APPLICATION OF CELLULAR AUTOMATA APPROACH FOR FAST FLOOD SIMULATION

Bidur Ghimire[1], Albert S. Chen[1], Slobodan Djordjević[1], Dragan A. Savić[1]
[1]Centre for Water Systems, University of Exeter, Harrison Building, North Park Road, Exeter EX4 4QF, UK.
b.ghimire@exeter.ac.uk, *a.s.chen@exeter.ac.uk*, *s.djordjevic@exeter.ac.uk*, *d.savic@exeter.ac.uk*

Abstract

The increasing pluvial flooding in many urban areas of the world has caused tremendous damage to societies and has drawn the attention of researchers to the development of a fast flood inundation model. Most available models are based on solving a set of partial differential equations that require a huge computational effort. Researchers are increasingly interested in an alternative grid-based approach called Cellular Automata (CA), due to its computational efficiency (both with respect to time and computational cost) and inherent parallel nature. This paper deals with the computational experiment with a new CA method for modelling 2D pluvial flood propagation. A Digital Elevation Model (DEM) comprising square grids forms the discrete space for the CA setup. Local rules are applied in the von Neumann Neighbourhood for the spatio-temporal evolution of the flow field. The proposed model is applied to a hypothetical terrain to assess its performance. The results from the CA model are compared with those of a physically based 2D urban inundation model (UIM). The CA model results are comparable with the results from UIM model. The advantages of low computational cost of CA and its ability to mimic realistic fluid movement are combined in a novel and fast flood simulation model.

Keywords
Cellular Automata, Urban Flood Simulation, Local Rules

1. INTRODUCTION

Urban flood modelling usually involves computational techniques to simulate spreading of water into a wide surface area containing many complex features. These techniques are solutions of the physically based flow governing equations: conservation of mass, momentum, and energy. Sometimes data-driven empirical stochastic models employing probabilistic approaches are also used to predict flood extent [1]. Due to the complexity of urban setup and infrastructures, accurate urban flood modelling has always been a challenging task. Recently, coupled 1D-2D and 2D models [2, 3] have been applied to urban flood modelling. However, their complexity, computation cost and significant data requirements limit their application. Many investigations have been carried out to develop models which allow the linking of complex physical processes involved in urban flooding. The modelling of interactions between 2D surface and 1D sewer flow phenomena has emerged as the main objective of recently completed and current research programs in hydraulics and urban hydrology [2, 3]. Chen et al. [3] developed an urban inundation model (UIM) coupling both surface (2D) and subsurface (1D) sewer flows. The model was successful in incorporating the bidirectional flow interaction between overland surface and underground sewer networks.

Various approaches have been devised to reduce the computational time of 2D flood modelling. Some models attempt this by dividing the overland flow area into various zones and then implementing a 1D-1D approach to simulate urban flood. Lhomme et al. [4] proposed a rapid flood spreading method that divides the domain into various impact zones (IZs) consisting of accumulation points and communication points where the transfer of water takes place. The method spills and merges IZs as the evolution of inundation proceeds. A similar model presented by Maksimović et al. [5] for urban pluvial flooding uses an automated overland flow pathway analysis in which a dual-drainage concept has been implemented using 1D-1D coupling of sewer and surface flow networks. The automated algorithm has been employed to create a surface network of ponds and surface flow paths. However, it seems a lot of pre-processing work for urban pluvial flood modelling is needed, thereby increasing the total runtime with little improvement over the results of the InfoWorks verified base model. Furthermore, 1D-2D modelling has been criticised as being too slow and difficult for effective real-time flood forecasting [6]. Vojinovic and Tutulic [6] proposed a method in which building elevations are raised and road surface were lowered to model urban flood. However, the method consumes a lot of time in pre-processing thus making the model computationally inefficient. In JFLOW-GPU, Lamb et al. [7] implemented the massive parallelisation using graphical processors to greatly reduce the computational time.

