The quantization algorithm impact in hydrological applications: Preliminary results

Alessio De Rango[1][0000−0002−7045−2702], Luca Furnari[1][0000−0002−7360−0811],
Donato D'Ambrosio[2][0000−0003−1701−2203],
Alfonso Senatore[1][0000−0001−9716−3532], Salvatore Straface[1][0000−0003−1081−6049],
and Giuseppe Mendicino[1][0000−0003−0353−5206]

[1] DIAm, University of Calabria, Italy
{alessio.derango,luca.furnari,alfonso.senatore,
salvatore.straface,giuseppe.mendicino}@unical.it
[2] DeMACS, University of Calabria, Italy
donato.dambrosio@unical.it

Abstract. A computationally efficient surface to groundwater coupled hydrological model is being developed based on the Extended Cellular Automata (XCA) formalism. The three-dimensional unsaturated flow model was the first to be designed and implemented in OpenCAL. Here, the response of the model with respect to small variations of the quantization threshold has been assessed, which is the OpenCAL's quantization algorithm's parameter used for evaluating cell's steady state condition. An unsaturated flow test case was considered where the elapsed times of both the non quantized execution and the execution run by setting the quantization threshold to zero (with respect to the moist content variable) were already evaluated. The model response has been assessed in terms of both accuracy and computational performance in the case of an MPI/OpenMP hybrid execution. Results have pointed out that a very good tradeoff between accuracy and computational performance can be achieved, allowing for a considerable speed-up of the model against a very limited loss of precision.

Keywords: Extended Cellular Automata · Unsaturated flow modelling · Computational Efficiency

1 Introduction

Due to climate change, water management has become a key factor for sustainable development scenarios [18, 1]. The development of increasingly efficient hydrological models is an essential element in the study of the water cycle dynamics. Many models have been proposed to predict these phenomena (cfr. [27]). Most of them use a physical approach based on partial differential equations (PDEs) to describe the real phenomenon. Nevertheless, since their analytical solution is often unknown, an approximate solution is obtained by applying a numerical method such as Finite Differences, Finite Elements, or Finite Volumes

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50436-6_14

(cf. [6, 14]). In most cases, a time-explicit recurrence relation is obtained, which expresses the next state of the generic (discrete) element of the system as a function of the current states of a limited number of neighboring elements. For this reason, the most explicit schemes can be formalized in terms of Cellular Automata (CA) [25], which is one of the best-known and most widely used decentralized discrete parallel calculation models (cf. [4, 5, 20, 22, 16, 29, 8, 9, 26]). Nevertheless, a kind of global control over the simulation is often useful to make the modeling of certain processes more straightforward. Therefore, both decentralized transitions and steering operations can be adopted to model complex systems (cfr. [10]).

Different software systems were proposed to model complex systems formalized in terms of Cellular Automata. Among them, *Camelot* was developed as a (commercial) integrated development environment (IDE) based on the CARPET C-like MPI-based language [2]. Another example is *libAuToti* [30], which represented a C++ free application program interface (API) based on MPI. Unfortunately, both systems are no longer developed. However, the *OpenCAL* (Open Computing Abstraction Layer) open source software library [7] has been recently proposed as an alternative to the aforementioned software systems. In particular, it permits to exploit heterogeneous parallel computational devices on clusters of interconnected workstations. Both CPUs and GPUs, as well as other many-core accelerators, are supported. Despite its recent release, OpenCAL was already applied to the simulation of different physical phenomena, including a debris flow evolving on real topographic surface, as well as to the implementation of graphics convolutional filters, fractal generation algorithms. In addition, a particle system based on the Discrete Element Method was also implemented and preliminarily tested [13, 12].

In [11] we developed a preliminary unsaturated model, $XCA - Flow$, based on the discretization of Darcy's law, by adopting an explicit Finite Difference scheme to obtain a discrete formulation [17, 23]. We then implemented it by using OpenCAL, and therefore applied the model to simulate a three-dimensional test case. Specifically, giving rise to a topologically connected phenomenon, it was simulated for assessing the computational advantage that the OpenCAL's *quantization algorithm* proved to be able to provide in similar cases [7]. The *quantization threshold* used to characterize the stationary cells was set to zero, meaning that the reference hydraulic head difference had to overcome zero in order to *activate* the cell. The highest speed-up (with respect to the parallel non quantized simulation) of about 4.4 was achieved by considering a hybrid distributed/shared memory execution on a dual socket Intel Xeon-based workstation, using 2 MPI processes and 16 OpenMP threads, for a total of 32 computing threads. In particular, the fastest execution of the non quantized simulation required about 535 seconds using 32 threads on a single MPI process, while the quantized simulation lasted 122 seconds only. This good result suggested us to further investigate the OpenCAL's quantization algorithm in order to understand if a good tradeoff between accuracy and computational performance was possible in the case non-zero quantization thresholds were considered.

