Numerical Model and Software for Simulation and Visualization of Flooding in the Urban Area

Saeful Bahtiar[1], Somporn Chuai-Aree[1], Anurak Busamun[1], and Areena Hazanee[1]

***Abstract*—**The shallow water equation system was applied in this study to model the water flow over complex topography in the urban area. A finite volume method based on the first order well-balanced scheme was used to solve the model. In order to upgrade the capability of flood control in urban areas, we developed in this work the numerical software for simulation and visualization of flooding in the urban areas. The developed software was used for the flood simulation in Jakarta, Indonesia. The simulation results can show effects of building for the urban flooding.

***Keywords*—** Shallow water equations, Finite Volume Method, Simulation, Visualization.

I. Introduction

FLOOD is one of the catastrophic events that can result to several effects such as loss of human life and loss of economics. Flood is the most common natural disaster which often occurs in almost countries all over the world. Flooding may be caused by several reasons, such as heavy rains, dam-break, flood embankments [10]. Previous studies have shown that the use of mathematical model can simulate flooding in urban areas, for example: [10], [9], and [3].

The shallow water equations (SWEs) are one of the mathematical models which describe the surface flow over complex topography and also used to simulate many phenomena of practical interest including river flood, tsunami propagation and dam break flow [5], [1], [13]. Solutions of shallow water equations are difficult to be solved analytically, and numerical methods are needed.

There are many approaches of numerical methods that are widely used for solving the SWEs. They include the finite element method (FEM), the finite volume method (FVM) and the finite different method (FDM), etc. The most of researchers selected to use the finite volume method for the water flow simulation. For the finite volume method, [8] developed a robust, accurate and computationally efficient numerical model to solve shallow water equations, and finite volume method based on triangular grid is used, while [2] studied the finite volume method to solve shallow water equations and Godunov-type method based on the approximate Riemann solver is used to compute the flux function.

For the numerical methods, the solutions obtained are always related to computer programing because numerical calculations are often repeated every time to calculate the solutions.

Computer simulation based on software development is normally used to minimize fatalities and damage to public service facility for disaster preparation and prevention process. Computer programs were developed in previous works for flooding simulation. [4] Developed lizard software for simulation of flooding, while [1] simulated and visualized rainwater flow in the surface by developing the software.

Moreover, numerical techniques were studied for improving the numerical model. [12], [11] and [7], provided two important factors for solving shallow water equations that are steady-state stationary and contact discontinuities. Both the completions are not always easy to be handled numerically using standard numerical schemes. This difficulty can be overcome by using the well-balanced schemes [7]. [12] used a second order scheme based on linear reconstruction for solving well-balanced scheme and prove the scheme can preserve the non-negativity water height. [11] Develop a first order scheme with hydrostatic reconstruction to solve well-balanced scheme. [6] proved that the first order based on hydrostatic reconstruction can preserve the computation in the wet-dry front area.

In order to upgrade the capability of flood control in the urban areas, we developed in this work a numerical model and software for simulation and visualization of flooding in the urban area. The FVM based on the first order well-balanced scheme is used to solve the shallow water equations.

The paper is organized as follows. Section 2 presents shallow water equations, while section 3 provides the details about the numerical method. Section 4, the application of city flood simulations with building and without building is shown. The results in application are given in section 5, and conclusion is presented in section 6.

II. Shallow water equations

The model in this work was developed based on shallow water equations for determining the behavior of water flow in the urban area. The model system is presented in vector form as follow:

Saeful Bahtiar[1] is with the Mathematics and Computer Science Department, Prince of Songkla University, Pattani, Thailand .

Somporn Chuai-Aree[1], Anurak Busamun[1] and Areena Hazanee[1] are with the Mathematics and Computer Science Department, Prince of Songkla University, Pattani, Thailand.

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

where $\vec{q}=[h\ uh\ vh]^T$ is the vector of dependent variables consisting of the water depth *h*, the discharge per unit width *uh* and *vh* with velocity component *u* and *v* in the *x* and *y* directions, and *t* is the time.

The vector $\vec{f}$ and $\vec{g}$ can be expressed in terms of the primary variables *u*, *v* and *h* as

$$\left.\begin{aligned}\vec{f}(\vec{q})&=\begin{bmatrix}uh\\ u^2h+\frac{1}{2}gh^2\\ uvh\end{bmatrix},\quad \vec{g}(\vec{q})=\begin{bmatrix}vh\\ uvh\\ v^2h+\frac{1}{2}gh^2\end{bmatrix},\\ \vec{S}(\vec{q})&=\begin{bmatrix}R\\ -S_{fx}\\ -S_{fy}\end{bmatrix},\quad \vec{Z}(\vec{q})=\begin{bmatrix}0\\ -ghS_{0x}\\ -ghS_{0y}\end{bmatrix}\end{aligned}\right\}. \quad (2)$$

where $\vec{f}(\vec{q})$ and $\vec{g}(\vec{q})$ are the flux vectors of the system in the directions of the coordinate axis *x* and *y*, respectively. *g* is the acceleration due to gravity, R is the rate of river increased by rainfall. S_{0x} and S_{0y} are the bed slope in *x* and *y* directions, respectively. S_{fx} and S_{fy} are the friction terms in *x* and *y* directions, can be estimated by using the Manning resistance law.

