Environmental Modelling & Software 84 (2016) 378–394

Contents lists available at ScienceDirect

Environmental Modelling & Software

journal homepage: www.elsevier.com/locate/envsoft

A weighted cellular automata 2D inundation model for rapid flood analysis

Michele Guidolin*, Albert S. Chen, Bidur Ghimire, Edward C. Keedwell, Slobodan Djordjević, Dragan A. Savić

Centre for Water Systems, College of Engineering, Mathematics and Physical Sciences, University of Exeter, Harrison Building, North Park Road, Exeter, EX4 4QF, United Kingdom

ARTICLE INFO

Article history:
Received 31 March 2015
Received in revised form
3 June 2016
Accepted 6 July 2016
Available online 22 July 2016

Keywords:
Flood modelling
Cellular automata
Weight-based model
GPU computing
Open source

ABSTRACT

To achieve fast flood modelling for large-scale problems, a two-dimensional cellular automata based model was developed. This model employs simple transition rules and a weight-based system rather than complex Shallow Water Equations. The simplified feature of cellular automata allows the model to be implemented in parallel environments, resulting in significantly improved modelling efficiency. The model has been tested using an analytical solution and four case studies and the outputs were compared to those from a widely-used commercial physically-based hydraulic model. Results show that the model is capable of simulating water-depth and velocity variables with reasonably good agreement with the benchmark model, using only a fraction of the computational time and memory. In the case of the real world example, the proposed model run times are up to 8 times faster. The rapid and accurate attributes of the model have demonstrated its applicability for quick flood analysis in large modelling systems.

Software availability

Name of software: CADDIES-caflood
Developers: Michele Guidolin, Albert S. Chen
Contact address: Centre for Water Systems, College of Engineering, Mathematics and Physical Sciences, University of Exeter, Harrison Building, North Park Road, Exeter, EX4 4QF, UK
Email: m.guidolin@exeter.ac.uk
Software required: OpenMP/OpenCL libraries
Hardware required: Multi-core CPU or OpenCL capable graphics card GPU
Programming language: C/C++, OpenCL
Program size: Around 20 MB
Availability: Open source MIT license

1. Introduction

The demand for performing two-dimensional (2D) flood modelling for extremely large spatial scale problems (large extent, fine grid resolution, or a large number of simulations) is growing in the flood risk management industry. The problems include flood risk assessment for up to continental size domains, flood simulations with high resolution terrain data (e.g., to reflect micro-features in urban environment), uncertainty analysis with a large number of input parameter combinations and the analysis of flood risk for future development and climate change scenarios. All of these need fast models that can run simulations accurately and efficiently.

Typical physically-based 2D flood models solve the Shallow Water Equations (SWEs), requiring high computational resources. Many of these models have been developed to obtain better performance, while maintaining the required accuracy, by reducing the complexity of the SWEs. This reduction is usually achieved by approximating or neglecting less significant terms of the equations (Hunter et al., 2007; Yen and Tsai, 2001). The JFLOW model (Bradbrook et al., 2004), Urban Inundation Model (UIM) (Chen et al., 2007), and the diffusive version of LISFLOOD-FP (Hunter et al., 2005) solve the 2D diffusion wave equations that neglect the inertial (local acceleration) and advection (convective acceleration) terms (Yen and Tsai, 2001). The inertial version of LISFLOOD-FP (Bates et al., 2010) solves the SWEs without the advection term. In either version of LISFLOOD-FP the flow is decoupled in the Cartesian directions. Other models use the full SWEs but focus on the use of multi resolution grids or irregular mesh, like InfoWorks ICM (Innovyze,

* Corresponding author.
E-mail address: m.guidolin@exeter.ac.uk (M. Guidolin).

http://dx.doi.org/10.1016/j.envsoft.2016.07.008

2012) and MIKE FLOOD (DHI Software, 2014; Hénonin et al., 2013). These last two models are commercial packages, and the code applied in the optimisation techniques is not in the public domain.

Many of these physically based models have benefited from parallelised computation, due to the recent advancement in parallel computing techniques and easy-to-access parallel capable hardware. For example, InfoWorks ICM (Innovyze, 2012), JFLOW-GPU (Lamb et al., 2009), and the Finite volume SWEs model of Smith et al. (2013) use the massive parallel computational power of Graphics Processing Units (GPUs) in modern graphics cards to reduce the computation time. LISFLOOD-FP uses the OpenMP (Neal et al., 2009) and the MPI libraries (Neal et al., 2010) and P-DWave (Leandro et al., 2014) uses the OpenMP library to take advantage of multi-core CPUs (MCs). To increase performance, other 2D models, such as the FloodMap-Parallel model (Yu, 2010) and CityCAT urban flood model (Glenis et al., 2013), have taken advantage of remote distributed computers or Cloud computing.

While the run time of a flood simulation is directly related to the spatial resolution of the data used, the complexity of the model also has a major impact. Considering this progress towards the development of a fast 2D model, physically based models still have some inherited performance disadvantages. Solving the SWEs, even in their reduced complexity formulations, is still computationally intensive due to the complex mathematical formulae that require expensive computing operations. Furthermore, these physically based models might not always be able to take full advantage of modern highly parallel computing techniques, like GPU computation, due to their need for inherently sequential computations.

In recent years, a number of studies have focused on developing simple 2D flood models using the cellular automata (CA) approach instead of solving the SWEs. The CA technique offers a versatile method for modelling complex physical systems using simple operations (Wolfram, 1984). This simplification dramatically reduces the computational load of a CA model compared with a physically based model. A CA model usually consists of five essential features: a set of cells that represent discretised space, each of which has a state, a distribution of neighbouring cells, a discrete time step, and a set of transition rules (Itami, 1994). The transition rules are composed of simple operations that govern the evolution of each cell's state, which make use of the previous state of the cell itself and of those in its neighbourhood. Since computing the new state of a cell depends only on the state of the neighbouring cells at the previous time step, CA algorithms are well suited to parallel computation. The CA approach used in this work differs from the related lattice-Boltzmann method (Chen and Doolen, 1998) that uses simple operations to compute the micro scale particles interaction to model the macro scale flow behaviour.

CA based models have been successfully used to simulate many types of water related problems. Thomas and Nicholas (2002) 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. (2007) developed the CA Evolutionary Slope and River (CAESAR) model to simulate the sediment evolution in rivers. Krupka et al. (2007) adopted a concept similar to CA that uses three states of a cell (dry, active and inactive) to develop a rapid inundation model. While not being CA based, the RSFM direct (Lhomme et al., 2008) and ISIS Fast (Halcrow Group Ltd, 2014) models use mass balance approach to spread flood volume to the linked depressions on the floodplain. However, these models can only determine the final inundation extent because of the lack of dynamic time variation modelling.

Austin et al. (2013, 2014) developed a series of CA models (CA1D, BCA1D, etc.), that simulate sewer network flow using various transition rules and demonstrated that these simplified CA models are capable of producing reliable results, compared with traditional 1D hydraulic models, with a lower computational cost. These models were developed as the 1D part of the Cellular Automata Dual-DraInagE Simulation (CADDIES, 2015) project which aimed to produce fast and accurate algorithms for flood modelling of coupled surface and sewer flow.

In terms of existing CA based 2D flood models, Dottori and Todini (2010, 2011) developed a CA model, which is similar to the storage cell models, such as LISFLOOD-FP. It employs Manning's equation for the computation of interfacial discharges between computing cells. Ghimire et al. (2013) developed the first version of the CADDIES-2D model (CA2D) that differed from Dottori and Todini's model in its approach by evaluating the volume transferred between cells using a ranking system instead of directly solving the Manning's equation. However, the ranking system equation was still solved for each time step, for each interfacial direction to limit the flow velocity. If the computed velocity was too high, the intercellular transfer volumes were recomputed. This model achieved high performance thanks also to the use of the massive parallelism of the GPU (Ghimire et al., 2013; Guidolin et al., 2012).

Both Dottori and Todini's, and Ghimire et al.'s CA models solve the Manning's equation for each direction of the cellular interfaces. The Manning's equation is relatively computationally expensive because it includes the power and square root operations that take more processing time than solving simpler operations. While in modern architecture, like GPU, the amount of memory used by a model has larger impact on the execution time than it had in the past, reducing the number of expensive operations can still significantly reduce the computation time.

The main idea of this work is to present a new CA2D model derived from Ghimire et al. (2013), which adopts a weight based system to further simplify the transition rules determining the flow movement and to further minimise the need for solving complex and computationally expensive equations. The aim is not to replace complex physically based models, but to achieve high performance while maintaining adequate accuracy for rapid flood analysis in large-scale applications.

The new model was applied to five different problems: the analytical solution proposed by Hunter et al. (2005), three examples from the Environment Agency (EA) benchmarking tests for 2D flood modelling (Néelz and Pender, 2013), and one real world case study in the area of Torquay in the UK. The results of the three 2D benchmarking test cases and the Torquay test case were compared with that of the widely used commercial model InfoWorks ICM 3.0 (Innovyze, 2012) to assess the performance and accuracy of the new CA model.

2. Model

The proposed model attempts to reduce the necessity for physically-based equations and complex mathematical operations in the transition rules used to simulate an inundation event. This model, which is also a part of the CADDIES 2D family, but improves upon the methodology used in the CA2D model (Ghimire et al., 2013) is referred to as the Weighted Cellular Automata 2D (WCA2D). The WCA2D model is a diffusive-like model that ignores inertia terms and momentum conservation. The model has been designed to work with various general grids, (e.g., rectangular, hexagonal or triangular grid) with different neighbourhood types (e.g., the five cells of the von-Neumann (VN) neighbourhood or the nine cells of the Moore neighbourhood). The major features of this new model are:

1. The ratios of water transferred from the central cell to the downstream neighbour cells (intercellular-volume) are calculated using a quick weight-based system;

2. The volume of water transferred between the central cell and the neighbour cells is limited by the Manning's formula and the critical flow equation.
3. Both the adaptive time step and the velocity are evaluated within a larger updated time step to speed up the simulation.

The pseudo-code of the main algorithm of the model is given in Fig. 1, where $t(s)$ is the simulation time, $\Delta t(s)$ is the adaptive time step, and $\Delta t_u(s)$ is the fixed update time step.

2.1. Intercellular-volume computation

CA2D (Ghimire et al., 2013) used a ranking technique to compute the volume of water transferring from the central cell to the neighbourhood, called intercellular-volume. This technique worked by sorting the water levels of the cells in the neighbourhood (central cell included) in ascending order. The cells ranked lower than the central cell were recipients of an intercellular-volume from the central cell. The volumes of water transferring at various interfaces were computed on a lower-cell-fill-first basis. The cell with the lowest rank received water until it reached the same level of the second ranked cells; then these two cells received water until they reached the level of the third ranked cells, and so on until there was no water left for distribution in the central cell or all cells had the same water level.

This process had two drawbacks: 1) it was prone to water level oscillations between time steps, and 2) a sorting algorithm was executed for each cell. The sorting algorithm is a computationally intensive process to execute because it generally needs $m\log(m)$ steps and at worst m^2 steps, where m is the number of cells in the neighbourhood. Although the sorting algorithm had small computational cost when considering only a single cell with a limited number of neighbourhood cells, the algorithm was executed for each cell at every time step, which affected the total run-time of the model significantly. The previous version of CA2D used a non-dimensional flow relaxation parameter, θ, which was determined empirically, to reduce the oscillations that may result in instability (Ghimire et al., 2013).

The new WCA2D model overcomes these two drawbacks by using a simplified weighting process which has four steps: 1) identify the downstream neighbour cells; 2) compute the specific weight of each downstream cell based on the available storage volume; 3) compute the total amount of volume that will leave the central cell; and 4) for each downstream cell, set the eventual intercellular-volume which depends on the previously computed weight and total amount of volume transferred. When compared with CA2D, internal tests show that the new model is less prone to oscillation, and has better accuracy and faster calculations, without using very small time step to ensure stable results.