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50436-6_14

The paper is organized as follows. Section 2 briefly outlines the $XCA-Flow$ unsaturated flow model, while Section 3 briefly presents the OpenCAL architecture and its quantization algorithm. Section 4 illustrates the case study, together with the outcomes and the accuracy evaluation, while Section 5 assesses the simulations outcomes and presents the computational performances achieved. Finally, Section 6 concludes the paper with a general discussion envisaging possible future developments.

2 The $\boldsymbol{XCA-Flow}$ Flow Model

The $XCA-Flow$ was developed for addressing several typical hydrological problems, from parcel/hillslope to large basin scales, benefiting from the computational advantages allowed by the XCA-based approach. The direct discrete formulation of the Richards' equation where the $XCA-Flow$ model relies on is thoroughly described by Mendicino et al. [23], who already used the XCA formalism for developing an unsaturated flow model using the CAMELot environment. Here, for the sake of clarity and completeness, the main equations of the model are summarized.

The Richards' equation is a non linear degenerate elliptic-parabolic partial differential equation [19] describing double-phases flow in porous media, it is given by combining the mass conservation equation with the momentum conservation equation represented by the Darcys' law. Considering the pressure head ψ as the dependent variable, the Richards' equation for an isotropic porous medium is written as [3]:

$$C_c(\psi)\frac{\partial\psi}{\partial t} - \nabla\big[K(\psi)\nabla\psi\big] - \frac{\partial K(\psi)}{\partial z} = 0 \tag{1}$$

where the pressure head ψ [m] is related to the hydraulic head h [m] by the equation $\psi = h - z$, being z [m] the elevation, $C_c(\psi)$ is the specific ritention capacity [m^{-1}], given by the relation $C_c(\psi) = d\theta/dh$, where in turns θ is the moisture content [m^3 m^{-3}] and $K(\psi)$ is the hydraulic conductivity [m s^{-1}].

The $XCA-Flow$ model solving Equation 1 is formally defined as:

$$XCA-Flow = \big\langle (D, S, I), X, Q, (\Sigma, \Phi), \Gamma, (T, t), \omega, \tau_{\mathbb{N}} \big\rangle$$

where:

- $D = [0, n_r - 1] \times [0, n_c - 1] \times [0, n_s - 1] \subset \mathbb{Z}^3$ is the three-dimensional discrete computational domain, with n_r, n_c, and n_s the number of rows, columns, and slices, respectively.
- $S = [0, n_r \cdot \Delta s] \times [0, n_c \cdot \Delta s] \times [0, n_s \cdot \Delta s] \subset \mathbb{R}^3$ is the continuum three-dimensional realm corresponding to D, subdivided in cubic cells of side Δs. The function μ defines the mapping to D as:

$$\mu : D \to \mathbb{R}^3$$

$$(\iota_1, \iota_2, \iota_3) \mapsto \big(\iota_1 \cdot \Delta s, \iota_2 \cdot \Delta s, (n_s - 1 - \iota_3) \cdot \Delta s\big)$$

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50436-6_14

- $I = \{I_\rho, I_\beta, I_\tau \subseteq D\}$ is the set of domain interfaces, where
    - $I_\rho \subseteq \pi_{s=0}$ is the boundary region, belonging to the top surface $\pi_{s=0}$, that is affected by rain;
    - I_β the remaining domain boundary region over which no input rain is considered;
    - $I_\tau = D \setminus I_\rho \cup I_\beta$ the set of cells belonging to the inner domain, where the system evolution occurs.
- $X = \{(0,0,0), (-1,0,0), (0,-1,0), (0,1,0), (1,0,0), (0,0,-1), (0,0,1)\}$ is the von Neumann neighborhood.
- $Q = Q_\theta \times Q_K \times Q_h \times Q_{C_c} \times Q_{conv}$ is the set of states for the cell, where:
    - Q_θ is the set of values representing the soil moisture content;
    - Q_K is the set of values representing the unsaturated hydraulic conductivity;
    - Q_h is the set of values representing the hydraulic total head;
    - Q_{C_c} is the set of values representing the specific retention capacity;
    - Q_{conv} is the set of values representing the temporal step size which guarantees the numerical convergence;
- $\Sigma = \{\sigma_\rho, \sigma_\beta, \sigma_\tau\}$ is the set of local transition functions or kernels. In particular:
    - $\sigma_\rho : Q_h \times Q_{C_c} \to Q_h$ accounts for input rain. Specifically, if h, C_c, r_{ir}, Δt, and Δs^2 denote hydraulic head, specific retention capacity, rain intensity rate, time interval corresponding to a transition step, and surface of the cell side, respectively, σ_ρ updates the hydraulic head within the cell as:

    $$h' = h + \frac{r_{ir} \cdot \Delta t}{\Delta s^2 \cdot C_c}$$

    - $\sigma_\beta : Q_h \to Q_h$ sets the boundary condition on the boundary cells that are not affected by rain. The Neumann boundary conditions, which fix the water flow to a constant value, for instance $h' = h$ represent a no flow condition; the Dirichlet boundary conditions, which fix the hydraulic head to a constant value, can be adopted.
    - $\sigma_\tau : Q^{|X|} \to Q$ corresponds to the discrete time explicit resolution of the equation 1 and can be written as:

    $$h'_c = h_c + \frac{\Delta t \left[\sum_{\alpha=1}^{6} \overline{K}_\alpha (h_\alpha - h_c) \right]}{\Delta s^2 C_c}$$

    where $\overline{K}_\alpha$ is the average hydraulic conductivity between the current cell c and the cell in the α neighbor calculated using:

    $$\overline{K}_\alpha = \frac{2 \Delta s^3}{\frac{\Delta s^3}{K_\alpha} + \frac{\Delta s^3}{K_c}}$$

    also the substates q_θ, q_K and q_{c_c} are updated according to the constitutive equations between ψ, θ and K proposed by [31], [24].

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50436-6_14

Finally the substate q_{conv} is updated adopting the Courant-Friedrichs-Lewy condition to achieve numerical convergence:

$$\Delta t \leq \frac{\Delta s^2 C_c}{\sum_{\alpha=1}^{6} \overline{K}_\alpha}$$

Optionally, the σ_τ kernel can add/remove cells to/from the set of active cells A in case the *quantization* algorithm is exploited in the OpenCAL implementation of $XCA-Flow$. In this case, a gradient threshold must be set on one or more cell substates that, if exceeded/not-exceeded, produces the cell activation/deactivation.

- $\Phi = \{\phi_\rho, \phi_\beta, \phi_\tau\}$ is the set of functions applying the local transition of Σ to the non-local domains defined by I. In particular:
  - $\phi_\rho : Q^{|I_\rho|} \to Q^{|I_\rho|}$ applies the σ_ρ local transition to the I_ρ interface, to account for input rain;
  - $\phi_\beta : Q^{|I_\beta|} \to Q^{|I_\beta|}$ applies the σ_β local transition to the I_β interface, to account for boundary conditions;
  - $\phi_\tau : Q^{|I_\tau^+|} \to Q^{|I_\tau|}$ applies the σ_τ local transition to the I_τ interface. Here, I_τ^+ denotes the union of I_τ and the set of cells belonging to $D \setminus I_\tau$ that are needed to guarantee a complete neighborhood to each boundary cell of I_τ. According to the Cellular Automata definition, only states of cells in I_τ are updated.
- $\Gamma = \{\gamma_t \mid \gamma_t : Q_{conv}^{|D|} \to \mathbb{R}\}$ is the set of global functions, where γ_t evaluates a reduction over the Q_{conv} substate in order to evaluate the physical time corresponding to a state transition of the automaton. Specifically, if Δt denotes the time step size, we have:

$$\Delta t = \min_{\iota \in D} q_{conv_\iota}$$

- $T = \phi_\rho \circ \phi_\beta \circ \phi_\tau \circ \gamma_t$ is the function determining the automaton global transition. It is obtained by preliminary applying the elementary processes to the related interfaces, and then the reduction function needed to evaluate the time interval corresponding to a computational step.
- $\Delta t \in \mathbb{R}^+$ is the quantity corresponding to the physical time interval simulated by a state transition of the automaton. It is evaluated step by step by considering the γ_t reduction function.
- $\omega : \mathbb{R} \to \{false, true\}$ is the termination criterion, based on the simulation elapsed time. When the prefixed simulated time interval is complete, ω returns $false$ and the simulation terminates.
- $\tau_\mathbb{N} : \mathbb{N} \times C \to C$ is the XCA *control unit*. At step $n = 0$, the XCA is in the initial configuration C_0. $\tau_\mathbb{N}$ is then applied at discrete steps, by producing a sequence of configurations $C_1, C_2, \cdots$, until the ω termination criterion is satisfied.

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50436-6_14

3 A Brief Overview of the OpenCAL Library and its Quantization Algorithm

OpenCAL (Open Computation Abstraction Layer) is an open source parallel C/C++ software library based on the Extended Cellular Automata (XCA) computational paradigm [15], which exposes as a Domain Specific Language (DSL). As a consequence, being the XCA formalism quite general, OpenCAL also provides support to other computational models like Cellular Automata, Finite Difference and, more in general, to structured grid-based methods.

Once the simulation model has been properly defined using the XCA formalism, OpenCAL allows for its straightforward implementation. As a matter of fact, the spatial computational domain, the variables representing the cell's state (substates), and the neighborhood can be easily defined, as well as the local state transition function and possible global operators. Moreover, the initial conditions of the system, and a termination criterion to the simulation can be defined with the minimum effort. Eventually, OpenCAL provides embedded optimization algorithms and allows for a fine grained control over the simulation [7].

