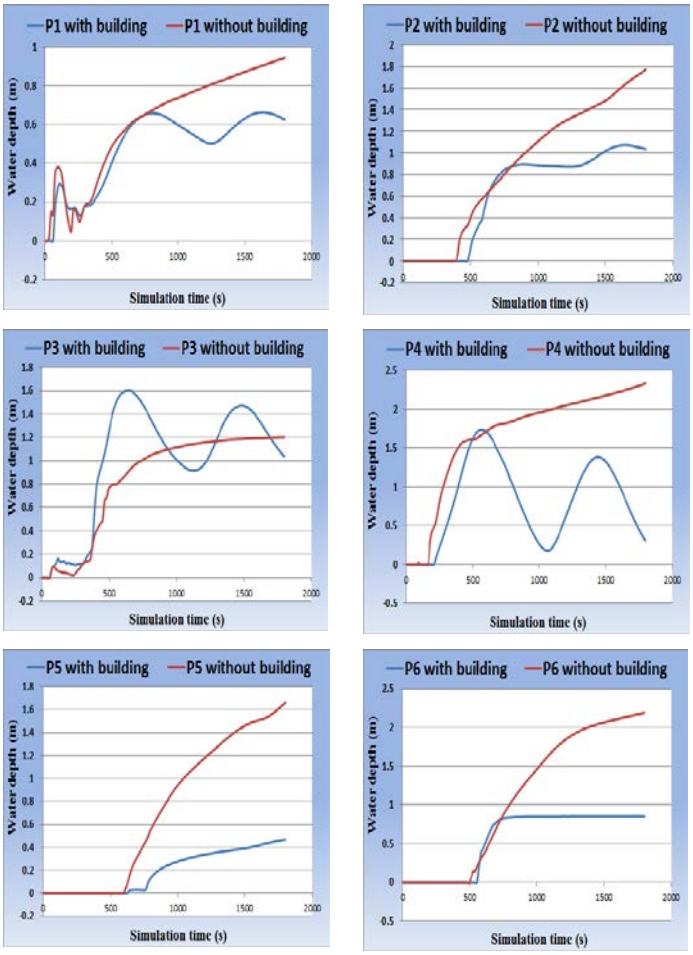


Fig. 9 Water depths at different points.

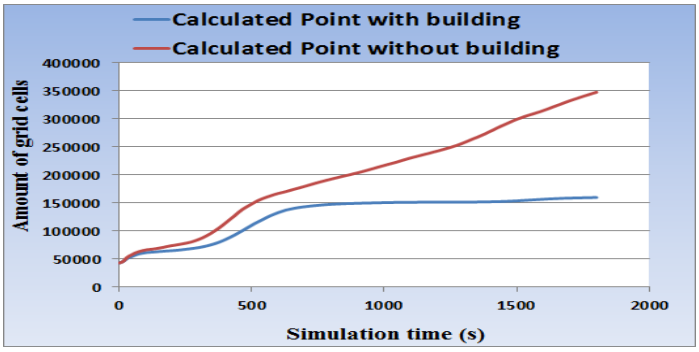


Fig. 10 Calculation points in the computation during simulation time.

Almost in all position, the depth of water without building is higher than the depth of water with buildings except for the position 3 as shown in Fig. 9. This is because of the position of that point P3 is close to the river and also in front of a building between buildings. The water collision towards the right-hand side is faster and will collide again with another building which thus leads to the conclusion mentioned above. Fig. 10 shows the calculation points that are involved in the computation during the simulation time. The result shows that the computational grids with buildings are less than without building. This is because when the calculation involves buildings, the flow of water is interrupted by building and it can be slow down calculation time.