In the first step, a downstream cell is identified by using water level differences between the central cell and the neighbour cells. A positive water level difference greater than a small tolerance τ indicates that the downstream cell will receive an outflow; a difference less than τ indicates that the cell will be ignored. The next step starts with finding volume differences between the downstream cells and the central cell, i.e., each positive difference in water level greater than τ is multiplied by the area of the respective neighbour cell. During this step, the minimum, maximum, and the total volume differences are evaluated. In this work, the volume difference is also called the available storage volume, which is the space in a neighbour cell that is available to receive water from the central cell. This is identified by the following equations:

$$\Delta l_{0,i} = l_o - l_i \;\forall i \in \{1 \dots m\}. \tag{1}$$

$$\Delta V_{0,i} = A_i \, max\{\Delta l_{0,i}, 0\} \;\forall i \in \{1 \dots m\}. \tag{2}$$

$$\Delta V_{min} = \min\{\Delta V_{0,i} \mid {}^{\Delta l_{0,i} > \tau}_{i=1\dots m}\} \tag{3}$$

$$\Delta V_{max} = \max\left\{\Delta V_{0,i} \mid {}_{i=1\dots m}\right\} \tag{4}$$

$$\Delta V_{tot} = \sum_{i=1}^{m} \Delta V_{0,i} \tag{5}$$

where m is the number of cells in the neighbourhood, i is the index of the neighbour cell analysed, l_0 (m) is the water level in the central cell, l_i (m) is the water level of the neighbour cell analysed, $\Delta l_{0,i}$ (m) is the difference in water level between the central cell and the neighbour cell analysed, A_i (m^2) is the area of the ith neighbour cell, $\Delta V_{0,i}$ (m^3) is the available storage volume between the central cell and the ith neighbour cell, ΔV_{min}(m^3) is the minimum available storage volume of the downstream cells, ΔV_{max} (m^3) is the maximum available storage volume of the downstream cells, and ΔV_{tot}(m^3) is the total available storage volume. In all of these and the following equations, variables without a superscript are considered to be at time t.

The weight of the downstream cell i is given by the ratio between the available storage volume $\Delta V_{0,i}$ and the total available storage volume of all downstream cells ΔV_{tot}. This weight represents the fraction of total intercellular-volume, i.e. the total volume to leave the central cell, which the neighbour cell will receive. As it is defined, the downstream cells could reach a higher water level

Initialise variables, cell mask and boundary conditions

1) Start time loop
   a. Update simulation time:
   b. For each cell compute the volume of water to transfer, i.e., intercellular-volume
   c. Update the water depth (intercellular-volume/rain/etc.) for the update step
   d. If the simulation time reaches the next update timing
      i. Compute the velocity vectors
      ii. Compute the next in each cell for the update step .
      iii. Retrieve the minimum for the whole computing domain
      iv. Check water volume balance

Fig. 1. The pseudo code of the WCA2D model.

than the central cell, which may cause oscillations. To minimise that problem, the central cell is considered to retain a fraction of the total intercellular-volume transferred. This is achieved by adding the minimum available storage volume to the total available storage volume, i.e., this minimum represents the weight of the central cell, for weight computing. Fig. 2 shows an example of how the weights are computed. This step is described by the following equation:

$$w_i = \frac{\Delta V_{0,i}}{\Delta V_{tot} + \Delta V_{min}}, \quad w_0 = \frac{\Delta V_{min}}{\Delta V_{tot} + \Delta V_{min}} \forall i \in \{1 \dots m\}. \tag{6}$$

where, w_i is the weight of the ith cell.

The total intercellular-volume, i.e., the volume of water that leaves the central cell, differs from the total available storage volume and it is calculated by Eq. (11) which takes the minimal value between three different terms.

In the first term, the total intercellular-volume is limited by the amount of water that exists in the central cell. In the second term of the equation, a physically based limitation is imposed on the total intercellular-volume by using the critical flow equation and the Manning's formula:

$$v_{crt} = \sqrt{gd} \tag{7}$$

$$v_{man} = \frac{1}{n} R^{\frac{2}{3}} S^{\frac{1}{2}} \tag{8}$$

where g (m s^{-2}) is the gravitational acceleration, d (m) is the water depth in the cell, v_{crt} (ms^{-1}) is the critical flow velocity, n is the Manning's roughness coefficient (m$^{-1/3}$s), R (m) is the hydraulic radius and S is the absolute value of the hydraulic gradient (−), v_{man} (ms^{-1}) is the cross-sectional average velocity.

Considering a typical square grid approach, the critical flow condition equation and the Manning's formula would be computed, on average, twice per central cell visited, with the outflow in each Cartesian direction. These equations are computationally expensive since they use the less efficient power and square root operations. By reducing the number of times Eqs. (7) and (8) are computed, the model performance can be significantly improved. The WCA2D uses, Eqs. (7) and (8) to calculate the maximum permissible intercellular velocity from the central cell into a neighbour cell, and thus the maximum intercellular-volume. The neighbour cell to receive this maximum volume of water is the cell with the largest weight. Therefore, the total intercellular-volume is limited by the value derived from the maximum individual intercellular-volume divided by the maximum weight. The intercellular-volume of the other downstream cells, i.e., with smaller weights, is limited by the ratio between their individual weights and the maximum weight.

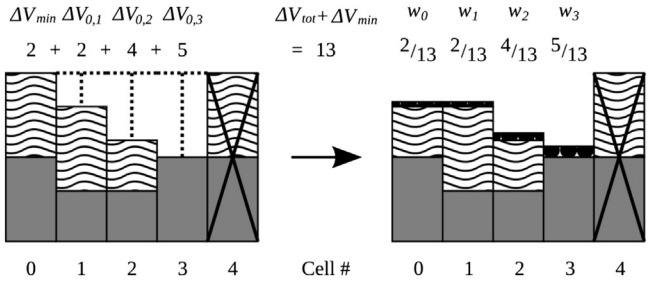

Fig. 2. Example of intercellular-volume computation. The dark shading represents the ground level in a cell, the wave pattern represents the amount of water available in a cell.

Thus, Eqs. (7) and (8) are computed only once per central cell visited using this weighting system.

In the third term of Eq. (11), the total intercellular volume to leave the central cell is limited by the minimum available storage volume ΔV_{min} plus the total intercellular-volume I_{tot} that left the cell at the time step t, which is determined during the previous time step iteration. The minimum volume ΔV_{min} is used to limit oscillations that may occur when a neighbour cell receives water from more than one cell, which results in the water level being higher than the central cell's in the next time step. The value I_{tot} is used to avoid large differences in the total amount of transferred volume between steps and it is computed using Eq. (11).

The total intercellular-volume is computed using the following equations:

$$v_M = min\left\{\sqrt{d_0 g},\ \frac{1}{n} d_0^{\frac{2}{3}} \sqrt{\frac{\Delta l_{0,M}}{\Delta x_{0,M}}}\right\} \tag{9}$$

$$I_M = v_M\, d_0 \Delta t\, \Delta e_M \tag{10}$$

$$I_{tot}^{t+\Delta t} = min(\, d_0 A_0,\ I_M\, / w_M, \Delta V_{min} + I_{tot}) \tag{11}$$

where, M is the index of the neighbour cell with the largest weight, v_M (m/s) is the maximum permissible intercellular velocity from the central cell into the neighbour cell with the largest weight, $\Delta l_{0,M}$ (m) is the difference in water level between the central cell and the cell with the largest weight (this might not be the maximum difference in water level between all the downstream cells), $\Delta x_{0,M}$ (m) is the distance between the centre of the central cell and the centre of cell with the largest weight, d_0 (m) is the water depth in the central cell, I_M (m^3) is the maximum intercellular-volume achievable into the neighbour cell with the largest weight, Δt (s) is the time step, Δe_M (m) is the length of a cell edge with the largest weight, A_0 (m^2) is the area of the central cell, w_M is the maximum computed weight in the neighbourhood, and $I_{tot}^{t+\Delta t}$ (m^3) is the total intercellular-volume that will leave the central cell at time $t + \Delta t$.

The final step is to compute the intercellular-volume of each downstream cell, by multiplying the weight of the cell with the total volume of water transferred. This is achieved by the following equation, where $I_i^{t+\Delta t}$ (m^3) is the intercellular-volume of the ith cell at time $t + \Delta t$:

$$I_i^{t+\Delta t} = w_i\, I_{tot}^{t+\Delta t} \ \forall i \in \{1 \dots m\}. \tag{12}$$

2.2. Depth updating and total intercellular-volume computation

In the WCA2D model, the updating of the water depth is achieved by simply subtracting the intercellular-volume of the neighbour cells from the water depth of the previous time step. Given that the total intercellular-volume from a cell is limited by the amount of water available in the cell itself as in the first term of Eq. (11), the total mass within the computing domain is always conserved between time steps. Furthermore, the water depth of the next time step is updated with any lateral inflow or outflow (e.g., rainfall and infiltration). The following equation is used to update the water depth:

$$d_o^{t+\Delta t} = d_0^t - \frac{\sum_{i=1}^{m} I_i^{t+\Delta t}}{A_0} + \frac{\Delta V_0^{in}}{A_0} - \frac{\Delta V_0^{out}}{A_0} \tag{13}$$

where m is the number of cells in the neighbourhood, $I_i^{t+\Delta t}$ (m^3) is the intercellular-volume of the ith cell, A_0 (m^2) is the area of the

central cell, ΔV_0^{in} is a lateral input volume of water into the central cell (e.g., precipitation, inflow from upstream catchments, or backflow from downstream boundaries), ΔV_0^{out} is an eventual output volume of water from the central cell (for example from infiltration or lateral outflow), d_0^t (m) is the water depth of the central cell at time t, $d_o^{t+\Delta t}$ (m) is the updated water depth of the central cell at the new simulation time.

2.3. Velocity and time step calculation

As shown in the main algorithm of Fig. 1, the velocity and the time step are computed only every update step Δt_u. However, they are considered instant values since they are calculated using the information from the last iteration at time t. The velocity is calculated using the intercellular velocity at time $t + \Delta t$ which is computed using the following equation:

$$v_i^{t+\Delta t} = \frac{I_i^{t+\Delta t}}{d_{0,i}^{t+\Delta t}\Delta e_i \Delta t} \forall i \in \{1 \dots m\}. \tag{14}$$

where $I_i^{t+\Delta t}$ (m^3) is the intercellular-volume transferred to the ith cell at the new simulation time, Δe_i (m) is the length of the ith cell edge, $d_{0,i}^{t+\Delta t}$ (m) is the arithmetic average between the water depth of the central cell and ith cell at the new simulation time, Δt (s) is the time step.

The velocity vector in polar coordinates (magnitude and angle) at the new simulation time is computed using the following equations:

$$a = \sum_{i=1}^{m} v_i^{t+\Delta t} \cos \phi_i, \quad b = \sum_{i=1}^{m} v_i^{t+\Delta t} \sin \phi_i \tag{15}$$

$$\boldsymbol{v}^{t+\Delta t} = (r, \theta) = \left(\sqrt{a^2 + b^2}, \tan^{-1}\frac{b}{a} \right) \tag{16}$$

where, m is the number of cells in the neighbourhood, $v_i^{t+\Delta t}$ (m/s) is the intercellular velocity between the central cell and the ith cell, ϕ_i is the angle between the centroid of the central cell and the centroid of the ith cell, a and b are the components of the velocity polar vector, $\boldsymbol{v}$ is the velocity polar vector, r is the magnitude of the vector and θ is the polar angle.