The Cellular Automata (CA) approaches to urban simulation have been limited mostly to land use change modelling. Those models are normally stochastic and/or expert rule-based. During the last decade, however, methods for describing physically-based, deterministic systems within the CA framework have become much more popular [8]. Frisch et al. [9] developed the earliest hydro-dynamical CA model to fulfil the isotropic Navier-Stokes equation. They employed a local particle collision rule on a hexagonal discrete grid to simulate fluid flows. In some complex natural phenomena, novel parallel computing models represent a valid alternative to standard differential equation methods. In particular, CA provides such an alternative approach for some complex natural systems, whose behaviour can be described in terms of local interactions of their constituent parts [10, 11]. Thomas and Nicholas [12] applied a CA model to simulate braided river flow by routing the flow from the cell under consideration into five downstream cells. Coulthard et. al [13] developed the CA Evolutionary Slope And River (CAESAR) model to simulate the sediment evolution along river channels. The four-sweep scanning algorithm employed for four directions in CAESAR impacts on its performance such that the model efficiency cannot be assured.

A Lattice Gas CA (LGCA) simulation model [14] was employed in real time computer games using GPU computation. In such an approach macroscopic density and velocity of particles define the new fluid configuration. A rapid inundation model as proposed by Krupka et al. [15] might be useful for flood risk management rather than for flood defence and analysis because it only calculates the final inundation extent employing a flood-storage cell algorithm. As such, the model does not allow dynamic interaction between river and flood plain flow. Krupka et al. adopt the CA-like concept such as three states of a cell, dry, active and inactive, to determine the flood inundation extent. However, the model lacks the timing of inundation and dynamic spreading.

The stability criterion in storage cell models like LISFLOOD-FP [16] is such that high resolution grids require smaller time steps relative to full 2D shallow water equation models. More recently Dottori and Todini [17] employed a modified local time step algorithm to improve the performance of their original cellular automata model [18]. The present work describes a reduced complexity model for urban flood inundation based on the CA approach. The motivation for this research is the development of a fast flood inundation model so that it can be used for flood risk calculations, where a large number of model run is needed. We propose a CA approach for pluvial flood modelling in which the spatio-temporal variations in flood depth and velocity are governed by local rules in the neighbourhood. The model is presented as an attempt to analyse the dynamics of the propagating flood water on the terrain due to heavy rainfall.

2. MODEL DEVELOPMENT

The proposed cellular automata model consists of a discrete space of square grid cells. The neighbourhood taken for developing the pluvial flood model is of von Neumann type. The initial condition of the terrain is assumed to be dry. The effective uniform rainfall lands directly on the whole area of the terrain considered. Hydrological losses are not considered in the model. The movement of water is modelled as occurring from the central cell. Only the outflows from the cell under consideration are calculated based on the ranks of the cells in the neighbourhood (NH). Any inflow fluxes into the cell under consideration are calculated as outflows from other locations in the cell neighbourhood. In this attempt, we dynamically apply the distribution algorithm according to the difference of the water surface elevation. Figure 1 illustrates the algorithm implemented.

Proposed Algorithm (pseudo code):

Program start

1. Initialization of variables and data input
2. Start of time loop {
3. Computation start in the local NH {
    i. Ranking of cells by their water surface elevations $\{r = 1, 2, 3, 4, 5\}$
    ii. Calculation of (r_c -1) outflows from the central cell with rank r_c
    iii. Distribute water fluxes within the NH
    iv. Find velocities at each cell face
4. End of local NH loop }
5. Determine time step Δt required for the distributions applied
6. Update time of simulation: $t = t + \Delta t$
7. Update the states (depths) for new time step
8. Data outputs for visualization and analysis
9. Repeat until the end of simulation time

    End of time loop }

End of Program

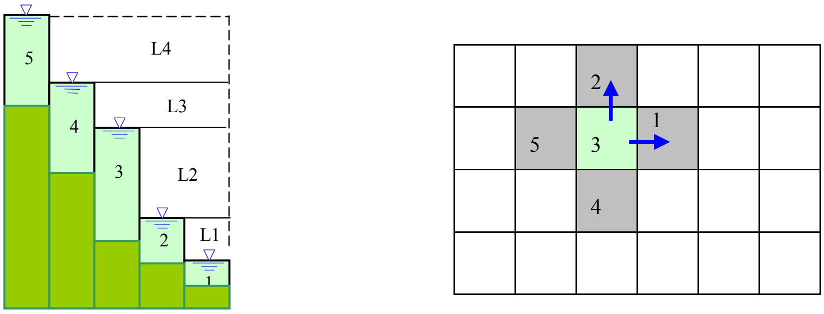

Figure 1. Definition sketch of layer wise distribution (left) and outflows to the neighbourhood from the central cell (right); the numbers shown are ranks in the neighbourhood

