Available online at www.sciencedirect.com
ScienceDirect
Simulation Modelling Practice and Theory 15 (2007) 315–327
SIMULATION MODELLING PRACTICE AND THEORY
www.elsevier.com/locate/simpat

Cellular automata algorithm for simulation of surface flows in large plains

Pablo R. Rinaldi [a], Diego D. Dalponte [a], Marcelo J. Vénere [b], Alejandro Clausse [b,*]

[a] *PLADEMA-CICPBA, Universidad Nacional del Centro, 7000 Tandil, BA, Argentina*
[b] *CNEA-CONICET, Facultad de Ciencias Exactas, Universidad Nacional del Centro, 7000 Tandil, BA, Argentina*

Received 8 April 2005; received in revised form 31 August 2006; accepted 24 November 2006
Available online 16 January 2007

Abstract

A hydrological model based in the cellular automata paradigm, capable to handle extended geographical domains discretized in a large number of cells is presented. The automata simulates the natural surface water flow by tracking local water stocks, book-keeping precipitation, infiltration, evapotranspiration, and intercellular flows. The flow resistance is represented with a relaxation local parameter, which takes different values for the watercourses and the terrain. Likewise, an infiltration function regulates the downward movement of water. The model was tested in the Santa Catalina Creek Basin (158 km^2, centre Buenos Aires Province in Argentina), showing excellent performance in reproducing the effect of various rainfall scenarios.

Keywords: Cellular automata; Surface flow; Hydrological models; Flooding

1. Introduction

Simulations of surface flow in landscape models play an important driving role in floodplain management. Although considerable efforts have been spent in creating adequate models for various landscapes, still there is not a universal code that can be easily adapted for a wide range of applications. Therefore, substantial efforts are required to calibrate the existing models to the specifics of each landscape and the goals of each study.

In general, hydrologic models are part of more complicated modeling structures, therefore requiring simple algorithms to run within the framework of the entire ecological scenario. Consequently, some details should be sacrificed in order to make the numerical calculation feasible. An important trade-off in hydrologic models is the coarser spatial and temporal resolution that should be employed (kilometers and days), in contrast to small scale flows (meters and seconds).

* Corresponding author. Tel./fax: +54 2293 442202.
E-mail address: clausse@exa.unicen.edu.ar (A. Clausse).

1569-190X/$ - see front matter © 2006 Published by Elsevier B.V.
doi:10.1016/j.simpat.2006.11.003

Nomenclature

$h(x, y)$	average vertical coordinate of spatial cell (x, y)
$w(x, y)$	water level in cell (x, y)
W_i^{old}	current water level of the i-cell
H	equilibrium surface water height
W	water volume of a partition divided by the unit-cell area
w_i^{drain}	drained water level of the i-cell
w_i^{new}	calculated water level of the i-cell
α_{terrain}	flow resistance relaxion parameter
α_{river}	flow resistance in the river cells
$I_n(x, y)$	infiltrated water level in cell (x, y) after n steps
$I_0(x, y)$	bias infiltration of cell (x, y)
β	soil saturation coefficient

Presently, it is still not known what process representation should be included in a floodplain inundation model to achieve given levels of predictive ability. Ultimately, the best model will be the simplest one that provides relevant information while reasonably fitting the available data.

Any software tool aimed to tackle the mentioned problems should be designed taking into account that ultimately it will be used together with a set of other applications. Accordingly, generic input and output data formats should be used in order to enable the flexible management of information. The object-oriented methodology provides efficient solutions to these problems, supporting the implementation of reusable models and flexible data structures. In the fifties Ulam and Von Neumann [2] conceived the idea of an ingenious mathematical tool called cellular automata. They realized that certain complex phenomena might be simulated as assemblies of finite cells, which interact according to a small number of simple rules based on heuristic considerations. The interaction rules, generally applied to the immediate neighbors, may or may not bear a resemblance to the actual physical laws governing the phenomena at hand. However, for fluids, it was found that the statistical averages tend to the solution of the partial differential equations known to govern the situation – typically, the Navier–Stokes equation [10]. More recently, numerical simulations of sand piles have shown that complex systems amenable to representation by cellular automata, exhibit certain regularities in their global behavior, thus rising hope that quantitative laws might be formulated to encompass disparate phenomena: earthquakes, stock markets, weather, biological systems and fluids.