Once the velocity is determined, as in the pseudo-code of the main algorithm shown in Fig. 1, the adaptive time step Δt is changed for the computing iterations for the next update step Δt_u, instead of after each iteration (as in the previous version of CA2D). Given that the WCA2D is a diffusive-like model, the time step Δt is calculated by computing on each cell of the grid the adaptive time step using the formula provided by Hunter et al. (2005):

$$\Delta t = \frac{\Delta x^2}{4} min\left(\frac{2n}{R^{5/3} S^{1/2}} \right), \; S > \sigma \tag{17}$$

The time step provided by this formula has shown stable results for diffusive-like problems; however it implies that the time-step reduces quadratically when the cell size decreases. Furthermore, as observed by Hunter et al. (2005), when the slope between two cells tends to zero, the time step tends to zero. The analytical solution case study, section 4.1, and empirical tests performed show that the WCA2D is less sensitive to the size of time step than a simple diffusive-like model that uses the Manning's formula to compute the intercellular-volume in each direction. Thus, if the free surface slope between two cells is less than a slope tolerance σ, the constraint in Eq. (17) does not apply. The slope tolerance prevents the minimum size of the time step from becoming too small thus avoiding the simulation time to become excessive. Adding a limiter to the time step is a common technique used in 2D hydraulic models (Hunter et al., 2005; Lamb et al., 2009).

The use of an update step Δt_u in the WCA2D model has improved the performance not only because the heavy computation of the polar velocity is performed less frequently, but also because the update of the time step is kept to a minimum. This reduced update of time step has a large performance advantage when the computation is executed on the GPU of a graphics card. When Eq. (17) is calculated in every cell of the grid, then the minimum Δt needs to be retrieved and broadcasted to the global domain, which leads to a bottleneck in the parallel performance of the GPU computation since it is an inherently sequential process. Thus the use of the update step minimises this performance bottleneck.

2.4. Implementation

The WCA2D model has been implemented using the CADDIES Application Programming Interface (API) framework (Guidolin et al., 2012) that defines a set of methods, data structures and variables to be used as the standard for developing parallel CA algorithms. The main idea of the CADDIES API is that a developer needs to write the code of the CA model only once. After that, the CADDIES API gives the flexibility to produce the same CA model for any type of CA grid, square/hexagonal/triangular grid, and to use different high performance acceleration techniques without changing the code or with minimum effort.

The WCA2D model has been designed to work on grids with different cells and neighbourhood types. However, in this work, the model was implemented using only a square cell grid with a von-Neumann neighbourhood. Furthermore, thanks to the CADDIES API, the WCA2D can be executed on a multi-core CPU only using OpenMP library (Dagum and Menon, 1998) and in both multi-core CPU and graphics card GPU using the OpenCL library (Munshi et al., 2011).

The implementation of the WCA2D model used in this paper, called CADDIES-caflood, will be publicly available using an open source license in the website of the Centre for Water Systems at: http://cws.exeter.ac.uk.

3. Test cases

The model developed in this work has been applied to five different case studies: the analytical solution proposed by Hunter et al. (2005), three test cases from the EA benchmarking tests for 2D flood modelling (Néelz and Pender, 2013), and one real world case study in the area of Torquay in the UK.

The WCA2D results of the EA benchmarking and Torquay test cases were directly compared to the results obtained by using the commercial software InfoWorks ICM 3.0 (referred to as IW; Innovyze, 2012). The IW software is widely used in the water industry in the UK, and is one of the 2D hydraulic models used in the EA benchmarking exercise (Néelz and Pender, 2013). InfoWorks utilises a finite volume numerical scheme to solve the full SWEs. Given that IW software is widely used and that it was designed to achieve fast computations using the latest GPU technology, it was considered a good benchmark to test the accuracy and performance of the new model. However, unlike the WCA2D, the IW software uses an irregular triangular mesh grid and computes the inertial term of the SWEs. Thus, some discrepancies between the results of the IW and the WCA2D were expected.

All these simulations were performed on a high-performance desktop machine with an Intel Core i7-4770 K CPU with four physical cores at 3.50 GHz, 32 GB of main memory and a Tesla K20c graphics card with 2496 CUDA cores and 5 GB of video memory.

3.1. Analytical solution

To rigorously analyse the performance of an inundation model, Hunter et al. (2005) adopted a one-dimensional analytical solution where the full Saint-Venant equations can be simplified to produce an ordinary non-linear differential equation. This can be solved analytically at any point in space x and in time t over a horizontal plane:

$$d_{x,t} = \left[\frac{7}{3}\left(C - n^2u^2(x-ut)\right)\right]^{3/7} \quad (18)$$

where d (m) is the water depth, u (ms^{-1}) is the component of the depth-averaged velocity in the x direction, C is a constant of integration, which can be determined by referring to the initial condition of the problem.

Eq. (18) has been used extensively in the literature (Hunter et al., 2005; Bates et al., 2010; Dottori and Todini, 2011) to test the ability of different models to simulate correctly wave-propagation in absence of a bed slope term. In this test case, the parameters in Eq. (18) were $u = 1(ms^{-1})$ and $n = 0.03$ ($m^{-1/3}s$) for a total duration of 60 min.

3.2. EA benchmarks test cases

The EA 2D benchmarking test cases have been applied to a number of 2D hydraulic models to test the models' capability and performance for simulating different hydraulic conditions. The detailed information about these test cases and the results obtained by various models analysed have been published in order to create a point of reference for evaluating various models (Néelz and Pender, 2013). The benchmarking test cases vary in complexity and nature, from the flooding of a simple disconnected water body to a dam break scenario. In this work, three test cases chosen are: Test 2 (EAT2), filling of floodplain depressions, Test 4 (EAT4), flood propagation over an extended floodplain, and Test 8a (EAT8a) runoff produced by rainfall and a point source in a small urban area. The first two tests were chosen because they were designed to test the ability of a model to solve a specific type of problem. While being of a small spatial scale, EAT8a was chosen because it represents a typical urban flood scenario that the WCA2D model was designed to solve at larger spatial scale. All the other test cases are either more complex, i.e., where multiple problem types need to be solved, or the WCA2D model was not designed to solve. For example, Test 3 requires momentum conservation over a small obstruction.

3.3. Real-world test case

The Torquay area, in the south-west of the UK, is a challenging test case due to rapid changes in its slope, i.e., a steep slope in the upper catchment and a gentle slope in the lower catchment. A design rainfall of 40 mm/h was applied to the whole area for one hour. The catchment was set to be 100% impervious. The modelling of the drainage system was not included since this is not implemented in the WCA2D model. The full simulation time was 12 h. Three Digital Elevation Models (DEMs) of respectively 8, 4 and 2 m resolutions, obtained from averaging of 1 m Light Detection and Ranging (LiDAR), were used as a basis for simulations using the WCA2D and the IW. The terrain (see Fig. 3) elevation ranges from 0 to 180 m, and the average slopes without buildings are 14.8% for the 8 m DEM, 18% for the 4 m DEM and 16.1% for the 2 m DEM. The average slope changes between resolutions due to the loss of information caused by the coarsening of the 1 m DEM. The catchment was delineated after the hydrological analysis so that the surface water at the upstream boundary, i.e., ridges, would flow toward the modelling domain, and the water reaching the downstream boundary, i.e., the coast, was free to leave the modelling domain. The terrain does not contain any rivers or channels and the main slope direction is from north to south where the Torquay harbour is located (close to point 10 in Fig. 3). A constant Manning roughness of 0.015 ($m^{-1/3}s$) was applied to the whole area. The inundation depth results of the WCA2D and the IW were compared at various times, respectively at 30, 60, 90, 120, 360, and 720 min. The maximum inundation depth and maximum velocity results were also compared. The hydrographs of water levels at various points of interest were also used to compare the results of the two models. These points of interest are located in the various areas of ponding of the main flow paths, as shown in Fig. 3.

3.4. Metrics

For the real test cases, the comparison between the base raster grid (converted from the IW results) and the tested raster grid (WCA2D results) is composed of four metrics:, 1) Root Mean Square Error (RMSE), 2) R-squared (R^2), 3) True Positive Rate (TPR) and 4) False Discovery Rate (FDR). These metrics have been selected from a wide range of measures, as suggested by Bennett et al. (2013), to characterise the performance of environmental models. All these metrics consider wet cells as those that have at least 0.1 m of water depth, the standard height of a kerb in the UK. The first two metrics consider only the cells that are wet in either result. The RMSE metric is calculated using Eq. (19), where Y_i^b is the value (i.e., water depth or velocity) of ith cell of the IW result, Y_i^T is the value of the ith cell of the WCA2D result, and p is the total number of wet cells.

The RMSE metric is calculated using the following equation:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{p}\left(Y_i^b - Y_i^T\right)^2}{p}} \quad (19)$$

The R-squared metric is calculated using Eq. (20), where, $\overline{Y^b}$ is the mean value between all the wet cells of the IW result.

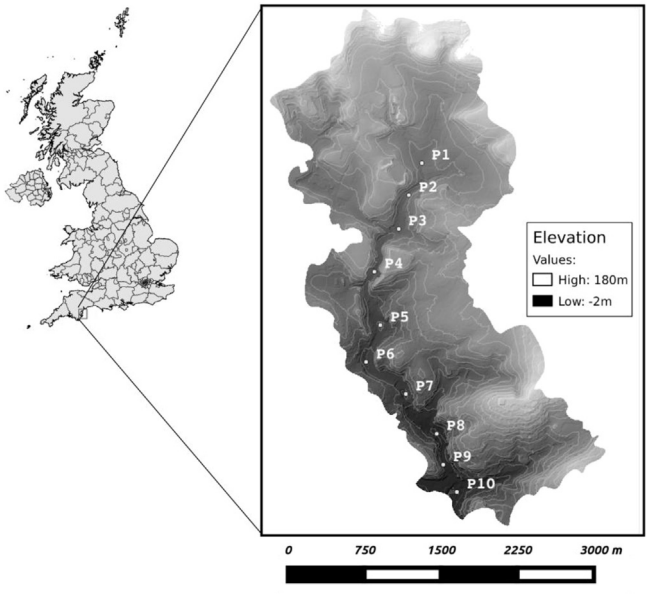

Fig. 3. The digital terrain model of the Torquay study area with hill-shade effect, contour lines at every 10 m, and the ten points of interest.

$$R^2 = 1 - \left[\frac{\sum_{i=1}^{p}\left(Y_i^b - Y_i^T\right)^2}{\sum_{i=1}^{p}\left(Y_i^b - \overline{Y^b}\right)^2}\right] \quad (20)$$

The TPR and FDR metrics, also respectively called probability of detection (hit rate) and false alarm ratio (Bennett et al., 2013), are used to compare the inundated area predicted by the two models. They are calculated using Eqs. (21) and (22), which are derived from the confusion matrix (contingency table). True Positive (TP) represents the number of cells that both the WCA2D and the IW identified as wet (hit). False Positive (FP) represents the number of cells that the WCA2D identified as wet while the IW identified as dry (false alarm). True Negative (TN) represents the number of cells that both the WCA2D and the IW identified as dry (correct negatives). False Negative (FN) represents the number of cells that the WCA2D model identified as dry while the IW software identified as wet (misses).

$$TPR = TP/(TP + FN) \quad (21)$$

$$FDR = FP/(TP + FP) \quad (22)$$

The TPR metric represents the agreement of the flooded area between the results from the IW and the WCA2D. The ideal value is one and it can be used to analyse the under-prediction of the flooded area by the WCA2D. The FDR metric is the ratio of the area identified as non-flooded in the IW software (FP) but flooded in the WCA2D model (TP + FP). The ideal value is zero and it can be used to analyse the over-prediction of the flooded area by the WCA2D.

4. Results and discussions

4.1. Analytical solution

Fig. 4(a) shows the analytical solution compared to the results obtained by the WCA2D model using two different grid resolutions, $\Delta x = 25$ m and $\Delta x = 10$ m, where Eq. (17) was used to compute the time step Δt and the update step Δt_u was equal to 60 s. Fig. 4(b) shows the enlarged-scale fronts of the results obtained by the WCA2D model, with $\Delta x = 10$ m, when the update step Δt_u has different values, from 0.5 s to 60 s. Table 1 summarises the RMSE, the minimum time step Δt for all the results and the run-times obtained by the WCA2D model when executed on a GPU.

