A parallel computational framework for the simulation of variably saturated flow based on the Cellular Automata concept using CUDA architecture.

Paolo Burlando[1], Grigorios G. Anagnostopoulos[1] and Adamos Kyriakou[2]

Institute of Environmental Engineering (1), Computer Vision Lab (2), ETH Zurich, Switzerland

correspondence: anagnostopoulos@ifu.baug.ethz.ch

ETH Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Abstract No: IN13A-1317

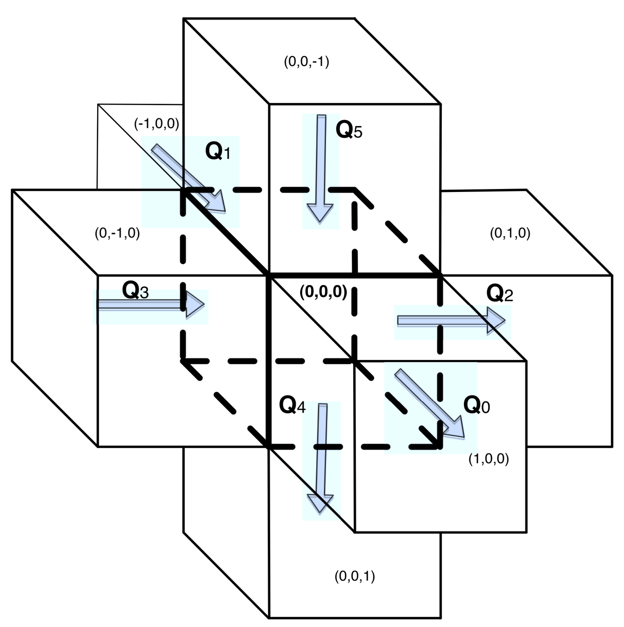
1. Introduction

A simple and efficient computational framework is presented for the simulation of variably saturated flow in porous media. In this modeling approach the Cellular Automata (CA) concept is implemented.

- It is efficient for the simulation of large scale phenomena.
- The inherent CA concept simplicity and its natural parallelism make its implementation easy within the CUDA framework.

2. Computational algorithm

According to the macroscopic CA notion the computational domain consists of a two or three dimensional lattice, which is composed by rectangular or prismatic cells respectively. Every cell of the lattice communicates with its neighbors only through its faces.



Coupling the discrete formulation of the mass balance of an arbitrary cell with the Darcy-Buckinghams law one can compute the head at time $t + \Delta t$:

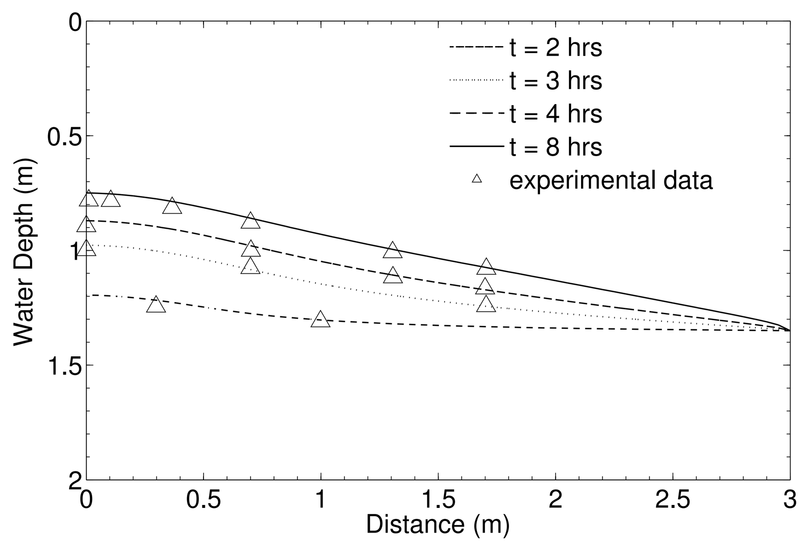
$$h_c^{t+\Delta t} = \frac{\sum_{\alpha \in I} \frac{\overline{K}_{\alpha c} A_{\alpha c}}{l_{\alpha c}} h_\alpha^t + \frac{V_c \sigma(\psi_c)}{\Delta t} h_c^t + \sum_{\alpha \in I'} Q_{bound}^{\alpha} + S_c}{\sum_{\alpha \in I} \frac{\overline{K}_{\alpha c} A_{\alpha c}}{l_{\alpha c}} + \frac{V_c \sigma(\psi_c)}{\Delta t}}$$

The above equation is applied in all the cells of the lattice except those, which have a Diriclets boundary condition, the hydraulic head of which is fixed throughout the simulation.