With the spreading of the software agent paradigm [9] as a by-product of the object orientation programming [7] new tools to tackle collective phenomena became available to the simulation community. Complex processes can be simulated as an assembly of finite entities that interact according to simple rules based on heuristic considerations [3].

In this article, a novel perspective in the modeling of surface flows in large plains, based on virtual landscapes created using cellular automata, is presented. The model is able to provide the evolution of the flow variables by tracking local water stocks, book-keeping precipitation, infiltration, evapotranspiration, and intercellular flows. Thus, while retaining the basic philosophy underlying cellular automata, a realistic model of surface flows is obtained, offering a useful tool for the modeling of floodings in plains.

2. The AQUA automata

The precipitations are stored in terrain depressions, forming shallow ponds and marshes. On sloping terrain this storage is generally negligible, but the amount of water accumulated in the local depressions of flat areas might substantially exceed the other terms of the water balance equation. The water exceeding the storage capacity of the depressions moves on the surface very slowly as sheet flow. Infiltration and evaporation is generally high because of the long duration of the pounded water. Given the negligible slope of the terrain, an important consequence can be drawn up from this fact: human activities modify the equilibrium of the plains; thus, any model has to be able to incorporate those actions (roads, channels, mainly).

Any model that simulates the dominant processes in the surface water in plains should:

(1) Be spatially distributed, for the lineal fluxes are not well defined.
(2) Incorporate a good spatial description of the terrain depressions, which in turn requires accurate digital elevation models with fine grid sizes. Following the mentioned criteria, let us describe the physical terrain by means of a digital elevation model (DEM), consisting of a scalar field, $h(x, y)$, associated with a two-dimensional grid. The field $h(x, y)$ represents the average vertical coordinate of each spatial cell.

Following the CA paradigm, the surface state of each cell is determined by a scalar $w(x, y)$, representing in our case the water level in the cell (x, y). Now let us imagine the cells connected by floodgates that are opened and closed in turns, allowing the water to flow driven by the elevation differences between cells. Additionally, water mass sources and sinks can be associated to each cell, accounting for infiltration, precipitation, and inflows and outflows due to seepage or evapotranspiration.

2.1. *Water distribution rule*

The surface flow is simulated by applying a basic water distribution rule on isolated partitions of the spatial domain. Several partition variants were investigated, leading to different neighborhood geometries as shown in Fig. 1. Not every partition leads to good numerical results. Theoretical and numerical studies show that the 3×3 neighborhood shows the best performances in cellular-automata surface flow simulations [11], and therefore that is the basic partition adopted in the present study (see Fig. 2). By means of the basic water distribution rule, the water is distributed inside each basic partition according to the following procedure:

1. Order the cells index according to the terrain height (see Fig. 3), that is

$$h_1 \leqslant h_2 \leqslant \cdots \leqslant h_9 \tag{1}$$

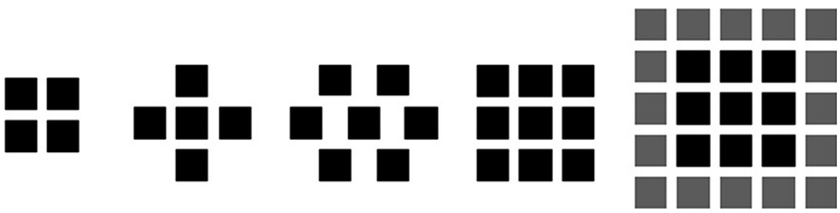


Fig. 1. Different neighborhood geometries for a basic partition.

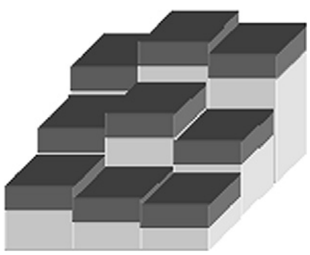


Fig. 2. Basic partition.

Fig. 3. Cells ordered according to their height.