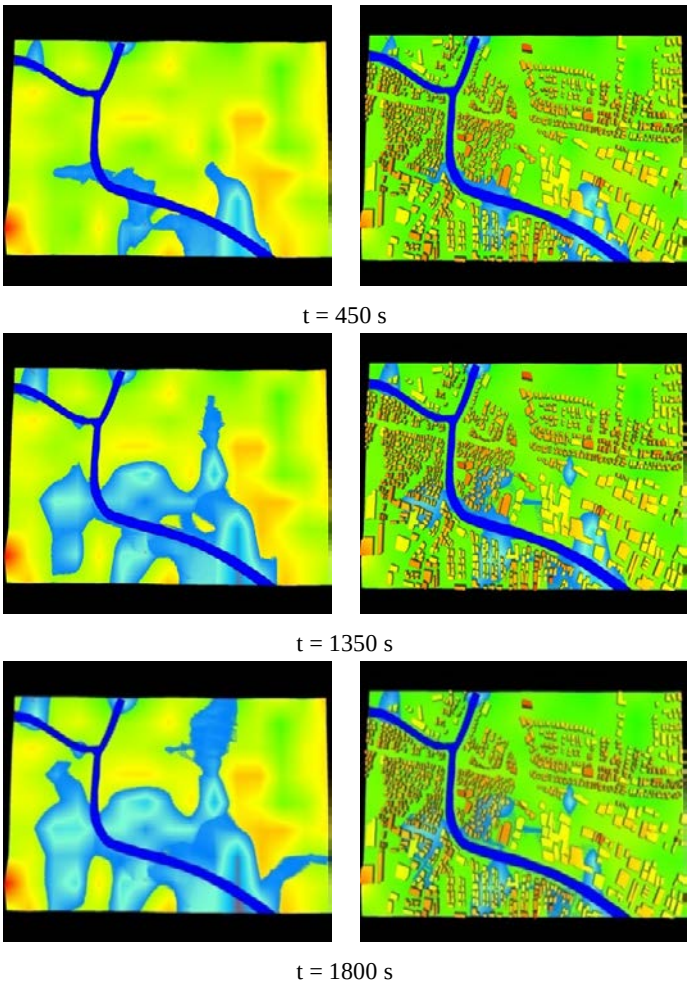


Fig. 11 Three-dimensional view of the flood maps with 3 difference time steps.

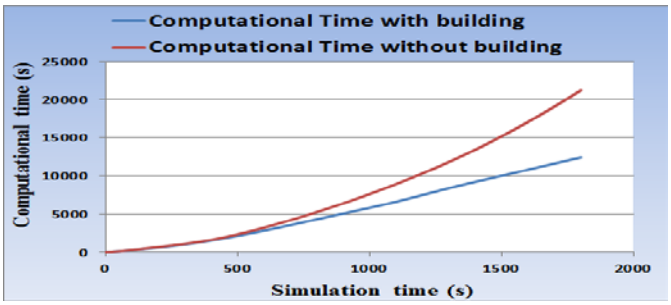


Fig. 12 Computation time during the simulation time.

Fig. 11 shows the water flow behavior. If buildings are not involved, then the flooding occurred in larger areas than with buildings involved. The depth of water would be changed depending on the selected position. Fig. 12 shows the computation time during the simulation time (1800s). The result shows the computation time when collision of water occurs with buildings. The simulation with building is less computational grid than without. The computational time will become slower if there are no buildings involved.

VII. Conclusion

The numerical algorithm and visualization software are investigated for flood simulation in an urban area with case study of West Jakarta, Indonesia. In this study, the area has the small slope of topography and also the source of water was considered to be from the river caused by heavy rainfall. The finite volume method based on the first order well-balanced scheme and equipped with dynamic domain defining method is used to solve the shallow water equations. The obtained results show that the water flow rate is faster when the simulation is carried out without the involvement of buildings since there is no any interruption. The comparison shows that the number of grid cells related to computational time. The limitation of this study is that only topography data was obtained and urban data was created. The result may be different when a real urban data is considered in another study. Furthermore the numerical algorithm in this study can be applied to other regions in the same manner with low slope of topography. This numerical algorithm could be useful for flood simulation and water flooding management in urban area

Acknowledgment

We would like to thank the editors and referees for their valuable comments, which have significantly for improved this paper. This study is supported by Thailand's Education Hub for ASEAN Countries and the Centre of Excellence in Mathematics, Commission on Higher Education, Thailand.