As expected, the WCA2D is sensitive to the grid resolution as shown in Fig. 4(a) and Table 1. These results are equivalent to the results obtained by Bates et al. (2010) using the diffusive version of the LISFLOOD-FP. Fig. 4(b) and Table 1 show that the differences in the results are minimal for different values of the update step Δt_u. When Δt_u is greater than 1 s, the differences are too small to be visible in Fig. 4(b). Thus the WCA2D model is insensitive to the value of the update step chosen in this test case. However, Table 1 shows that this parameter has an impact on the run time of the model; the execution with $\Delta t_u = 60$ s is over 10% faster than the one with $\Delta t_u = 0.5$ s. The increase in run-time with would be even greater if the time step was updated after every iteration of the main loop.

Table 1
RMSE values and minimum time step of the WCA2D at different grid resolutions and update time step values.

Δx (m)	Δt_u	RMSE (m)	Min Δt (s)	Run time (s)
25	60	0.0313	0.038	3.99
10	60	0.0139	0.006	30.05
10	10	0.0141	0.006	30.59
10	1	0.0140	0.006	32.41
10	0.5	0.0136	0.006	34.47

The use of large values of the update step, Eq. (17) produces a time step Δt that is optimal for the first iteration after the update, but this Δt should become less optimal toward the later iterations. However, numerical experiments performed on the other test cases also show the solutions are insensitive to the update step if its value is maintained across different problems. To test the impact of using sub-optimal time step, the results of different versions of the WCA2D model using large fixed time step Δt, with $\Delta x = 10$ m, are compared with the analytical solution. The fixed Δt values used are 0.018 s and 0.024 s, which are respectively three and four times the minimum step as shown in Table 1.

The various versions of the WCA2D model tested differ on the operations used to compute the intercellular-volume step and they are: 1) the full WCA2D model; 2) the WCA2D model where the total volume to transfer from a cell is computed considering only the total amount of water available and the maximum flux calculated using Eq. (10), i.e., when the third term of the minimisation of Eq. (11) is removed; and 3) a simple diffusive-like model that uses the Manning's formula to compute the intercellular-volume in each direction, similar to how the diffusive version of LISFLOOD-FP works.

Fig. 5(a) and (b) show the results of the various versions at $\Delta t = 0.018$ s and 0.024 s, respectively. The version that uses the

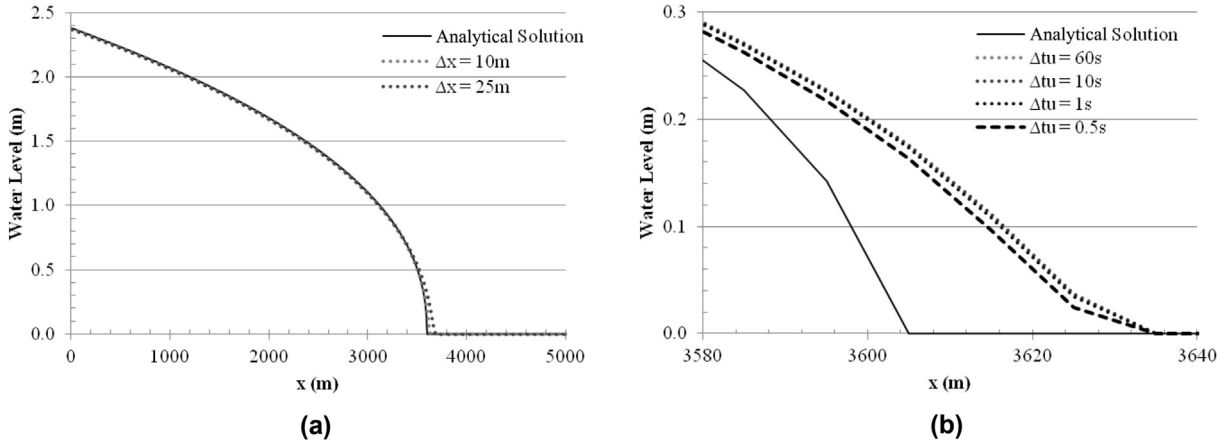

Fig. 4. Comparison between analytical solution and the WCA2D model at t = 3600 s for: (a) different grid resolutions; (b) different update step values at Δx = 10 m.

Manning's formula to compute the intercellular-volume largely under-predicts the flood wave volume and, in the case of the larger time step, under-predicts also the front location. The second version, which uses the weighted system and the maximum flux to compute the intercellular-volume, is shown to reduce the under-prediction; however, this is still significant. Fig. 5 shows that the full WCA2D model under predicted the flow wave volume, but only marginally, while using a time step that is three times larger than the minimum one given by Eq. (17). Furthermore, the WCA2D predicts the right front location also when the time step is four times larger. Fig. 5(b) shows also that with the larger time step there are no oscillations in the water level predicted by the WCA2D model, while the other two versions produced some oscillations.

The weighting system of Eq. (6) was implemented in the WCA2D model to reduce the number of computational expensive operations performed, such as power and square root. Fig. 5 shows that this weighting system has also the advantage of lower model sensitivity to the size of the time step in comparison to a diffusive model that uses Manning's formula to compute the intercellular-volume in each direction. Fig. 5 also shows that by using the minimum positive transferrable volume ΔV_{min} plus the total intercellular-volume I_{tot} (that left the cell during the previous step) to compute the total volume to leave the central cell further lowers the sensitivity of the WCA2D to the size of the time step. However, the WCA2D model is not completely insensitive to the size of the time step Δt.

4.2. EA benchmarks test cases

Table 2 shows the parameters (Δt_u, σ) of various simulations used to solve the EA Benchmarks test cases. An important configuration step is to decide the value of slope tolerance σ for each test case. This value has a large impact on the final result and on the computational time, since it influences the time step used during the simulation. In order to analyse this impact, for all these test cases, two values of σ are used that produce two types of results: 1) fine precision, but high run time, and 2) coarse precision, but fast run time. These values are evaluated empirically, but the coarse precision ones are determined as an order of magnitude lower than the average terrain slope of each case. The EAT4 test case is an exception of this rule due to the terrain being a horizontal plane.

Fig. 6 shows the DEM map with the output point locations for the three EA benchmarks test cases: (a) the map for the EAT2 problem with the upstream boundary condition (red line) and the ground elevation contour lines at every 0.05 m; (b) the map for the EAT4 problem with the location of the inflow and the possible 10 cm and 20 cm contour lines at 1 h (dashed) and 3 h (solid); (c) the map for the EAT8a problem with the location of the inflow and purple lines for the outline of road and pavements (Néelz and Pender, 2013).

Table 2
Information about the WCA2D simulations of the EA Benchmarks test cases.

Parameters/Test case	EAT2		EAT4		EAT8a	
# of data cells	10,000		80,000		97,000	
Event duration	48 h		5 h		5 h	
Average slope	0.22%		0%		5.28%	
Update time step Δt_u	60 s		60 s		60 s	
Type of σ	Fine	Coarse	Fine	Coarse	Fine	Coarse
Value of σ	0.004%	0.022%	0.001%	0.01%	0.1%	0.528%
Mean time step Δt	1.28s	5.17s	0.036s	0.057s	0.043s	0.138s

The EAT2 test consists of a gently sloping squared area with a 4×4 matrix of ~0.5 m deep depressions. There is an inflow from the north-west corner which produces a low momentum flow. A constant Manning roughness 0.03 ($m^{-1/3}s$) was applied to the whole area and the DEM resolution is 20 m. The original problem specifies 16 output points at the centre of the depressions. Fig. 7 shows the comparison of the water levels versus time at points 4, 7, 12, 10, 1, and 5, see Fig. 6(a); these are ordered by distance from the inflow source. The points on the left-hand side of the domain, 13–16, are not shown since they stayed dry for both models. There is a good agreement between the results of the WCA2D with slope tolerance $\sigma = 0.004\%$ and the IW, mainly in the shape of the hydrographs and in the maximum inundation values. Some discrepancies exist between the two models, mainly in terms of the arrival time of the front at the location points far from the inflow source (1, 5 and 10) and in the maximum inundation value at point 5. The results of the WCA2D with the larger slope tolerance $\sigma = 0.022\%$ are still in good agreement with the IW. However, increasing the slope tolerance decreased the average time step and produced hydrographs with a slightly delayed arrival of the front at the points far from the inflow source. The maximum inundation values are similar to the ones found by the WCA2D using the smaller time steps; apart from point 4 where the larger time step produced an imperceptible higher peak. The results of Fig. 7 show that WCA2D is sensitive to the value of the slope tolerance and thus it is not completely insensitive to the size of the time step Δt.

The EAT4 test consists of a flat floodplain and it is intended to test a case when a flood wave occurs following an embankment defence failure by breaching or overtopping. The boundary condition consists of an inflow from the central west border. A constant Manning roughness of 0.05 ($m^{-1/3}s$) was applied to the whole area

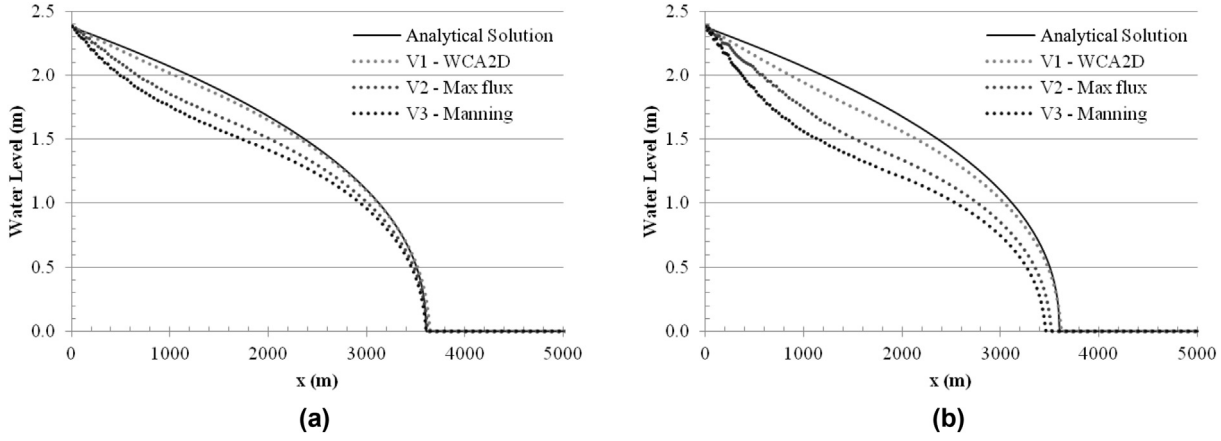

Fig. 5. Comparison between analytical solution and various versions of the WCA2D at t = 3600 s with $\Delta x = 10$ m using: (a) a fixed Δt of 0.018 s; (b) a fixed Δt of 0.024 s.