2. Let w_i^{old} be the current water level of the *i*-cell.
3. Calculate the equilibrium surface water height, H, as the height the water would reach, provided that the total water volume contained in the partition drains down to the lowest possible locations (Fig. 4), that is

$$H = \frac{W + \sum_{i=1}^{k} h_i}{k} \tag{2}$$

where W is the water volume contained in the partition divided by the unit-cell area that can be calculated as

$$W = \sum_{i=1}^{9} (w_i^{\text{old}} - h_i) \tag{3}$$

and k is the number of cells that remains wet after the water drains down, which is the largest cell index satisfying:

$$W \geqslant \sum_{i=1}^{k} (h_k - h_i) \tag{4}$$

4. Calculate the drained water level, w_i^{drain}, that would have the cells if the water volume contained in the partition drains down to the lowest locations (Fig. 4), that is:

$$\begin{aligned} w_i^{\text{drain}} &= H - h_i \quad && \text{if } i \leqslant k \\ w_i^{\text{drain}} &= 0 && \text{if } i > k \end{aligned} \tag{5}$$

5. Calculate the new water level of the cells as a linear combination of w_i^{old}, and w_i^{drain}:

$$w_i^{\text{new}} = \alpha w_i^{\text{old}} + (1 - \alpha) w_i^{\text{drain}} \tag{6}$$

where α is a relaxation parameter ($0 < \alpha < 1$) which represents the flow resistance, and it is modeled as a cell attribute. In Eq. (6), the value of α corresponds to the center cell of the partition. In principle, a different α value can be assigned to every cell. Furthermore, for the purpose of modeling especial situations, the α-field can be treated as a function of the water level. The latter will be a powerful tool in simulating water streams.

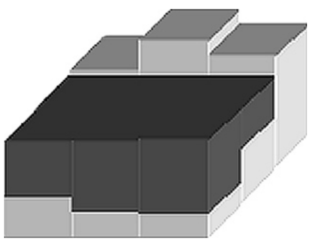


Fig. 4. Equilibrium state when the water is drained down in a basic partition.

2.2. Surface flow rule

Niet 100% mee met deze regel!

The surface flow is simulated by applying the water distribution rule acting sequentially on nine partitions of the spatial domain, in such a way that every cell occupies all the possible locations in the 3×3 basic partition (Fig. 5). The computational cost of ordering the cells in the basic partition is reduced by computing all the possible configurations once before the numerical calculation starts.

The time-step acts once the distribution rule is applied to the complete sequence of nine partitions. The set of intermediate distributions is not considered "observable", and should not be viewed as intermediate times, but only as auxiliary calculations (similarly to the sub-steps in the Runge–Kutta method [1]).

2.3. Water sources

Eigen opmerking: je zou hier een evaporatie term aan kunnen toevoegen!

Water additions to or subtractions from the cells, due to the precipitation and infiltration processes, are modeled as sources/sinks. Precipitations are simply added to each cell according to a pre-determined time schedule, which can represent either actual measurements or hypothetical scenarios.

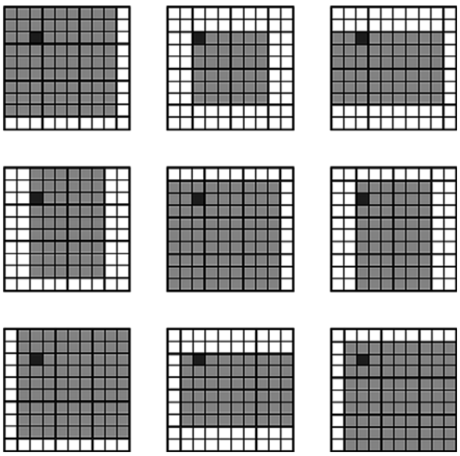
The infiltration process is more complicated than the precipitation, for its rate depends on the saturation state of the soil, which in turn changes when water infiltrates. In order to simulate this effect, it is necessary to keep track of the inventory, $I(x,y)$, of water infiltrated in each cell. The current infiltration volume at time-step n is calculated as

$$I_n(x,y) = \begin{cases} 0 & \text{if } w(x,y) = 0 \\ w(x,y) & \text{if } 0 < w(x,y) < \beta I_{n-1}(x,y) + I_0(x,y) \\ \beta I_{n-1}(x,y) + I_0(x,y) & \text{if } w(x,y) \geqslant \beta I_{n-1}(x,y) + I_0(x,y) \end{cases} \tag{7}$$

where $I_0(x,y)$ is a bias infiltration, and β is a coefficient representing soil saturation characteristics ($0 < \beta < 1$). Smaller β values mean that the soil would approach saturation faster. The initial infiltration, I_i, should be provided by the analyst, and it is likely to expect a dependence on the initial soil conditions.