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

In which n is the Manning resistance coefficient.

III. Numerical Method

The finite volume methods based on the first order well-balanced scheme is used to solve the model. The details can be shown as the followings.

A. Finite Volume Discretization Over Structured

For the finite volume method, the numerical domain is subdivided into rectangular (interval) grid cells of the form $C_{i,j}=\left[x_{i-\frac{1}{2}},x_{i+\frac{1}{2}}\right]\times\left[y_{j-\frac{1}{2}},y_{j+\frac{1}{2}}\right]$ as show in Fig. 1(a). Let $\Delta x=x_{i+\frac{1}{2}}-x_{i-\frac{1}{2}}$ and $\Delta y=y_{j+\frac{1}{2}}-y_{j-\frac{1}{2}}$ are the length of the cells.

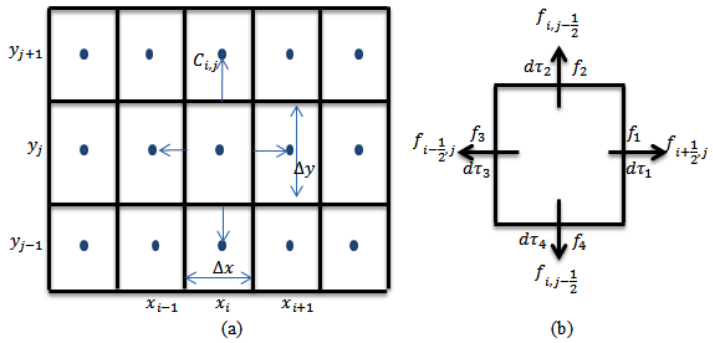


Fig. 1 The notation used for a Cartesian 2D grid (a) and typical structured grids (b)

Integrating the conservation law in (1) over each grid cell, and by applying Gauss divergent theorem to the second and third terns, we obtain:

$$\iint_{C_{i,j}}\frac{\partial \vec{q}}{\partial t}dxdy+\oint_{\tau}\vec{F}\cdot\vec{n}\,d\tau=\iint_{C_{i,j}}z(\vec{q})dxdy+\iint_{C_{i,j}}s(\vec{q})dxdy \quad (4)$$

where $C_{i,j}$ is the control volume, τ is the boundary of the $C_{i,j}$. $\vec{F}=\left(\vec{f}(\vec{q}),\vec{g}(\vec{q})\right)$ is the flux vectors at each interface of the cell boundary. Dividing with the cell area $\Delta_{i,j}=\Delta x_{i,j}\times\Delta y_{i,j}$, and denoting $\overline{Q}_{i,j}$, $\overline{Z}_{i,j}$, and $\overline{S}_{i,j}$ as the average of $\vec{q}$, $\vec{z}(\vec{q})$ and $\vec{s}(\vec{q})$ over the cell $C_{i,j}$ respectively, the (4) becomes,

$$\frac{\partial \vec{Q}_{i,j}}{\partial t}=-\frac{1}{\Delta_{i,j}}\oint_{\tau}\vec{F}\cdot\vec{n}\,d\tau+\vec{Z}_{i,j}+\vec{S}_{i,j} \quad (5)$$

where subscripts *i*, *j* denote spatial index of the cell.

The line integral in (4) can be calculated by the sum of the fluxes over the four walls around the cell as shown in Fig. 1 (b). By using the first order well-balanced scheme in [11], the gravity force $\overline{Z}_{i,j}$ can be distributed to the numerical fluxes for each sub-interface, and Euler's method can be used for approximation of the time derivative, so we can write the finite volume formulation as

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

where $\vec{f}_{i\pm\frac{1}{2},j}^{*}$ and $\vec{g}_{i,j\pm\frac{1}{2}}^{*}$ are the numerical fluxes function depending upon the chosen scheme. $\vec{Q}_{i,j}^{n}$ represents the cell average value over the *(i,j)* grid cell at time t_n.