One of the most important advantages of using OpenCAL is that, once the serial implementation of a model has been completed, different parallel versions can be easily obtained, including those for multi- and many-core shared memory devices, as well as for distributed memory systems, thanks to the adoption of parallel underlying APIs like OpenMP, OpenCL, and MPI. Furthermore, the embedded optimizations adopted in the serial version are transparently translated in the different parallel execution contexts.

Among the above cited optimizations, the quantization algorithm can provide a significant speed-up in the simulation of topologically connected phenomena. Chosen a model's substate and a stationary condition threshold for the cell, the algorithm allows to skip stationary cells, meaning that not only the transition function is not computed, but also that stationary cells are completely skipped, i.e., they are not visited by the loops spanning the domain. To this end, the set of *active* cells, A, must be preliminary defined, usually at the initialization stage. Only cells belonging to A are processed. The set A must therefore be kept updated during the simulation by referring to specific add/remove API functions. Similarly to the case of the substates, even the active cell's data structure needs to be updated before applying the next transition. Further details regarding the quantization algorithm can be found in [7].

4 Description of the Test Case

The case of study investigated in this work is a three-dimensional test based on a real experiment conducted in the Jornada Test Site near Las Cruces, New Mexico, by the University of Arizona and reproduced using two-dimensional numerical modeling by [28] and [21]. It concerned a heterogeneous terrain with very dry initial conditions, having three main horizontal soil layers and a fourth soil

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50436-6_14

type inserted in the deepest of them (Figure 1). The main hydraulic properties of the four soil types are reported in Table 1. The original two-dimensional test case domain was 8.00 m long and 6.50 m deep, with a uniform rain rate r_{ir} of 0.02 m d^{-1} for a length of 2.25 m along the left side of the upper boundary layer (Figure 1). Along the other boundaries, Neumann's conditions (no flow) were imposed.

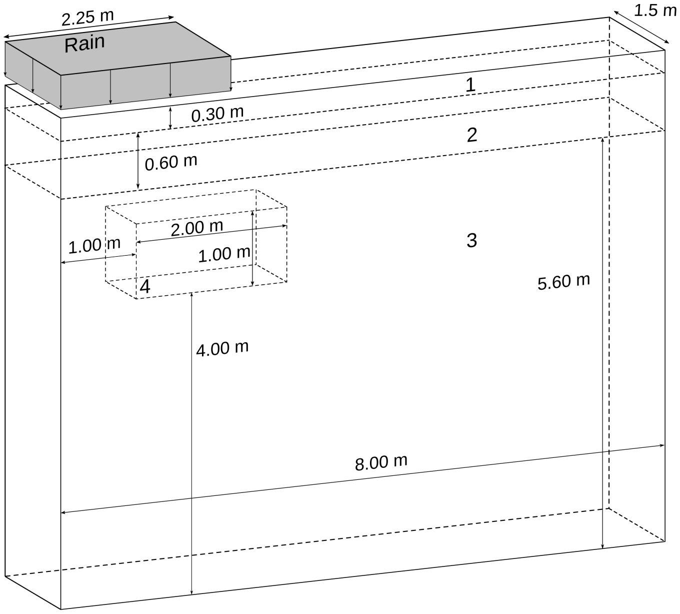

Fig. 1. Cross-section of the computational domain. It shows the four different zones, the first is located on the top of the domain and extends vertically for 0.30 m, the second extends for 0.60 m, the third for the remaining 5.60 m. The fourth zone is inserted and surrounded by the third zone. The grey zone on the top of the domain highlights the surface area directly affected by the rain.

As in [11], the original domain was extended along the third dimension in order to obtain a 1.50 m wide three-dimensional grid of cubic cells with edge length $\Delta s = 0.05$ m. Accordingly, a grid of $n_r = 160$ rows, $n_c = 30$ columns, and $n_s = 130$ slices was obtained. The input rain was accordingly extended to the third dimension (Figure 1). The system initial conditions were defined by imposing a constant total head value of -7.3 m all over the domain, coinciding with the test case already considered in [11]. This configuration permitted to exploit the OpenCAL's quantization algorithm i) by initializing the set of non-stationary cells, A, to the domain interface affected by rain, and ii) by activating a stationary cell c each time the $\Delta h_c > \tau$ condition was satisfied, where Δh_c and τ are the hydraulic head difference between the central (already active) cell and

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50436-6_14

Zone	θ_r	θ_s	α (m^{-1})	n	K_s (m d^{-1})
1	0.1020	0.368	3.34	1.982	7.909
2	0.0985	0.351	3.63	1.632	4.699
3	0.0859	0.325	3.45	1.573	4.150
4	0.0859	0.325	3.45	1.573	41.50