2.1 Flux calculation

The calculation process starts with cells being ranked locally in the neighbourhood based on the water surface elevation. For the von Neumann neighbourhood, there will be five discrete states of cell ranks (r = 1, 2, 3, 4, 5). Other global state variables such as depth of flow and ground elevations are continuous in nature. If the rank of the central cell is r_c, there will be r_c - 1 number of cells getting water as flux through the cell boundaries in the neighbourhood considered. In the layer wise calculations, there will be r_c - 1 layers (at most) to receive water, if enough volume is available. Thus outflow volumes for i^{th} layer (see Figure 1) can be given by the following recursive formula that is applied locally for each cell considered.

$$\Delta dL_i = \min\left\{ d_0 - \sum_{k=1}^{i-1} \Delta dL_k, \frac{i}{(i+1)} * \left(\Delta WL_i - \sum_{k=1}^{i-1} \Delta dL_k \right), i * \left(\Delta WL_i - \Delta WL_{i+1} \right) \right\} \tag{1}$$

where ΔdL_i is the volume distributed to i^{th} layer from the central cell having rank r_c, d_0 is the depth of water in the central cell before any spillage into the neighbouring cells takes place. Thus fluxes for each of the cells are calculated as outflow flux to the cell of rank r given by:

$$F_r = \sum_{k=r_c}^{N-1} \frac{\Delta dL_k}{k} \tag{2}$$

where, N is the total number of neighbourhood cells, i.e., 5 for the von Neumann neighbourhood considered.

2.2 Depth calculation

A very important step in the CA approach is its state update. In present CA calculations the global continuous state is the flow depth in a cell, which is updated for every new time step n+1.The following transition rule is used to update the flow depth:

$$d^{n+1} = d^n + \frac{q\Delta t}{\Delta A} \tag{3}$$

where, q is the net inflow flux into the cell; ΔA is the area of the cell. Equation (3) is equivalent to the following continuity equation:

$$\frac{\partial W}{\partial t} + \nabla \bullet \mathbf{q} = S_r \tag{4}$$

where, W is the volume of water in the cell, **q** is the flux vector at cell boundary and S_r is the source term. To control the outflows from the cell and avoid abrupt jumps in depth, a non-dimensional parameter is used to limit the flux through the boundaries. The transition rule for updating the global state of cells is then given by:

$$d = d_0 + \frac{\sum F_i}{\Delta A} \tag{5}$$

2.3 Time-step calculation

Many studies have been carried out by the research community to reduce the simulation time and predict reliably the timing of flood inundation. This is mainly due to the fact that as higher and higher resolution DEM data is being used, the shorter will be the time step required for stability of model computations. The time increment determined in this way, i.e., the smallest anywhere in the whole domain that satisfies the stability criteria, implies that for most of the cells only a fraction of the locally allowable time step is used to integrate the solution in time. This represents a waste of computational effort [19] and limits the applicability of the method. The use of spatially variable local time step and constrained time steps (in between the shortest and longest time steps) can be found in literature. A spatially varying time step can increase solution accuracy and reduce computer run times [20, 21]. Dottori and Todini [17] employed a local time step algorithm to speed up the originally formulated LISFLOOD-FP model by Bates et al. [22] and suggested that there is no need to limit the ratio between maximum and minimum time step. In particular, there is no need to divide cells in groups with different time steps, since the maximum allowable time step is directly assigned to each cell interface and stored after each computational step. In the present work the maximum allowable time step is calculated so that water across a single cell width is distributed within the neighbourhood, thereby automatically satisfying the conventional Courant-Friedrichs-Lewy (CFL) criterion. The velocity at a cell boundary is calculated based on the flux transferred through the cell given by following formula:

$$v = \frac{F}{d.\Delta x.\Delta t} \tag{6}$$

where F is the flux volume being transferred between two cells, Δx is the cell size and Δt is the time step. The maximum allowable velocity is limited by the maximum velocity calculated based on the Manning's formula and wave celerity given by:.

$$v_{\max} = \left\{ \frac{1}{n} R^{\frac{2}{3}} S^{\frac{1}{2}}, \sqrt{gd} \right\} \tag{7}$$

where n is Manning's roughness coefficient, R is the hydraulic radius, S is the slope of water surface, g is the gravitational acceleration, and d is the water depth. The time step is then set to ensure that the maximum flux can be transferred within single time step. So, each time the state transition rule is applied, the global time is updated by the time step value obtained from Equation (8). Since the time step is based on the maximum flux transferred through the cell boundaries into the neighbourhood, it is also valid for other intermediate flux values calculated during current time step because water distribution is allowed within a single cell width only.

$$\Delta t = \frac{\Delta x}{v_{\max}} \tag{8}$$

The inherent nature of the CA approach makes it highly suitable for parallelisation. Because of the capability of synchronous and independent updating of the states of each cell, it is computationally efficient and may find its application into cloud computing and GPU computations. The reduction in the model run time through parallelisation would increase the utility of such models.