B. Interface Fluxes Calculation

The fluxes calculation based on the first order well-balanced scheme was used in this work. The fluxes with gravity force at each interface are computed based on the Harten, Lax and Van Leer (HLL) Riemann solver with good robustness and accuracy

as showed in [9]. The application of this approach to the numerical fluxes can be expressed as:

$$\vec{f}^{\,*}_{i\pm\frac{1}{2},j} = \vec{f}^{\,HLL}\left(\vec{U}_{i\pm\frac{1}{2},j-}\ ,\ \vec{U}_{i\pm\frac{1}{2},j+}\right)+\vec{L}^{x}_{i\pm\frac{1}{2},j} \tag{7}$$

$$\vec{g}^{\,*}_{i\pm\frac{1}{2},j} = \vec{g}^{\,HLL}\left(\vec{U}_{i,j\pm\frac{1}{2}-}\ ,\ \vec{U}_{i,j\pm\frac{1}{2}+}\right)+\vec{L}^{y}_{i,j\pm\frac{1}{2}} \tag{8}$$

Where $\vec{L}^{x}_{i\pm\frac{1}{2},j}$ and $\vec{L}^{y}_{i,j\pm\frac{1}{2}}$ are well-balanced, can be expression based on work [11] with

$$\vec{L}^{x}_{i\pm\frac{1}{2},j} = \begin{bmatrix} 0 \\ \frac{g}{2}\hat{h}^{2}_{i\pm\frac{1}{2},j\mp} \\ 0 \end{bmatrix} \quad , \quad \vec{L}^{y}_{i,j\pm\frac{1}{2}} = \begin{bmatrix} 0 \\ 0 \\ \frac{g}{2}\hat{h}^{2}_{i,j\pm\frac{1}{2}\mp} \end{bmatrix} \tag{9}$$

where $\vec{f}^{\,HLL}$ and $\vec{g}^{\,HLL}$ are approximate Riemann solver based on Harten, Lax and Van Leer (HLL) and can be defined by

$$\vec{f}^{\,HLL}\left(\vec{U}_L,\vec{U}_R\right)=\frac{\vec{f}\left(\vec{U}_L\right)S_R-\vec{f}\left(\vec{U}_R\right)S_L+S_LS_R\left(\vec{U}_R-\vec{U}_L\right)}{\left(S_R-S_L\right)} \tag{10}$$

$$\vec{g}^{\,HLL}\left(\vec{U}_D,\vec{U}_U\right)=\frac{\vec{g}\left(\vec{U}_D\right)S_U-\vec{g}\left(\vec{U}_U\right)S_D+S_DS_U\left(\vec{U}_U-\vec{U}_D\right)}{\left(S_U-S_D\right)} \tag{11}$$

They are computed with the hydrostatic reconstruction state by $\vec{U}=\left[\hat{h}\ u\hat{h}\ v\hat{h}\right]^T$ when $\hat{h}$ is a particular value based on Audusse's scheme for preserving the lake-at-rest condition and guarantee that water depth is nonnegative. The formulas for $\hat{h}$ are given by

$$\left.\begin{aligned}\hat{h}_{i\pm\frac{1}{2},j\mp} &= \max\left(0,\ h_{i,j}+z_{i,j}-\max\left(z_{i,j},z_{i\pm1,j}\right)\right)\\ \hat{h}_{i,j\pm\frac{1}{2}\mp} &= \max\left(0,\ h_{i,j}+z_{i,j}-\max\left(z_{i,j},z_{i,j\pm1}\right)\right)\end{aligned}\right\} \tag{12}$$

In order to completely determine the numerical flux in HLL Riemann solver, there is need to estimate the wave speeds S_R, S_L, S_U and S_D. The wave speeds are assigned based on the work of [14] as follows:

$$S_R=\max\left(u_L+\sqrt{g\hat{h}_L}\ ,u_R+\sqrt{g\hat{h}_R}\ ,0\right) \tag{14}$$

$$S_L=\min\left(u_L-\sqrt{g\hat{h}_L}\ ,u_R-\sqrt{g\hat{h}_R}\ ,0\right) \tag{15}$$

$$S_U=\max\left(v_D+\sqrt{g\hat{h}_D}\ ,v_U+\sqrt{g\hat{h}_U}\ ,0\right) \tag{16}$$

$$S_D=\min\left(v_D-\sqrt{g\hat{h}_D}\ ,v_U-\sqrt{g\hat{h}_U}\ ,0\right) \tag{17}$$

When $u-\sqrt{g\hat{h}}$ and $u+\sqrt{g\hat{h}}$ are the smallest and largest eigenvalues of Jacobian matrix $\frac{\partial\vec{f}\left(\vec{q}\right)}{\partial x}$ and for $\frac{\partial\vec{g}\left(\vec{q}\right)}{\partial y}$ gives $v-\sqrt{g\hat{h}}$ and $v+\sqrt{g\hat{h}}$.