Table 1. Properties of the four different soil types. Each row indicates a different zone from the first to the fourth, each column represents a soil property. θ_r is the residual soil moisture content, θ_s is the satutared soil moisture content, α [m^{-1}] and n are soil parameters adopted in the Van Genuchten model, K_s [m d^{-1}] is the saturated hydraulic conductivity.

the neighboring cell c, and τ the fixed activation threshold, respectively. Since h-form of the Richards' equation [3] was considered to solve the unsaturated flow, the hydraulic head has been used to define the activation condition. Note that, each time a new cell is activated, the model activates its neighboring cells as well. This is needed to guarantee mass conservation. In fact, if a cell c losses some mass due to the hydraulic head condition with respect to a neighboring cell c_n, this latter must be active at the same step to receive the mass from c. In particular, the problem could occur at the phenomenon's propagation front.

As in [11], a total of 30 days were simulated for each experiment, which required a total of 25,986 computational steps, with an average time step $\Delta t =$ 99.75 s. The outcome is shown in Figure 2. The phenomenon propagates up to about 3 m depth, not affecting the deepest layers.

5 Experiments, Accuracy and Computational Performance

Different simulations of the test case described in Section 4 were performed by varying the hydraulic head quantization threshold, with the purpose to assess both the accuracy and the computational performance with respect to the non quantized (reference) simulation. The values adopted for the quantization threshold are listed in Table 2. Note that, the set of experiments here considered extends the study described in [11], where the only $\tau = 0$ threshold was taken into account.

Figure 3 shows the differences of the variable ψ between the reference and the $\tau = 0.1, 0.25, 0.5, 1$ quantized simulations. The error, which is limited to about -3.3 m, increases (in absolute value) with the quantization threshold. In particular, the differences are close to zero in most parts of the domain, with the major errors localized near the phenomenon propagation front (cf. Figures 2 and 3).

Regarding the computational aspects, the OpenMP/MPI OpenCAL component was used to implement the XCA-Flow model, and a workstation running Arch Linux, equipped whit two Intel 8 core (16 threads) Xeon E5-2650 sockets, was used to run the simulations. In particular, the experiments were performed

DOI: 10.1007/978-3-030-50436-6_14

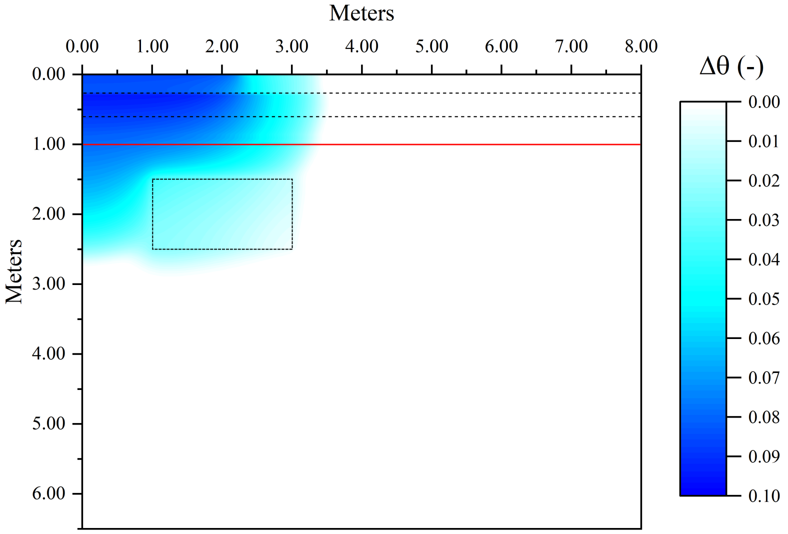

Fig. 2. The soil moisture content variation $\Delta\theta$ [-] on a vertical section at the final step of the 30-days reference simulation. Black dotted lines indicate the different zones, the red line shows the domain decomposition between the two nodes considered.

by referring to a 2 MPI processes/16 OpenMP threads parallel configuration. Specifically, the 3D domain was decomposed along the third (vertical) dimension over the 2 MPI processes, resulting in two sub-domains of 20 and 110 slices, respectively (cf. Figure 2). Note that, the adopted parallel set up is the same of [11], which permitted to achieve the best performance on the considered hardware by executing the reference simulation in 729.15 s. The results achieved are summarized in Table 2 in terms of elapsed times. Figure 4 shows both the mean square error of the ψ variable with respect to the reference simulation and the corresponding elapsed times of the simulations executed by considering the different quantization thresholds listed in Table 2. The error is close to zero up to $\tau = 0.1$, afterward it increases up to the value of about $2.6 \cdot 10^{-3}$ m^2. Differently, the elapsed times decrease with the quantization threshold, from a value of about 300 s to a value of less than 100 s.