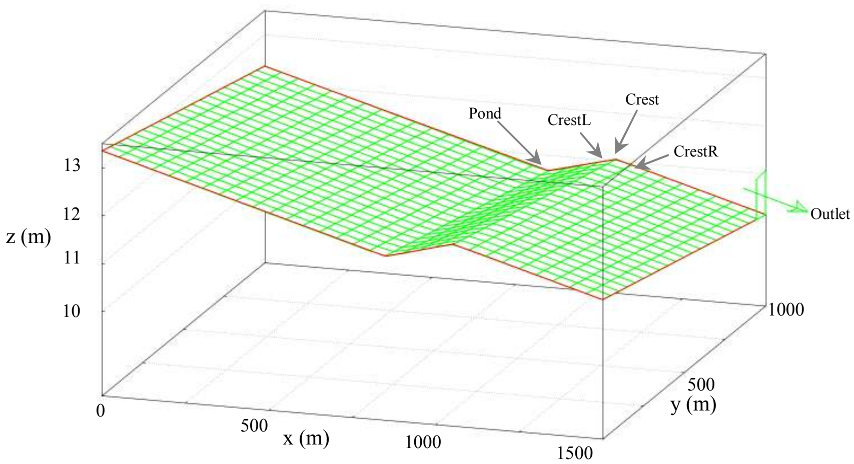

Figure 2. Hypothetical terrain

3. MODEL APPLICATION

The proposed model was tested for a hypothetical terrain as shown in Figure 2. The terrain consists of both forward and reverse slopes of 0.2 %. It also has a lateral slope of +0.1% toward the reader. There is no flow allowed across boundaries, i.e., flow is enclosed by the boundaries. There is an outlet at the lowermost end of the terrain. The main data inputs to the model are:

(i) Rainfall: A net rainfall of 20 mm per hour uniform intensity fed to the whole terrain. The rainfall lasts for one hour and then stops.

(ii) Digital Elevation Model: A DEM consisting of 600 square cells (50mx50m) is given as an input to the model.
(iii) A constant value of Manning's coefficient of roughness is given for all the cells, $n = 0.10$.

The above mentioned inputs were provided to the model and various outputs and results calculated. Those include, but are not limited to, the depth distribution at checkpoints, flow profile, flow rate across any boundary, total inundated area, and outflow volume from the terrain, etc. Some of the results will be presented in the subsequent sections. To achieve the uninterrupted computation for all the grids inside the flow domain and to provide a realistic boundary condition, a layer of grid cells is added as ghost cells around the whole domain of calculation. This layer acts as neighbouring cells for the edge cells, but does not affect the results. Thus, for no-flow boundary, the water surface elevation of the ghost cell is made equal or higher to the adjacent (inner) edge cell so that there is no flow or no gradient available for any crossflow to occur. For the outflow cell a surface elevation gradient is maintained to ensure the continuity of flow.

4. RESULTS AND DISCUSSION

The results obtained from the proposed CA model are compared with the results of UIM, a 2D non-inertia overland flow model that neglects the acceleration terms in the shallow water equations. The flow profile after the onset of rainfall is shown in Figure 3. The water is flowing down along the steepest slope and accumulating in the lowermost pond area. The drying and wetting process during this spatio-temporal flow evolution is reproduced well by the model. It can also be observed that the water exchange between two pond areas takes place through the narrow strip on the lower side of the area.

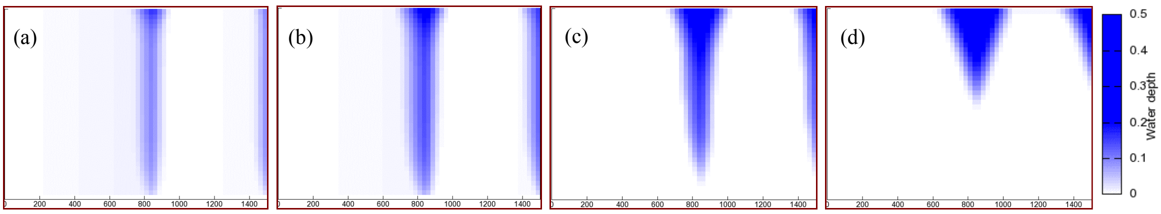

Figure 3. Flow and depth profiles after the onset of rainfall at (a) 40 (b) 60 (c) 100 & (d) 200 minutes.