C.Source Terms Computation

The source term vectors $\overline{S}_{i,j}$ in (2) consists of the rate of river increased by rainfall and the friction force, For the first one, because this one is water depth that is vertically added to control volume per unit time, we update a new value of $h^{n+1}_{i,j}$ in (6), as

$$h^{n+1}_{i,j}\leftarrow h^{n}_{i,j}+\Delta tR^{n}_{i,j} \tag{18}$$

Where $R^{n}_{i,j}$ is the average rate of river increased by rainfall, in each cell (*i* ,*j*) in the range of time step for *n* to *n+1*. For the second one, the friction forces can be computed by the semi implicit method as showed in [1]. It update a new value of $uh^{n+1}_{i,j}$ in (6) as follows:

$$uh^{n+1}_{i,j}\leftarrow\frac{uh^{n+1}_{i,j}}{1+\Delta t\dfrac{gn^2\sqrt{\left(u^{n}_{i,j}\right)^2+\left(v^{n}_{i,j}\right)^2}}{\left(h^{n+1}_{i,,j}\right)^{4/3}}} \tag{19}$$

Similarly, $vh^{n+1}_{i,j}$ can be compute using formula so we can get

$$vh^{n+1}_{i,j}\leftarrow\frac{vh^{n+1}_{i,j}}{1+\Delta t\dfrac{gn^2\sqrt{\left(u^{n}_{i,j}\right)^2+\left(v^{n}_{i,j}\right)^2}}{\left(h^{n+1}_{i,,j}\right)^{4/3}}} \tag{20}$$

The semi implicit method allows to preserve stability and steady state at rest.

D. Stability Condition

The Courant Friedrich Lewy (CFL) condition, which is the stability criterion of explicit numerical schemes is used to define the time step Δt where

$$\Delta t\le 0.5\frac{\Delta A_{\min}}{\lambda_{\max}} \tag{21}$$

In the (21) $\Delta A_{\min}=\min\left(\Delta x_{i,j},\Delta y_{i,j}\right)$, for all *(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 maximum of max (S_R, S_L, S_U, S_D) for all the interface of all the cells.

E. Boundary Condition

This study used open boundary conditions, and can be given by

$$\vec{q}_{0,j} = \vec{q}_{1,j}\ ,\ \ \vec{q}_{mx,j} = \vec{q}_{mx-1,j}\ ,\ \ \vec{q}_{i,0} = \vec{q}_{i,1}\ ,\ \ \vec{q}_{i,my} = \vec{q}_{i,my-1} \quad (22)$$

where $\vec{q}_{0,j}$, $\vec{q}_{mx,j}$, $\vec{q}_{i,0}$, $\vec{q}_{i,1}$ and $\vec{q}_{i,my}$ are the vector of dependent variables for the boundary cells.

F. Topography Interpolation

In this work, Shuttle Radar Topography Mission (SRTM) topography data is used, and the data represent in the form of a data grid cells for each 90 meter. Topography data is better modeled by the smaller distance. Therefore, the topography interpolation is needed. In this study, we used the bilinear interpolation technique for obtaining the interpolated data of topography.

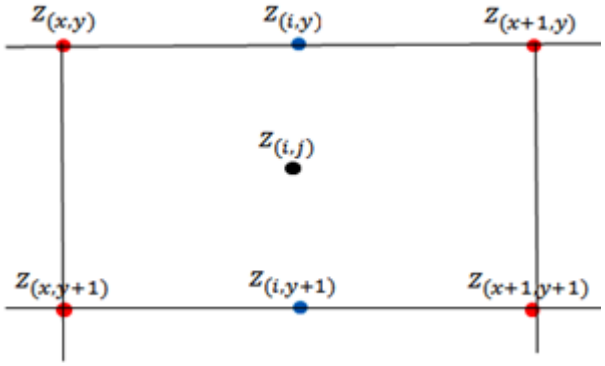


Fig. 2 The bilinear interpolation

Our purpose is to find the value of $z(i,j)$ as shown in Fig. 2, before we find the value of $z(i,j)$, we have to find value of $z(i,y)$ and $z(i,y+1)$ by using the linear interpolation. ,and then the approximated value of $z(i,j)$ is obtained by the linear interpolation $z(i,y)$ and $z(i,y+1)$. The formula for $z(i,j)$ can be written as

$$\begin{aligned} z_{(i,j)} = {} & z_{x',y'}(x'+1-m)(y'+1-n) + z_{x'+1,y'}(m-x')(y'+1-n) \\ & + z_{x',y'+1}(x'+1-m)(n-y') + z_{x'+1,y'+1}(m-x')(n-y') \end{aligned} \quad (23)$$

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

$$m = i \times n^{x'} / n^{x}\ ,\ \ n = j \times n^{y'} / n^{y} \quad (24)$$

are the mapped indicates of the computational grid cell position (x_i, y_i) to the topographic data grid. $n^{x'}$ and $n^{y'}$ are numbers of columns and rows of the topographic data grid, while n^{x} and n^{y} are number of columns and rows of the computational grid.