2.4. Creeks and rivers modeling

Streams and rivers can be simulated in the AQUA environment by reducing the flow resistance along the corresponding water channels. Accordingly, the current value of α in every cell located along a river path is calculated as a function of the local water level, which accounts for the influence of the shape of the river bed on the flow rate.



the 9 partitions of the physical domain

Fig. 5. Sequence of partitions covering a step.

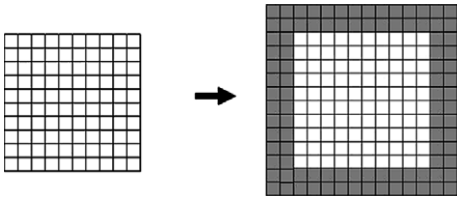


Fig. 6. Treatment of the boundary conditions.

Since there are a number of factors affecting this relation (bed profile, soil characteristics, aquatic vegetation, curvature, etc.) a comprehensive model would require the definition of a function for each cell. However, this is practically impossible when modeling large extensions of terrain. Alternatively, one can define regional families of α-functions, whose parameters can be determined by comparing the numerical calculation with experimental data.

A function family that shows good agreement with flow measurements in southern streams of the Argentine Humid Pampas is the following:

$$\alpha_{\text{river}} = 1 - \alpha_0 \left(\frac{w}{w_0} \right)^n \tag{8}$$

Varieert dus in de tijd!
w0 = waterhoogte op tijd 0. w = waterhoogte op tijd t!

where α_0, w_0 and n are effective constant parameters.

2.5. Boundary conditions

Open boundary conditions are applied to the external contour of the simulated region. This is implemented by adding an auxiliary border consisting of two lines of cells, with elevations substantially lower that the adjacent terrain (Fig. 6). The water volume contained in the added border is eliminated after each calculation step, in order to avoid long term accumulations. In order to achieve good global mass balances, it is very important that the simulation domain is much larger than the basin under study, leading to the automatic determination of the basin boundaries.

3. Object-Oriented implementation

The set of algorithms of the AQUA-II model was implemented in C++ using the object-oriented paradigm for the sake of flexibility and reusability. The application was provided with a graphic interface to facilitate the flow visualization.

3.1. Design and architecture

A model view controller (MVC) architecture was used in the application design [5], which keeps the logic and the internal representation of the model separated from the visualization and the calculation management. This architecture enables the coupling of the model to multiple views in order to provide different representations.

For the simulation model to be ready to evolve towards more complex stages, the implementation should be able to integrate further extensions and new visual features. In order to comply with this requirement, the calculation process was implemented following the design pattern strategy, which facilitates defining and encapsulating families of algorithms [5].

3.2. Visual interface

The visualization of the numerical results provides a useful tool for the analysis of surface flow events. In the present implementation, a global two-dimensional view of the water table was implemented, showing the water height with a non-linear scale of blues that provide more detail to the lower levels. In addition, three-dimensional visualizations were found practical from the point of view of the presentation, for they give a better contrast of the influence of the terrain on the flow direction. A 3D implementation was included in the application using OpenGL graphic language [12], which is a standard for most of the video cards and it is available in numerous platforms. The 3D interface provides interactive controls as rotations and zooms into the image. Good image resolutions without loosing fast representations are achieved using a single mesh, the color scale of the triangles varying from green to blue depending on the water level.

4. Results

4.1. Application of the model to real cases

The AQUA model was applied to a plain region located at the centre of Buenos Aires Province, Argentina, between latitude south 36°8′ and 37°22′, and between longitude west 58°49′ and 60°10′ (Fig. 7). Its extension is about 150 km. in Southwest–Northeast direction, and 40 km. wide, consisting of a large plain with a low hilly area at the southern boundary, where the Azul River headwaters occur. The hilly area is physiographically connected to the plains by pediments. Typical slopes are 5% in the south and less than 0.2% in the plain landscape [8].