References

[1] E. Audusse, F. Bouchut, M. O. Bristeau, R. Klein, B. Perthame, A fast and stable well-balanced scheme with hydrostatic reconstruction for shallow water flows, *Siam J. Sci. Comput*, V. 25, no. 6, 2004, pp. 2050-2065.

[2] S. Bryson, Y. Epshteyn, A. Kurganov, G. Petrova, Well-balanced positivity preserving central-upwind scheme on triangular grids for the saint-venant system, *Mathematical Modeling and Numerical Analysis*," V. 45, no. 3, 2010, pp. 423-446.

[3] A. Busaman, K. Mekchay, S. Siripant, S. Chuai-Aree, Dynamically adaptive tree grid modeling for simulation and visualization of rainwater overland flow, *International Journal for Numerical methods in fluids*, V. 79, no. 11, 2015, pp. 559-579.

[4] D. L. George, Adaptive finite volume methods with well-balanced Riemann solvers for modeling floods in rugged terrain: Application to the Malpasset dam-break flood (France, 1959), *International Journal for Numerical Methods in Fluids*, V. 66, no. 8, 2011, pp. 1000-1018.

[5] Harten A, Lax PD, Leer BV, On upstream differencing and Godunov-type scheme for hyperbolic conservation laws, *SIAM Review*, V. 25, no. 1, 1983, pp. 35-61.

[6] C. J. Huang, M. H. Hsu, A. S. Chen, C. H. Chiu, Simulating the storage and the blockage effect of buildings in urban flooding modeling, *Terr. Atmos. Ocean. Sci*, V. 25, no. 4, 2014, pp. 591-604.

[7] K. M. T. Kleefsman, G. Fekken, A. E. P. Veldman, B. Iwanowski, B. Buchner, A volume-of-fluid based simulation method for wave impact problems, *J. Comp. Phys*, V. 206, 2005, pp. 363–393.

[8] A. Kurganov, D. Levy, Central-upwind schemes for the Saint-Venant system, *ESAIM: Mathematical Modeling and Numerical Analysis*, V. 36, no. 3, 2002, pp. 397-425.

[9] A. Kurganov, S. Noelle, G. Petrova, Semidiscrete central upwind schemes for hyperbolic conservation laws and Hamilton Jacobi equation, *Society for Industrial and Applied Mathematics*, V. 23, no. 3, 2001, pp. 707-740.

[10] B. S. M. Kusuma, P. H. Rahayu, M. Farid, B. M. Adityawan, T. Setiawati, M. R. Silasari, Study the development of the index flood map risks in the village of Bukit Duri, Jakarta (Studi pengembangan peta indeks resiko banjir pada Kelurahan Bukit Duri Jakarta), *Journal of Theoretical and Applied Civil Engineering Sector* (*Jurnal Teoretis dan Terapan Bidang Rekayasa Sipil)*, V. 17, no 2, 2010, pp. 123-134.

[11] A. Roshandel, N. Hedayat, H. Kiamanesh, Simulation of dam break using finite volume method, *International Journal of Mathematical, Computational, Physical, Electrical and Computer Engineering*, V. 4, no. 11, 2010, pp. 1406-1409.

[12] L. Song, J. Zhou, Q. li, X. Yang, Y. Zhang, An unstructured finite volume model for dam-break floods with wet/dry fronts over complex topography, *International Journal for Numerical Methods in Fluids*, V. 67, no. 8, 2010, pp. 960-980.

[13] M. Szydlowski, Numerical simulation of extreme flooding in a built-up area, *Archives of Hydro-Engineering and Environmental Mechanics*, V. 52, no. 4, 2005, pp. 321-333.

[14] Z. Xinhua, L. Wenfei, X. Heping, Z. Jiahua, W. Jiangping, Numerical simulation of flood inundation processes by 2D shallow water equations, *Front. Archit. Civ. Eng. China*, V. 1, no. 1, 2007, pp. 107-113.

[15] S. Yamaguchi, T. Ikeda, Iwamura, Rapid Flood Simulation Software for Personal Computer with Dynamic Domain Defining Method, K. 4th. International symposium of flood defence: Managing Flood Risk, Reliability and Vulnerability. Toronto, Ontario Canada, 2008.