G. Algorithm Overview

The developed algorithm consists of several steps describing the calculation procedures, as illustrated in Fig. 3, the detail for each step is described in the following steps:

- Step 1: Read data and topography interpolation.
- Step 2: Make initial data and create building (the details of the creating building will be shown in the next section).
- Step 3: Set boundary condition.
- Step 4: Calculate the flux by using the first order well-balanced scheme, and find the max wave speed.

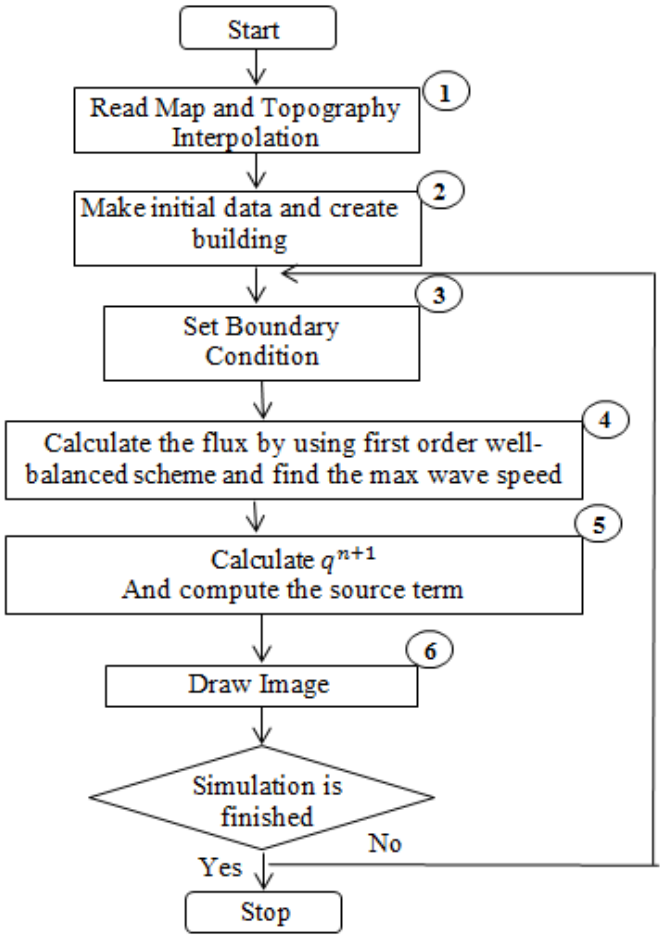


Fig. 3 The algorithm overview

- Step 5: Calculate q^{n+1}, and compute the source term.
- Step 6: Visualize the simulation results in 3D by OpenGL.
- Repeat steps 3-6 until the simulation is finished.

IV. Application City Flood Simulation With Building And Without Building

This section will present how to set up the simulation of flooding in urban area by using our software developed in this work, including how to make the building, river and road data in the software. The software was developed by Lazarus.

A. Characteristics

Fig. 4 shows the user interface of the software. Before we start the simulation, we have to know the step of the creating simulation. This software starts from the creation of topography by using the interpolating the SRTM data and creation urban data (shown in subsection B). Next, it computes and creates the information which is needed, such as velocity field, water depth, water level, arrival time and so on.

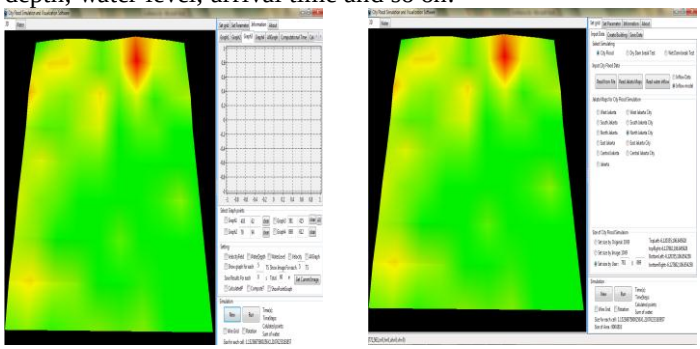


Fig. 4 The user interface of the developed software

B. Creating Building, River and the Road Data

For the flood simulation in urban, the creating of urban data which are building, rivers, and road data is very essential. Before we create the urban data, we have to know their respective positions in the real world. By using the Google earth map we can know the precise position that we want (river, road, and building) and we can also know the latitude and longitude the area for the respective data. Fig. 5 shows the input pictures from Google earth map (right) where the latitude and longitude was considered for creating the road, rivers, and buildings.