3. Verification of the algorithm

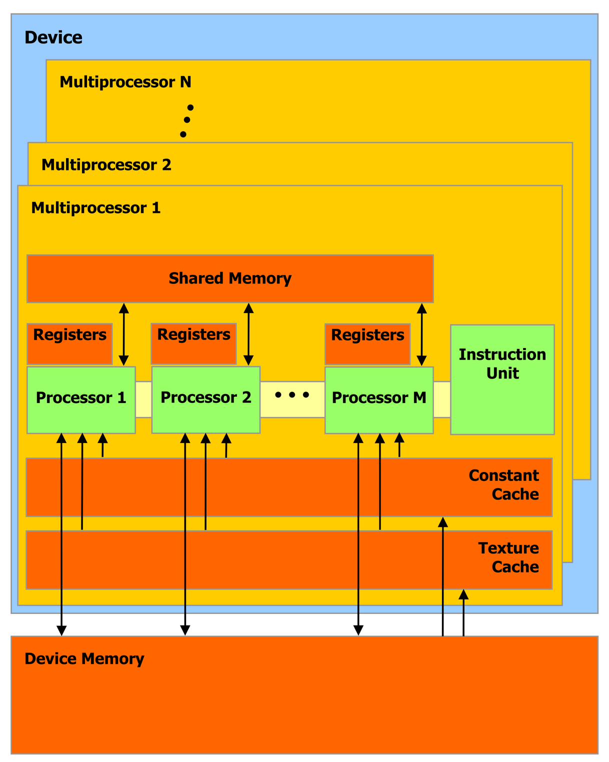
The presented algorithm was tested against known benchmark cases available from the literature, in order to evaluate its performance. These include experimental data, analytical solutions and numerical experiments (Anagnostopoulos and Burlando, 2011).

An example is the infiltration experiment of Vauclin et al (1979), which is used to evaluate the ability of the model simulating the transient position of the water table in a laboratory scale soil box.



4. CUDA Architecture

CUDA is a general purpose parallel computing architecture that leverages the parallel compute engine in NVIDIA GPUs to solve many complex computational problems in a more efficient way than on a CPU.



CUDA comes with a software environment that allows developers to use C as a high-level programming language.

5. Implementation and performance

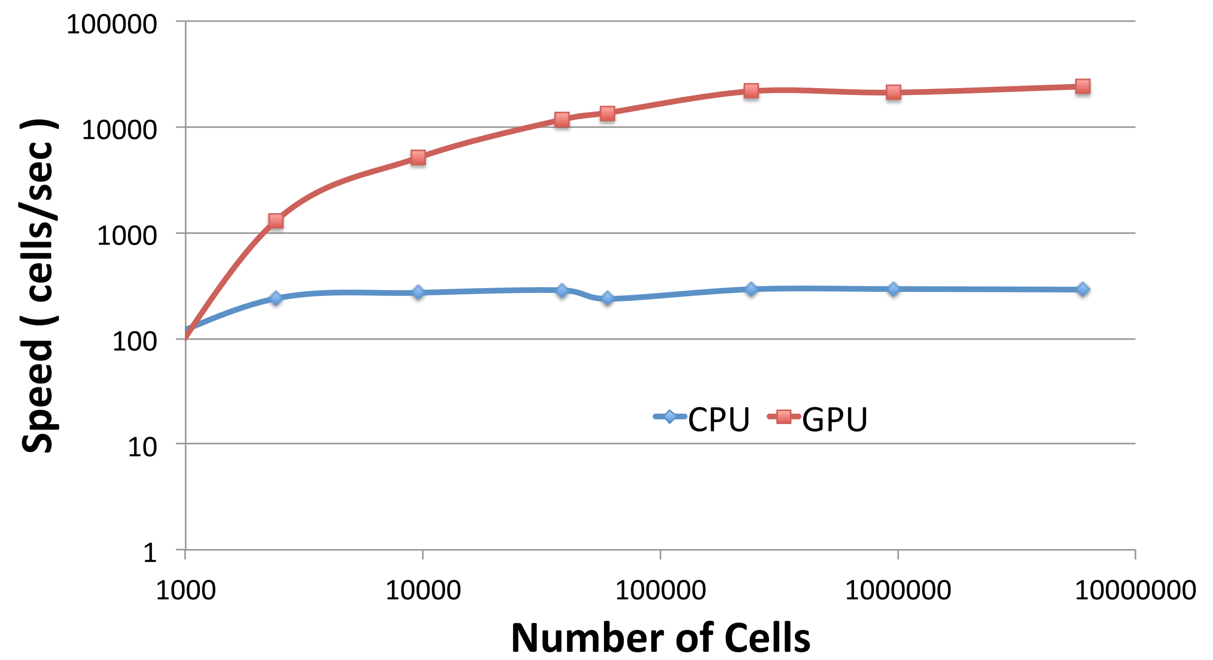
The most challenging issue is the fact that the domain can have irregular geometry, which can make more difficult the exploitation of locality at the thread computations and the use of the shared memory.