Fig. 10 shows the digital elevation model (DEM) constructed for this region. Given the extreme flatness of the terrain, the surface flow is very sensitive to minor topographic variations, being important that the DEM appropriately capture those variations. Unfortunately, the conventional technique of interpolation between the cartographic contour lines is inadequate, for all the details are lost in the process. In the present case, an appropriate DEM was produced by means of radar interferometry constructed from satellite images corresponding to an ERS Tandem mission in 1997. Synthetic aperture radar (SAR) interferometry is based on the relation between the phase difference of two radar scenes covering the study area and the corresponding terrain elevation. Models with very good vertical resolutions can be obtained provided good image qualities and appropriate processing, including focusing and filtering, phase unwrapping, and geocoding. The processing details involved in the DEM construction are described in [4]. The cell size of the resulting model shown in Fig. 8 is 80 m. A number of small depressions can be observed in the lower region, which later will be responsible for water stagnations.

The region modeled in this study is the basin of the Santa Catalina Creek, which is a tributary of the Azul River. The length of the creek is 32 km, and drains 158 km^2 of the upper and middle sectors of the Azul River basin. Typical slopes in the upper basin are around 1–10% (rocky outcrops), whereas values of 0.1–1% characterize the middle sector. As insignificant as it may seem given the mean discharge, the Santa Catalina Creek has contributed to relevant floodings of the Azul Town, besides producing waterlogging of the rural areas it runs through.

The model was validated against major floods events that took place in May, August and October, 2002. The following hydrological data is available for each event:

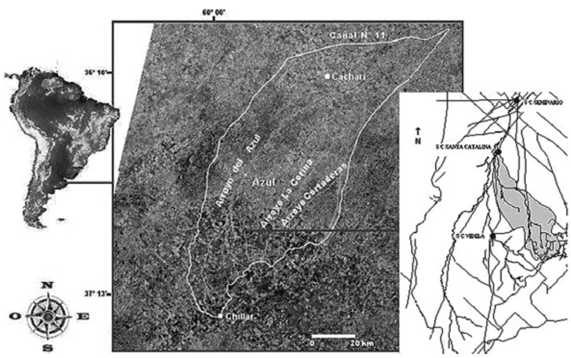


Fig. 7. Geographic location of the Santa Catalina Creek.

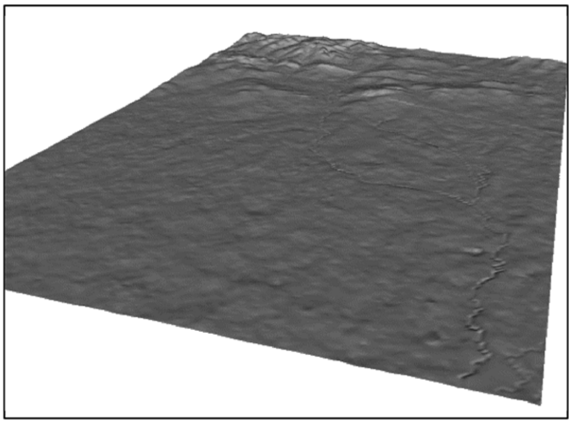


Fig. 8. Digital elevation model of the Santa Catalina sub-basin.

- Integrated precipitation from pluviometers distributed along the basin.
- Temporal evolution of the rains measured in two points located at the upper part and at lower part of the main river basin.
- Hydrograms of the Santa Catalina Creek, calculated from the HQ curve using water high data measured at the creek exit.
- Phreatic aquifer level previous to the rains.

Figs. 9–11 show the main features of each event and the comparison with the numerical calculations. The bars in each graphic indicate the precipitation in water millimeters per hour (right-axis scale). The solid points represent the water flow discharged by the creek to the main river. It can be seen that the response typically produce a downstream flow wave after the heavy rains.

The October event (Fig. 9) was a typical mild and short spring rain, occurring during the first 20 h of the event. The August case (Fig. 10) is a typical winter event, with continuous precipitations during the first 25 h of the event and a mild recurrence between 30 and 40 h.

The May event (Fig. 11) is very interesting, for there were actually two consecutive rains separated by 80 h. The first rain caused the saturation of the soil absorption capacity, which changes the infiltration parameters. It can be seen that the infiltration algorithm is able to handle the whole event reproducing both waves with excellent agreement. It is worth to note that this type of events are very difficult to simulate with models that use hyetographs based in curve numbers accounting for infiltrations and vegetation cover.