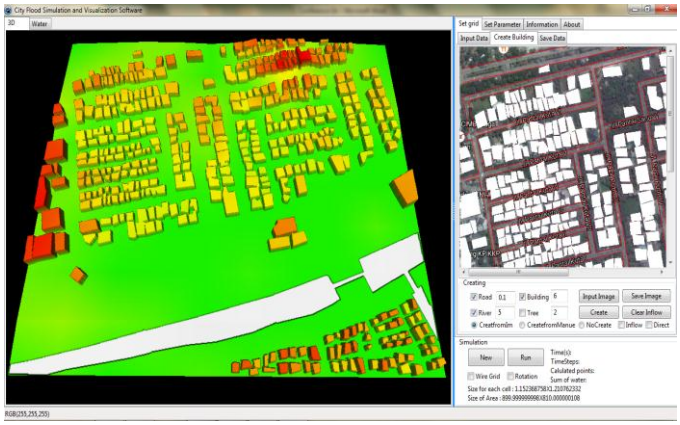
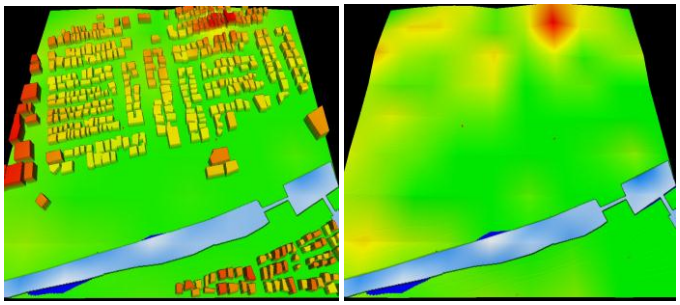


Fig. 5 The 3D represent of creating river, road, and Building data (left) with consider map by using Google Earth (right)

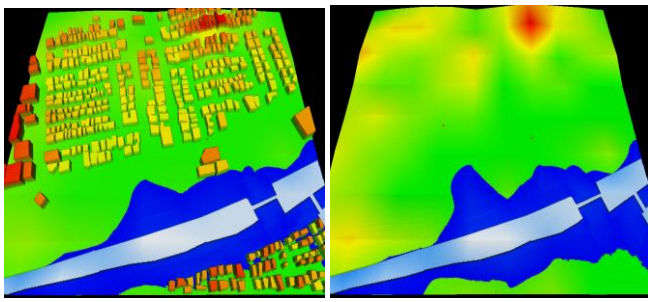
V. Result in Application

For this application, the simulation area used in this paper is in some part of North Jakarta in Indonesia. The original topography data obtained from SRTM data. We make the depth of river 6 meter and using the initial water depth in the river 5 meter, and assume the river increase by rain 0.1 m^3/s. The size of cell grid is 781 x 669. We defined four points, P1, P2, P3, and P4, to measure the results. The points are shown in Fig. 5, the positions of P1, P2, P3 and P4 are (636,609), (403,664), (501,386) and (269,359), respectively. The simulation used the Maning's coefficient is 0.01 (1/s), with the duration of the simulation is 1800 s. The simulation results consist of:

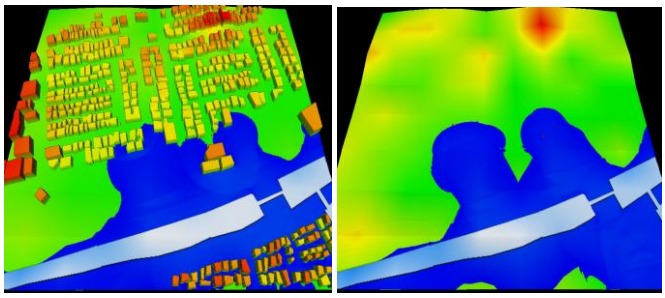
A. Flood Maps



$t = 840s.$



$t = 3360s.$



$t = 7560s.$

Fig. 6 The 3D represent of flood simulation by considering building (left) and without building (right) at t=840, 3360, and 7560s (from top to bottom)

In Fig. 6 at t=840s, indicates that the initial flow has exceeded the river. At t=3360s, the water that has flowed into the surface and will soak in some areas of the surface. And at t=7560s, the water already soaking at some point.