The results for flow depths at various locations are examined to assess the model performance. Various check points where depths are calculated are shown on Figure 2. The depth of water computed by both models at the lowest point (pond) against the time is plotted. The results are in good agreement (see Figure 4(a)). The depth at the lowermost point, upstream of the crest, increases rapidly at the beginning (after the onset of rainfall), but when water starts transferring to downstream sub-catchment through the crest, the rate becomes lower and then remains almost constant. The variation of flow depths at various other points are shown in Figure 4(b). Both results from the current CA model and UIM are plotted for comparison. The CA results follow the trend of UIM model results. However, the peak depths are underestimated by the CA model though the discrepancies are not so significant. The plot shows the depths at crest and both of its left and right cells. When compared with the computational speed, the current CA model is 15 times faster than UIM (e.g. 2 seconds versus 30 seconds).

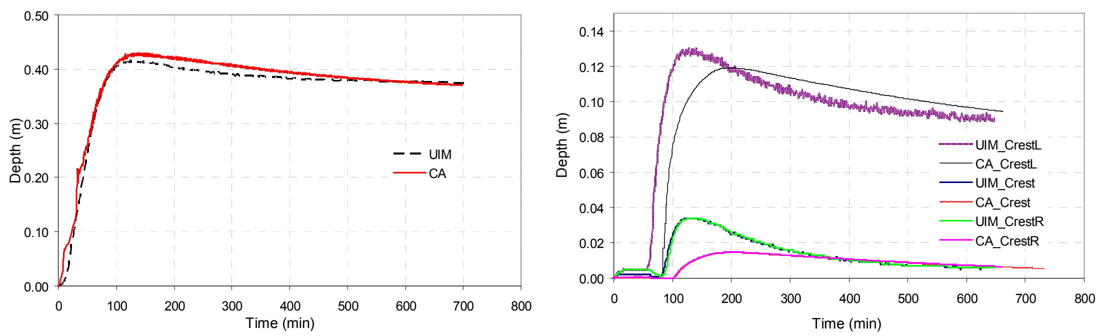

Figure 4. (a) Temporal variation of water depth at pond. (b) Temporal variation of water depth at various check point

5. CONCLUSION

The developed 2D cellular automata model has been applied to a simple hypothetical terrain. Numerical results obtained are compared with those of a 2D UIM that employs Saint-Venant's equation for flood inundation prediction. The depths obtained at various check points show a good agreement with the results of UIM.

At present, the model is undergoing refinements so that it could be used on a large area and still achieve fast computations. It should be noted that the proposed CA algorithm is particularly suitable for parallelization and GPU computations, which could increase the appeal of the model for a range of applications. The CA method could be further developed to include not only the overland flow but also associated processes, such as sewer flow, infiltration and sediment transport, etc. The model allows managing wetting and drying effects over wide areas. However, it needs to be demonstrated and validated on a wide variety of practical applications that are planned for the future. Therefore, future work will focus on the application of the CA model to real flood situations and evaluation of its effectiveness.

Acknowledgement

The authors would like to acknowledge the funding provided by by the U.K. Engineering and Physical Sciences Research Council, grant GR/J09796 (Cellular Automata Dual Drainage Simulation, CADDIES).

References

[1] L.C. Chang, H.Y. Shen, Y.F. Wang, J.Y. Huang, Y.T. Lin, "Clustering-based hybrid inundation model for forecasting flood inundation depths, *Journal of Hydrology*, vol. 385, 2010, pp. 257-268.

[2] S. Djordjevic, D. Prodanovic, and C. Maksimovic, "An approach to simulation of dual drainage," *Water Science and Technology*, vol. 39, no. 9, 1999, pp. 95-103.

[3] A. Chen, S. Djordjevic, J. Leandro, and D. Savic, "The urban inundation model with bidirectional flow interaction between 2D overland surface and 1D sewer networks," in *proc. Novatech*, Lyon, France, 2007, vol. 1, pp. 465-472.