DOI: 10.1007/978-3-030-50436-6_14

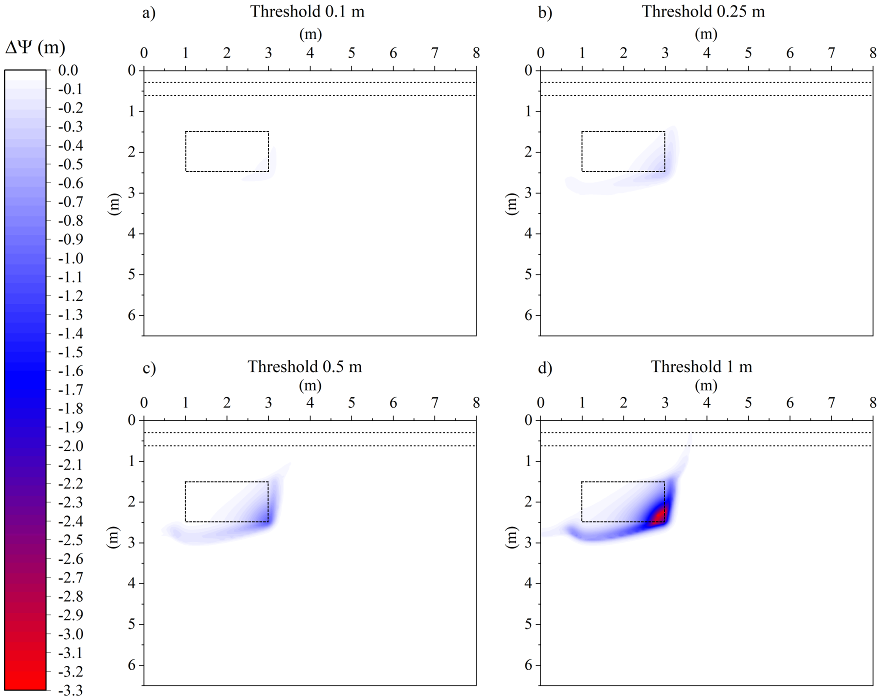

Fig. 3. Vertical sections showing the differences of the $\Delta\psi$ variable [m] between the final configurations of the reference simulation and the simulations performed by considering the τ a) 0.1 m, b) 0.25 m, c) 0.5 m, and d) 1 m quantization thresholds.

Threshold (m)	Mean square error (m^2)	Elapsed time (s)
0	0	291.77
0.001	1.74e-10	183.47
0.01	2.92e-08	155.69
0.1	6.23e-06	137.23
0.25	6.46e-05	122.89
0.5	4.36e-04	109.47
1.0	2.59e-03	99.15

Table 2. Threshold values τ [m] adopted in this work with the corresponding mean square error [m^2], based on the ψ variable calculated only on the active cells, and elapsed time [s].

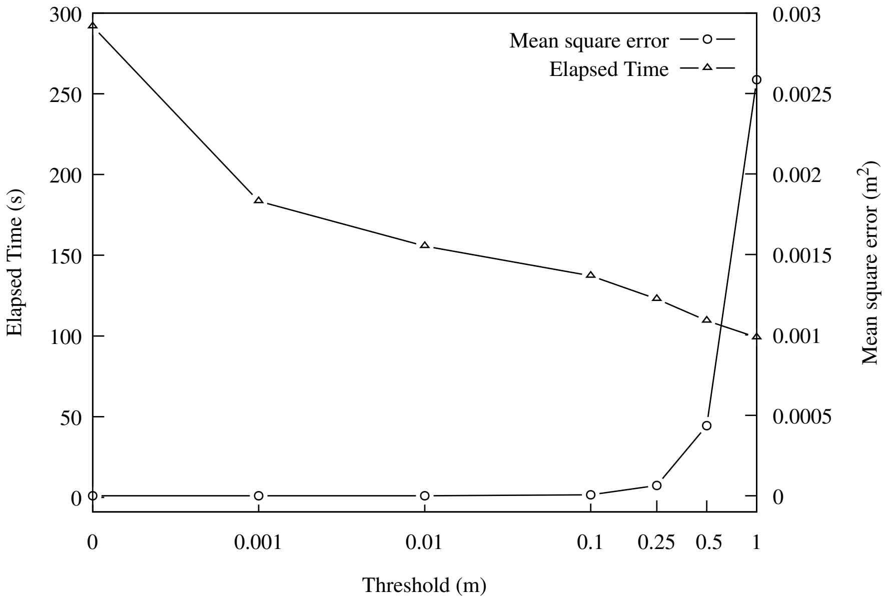

Fig. 4. Mean square error [m^2], based on the ψ variable calculated only on the active cells, and elapsed time [s] achieved using different quantization threshold. A further simulation with no threshold has been executed with an elapsed time of 729.15 s.

6 Conclusions

In this paper, the implementation of an unsaturated water flow model is presented based on the OpenCAL library, which allows exploiting the XCA formalism. Specifically, the impact of the quantization algorithm is presented, which allows reducing the computational effort at the expense of the accuracy level.