B. Arrival Time

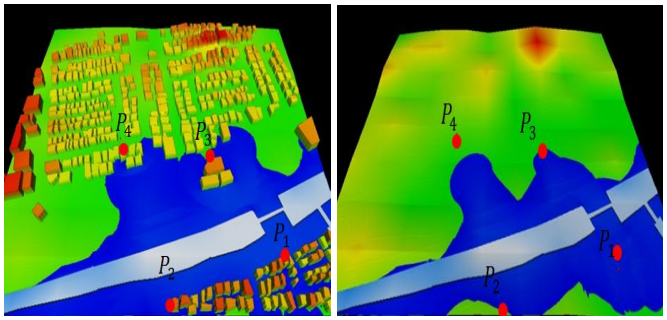


Fig. 7 The 3D representation of each point evaluated the arrival times

Fig. 7 shows the arrival time for each points of the flood simulation with building and without building. Fig. 8 shows that at the point 3, the arrival time of water flow in the urban area without building is faster than when a building is involved. This is because when a building is involved, once the water flows from the river; it is blocked by this building and the water coming from the river will collide back in opposite direction, thus making the flow of water very slow. The general results in the Fig. 8 shows that (blue diamond with building and red rectangle without building) the arrival time without building is faster than arrival time with building; and this it is caused by the influence of the collision flow of water to the building.

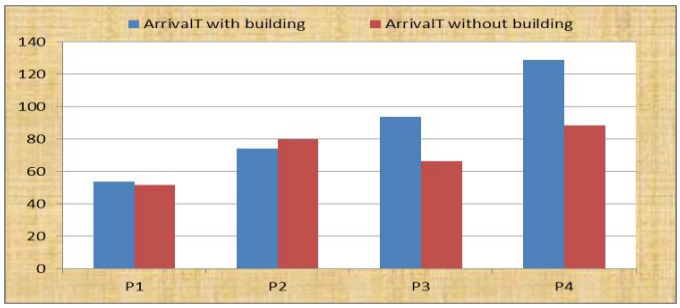
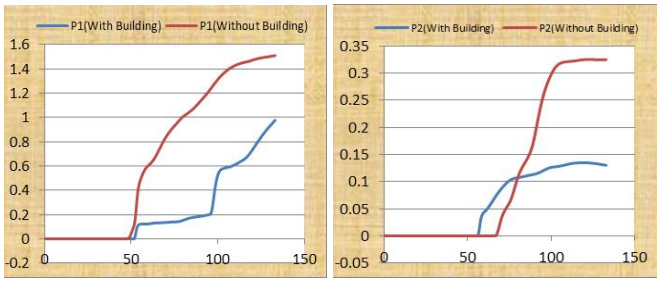


Fig. 8 The arrival time for each points of the flood simulation with building and without building

C. Water Depth



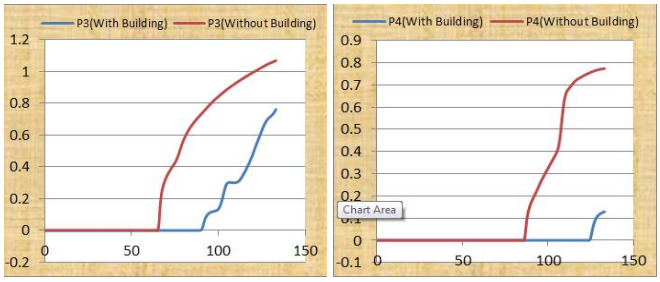


Fig 9. The depths of water for the flood simulation with buildings and without buildings, evaluated at P1, P2, P3 and P4, respectively (from left to right and top to bottom)

Fig. 9 shows the water depth results for each points. The results show that the water depth without building deeper than the depth of waters that building. Because we set up the evaluated points that has building block the water flow. In the point 2 where the position of the point is (403, 664), the flow of water that crashed into the building on the right side a little deeper than the flat water flow.

D. Velocities Vector Field

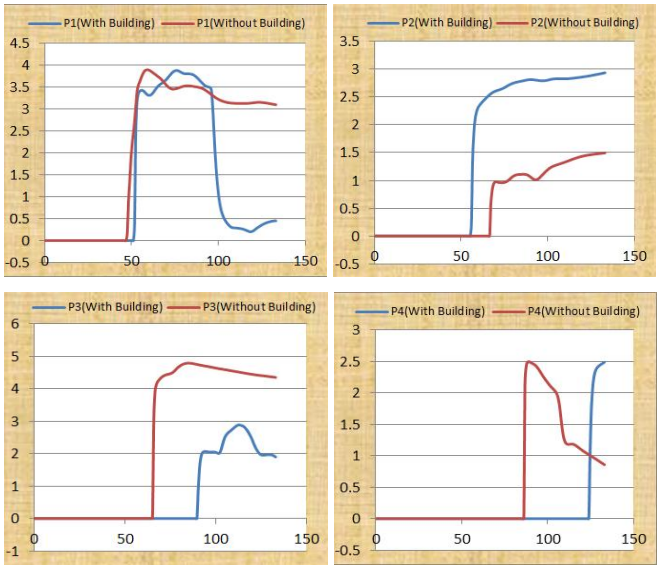


Fig. 10 The velocities of water for the flood simulation with buildings and without buildings, evaluated at P1, P2, P3 and P4, respectively (from left to right and top to bottom)

Fig. 10 shows the velocities of the flood simulation when we consider area with building and without building. The results show that the water flow rate will be faster flowing in the free surface without interruption, but when there are have distractions such as building, the water flow will more slowly after hit the building, because that the water hit the building will be in the opposite direction which causes the speed to be slow.

https://doi.org/10.15242/IIE.E1216030

VI. Discussion and Conclusion

In this paper, we applied shallow water equations to simulate and visualize water flow in urban area. The finite volume method based on the first order well-balanced scheme is used to approximate the solutions. The results obtained show that the water flow rate is more faster when the simulation is carried out without the involvement of buildings. Furthermore this is more efficient as the water moves much faster without any interruption.