[4] J. Lhomme, P.B. Sayers, B. Gouldby, P. Samuels, and J.M. Marti, "Recent developments and application of a rapid flood spreading method," in *proc. European conf flood risk mgmt: res. and Practice*, Taylor and Francis, 2008.

[5] C. Maksimović, D. Prodanović, S.B. Aroonnet, J. Leitão, S. Djordjević, and R. Allitt, "Overland flow and pathway analysis for modelling of urban pluvial flooding," *Journal of Hydraulic Research*, vol. 47, no. 4, 2009, pp. 512-523.

[6] Z. Vojinovic and D. Tutulic, "On the use of 1D and coupled 1D-2D modelling approaches for assessment of flood damage in urban areas," *Urban Water Journal*, vol. 6, no. 3, 2009, pp. 183-199.

[7] R. Lamb, M. Crossley, and S. Waller, "A fast 2D floodplain inundation model, in *Proc. the Institution of Civil Engineers, Water Management*, vol. 162, no. 800062, 2009. doi: 10.1680/wama.2009.162.1.1

[8] M.A. Fonstad, "Cellular automata as analysis and synthesis engines at the geomorphology ecology interface," *Geomorphology*, vol. 77, 2006, pp. 217-324.

[9] U. Frisch, B. Hasslacher, and Y. Pomeau, "Lattice-Gas Automata for the Navier-Stokes Equations," *Physical Review Letters*, vol. 56, no. 14, 1986, pp. 1505-1508.

[10] S.D. Gregorio and R. Serra, "An empirical method for modelling and simulating some complex macroscopic phenomena by cellular automata," *Future Generation Computer Systems*, vol.16, nos. 2-3, Dec. 1999, pp. 259-271.

[11] D. D'Ambrosio, S.D. Gregorio, G. Iovine, V. Lupiano, R. Rongo, and W. Spataro, "Fisrt simulations of the Sarno debris flows through Cellular Automata modelling," *Geomorphology*, vol. 54, nos. 1-2, 2003, pp. 91-117.

[12] R. Thomas and A.P. Nicholas, "Simulation of braided river flow using a new cellular routing scheme," *Geomorphology*, vol. 43, 2002, pp. 179-195.

[13] T.J. Coulthard, M.G. Macklin, and M.J. Kirkby, "A cellular model of Holocene upland river basin and alluvial fan evolution," *Earth Surface Processes and Landforms*, vol. 27, 2002, pp. 269-288.

[14] S.F. Judice, B. Barcellos, and G.A. Giraldi, "A cellular automata framework for real time fluid animation," in *Proc. Brazilian Symposium on Computer Games and DigitalEntertainment*, 2008, pp. 169-176.

[15] M. Krupka, G. Pender, S. Wallis, P.B. Sayers, and J.M. Marti, "A rapid flood inundation model," in *proc. 32nd IAHR Congress,* Venice, Italy, 1-6 Jul. 2007.

[16] N.M. Hunter, M.S. Horritt, P.D. Bates, M.D. Wilson, and M.G.F. Werner, "An adaptive time step solution for raster-based storage cell modelling of floodplain inundation," *Adv. in Water Res.*, vol. 28, 2005, pp. 975-991.

[17] F. Dottori and E. Todini, "Developments of a flood inundation model based on the cellular automata approach: Testing different methods to improve model performance," *J. Phys. Chem. Earth*, 2011.

[18] F Dottori, E. Todini, "2D flood inundation model based on cellular atomata approach" in *proc. XVIII International Conference on Water Resources*, *CMWR*, Barcelona, Spain, 2010.

[19] X.D. Zhang, J.Y. Trépanier, M. Reggio, and R. Camarero, "Time-accurate local time stepping method based on flux updating," *AIAA J.* vol. 30, no.8, 1994, pp. 1980-1985.

[20] N.G. Wright, C.D. Whitlow, and A.J. Crossley, ''Locally adapted time steps for open channel flow with transitions,'' in *Proc. 28th Congress of the IAHR,* Graz, Austria, 1999, pp. 364-369.

[21] A.J. Crossley, N.G. Wright, D. Chris and C.D. Whitlow, "Local time stepping for modeling open channel flows," *J. Hydraul. Eng.*, vol. 129, no. 6, 2003, pp. 455-462.

[22] P.D. Bates, A.P.J. De Roo, "A simple raster-based model for flood inundation simulation," *Journal of Hydrology*, vol. 236, nos. 1-2, Sep. 2000, pp. 54-77.