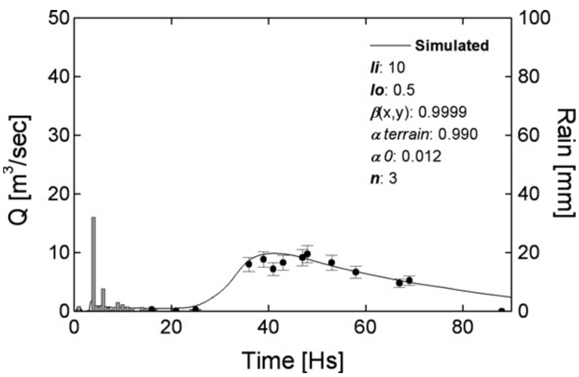


Fig. 9. Measured (dots) and simulated (curve) outlet flow rate of the Santa Catalina Creek during the event of October 2002. The bars indicate the measured precipitation rate.

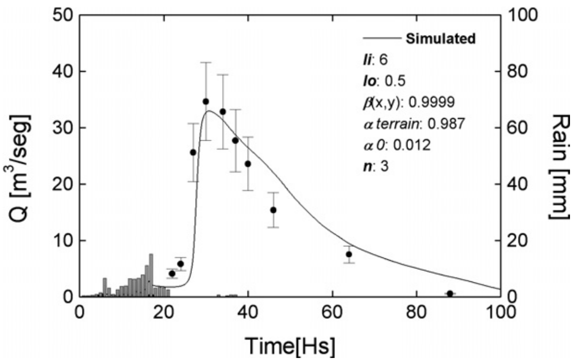


Fig. 10. Measured (dots) and simulated (curve) outlet flow rate of the Santa Catalina Creek during the event of August 2002. The bars indicate the measured precipitation rate.

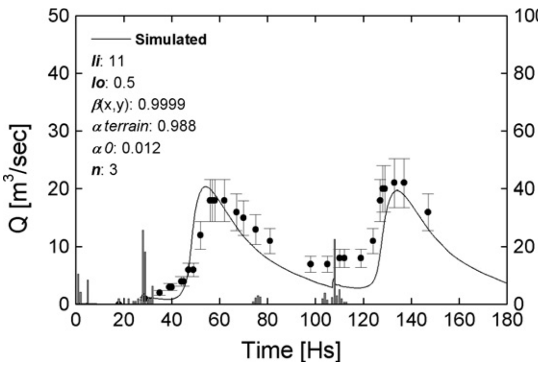


Fig. 11. Measured (dots) and simulated (curve) outlet flow rate of the Santa Catalina Creek during the event of May 2002. The bars indicate the measured precipitation rate.

In general, it can be seen that the simulation reproduce fairly well the measurements, using uniform sets of parameters over the whole creek basin (see Table 1). Season dependence was found in the initial infiltration and in the surface-flow resistance parameter, which is reasonable since both are influenced by the current soil conditions. Indeed, the initial infiltration capacity of the soil depends on the soil water content, and the flow resistance is strongly influenced by the vegetation, which varies substantially along the year.

4.2. Sensitivity analysis

In order to determine the robustness of the model, the sensitivity of the simulation results to variations of the relevant parameters was studied. The parameters analyzed were the river relaxation factor (α_0) and the infiltration saturation coefficient (β). Each parameter was varied about a reference value, keeping constant

Table 1
Parameters used in the simulations of the real cases

Parameter	May 2002	August 2002	October 2002
$\alpha_{terrain}$	0.988	0.987	0.990
α_0	0.012	0.012	0.012
n	3	3	3
I_0(mm/h)	0.5	0.5	0.5
I_i(mm/h)	11	6	10
β	0.9999	0.9999	0.9999

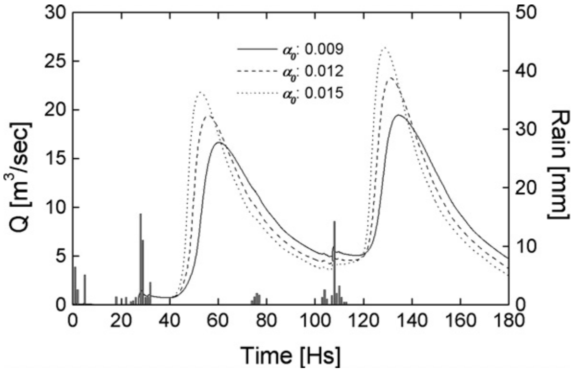


Fig. 12. Sensibility to the infiltration coefficient α_0.

all the other parameters. The May event was chosen as reference, since the two sequential flooding waves that were registered during that event are useful to observe the sensitivity at different time scales.