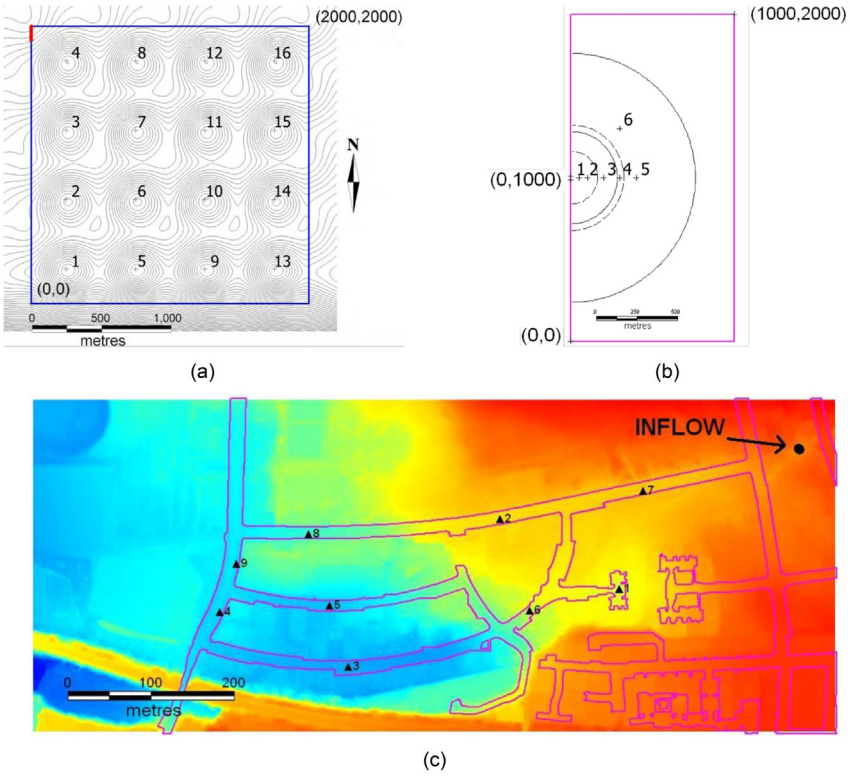

Fig. 6. The DEM map of the EAT2 (a), EAT4 (b) and EAT8a (c) problems; figures taken from (Néelz and Pender, 2013).

and the DEM resolution is 5 m. The original problem specifies 6 output points. Fig. 8 shows the water levels versus time (left column) and velocity versus time (right column) at points 1, 5, and 6, see Fig. 6(b). The results obtained by the WCA2D model with both values of slope tolerance are in good agreement with the IW results. However, there is a small discrepancy in water level compared with the IW results during the drying phase. The problem during the drying phase of the WCA2D model is more noticeable on the velocity plots where there are some visible oscillations at points 1 and 5 and large oscillations at point 6. The problem during the drying phase is caused by the lack of momentum terms in the WCA2D and by the use of a non-optimal time step during this phase, i.e., it being too large. In order to minimise the oscillations, it could be possible to reduce the time step by changing the slope tolerance. However, this would have a negative impact on performance. The points 5 and 6 in Fig. 6 are at the same distance from the inflow location, thus the resulting hydrographs should be the same in Fig. 8. The WCA2D model shows a small difference in the water levels between these two points (in the order of millimetres). This difference is more visible in the velocity plots. Thus, in the WCA2D model, there is a small asymmetry in the spread of the flow, which is due to the way that the intercellular-volume transfer is decoupled directionally.

The EAT8a test consists of an approximately 0.4 km by 0.96 km urban area in Glasgow, UK. The boundary conditions involve two sources, a uniformly distributed rainfall and an inflow from a point source (sewer overflow) in the top left corner of Fig. 6(c); the catchment is 100% impervious. The DEM resolution was 2 m and two Manning's roughness values were used: 0.02 ($m^{-1/3}s$) for road and pavement, 0.05 ($m^{-1/3}s$) elsewhere. The original problem specifies 9 output points. Fig. 9 shows the water level (left column) and velocity hydrograpshs (right column) in the ponding areas of points 1 and 3 and in the fast flow areas of points 2 and 6 located in the road, see Fig. 6(c). The water level results obtained by the WCA2D model with slope tolerance $\sigma = 0.1\%$ are in good agreement with IW. In the fast flow areas of point 2 and 6, the WCA2D predicted a faster flood wave than IW; this is more visible in the temporal plots of the velocity. Some oscillations are observed in the velocity predicted by the WCA2D in points 1 and 3. When a larger slope tolerance is used in the WCA2D, $\sigma = 0.528\%$, there are some small visible differences in the water level results; mainly at point 1 where the level was overpredicted after the peaks. In the case of the predicted velocities, the use of larger time steps caused some extra oscillations during the drying phase as shown at point 2.

Table 3 shows the run times for the three EA benchmark simulations for the two models on both MC and GPU executions. When compared to IW, the WCA2D run times were shorter for the EAT2 benchmark (for both MC and GPU executions) and also when the fine tolerance was used. In the case of the EAT4, the computational performance obtained by the WCA2D was comparable to the IW when a large tolerance value were used; the MC performance of WCA2D was not as good as in the first test case due to the particular characteristics of the EAT4 problem, i.e., the entirely flat terrain and a fast moving front. The WCA2D is a diffusive-like model which ignores any inertia terms and momentum conservation. Thus it needs to use very small time steps to move enough water between cells through the flat plane to reproduce the front. This increase in the number of steps had a large impact on the computational performance of the model.

In the case of the EAT8a, the run-times obtained by the WCA2D were slightly shorter than those obtained by the IW when the simulation was executed using a small tolerance value. When the WCA2D was executed using a large tolerance value the run times were significantly faster than the IW, over 4 times in the case of GPU execution. This test case represented a good indicator of the possible computational performance of the WCA2D model in a typical urban flood scenario even when the case study size is small.

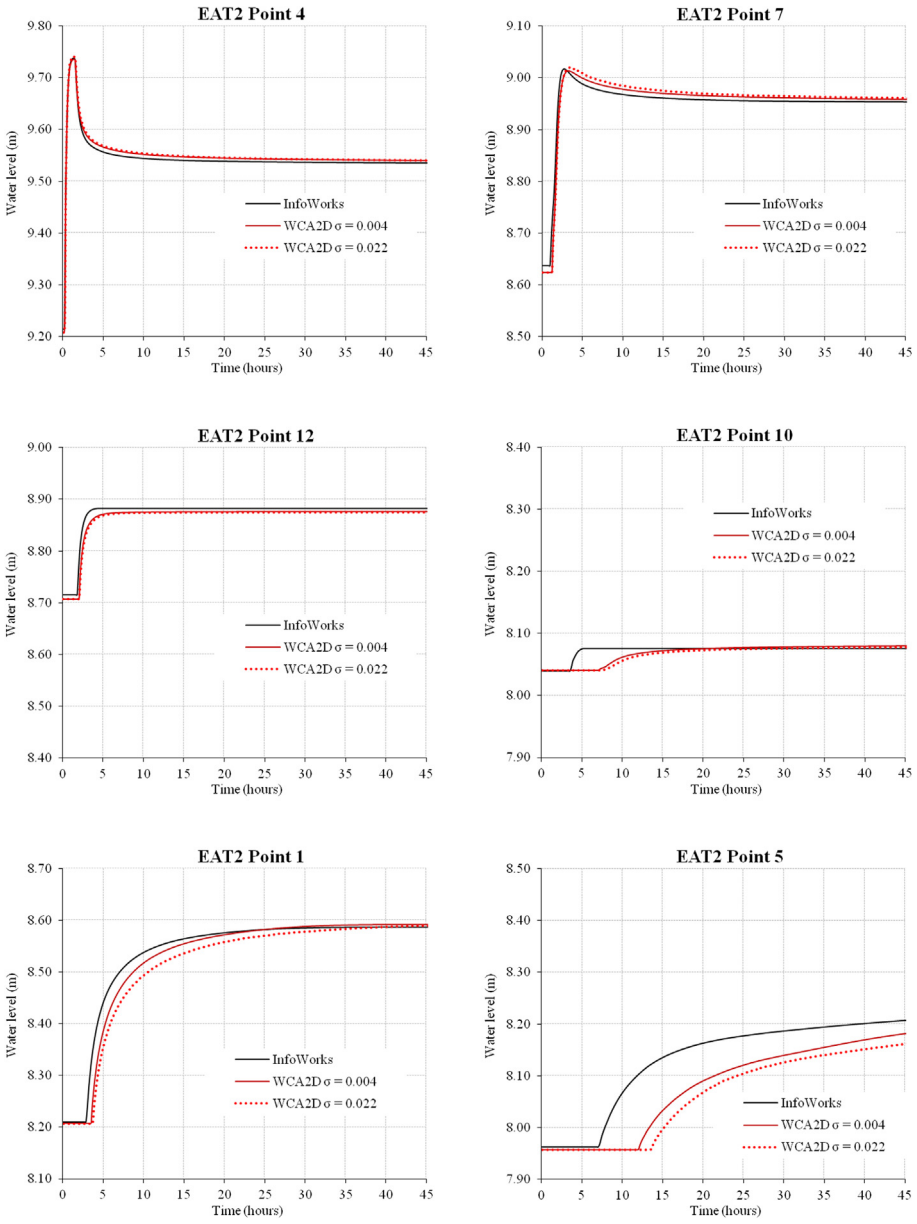

Fig. 7. Temporal variation of water level for EAT2 at point 4, 7 (1st row), 12, 10 (2nd row), 1 and 5 (3rd row); comparison between the WCA2D and the IW models.

The previous test cases show that by using a large tolerance value σ, i.e., a large time step during the simulations, it is possible to achieve faster run times than the IW while experiencing only minor loss in the accuracy of the predicted water levels and velocities. The WCA2D model predicted slightly delayed front of the flood wave with no oscillations on the water level, but it exhibited some visible oscillations in the velocity during the drying phase, which did not compromise the good agreement in the water elevations. By choosing the size of slope tolerance, a user of the WCA2D model will have to balance between reducing the computational time of the model and the possibility to have some oscillations in the results.

4.3. Torquay test case

In the Torquay test case, the WCA2D model runs were performed using DEMs with three different spatial resolutions, 2 m, 4 m and 8 m. The results were compared to the IW simulations with a similar number of cells at the three resolutions. The 2 m resolution was the highest resolution that the IW could cope with, given the 5 GB video memory of the graphics card available. The IW computational meshes for the different resolutions were generated directly from the 1 m LiDAR DEM without using the 'terrain-sensitive meshing' option; i.e., all the cells in a mesh have similar individual area even when there is a large variation in elevation.

The results from the IW were only provided in a triangular mesh (polygons) format. These results were rasterised from the polygon shape into a square grid raster using the Geospatial Data Abstraction Library (GDAL) version 1.8.0 (GDAL, 2013). This application uses the centre of the cell to identify the polygon from where to retrieve the value of the cell.

One common metric which was not used in this work to analyse the results is the direct cell-to-cell comparison of the Absolute Maximum Differences (Bennett et al., 2013). This was due to difficulties in transferring results between the triangular and square grid meshes. Fig. 10 shows an example of a large difference that was caused by this transfer. The triangular mesh results of the IW

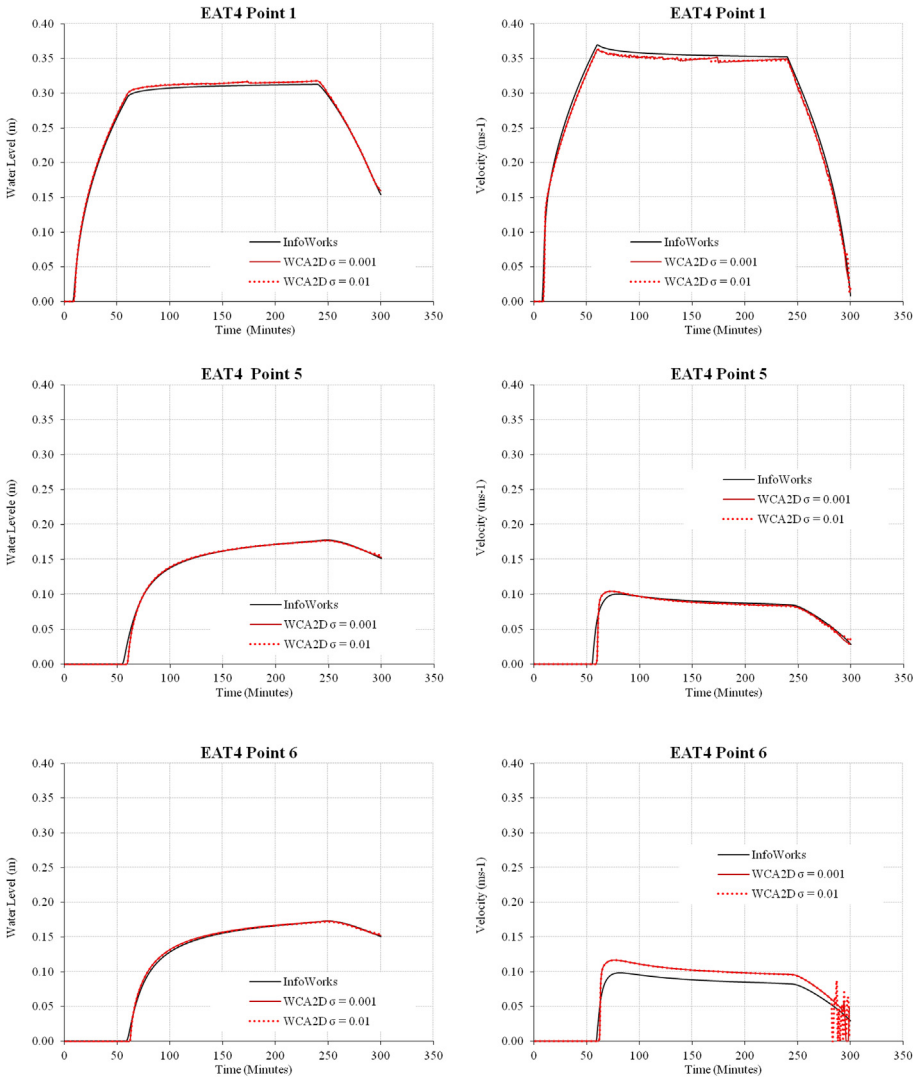

Fig. 8. Temporal variation of water level (left column) and velocity (right column) for EAT4 at point 1 (1st row), 5 (2nd row) and 6 (3rd row).