A three-dimensional test case was considered to assess computational performance and accuracy, which simulates the infiltration in the first soil layers produced by a uniform rain rate. Small variations of the quantization threshold τ were assessed in order to activate the non-stationary cells. In particular, τ refers to the hydraulic head variation between adjacent cells.

From an accuracy assessment point of view, it was observed that the quantization algorithm produces a delay in the propagation front since the non-active cells, which are not taken into account during the computation, become active when the quantization threshold is reached. As a consequence, there is a time cumulative error near the propagation front, which is tightly related to the chosen quantization threshold. In particular, most of the thresholds used for this specific case study, on a 30 days simulation, generate a low relative error. The greatest error of about 3.3 m has been obtained by adopting a threshold $\tau = 1$ m, which could not be tolerable for some applications. Although, the other thresh-

DOI: 10.1007/978-3-030-50436-6_14

olds $\tau \in [0.001, 0.5]$ m generate smaller errors. The acceptability of the results achieved, also in these cases, relies on the specific application. Therefore, the choice of the quantization threshold turns out to be a key factor and depends mainly on the model application.

Regarding the computational performance, the parallel implementation of the $XCA - Flow$ model proposed in [11] has been adopted, by varying the quantization threshold to further speeding-up the execution. The best results have been achieved using the threshold $\tau = 1$ m, which permitted a 66 percent execution boost compared to the $\tau = 0$ parallel simulation. The results pointed out that a good trade off between accuracy and computational performance can be achieved, allowing a considerable speed-up of the model against a limited loss of precision.

The future outlook will regard the application of dynamic quantization thresholds, which will permit to reduce the error during the simulation. Moreover, further hydrological variables will be considered to define the quantization thresholds in order to assess if there is any further advantage in terms of computational performance and accuracy. Finally, $XCA - Flow$ will be applied to a real basin scale case study. In this context, the quantization algorithm will be a crucial factor to reduce the expected high computing time.

Acknowledgements

Luca Furnari acknowledges for the program “POR Calabria FSE/FESR 2014/2020 - Mobilità internazionale di Dottorandi e Assegni di ricerca/Ricercatori di Tipo A” Actions 10.5.6 and 10.5.12.

References

1. Abd-Elaty, I., Sallam, G.A., Straface, S., Scozzari, A.: Effects of climate change on the design of subsurface drainage systems in coastal aquifers in arid/semi-arid regions: Case study of the nile delta. Science of The Total Environment **672**, 283 – 295 (2019)
2. Cannataro, M., Di Gregorio, S., Rongo, R., Spataro, W., Spezzano, G., Talia, D.: A parallel cellular automata environment on multicomputers for computational science. Parallel Computing **21**(5), 803–823 (1995)
3. Celia, M.A., Bouloutas, E.T., Zarba, R.L.: A general mass-conservative numerical solution for the unsaturated flow equation. Water Resources Research **26**(7), 1483–1496 (1990)
4. Cervarolo, G., Mendicino, G., Senatore, A.: A coupled ecohydrological-three-dimensional unsaturated flow model describing energy, H2O and CO2 fluxes. Ecohydrology **3**(2), 205–225 (2010)
5. Cervarolo, G., Mendicino, G., Senatore, A.: Coupled Vegetation and Soil Moisture Dynamics Modeling in Heterogeneous and Sloping Terrains. Vadose Zone Journal **10**, 206–225 (2011)
6. Chang, K.S., Song, C.J.: Interactive vortex shedding from a pair of circular cylinders in a transverse arrangement. International Journal for Numerical Methods in Fluids **11**(3), 317–329 (1990)