The coefficient α_0 is defined in Eq. (8) controls the flow resistance of the river. Fig. 12 shows the sensitivity of the simulation results to variations of α_0. It can be seen that larger values of α_0 produce higher flow rates in the river and narrower peaks. The latter is due to the fact that the river flow rate accelerates as the water level increases, and this effect is enhanced at larger α_0.

Fig. 13 shows the sensitivity of the simulation to variations of the saturation coefficient β. Larger values of β induce faster saturation of the soil, and consequently the impact of the infiltration is larger in magnitude and duration.

4.3. *Comparison against other surface flow models*

cell size wordt niet vermeld, ik vermoed 80m zoals in vorige implementatie

In order to test the coherence of the theory, the present model was also compared against HEC-1 (HEC-HMS), a well known surface water flow simulator. The test was performed in the synthetic scenario shown in Fig. 14, representing a simple plane with an adjacent channel. The size of the plane is 8800 by 11,000 m with a slope 0.0085. The water drains to a rectangular channel 5 m wide with a slope of 0.0025.

The parameters used in the simulation are shown in Table 2. The equivalent *Manning* coefficients used by HEC-1 in the plane and the channel are 0.02 and 0.01, respectively, corresponding to the minimum roughness. Figs. 15 and 16 show the channel outlet flow for two different events, both corresponding to a uniform rain over the plane. It can be seen that the present model matches pretty fairly the standard results.

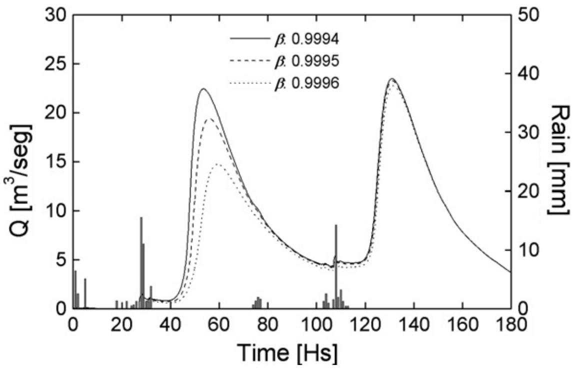


Fig. 13. Sensibility to the saturation coefficient β.

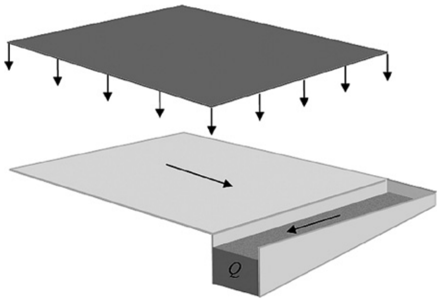


Fig. 14. Ideal surface flow case.

Table 2
Parameters used in the simulations of the synthetic case

$\alpha_{terrain}$	0.988
α_0	0.012
n	3
I_0(mm/h)	0
I_i(mm/h)	0
β	0.9999

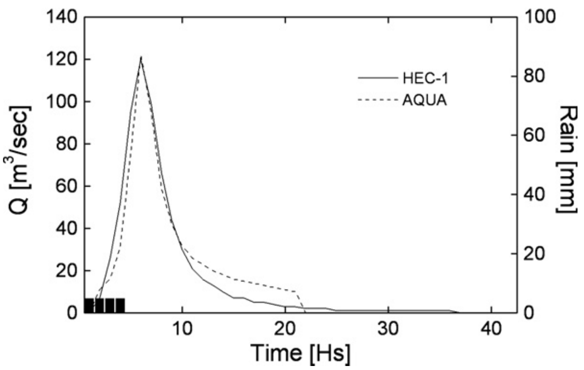


Fig. 15. Channel flow after a constant rate of 5 mm/h during 5 h in the plain (Fig. 14).

Meeste interessante
figuren om initieel na te bootsen!
eigen DEM = enkel wanneer tijd!