(Fig. 10(a)) show two triangles with a large difference in water depth due to a sharp change in elevation around a ponding area. In the results transferred from the IW to the square grid (Fig. 10(b)), the highlighted cell, which covered the area where the transition occurred, has a large part of its area covered by the triangle with greater water depth. However, its centre is located in the triangle with the lowest water depth such that the transferred cell water depth was small. The WCA2D square grid results (Fig. 10(c)) show that the CA model correctly identified the large water depth in the ponding area, including the highlighted cell. Consequently, the comparison between the IW and WCA2D results (Fig. 10(d)) shows that while there was a small difference between the two models when comparing the cell depths in the ponding area, in the highlighted cell (pond boundary) covering the sharp elevation change, there was a significant difference in water depth. Even when an interpolation technique was used to transform the results from triangular mesh to square grid, instead of using the centre of the cell, large differences in water depth due to the terrain variations did not disappear.

Table 4 shows the basic characteristics of the various simulations. For all simulations, the total duration was set at 720 min and floating point precision was set to single (32 bit) as it was the only option available in the IW for the GPU simulations. The update time step used in the WCA2D was 60 s and the maximum time step used in the IW was 30 s. In the case of the WCA2D, the slope tolerance, σ, has to be selected. As described earlier, this value has a large impact on the final result and on the simulation time, since it influences the mean time step. In this test case, the σ value was set to be an order of magnitude lower than the average slope of the terrain considered. This is the same process used to set the large σ value in the EA Benchmark test cases that produced results comparable to the IW while achieving faster run times. All the other parameters of the IW were kept as default values, unless indicated otherwise.

Table 5 shows the metrics values from the comparison between the IW and WCA2D results using 2 m, 4 m and 8 m grid resolutions. These metrics compare the maximum predicted values of the water depth and velocity. Furthermore, they are used in a temporal analysis by comparing the water depth values at 30, 60, 90, 120, 360 and 720 min. These represent the inundation extent before, during and after the peak time, which is between 60 and 90 min, and to compare the final inundation extent.

In the case of RMSE at 8 m resolution, the error ranges from 0.20 m to 0.36 m. For the 4 m and 2 m resolution the error is smaller, respectively below 0.20 m and below 0.15 m. A large contributor to these errors is probably the water difference in the peripheral cells that cover sharp elevation changes as shown in Fig. 10. For the R^2

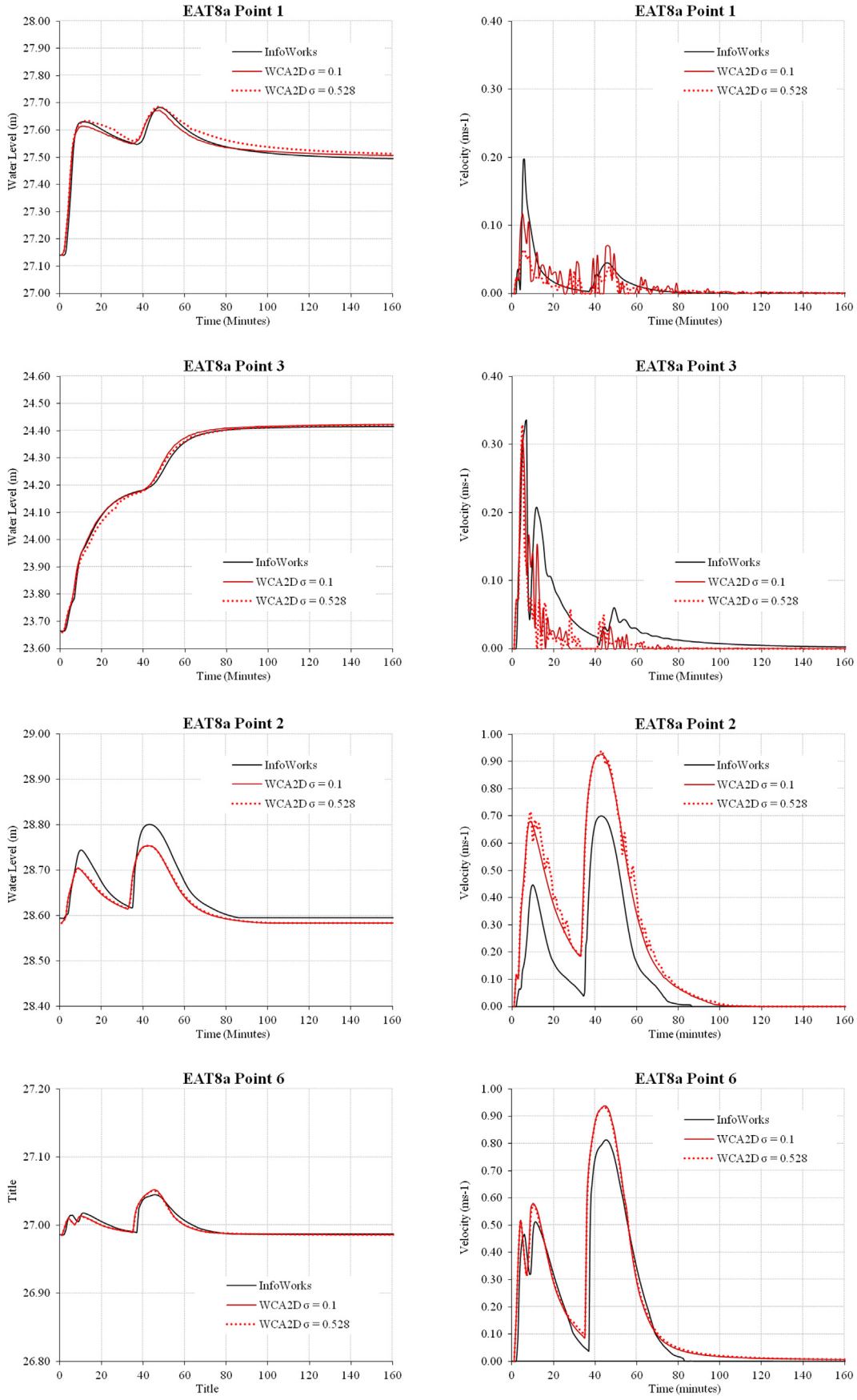

Fig. 9. Temporal variation of water level (left column) and velocity (right column) for EAT8a at point 1 (1st row), 3 (2nd row), 2 (3rd row) and 6 (4th row).

Table 3
Comparisons of the WCA2D run time for the EA test cases versus IW run time.

	Run time (seconds)					
	EAT2		EAT4		EAT8a	
Computation type	MC	GPU	MC	GPU	MC	GPU
WCA2D Fine σ	15.3	4.7	590.1	38.0	390.4	37.4
WCA2D Coarse σ	4.5	2.2	312.3	23.8	124.2	12.5
IW	20.1	9.3	260.9	22.4	448.4	58.8

metric, as expected the best agreement is between the models at 2 m resolution and it deteriorates for coarser resolutions. The values of the R^2 metric are over 0.95, around 0.90 and lower than 0.90 for respectively the 2 m, 4 m and 8 m resolutions. The agreement between the two models is always lower at 30 min of the simulations for all the resolutions, with the extreme case of the 8 m one.

The low values of R^2 at 30 min of the simulation could be caused by the differences in predicting the time of arrival of the various fronts between the two models. In the graphs showing the temporal variation of the water level of Fig. 12 it is possible to see that there is a sharp increase in the front around the 30-min mark. Thus, even a slight difference between the two models in predicting the time of arrival of the front, could result in a large difference in the water level. Another possible explanation is that the R^2 metric is sensitive to extreme values mainly when there are not enough data points, as it is in the case of the 8 m test cases and at 30 min of the simulations.

The TPR and FDR metrics are used to compare the inundation extend between the two models. The TPR metric shows that the WCA2D model predicts as inundated over 70% of the area identified by the IW in the simulation, with a maximum of 93% for the 2 m resolution test case. The FDR metrics shows the percent of area not

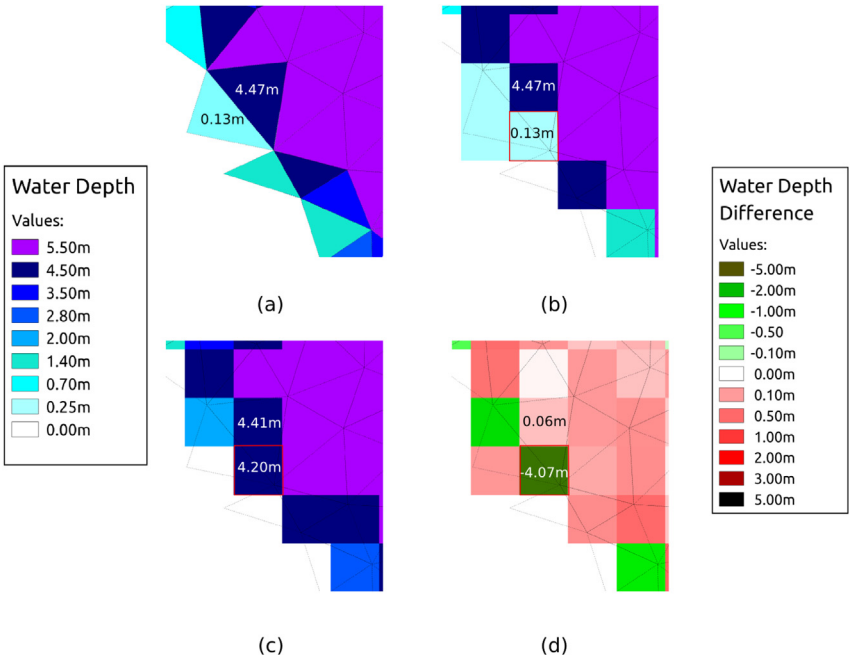

Fig. 10. Example of a large difference (d) in water depth between the square grid results of IW (b) and WCA2D (c) due to the transformation of the IW triangular mesh results (a).

Table 4
Information about the WCA2D and IW simulations of the Torquay test case.

Parameters/Model (Resolution)	WCA2D (8 m)	WCA2D (4 m)	WCA2D (2 m)	IW (8 m)	IW (4 m)	IW (2 m)
# of data cells/ = triangles	123,080	492,377	1,969,477	123,874	490,997	1,964,144
Avg. cell/triangle area	64.00 m^2	16.00 m^2	4.00 m^2	61.31 m^2	15.47 m^2	3.87 m^2
Type of time step	σ = 1.48%	σ = 1.8%	σ = 1.61%	Adaptive	Adaptive	Adaptive

Table 5
Metrics values of the comparison between IW and WCA2D results using 8 m, 4 m and 2 m grid resolutions.