7. D'Ambrosio, D., De Rango, A., Oliverio, M., Spataro, D., Spataro, W., Rongo, R., Mendicino, G., Senatore, A.: The Open Computing Abstraction Layer for Parallel Complex Systems Modeling on Many-Core Systems. Journal of Parallel and Distributed Computing **121**, 53–70 (2018)
8. D'Ambrosio, D., De Rango, A., Rongo, R.: Opencal simulation of the 1992 tessina landslide. In: 30th European Modeling and Simulation Symposium, EMSS 2018. pp. 210–217 (2018)
9. D'Ambrosio, D., De Rango, A., Spataro, D., Rongo, R., Spataro, W.: Applications of the opencal scientific library in the context of cfd: Applications to debris flows. In: 2017 IEEE 14th International Conference on Networking, Sensing and Control (ICNSC). pp. 720–725 (2017)
10. Dattilo, G., Spezzano, G.: Simulation of a cellular landslide model with CAMELOT on high performance computers. Parallel Computing **29**(10), 1403–1418 (2003)
11. De Rango, A., Furnari, L., Giordano, A., Senatore, A., D'Ambrosio, D., Spataro, W., Straface, S., Mendicino, G.: OpenCAL System Extension and Application to the Three-Dimensional Richards Equation for Unsaturated Flow. Computers & Mathematics with Applications (under revision)
12. De Rango, A., Napoli, P., D'Ambrosio, D., Spataro, W., Di Renzo, A., Di Maio, F.: Structured Grid-Based Parallel Simulation of a Simple DEM Model on Heterogeneous Systems. pp. 588–595 (2018)
13. De Rango, A., Spataro, D., Spataro, W., D'Ambrosio, D.: A first multi-gpu/multi-node implementation of the open computing abstraction layer. Journal of Computational Science **32**, 115 – 124 (2019)
14. Deng, X., Min, Y., Mao, M., Liu, H., Tu, G., Zhang, H.: Further studies on geometric conservation law and applications to high-order finite difference schemes with stationary grids. Journal of Computational Physics **239**, 90–111 (2013)
15. Di Gregorio, S., Serra, R.: An empirical method for modelling and simulating some complex macroscopic phenomena by cellular automata. Future Generation Computer Systems **16**, 259–271 (1999)
16. Filippone, G., D'Ambrosio, D., Marocco, D., Spataro, W.: Morphological coevolution for fluid dynamical-related risk mitigation. ACM Transactions on Modeling and Computer Simulation (TOMACS) **26**(3), 18 (2016)
17. Folino, G., Mendicino, G., Senatore, A., Spezzano, G., Straface, S.: A model based on cellular automata for the parallel simulation of 3D unsaturated flow. Parallel Computing **32**(5), 357 – 376 (2006)
18. Kundzewicz, Z., Mata, L., Arnell, N., Döll, P., Jimenez, B., Miller, K., Oki, T., Z.Şen, Shiklomanov, I.: The implications of projected climate change for freshwater resources and their management. Hydrological Sciences Journal **53**(1), 3–10 (2008)
19. List, F., Radu, F.A.: A study on iterative methods for solving richards' equation. Computational Geosciences **20**(2), 341–353 (Apr 2016)
20. Lucà, F., D'Ambrosio, D., Robustelli, G., Rongo, R., Spataro, W.: Integrating geomorphology, statistic and numerical simulations for landslide invasion hazard scenarios mapping: An example in the Sorrento Peninsula (Italy). Computers and Geosciences **67**(1811), 163–172 (2014)
21. McCord, J.T., Goodrich, M.T.: Benchmark testing and independent verification of the VS2DT computer code. Tech. rep., Sandia National Labs. (1994)
22. Mendicino, G., Pedace, J., Senatore, A.: Stability of an overland flow scheme in the framework of a fully coupled eco-hydrological model based on the Macroscopic Cellular Automata approach. Communications in Nonlinear Science and Numerical Simulation **21**(1-3), 128–146 (2015)

23. Mendicino, G., Senatore, A., Spezzano, G., Straface, S.: Three-dimensional unsaturated flow modeling using cellular automata. Water Resources Research **42**(11), 2332–2335 (2006)
24. Mualem, Y.: A new model for predicting the hydraulic conductivity of unsaturated porous media. Water Resources Research **12**(3), 513–522 (1976)
25. von Neumann, J.: Theory of Self-Reproducing Automata. University of Illinois Press, Champaign, IL, USA (1966), edited by Arthur W. Burks
26. Senatore, A., D'Ambrosio, D., De Rango, A., Rongo, R., Spataro, W., Straface, S., Mendicino, G.: Accelerating a three-dimensional eco-hydrological cellular automaton on GPGPU with OpenCL. In: AIP Conference Proceedings. vol. 1776, p. 080003 (2016)
27. Senatore, A., Mendicino, G., Smiatek, G., Kunstmann, H.: Regional climate change projections and hydrological impact analysis for a Mediterranean basin in Southern Italy. Journal of Hydrology **399**(1), 70 – 92 (2011)
28. Smyth, J., Yabusaki, S., Gee, G.: Infiltration evaluation methodology-letter report 3: Selected tests of infiltration using two-dimensional numerical models. Pacific Northwest Laboratory, Richland, WA (1989)
29. Spataro, D., D'Ambrosio, D., Filippone, G., Rongo, R., Spataro, W., Marocco, D.: The new SCIARA-fv3 numerical model and acceleration by GPGPU strategies. International Journal of High Performance Computing Applications **31**(2), 163–176 (2017)
30. Spingola, G., D'Ambrosio, D., Spataro, W., Rongo, R., Zito, G.: Modeling Complex Natural Phenomena with the libAuToti Cellular Automata Library: An example of Application to Lava Flows Simulation. In: PDPTA - International Conference on Parallel and Distributed Processing Techniques and Applications. pp. 277–283 (2008)
31. Van Genuchten, M.T.: Calculating the unsaturated hydraulic conductivity with a new closed-form analytical model. Researh Reprot - Water Resour. Program, Dep. of Civ. Eng., Princeton Univ., Princeton, NJ (1978)