Bijvoorbeeld een idee: neem DEM van de
Zwalm (zie hydrologisch modelleren), bepaal
riviernetwerk en geef

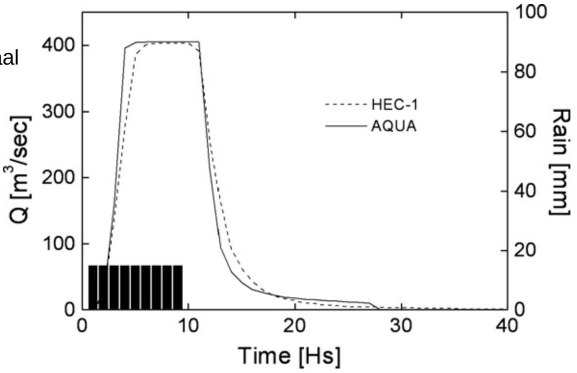


Fig. 16. Channel flow after a constant rate of 15 mm/h during 10 h in the plain (Fig. 14).

Unfortunately HEC-1 requires the use of different effective slopes in order to reproduce the real cases shown in Section 4.1. For this reason, in these cases the comparison of the present flow automata model against the standard model would not be consistent.

5. Conclusions

A cellular automata evolving on a regular grid representing the surface flow along large plains was presented. Simple local rules simulate the natural flow of water towards the lower locations, reproducing in the global picture the macroscopic behavior of the actual surface flow in real terrains. The soil resistance to the flow is characterized in each cell by one relaxation parameter determining how fast the water moves onto the land or in the watercourses. The infiltration properties are modeled by means of a function simulating the soil saturation process.

Three flooding events were simulated and compared with consistent and complete available data records. The model can describe fairly well flooding waves, including a double peak event where the first flood changed the soil water content for the second wave.

It is interesting to note that although the present model does not include a surface storage coefficient in the cells (as most models do), the measurements are reproduced fairly accurately. The reason for this performance is that the calculation is based on a digital topological representation defined over a very fine mesh, which capture the terrain depressions responsible of the surface water storage.

The model presented here should be seen as complementary to other modeling strategies, such as finite elements or finite differences methods. By means of this development, without invalidating the important advances in other directions, the present model can be very profitable regarding the computer modeling and simulation of surface flow processes in complex scenarios.

Glossary

DEM digital elevation model
Interferometry interference patterns obtained from two radar images
Phreatic aquifer level distance between the terrain surface and underground water
Hyetographs graphic representation of the time distribution of rain
Evapotranspiration summation of evaporation and plant transpiration
Manning coefficients roughness coefficient used in the Manning's formula for flow calculation in open channels

Acknowledgements

This project was supported by the Scientific Research Council of the Buenos Aires Province (CICPBA) and the National Atomic Energy Commission from Argentina (CNEA). The satellite images were provided by the National Space Agency of Argentina (CONAE). The authors especially thank the support of the Instituto de Hidrologia de Llanuras (IHLLA) who was responsible for the data showed in this article, and for their helpful discussions related to the physical aspects of the present model.

References

[1] J. Butcher, Numerical Methods for Ordinary Differential Equations, 2nd ed., John Wiley & Sons, 2003.
[2] N.G. Cooper, Los Alamos Science Special Issue, Los Alamos National Laboratory, New Mexico, 1987.
[3] A. Díaz, F. Trilnik, M. Campo, A. Clausse, flows simulations using object-oriented software agents, Inter J Heat Technol 21 (2003) 77–84.
[4] P. Euillades, M. Vénere, Corrección de modelos de elevación en base a un conjunto de puntos seguros, Revista internacional de Métodos numéricos para Cálculo y Diseño en ingeniería 19 (2003) 33–43.
[5] E. Gamma, R. Helm, R. Johnson, J. Vlissides, Design Patterns Elements of Reusable Object-oriented Software, Addison-Wesley, 1995.
[6] HEC-HMS. Hydrological Model Developed by The U.S. Army Corps of Engineers.

[7] B. Meyer, Object-oriented Software Construction, Second ed., Prentice-Hall, 1997.
[8] J.M. Sala, E. Kruse, R. Aguglino, Investigación hidrológica de la Cuenca delArroyo Azul, Provincia de Buenos Aires, CIC, Informe 37 (1987).
[9] Y. Shoham, Agent-oriented programming, Artif Intell 60 (1993) 51–92.
[10] S. Wolfram, Cellular automaton fluids, J Stat Phys 45 (1986) 3–4.
[11] J.G. Zhou, Lattice Boltzmann Methods for Shallow Water Flows, Springer, 2004.
[12] M. Woo, J. Neider, T. Davis, D. Shreiner, OpenGl Programing Guide, Addison-Wesley, 2000.