Models comparison time/Attribute	IW 8 m – WCA2D 8 m				IW 4 m – WCA2D 4 m				IW 2 m – WCA2D 2 m			
	RMSE	R2	TPR	FDR	RMSE	R2	TPR	FDR	RMSE	R2	TPR	FDR
30 min.	0.20 m	0.59	0.70	0.34	0.13 m	0.86	0.82	0.28	0.10 m	0.93	0.88	0.26
60 min.	0.23 m	0.84	0.81	0.23	0.16 m	0.93	0.89	0.21	0.12 m	0.96	0.92	0.19
90 min.	0.30 m	0.86	0.76	0.18	0.18 m	0.95	0.85	0.12	0.14 m	0.97	0.88	0.08
120 min.	0.31 m	0.86	0.80	0.20	0.18 m	0.96	0.87	0.12	0.12 m	0.98	0.91	0.08
360 min.	0.35 m	0.82	0.78	0.25	0.20 m	0.95	0.88	0.15	0.12 m	0.98	0.93	0.08
720 min.	0.36 m	0.82	0.78	0.25	0.20 m	0.95	0.88	0.15	0.13 m	0.98	0.93	0.09
Max. depth	0.26 m	0.88	0.83	0.20	0.17 m	0.95	0.89	0.17	0.13 m	0.97	0.92	0.16
Max. velocity	0.30 m/s	0.65	–	–	0.35 m/s	0.65	–	–	0.43 m/s	0.60	–	–

predicted to be inundated by the IW but predicted to be inundated by the WCA2D model. This over-prediction ranges from a maximum of 34% of the area to a minimum of 8% for the 2 m resolution. Furthermore, the over-prediction happens mainly at the beginning of the simulations which could also be another factor influencing the values of R^2 at the 30-min mark.

The TPR and FDR results show that there is a good agreement in the inundation extent between the two models. Fig. 11 shows an example at 8 m and 4 m resolutions of the locations where the two models disagree about the flooded areas. The figure is focused on the downstream part of the terrain, near the Torquay harbour, and it shows in orange the area that has been predicted as flooded by both models, in red the area predicted as flooded only by WCA2D (over-prediction) and in black the area predicted as flooded only by IW (under-prediction by the WCA2D model). It is possible to see that the differences between the two models are mainly where there are sharp elevation changes, e.g, along various channels.

In the case of the temporal variation of the water level at various points of interest (see Fig. 3), Fig. 12 shows a good agreement between the results of the WCA2D model and the IW software for the 4 m and 2 m resolutions, mainly in the shape of the hydrographs (8 m resolution results are not shown for clarity of the pictures). There are some minor discrepancies in term of maximum inundation values and in the rate of discharge. There is a good agreement between the two models in term of the final inundation level.

Table 6 shows the computational performance of the two models obtained during the simulations for both MC and GPU executions. The run-time includes the pre-processing of the data and the model computation. However, it does not include the post-processing of the data and the creation of the 2D triangular mesh in IW.

In all these test cases, the WCA2D model is faster than IW. The WCA2D 8 m simulation obtains an increase in speed of four to eight times in comparison with the IW 8 m simulations. In the case of 4 m resolution simulations, the WCA2D model was from over two to almost four times faster than IW. Finally, for the 2 m resolution simulations, WCA2D was over twice as fast as IW. The WCA2D model achieved large speed ups when the computation is performed on the graphics card. This is due to the intrinsic parallelism of CA algorithms and to the minimal use of global sequential operation when computing the new time step by using the larger update step.

Table 6 shows that, while the WCA2D model obtains faster run-times, there is a decrease in the computational speedup at higher resolutions. Given that WCA2D is a diffusive-like model, it suffers from the same problem, like other diffusive storage cell codes where the size of time-step reduces quadratically with the decrease in distance between two adjacent cells. Further advantage of the WCA2D model in comparison with IW is the reduced memory needed to run the simulation as shown in Table 6. In the case of the Torquay case, the WCA2D model would be capable of running a simulation at 1 m resolution, even on a mass-market GPU with around 2 GB of main memory, while IW could not achieve this efficiently on a professional GPU with 5 GB of memory. As shown in Table 6, the computational impact of the use of a GPU is significant, since the GPU run-times are, on average, eight times faster than MC run-times for all the simulations.

5. Conclusions

This paper presents a new CADDIES 2D flood model WCA2D based on cellular automata that minimises the use of complex and time consuming equations in order to achieve fast computational performance without a significant sacrifice of accuracy. The new model was compared to an analytical solution and it has been applied to three test cases from the 2D benchmarking tests for 2D flood modelling proposed by the EA, and one real world case study in the area of Torquay in the UK.

The comparison with the analytical solution shows that the WCA2D model produces accurate results which are in line with those obtained by other diffusive-like model like the diffusive version of LISFLOOD-FP (Hunter et al., 2005). Furthermore, it shows that, thanks to the new weight system employed, the WCA2D model is less sensitive to the size of the time step in comparison to a diffusive-like model that uses the Manning's formula to compute the intercellular-volume in each Cartesian direction.

Numerical results obtained for the three 2D benchmarking test cases and for the real world case study were compared with those of a physically based model IW, which is an industry standard for 2D flood modelling, which computes the full SWEs on an irregular

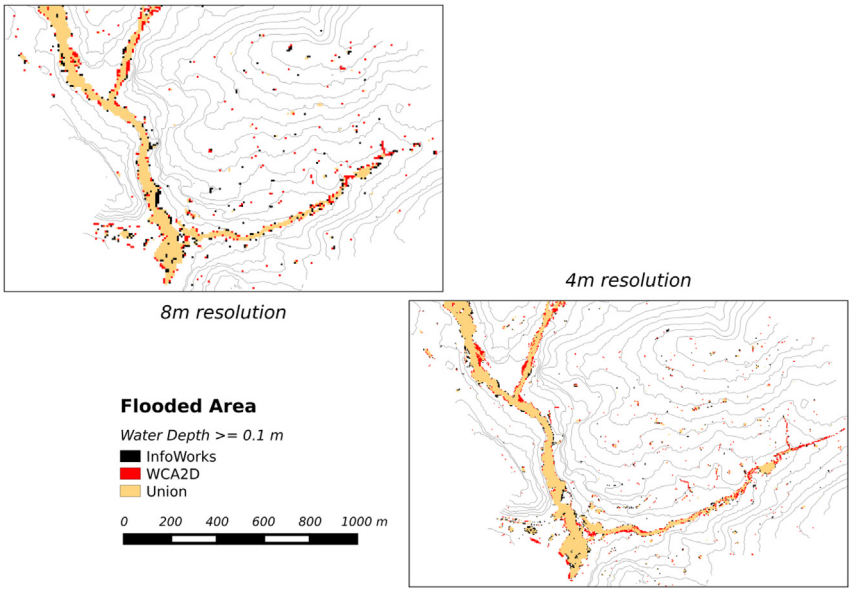

Fig. 11. Area predicted as flooded by IW (black pixel), by WCA2D (red pixel) and by both models (orange pixel) at 8 m (left) and 4 m (right) resolutions.

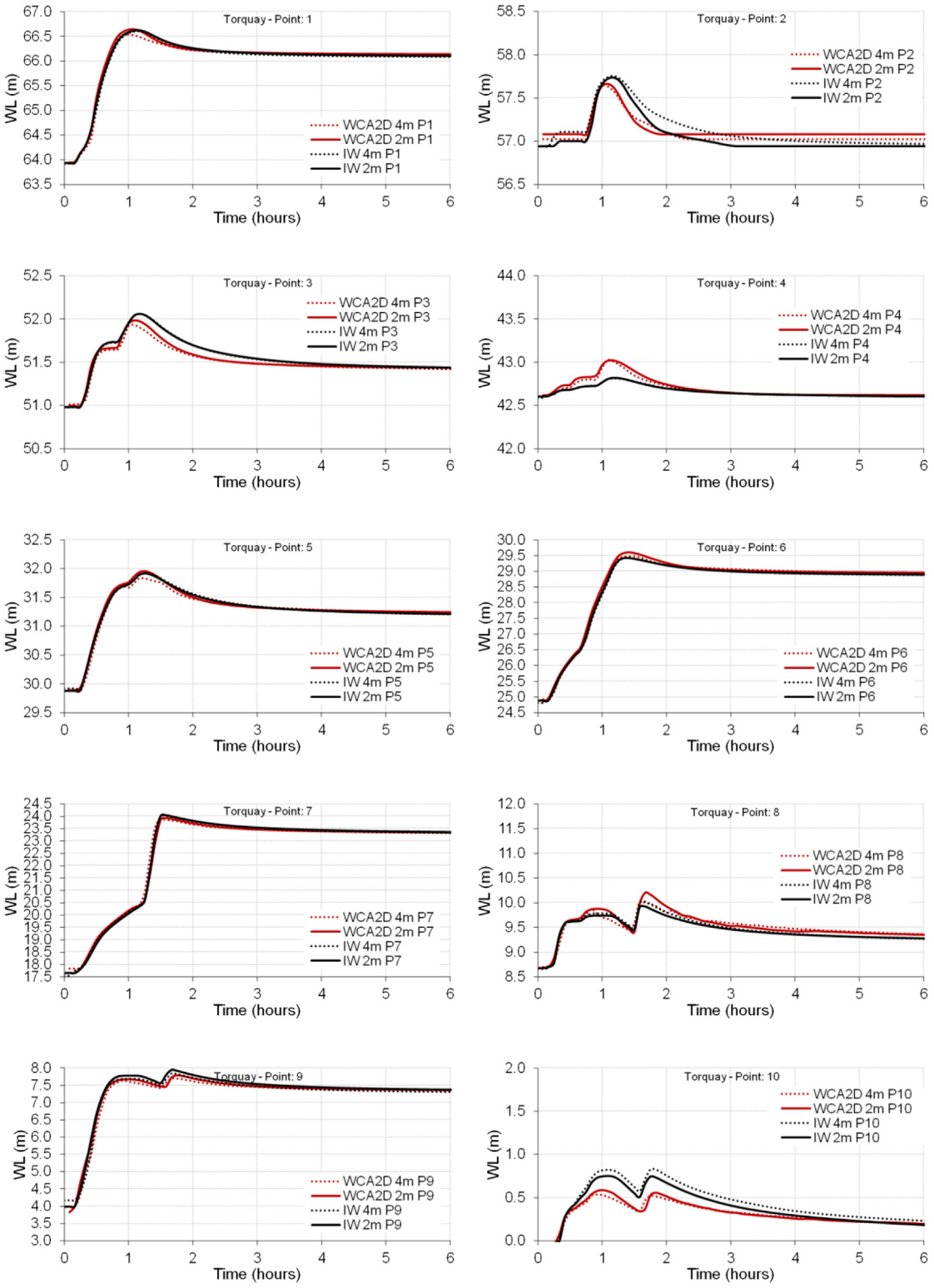

Fig. 12. Temporal variation of water level for Torquay problem at various point of interest; comparison between WCA2D (red line) and IW (black line) at 4 m (dotted line) and 2 m (solid line) resolutions.

triangular mesh grid. The water level hydrographs obtained at various points for the three EA test cases show results consistent with the IW model. In the case of the temporal variation of the velocity, the WCA2D shows some visible oscillation when the time step used is large, but these oscillations did not compromise the water elevation results. In the real-world case of Torquay, the depths at various points in time and the maximum inundation depths show a satisfactory agreement between the WCA2D and IW results. The comparison of the inundation extent predicted by the two models show that WCA2D model identify as flooded from

Table 6
Computational performance of the WCA2D and IW simulations of the Torquay test case.