Acknowledgment

We would like to thank the department of Mathematics and Computer Science for their assistance and guidance as well as Pattani Bay Watch (PB Watch) Project at the Prince of Songkla University, Pattani campus. Thailand.

References

[1] A. Busaman, K. Mekchay, S. Siripant, and S. Chuai-Aree. Dynamically adaptive tree grid modeling for simulation and visualization of rainwater overland flow. International Journal for Numerical methods in fluids. Vol 79 no. 11 2015. 559-579.
https://doi.org/10.1002/fld.4064

[2] E. Audusse, C. Chalons, and P. Ung. A simple well-balanced and positivity numerical scheme for the shallow water system. Commun. Math. Sci vol 13 no. 5. 2015. 1317-1332.
https://doi.org/10.4310/CMS.2015.v13.n5.a11

[3] C. Huang, M. Hsu, A. S. Chen, and C. H. Chiu. Simulating the storage and the blockage effect of buildings in urban flooding modeling. Terr. Atmos. Ocean. Sci., Vol. 25, No. 4, 591-604, August 2014.
https://doi.org/10.3319/TAO.2014.02.11.01(Hy)

[4] J. G. Lesken, M. Brugnach, A. Y. Hoekstra, and W. Schuurmans. Why are decisions in flood disaster management so poorly supported by information from flood models?. Environmental modeling and software, Vol. 53 (2014), 53-61.
https://doi.org/10.1016/j.envsoft.2013.11.003

[5] M. F. Ahmad, M. Mamat, W. B. Wan Nik, and A. Kartono. Numerical method for dam break problem using Godunov approach. App. Math. And Comp. Intel., Vol. 2(1)(2013) 95-107.

[6] O. Delestre, S. Cordier, F. Darboux, and F. James. A limitation of some well-balanced scheme for shallow water equations. CR Acad. Sci. Paris, Ser. I. 2012. Vol 350, 677-681.

[7] S. Bryson, Y. Epshteyn, A. Kurganov, and G. Petrova. Well-balanced positivity preserving central-upwind scheme on triangular grids for the Saint-Venant system. ESAIM. Mathematical modeling and Numerical Analysis, 45(3) 2010, 423-446.
https://doi.org/10.1051/m2an/2010060

[8] X. Ying, J. Jorgeson and S. S. Y. Wang. Modeling dam break flows using finite volume method on unstructured grid. Engineering applications of computational fluid mechanics. Vol 3, No. 2, pp. 184-194. (2009).
https://doi.org/10.1080/19942060.2009.11015264

[9] Z. Xinhua, L. Wenfei, X. Heping, Z. Jiahua, and W. Jiangping. Numerical simulation of flood inundation processes by 2D shallow water equations. Front. Archit. Civ. Eng. China 2007, 1(1), 107-113.
https://doi.org/10.1007/s11709-007-0011-5

[10] M. Szydlowski. Numerical Simulation of Extreme flooding in a built-up area. Archives of hydro-engineering and environmental mechanics. Vol 52 (2005), No. 4, pp. 321-333.

[11] E. Audusse, F. Bouchut, M. O. Bristeau, R. Klein, and B. Perthame. A fast and stable well-balanced scheme with hydrostatic reconstruction for shallow water flows. Siam J. Sci. Comput. Vol. 25, no 6, 2004. pp. 2050-2065.
https://doi.org/10.1137/S1064827503431090

[12] A. Kurganov and D. Levy. Central-upwind schemes for the Saint-Venant system. Mathematical modeling and numerical Analysis. Vol 36 no. 3. 2002. 397-425.

[13] Z. Horvath, J. Waser, R. A. P. Perdigao, A. Konev and G. Bloschl. A two-dimensional numerical scheme of dry/wet fronts for the Saint-Venant system of shallow water equations. International Journal for numerical methods in fluids. 2015: 77: 159-182.
https://doi.org/10.1002/fld.3983

[14] A. Kurganov, S. Noelle, and G. Petrova. Semidiscrete central-upwind schemes for hyperbolic conservation laws and Hamilton-Jacobian equation. Society for Industrial and Applied Mathematics. Vol. 23 No.3, 707-740.