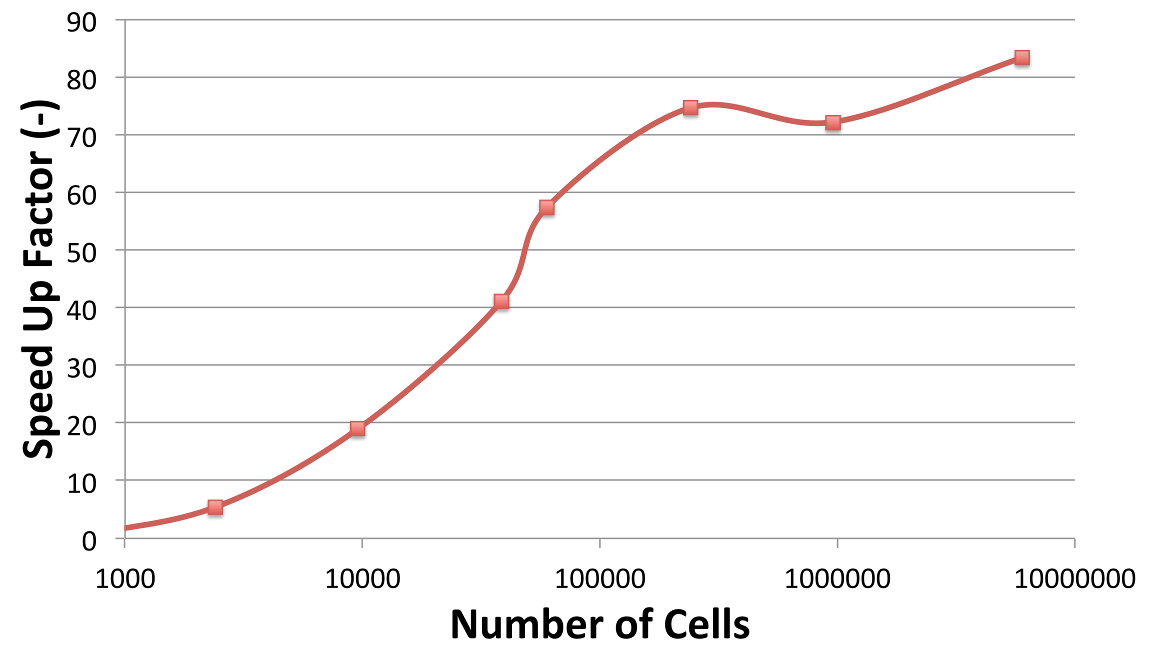
For the runs we used a Nvidia Quadro 2000 graphics card with 192 CUDA cores installed in a pc with an Intel Xeon processor at 2.93 GHz. The benchmark case of Vauclin et al (1979) was used for assessing the performance of the code for grid dimensions of increasing size (scale effect).

Parallelization strategy:

- The cell values are stored in a 1D array and for each cell the indexes of its neighboring cells were also stored. Both of these matrices reside in the **global memory**.
- Simulation constants are stored in the **constant** memory.
- Soil properties for each soil class are stored in the **texture** memory.
- **Atomic operations** are used in order to check for convergence at every iteration.
- The **shared** memory is used to accelerate the atomic operations and the block's memory accesses.

Results and conclusions:





- The speed up factor increases with grid dimension. As the domain size increases more computational resources of the GPU are exploited.
- Our framework is very attractive for basin scale simulations (e.g. in natural hazards assessment) where the grid sizes can become excessively large.

References

[1] G.G. Anagnostopoulos, P. Burlando, (2011). Object-oriented computational framework for the simulation of variably saturated flow, using a reduced complexity model, *Submitted in Environmental Modelling & Software*

[2] M. Vauclin, D. Khanji, G. Vachaud, (1979). Experimental and numerical study of a transient, two-dimensional unsaturated-saturated water recharge problem. *Water Resources Research, Vol 15*

[3] NVIDIA (2010). Cuda programming guide, 3.0, *Available: http://developer.download.nvidia.com/compute/cuda/3_0/toolkit/docs/NVIDIA_CUDA_ProgrammingGuide.pdf*