Parameters/Model (Resolution)	WCA2D (8 m)	WCA2D (4 m)	WCA2D (2 m)	IW (8 m)	IW (4 m)	IW (2 m)
Memory used	~85 MB	~131 MB	~308 MB	~230 MB	~900 MB	~3600 MB
Computation type	MC	MC	MC	MC	MC	MC
Run time (minutes)	1.78	21.45	284.80	7.26	62.25	609.90
Speed-up vs IW	4.09	2.90	2.14	–	–	–
Computation type	GPU	GPU	GPU	GPU	GPU	GPU
Run time (minutes)	0.19	2.28	26.34	1.52	8.98	76.25
Speed-up vs IW	8.00	3.94	2.89	–	–	–

between 70% and 90% of the area identified as flooded by IW; with the main differences arising on the sides of the flood channels. Furthermore, the numerical results show a very good agreement between WCA2D and IW in the case of the temporal variation of water level at various points of interest. Only the maximum velocity results predicted by WCA2D are less in agreement with the results predicted by IW.

The WCA2D run-times were compared to those obtained by IW. In the EAT4 case, due to the characteristics of the problem with a flat free surface gradient terrain and a fast moving front, the run-times obtained by WCA2D using the multi core processor and a graphics card were slightly slower than the ones obtained by IW. The WCA2D model obtained shorter run-times in the EAT2 case and in the case of the urban flood modelling scenario of the EAT8a case. In the Torquay case study, the WCA2D run-times were up to 8 times faster than the ones of IW. This obviously depends on the terrain resolution and parallelization technique implemented. Furthermore, it was demonstrated in this work that the proposed model is particularly suitable for parallelization by using either a MC or a GPU implementation of the model.

This work showed that the WCA2D model could be used to perform 2D flood simulations at a large scale due to its high computational performance and low memory requirement with a minimal compromise in accuracy. While the improvements obtained by WCA2D in comparison to a full SWE model might not influence the choice of the model for a relatively small domain areas, or when only few simulations are performed, the performance improvements of WCA2D could be very significant for large domains with a significantly large number of simulations (e.g., for risk analysis).

The next fundamental step in the development of the WCA2D is to implement an inertial like effect. This model considers only the water surface slope and ignores any momentum of the flow; thus it suffers from the same problem of other diffusive wave system models where the time step reduced quadratically with the decrease in distance between two adjacent cells. By adding an inertial like effect, the time step could be reducing linearly with the decrease of spatial resolution of the terrain modelled as showed by Bates et al. (2010) in the inertial version of LISFLOOD-FP. This would reduce significantly the run-times of the model at higher resolutions. One important research objective will be to transfer into an inertial like model, the advantages obtained by the WCA2D model, i.e., reduced sensitivity to the size of the time step, reduced use of computational intensive operations and the ability to highly parallelise the computation.

Further development steps could be to implement the model using different neighbourhood and cell types (hexagonal, triangular) to identify the grid characteristics that offer the best performance and accuracy and to implement a Building Coverage Ratio (BCR) technique (Chen et al., 2012) or a depth-area/depth-volume relationship technique (Vojinovic et al., 2013), which would allow to address urban features or high resolution features in a coarse grid resolution DEM.

Finally, the 2D model presented here for overland flood modelling will be integrated with the CADDIES 1D model for sewer network modelling developed by Austin et al. (2014), to produce a fast simplified dual-drainage model for urban flood modelling.

Acknowledgements

The authors would like to acknowledge the funding provided by the UK Engineering and Physical Sciences Research Council, grant EP/H015736/1 (Simplified Dual-Drainage Modelling for Flood Risk Assessment in Urban Areas). The authors would also like to thank the UK Environment Agency for the EA benchmarks datasets and the Torquay Council for the LIDAR datasets. Furthermore, the authors would like to thank Mike Gibson for the help given during the development of the OpenMP implementation of the CADDIES CA API and for the helpful comments about this document. Finally, the authors would like to acknowledge the support of NVIDIA Corporation with the donation of the Tesla K20c GPU used in this research and the support of Innovyze for the license of the InfoWorks ICM 3.0 software. Material deployed in the work can be accessed by direct communication with the corresponding author.

References

Austin, R., Chen, A.S., Savic, D.A., Djordjevic, S., 2013. Fast Simulation of Sewer Flow Using Cellular Automata. NOVATECH 2013.

Austin, R., Chen, A.S., Savić, D.A., Djordjević, S., 2014. Quick and accurate cellular automata sewer simulator. J. Hydroinformatics 16 (6), 1359. http://dx.doi.org/10.2166/hydro.2014.070.

Bates, P.D., Horritt, M.S., Fewtrell, T.J., 2010. A simple inertial formulation of the shallow water equations for efficient two-dimensional flood inundation modelling. J. Hydrol. 387, 33–45. http://dx.doi.org/10.1016/j.jhydrol.2010.03.027.

Bennett, N.D., Croke, B.F., Guariso, G., Guillaume, J.H., Hamilton, S.H., Jakeman, A.J., Marsili-Libelli, S., Newham, L.T., Norton, J.P., Perrin, C., et al., 2013. Characterising performance of environmental models. Environ. Model. Softw. 40, 1–20.

Bradbrook, K.F., Lane, S.N., Waller, S.G., Bates, P.D., 2004. Two dimensional diffusion wave modelling of flood inundation using a simplified channel representation. Int. J. River Basin Manag. 2, 211–223.

CADDIES, 2015. Cellular Automata Dual-drainage Simulation [WWW Document]. http://emps.exeter.ac.uk/engineering/research/cws/resources/caddies-framework/ (accessed 01.03.15.).

Chen, S., Doolen, G.D., 1998. Lattice Boltzmann method for fluid flows. Annu. Rev. Fluid Mech. 30 (1), 329–364.

Chen, A.S., Djordjević, S., Leandro, J., Savić, D.A., 2007. The Urban Inundation Model with Bidirectional Flow Interaction between 2D Overland Surface and 1D Sewer Networks. NOVATECH 2007, Lyon, France, pp. 465–472.

Chen, A.S., Evans, B., Djordjević, S., Savić, D.A., 2012. A coarse-grid approach to representing building blockage effects in 2D urban flood modelling. J. Hydrol. 426–427, 1–16. http://dx.doi.org/10.1016/j.jhydrol.2012.01.007.

Coulthard, T.J., Hicks, D.M., Van De Wiel, M.J., 2007. Cellular modelling of river catchments and reaches: advantages, limitations and prospects. Geomorphology 90, 192–207. http://dx.doi.org/10.1016/j.geomorph.2006.10.030.

Dagum, L., Menon, R., 1998. OpenMP: an industry standard API for shared-memory programming. Comput. Sci. Eng. IEEE 5, 46–55.

DHI Software, 2014. MIKE FLOOD [WWW Document]. http://www.mikebydhi.com/products/mike-flood (accessed 10.23.14.).

Dottori, F., Todini, E., 2010. A 2D flood inundation model based on cellular automata approach. In: Carrera, J. (Ed.), Proc. XVIII International Conference on Water Resources (Barcellona).

Dottori, F., Todini, E., 2011. Developments of a flood inundation model based on the cellular automata approach: testing different methods to improve model performance. Phys. Chem. Earth Parts ABC 36, 266–280.

GDAL, 2013. Geospatial Data Abstraction Library [WWW Document]. http://www.gdal.org/ (accessed 12.18.13.).

Ghimire, B., Chen, A.S., Guidolin, M., Keedwell, E.C., Djordjević, S., Savić, D.A., 2013. Formulation of a fast 2D urban pluvial flood model using a cellular automata approach. J. Hydroinformatics 15, 676. http://dx.doi.org/10.2166/hydro.2012.245.

Glenis, V., McGough, A.S., Kutija, V., Kilsby, C., Woodman, S., 2013. Flood modelling for cities using Cloud computing. J. Cloud Comput. 2, 1–14. http://dx.doi.org/10.1186/2192-113X-2-7.

Guidolin, M., Duncan, A., Ghimire, B., Gibson, M., Keedwell, E., Chen, A.S., Djordjevic, S., Savic, D., 2012. CADDIES: a new framework for rapid development of parallel cellular automata algorithms for flood simulation. In: Presented at the 10th International Conference on Hydroinformatics (HIC 2012). IWA (International Water Association), Hamburg, Germany.

Halcrow Group Ltd, 2014. ISIS FAST [WWW Document]. http://www.isisuser.com/isis/isisfast.asp (accessed 10.21.14.).

Hénonin, J., Ma, H., Yang, Z.-Y., Hartnack, J., Havnø, K., Gourbesville, P., Mark, O., 2013. Citywide multi-grid urban flood modelling: the July 2012 flood in Beijing. Urban Water J. 1–15. http://dx.doi.org/10.1080/1573062X.2013.851710.

Hunter, N.M., Horritt, M.S., Bates, P.D., Wilson, M.D., Werner, M.G.F., 2005. An adaptive time step solution for raster-based storage cell modelling of floodplain inundation. Adv. Water Resour. 28, 975–991. http://dx.doi.org/10.1016/j.advwatres.2005.03.007.

Hunter, N.M., Bates, P.D., Horritt, M.S., Wilson, M.D., 2007. Simple spatially-distributed models for predicting flood inundation: a review. Geomorphology 90, 208–225. http://dx.doi.org/10.1016/j.geomorph.2006.10.021.

Innovyze, 2012. InfoWorks ICM Help v3.0.

Itami, R.M., 1994. Simulating spatial dynamics: cellular automata theory. Landsc. Urban Plan. 30, 27–47.

Krupka, M., Pender, G., Wallis, S., Sayers, P.B., Mulet-Marti, J., 2007. A rapid flood inundation model. In: Proceedings of the Congress-International Association for Hydraulic Research, p. 28.

Lamb, R., Crossley, M., Waller, S., 2009. A fast two-dimensional floodplain inundation model. Proc. ICE – Water Manag. 162, 363–370. http://dx.doi.org/10.1680/wama.2009.162.6.363.

Leandro, J., Chen, A.S., Schumann, A., 2014. A 2D parallel diffusive wave model for floodplain inundation with variable time step (P-DWave). J. Hydrol. 517, 250–259. http://dx.doi.org/10.1016/j.jhydrol.2014.05.020.

Lhomme, J., Sayers, P.B., Gouldby, B.P., Samuels, P.G., Wills, M., Mulet-Marti, J., 2008. Recent Development and Application of a Rapid Flood Spreading Method, in FLOODrisk 2008. Keble College, Oxford, UK.

Munshi, A., et al., 2011. The OpenCL Specification Version 1.1 (Khronos OpenCL Work. Group).

Neal, J., Fewtrell, T., Trigg, M., 2009. Parallelisation of storage cell flood models using OpenMP. Environ. Model. Softw. 24, 872–877.

Neal, J.C., Fewtrell, T.J., Bates, P.D., Wright, N.G., 2010. A comparison of three parallelisation methods for 2D flood inundation models. Environ. Model. Softw. 25, 398–411. http://dx.doi.org/10.1016/j.envsoft.2009.11.007.

Néelz, S., Pender, G., 2013. Benchmarking the Latest Generation of 2D Hydraulic Modelling Packages. Environment Agency, Horison House, Deanery Road, Bristol, BS1 9AH.

Smith, L.S., Liang, Q., Quinn, Paul F., 2013. A flexible hydrodynamic modelling framework for GPUs and CPUs: application to the Carlisle 2005 floods, in: experience in Asia and Europe. In: Presented at the International Conference on Flood Resilience, Exeter, UK.

Thomas, R., Nicholas, A.P., 2002. Simulation of braided river flow using a new cellular routing scheme. Geomorphology 43, 179–195.

Vojinovic, Z., Seyoum, S., Salum, M.H., Price, R.K., Fikri, A.K., Abebe, Y., 2013. Modelling floods in urban areas and representation of buildings with a method based on adjusted conveyance and storage characteristics. J. Hydroinformatics 15 (4), 1150–1168.

Wolfram, S., 1984. Cellular automata as models of complexity. Nature 311, 419–424.

Yen, B.C., Tsai, C.W.S., 2001. On noninertia wave versus diffusion wave in flood routing. J. Hydrol. 244, 97–104. http://dx.doi.org/10.1016/S0022-1694(00)00422-4.

Yu, D., 2010. Parallelization of a two-dimensional flood inundation model based on domain decomposition. Environ. Model. Softw. 25, 935–945. http://dx.doi.org/10.1016/j.envsoft.2010.03.003.