



Jul 1st, 12:00 AM

Simulation of Pesticide Percolation in the Soil Based on Cellular Automata

Stefania Bandini

Giulio Pavesi

Follow this and additional works at: <https://scholarsarchive.byu.edu/iemssconference>

Bandini, Stefania and Pavesi, Giulio, "Simulation of Pesticide Percolation in the Soil Based on Cellular Automata" (2002). *International Congress on Environmental Modelling and Software*. 172.
<https://scholarsarchive.byu.edu/iemssconference/2002/all/172>

This Event is brought to you for free and open access by the Civil and Environmental Engineering at BYU ScholarsArchive. It has been accepted for inclusion in International Congress on Environmental Modelling and Software by an authorized administrator of BYU ScholarsArchive. For more information, please contact scholarsarchive@byu.edu, ellen_amatangelo@byu.edu.

Simulation of Pesticide Percolation in the Soil Based on Cellular Automata *

Stefania Bandini, Giulio Pavesi ^a

^aDept. of Computer Science, Systems and Communication
University of Milan–Bicocca,
Milan, Italy
e-mail: {bandini,pavesi}@disco.unimib.it

Abstract: In the last decades, the use of pesticides has increased exponentially agricultural production. However, it is a well known fact that excessive use of pesticides can be hazardous both for the environment and human health. It is thus essential to assess thoroughly the impact on the environment of a given pesticide before its application. When they are sprayed on crops, pesticides are absorbed by the soil. Then, when water flows through the soil because of rain or floods, chemical substances can be released into it. The amount of pesticide released into water mainly depends on the chemical properties of the pesticide itself and the physical and morphological properties of the soil. Finally, water carries the pesticides to the groundwater layer, polluting it. Simulation models provide a useful tool for the evaluation of the pollutive potential of a pesticide, and in the absence of experimental data are often the only option available. The model we present is based on Cellular Automata. It permits to reproduce explicitly the flow of water, and the chemical reactions taking place between water and soil. Moreover, with Cellular Automata is possible to generate artificial percolation beds, reproducing in detail the morphological features of real types of soil, thus reducing the need to collect experimental data. The results obtained from simulations are consistent with real case studies.

1 THE PROBLEM

In the last decades, the use of pesticides has increased exponentially agricultural production. However, it is a well known fact that excessive use of pesticides can be hazardous for the environment, non–target organisms, and eventually human health.

One of the potential dangers deriving from the application of pesticides is the pollution of groundwater (Gilliom et al. [1999]). After their application to crops, pesticides are absorbed by soil. Then, if water flows (*percolates*) through the soil because of rain or floods, pesticides can be released (*leach*) into it. The amount of pesticide leached changes according to the chemical properties of the pesticide itself and the physical and morphological properties of the soil. Water containing pesticide may eventually reach the underlying groundwater. It is thus essential to assess thoroughly the impact on the environment of a given pesticide before its application (Butler [1969]).

Computational models are a very useful tool for assessing the leaching potential of a pesticide into groundwater, especially in the absence of experimental data. Several models based on differential equations have been proposed for predicting pesticide concentration in surface water and groundwater. A comparative review is given for example by Vancloster et al. [2000]. The main differences lie in the approach, in the complexity of the equations used, and the amount of data required. Unfortunately, data required by these models are only partial, difficult to obtain, and have a variable degree of quality, while high quality data are needed to obtain a reliable prediction. Moreover, available data usually refer to specific locations (types of crops and soil) and to specific pesticides. In the absence of all of part of the data needed to fine–tune the parameters of analytical models, qualitative models, like the one presented in this paper, are perhaps the only option available for researchers. The goal of this kind of models, rather than replacing real experiments and tests, is to point out which are potentially the most dangerous scenarios for the application of a given pesticide, and to provide directions for the planning of experimental campaigns.

*This work was part of the TTN/CAPP project, supported by European Commission DG3.

2 THE MODEL

The extraction process of soluble substances (in our case, pesticides) from the *percolation bed* (the soil) can be divided into two main phases: *washing* and *diffusion*. Washing corresponds to the reaction taking place between water flowing into the percolation bed and the surface of the soil particles. This phenomenon causes the release of pesticide from the soil to the water. Diffusion is due to the movement of water through the channels of the percolation bed, determined by gravity. Percolation is a very complex physical phenomenon, that can be studied from theoretical (Stauffer and Aharony [1992]) or applicative (Sahimi [1994]) points of view. Percolation has been simulated by numerical analysis or by fractal theory. More recently, *Cellular Automata* (CA) have offered new approaches to the modeling and the simulation of the process.

In this case, we modeled the process with two-dimensional Cellular Automata (Wolfram [1994]). A Cellular Automaton consists of a regular discrete lattice of *cells*, where each cell is characterized by a state taken from a finite set of states. Each cell evolves (changes its state) according to a given *update rule*, which depends only on the state of the cell and a finite number of neighboring cells. All the cells of the automaton follow the same update rule. The automaton evolves through a sequence of discrete time steps, where the state of the cells is updated simultaneously. In our model, cells are arranged on a two-dimensional square lattice, and we adopted the von Neumann neighborhood, where every cell has four neighbors (see Fig. 1) that influence its evolution. CA have been widely used in the discrete modeling of complex systems (see for example Gutowitz [1991]), and several works concerning the modeling of reaction-diffusion phenomena with CA have been developed (see for example Dab and Boon [1990]). The main contribution of our approach consists of a model where chemical reactions and the movement of fluid particles in a porous medium can be explicitly described and simulated.

The states of the cells represent the different entities (water, soil and empty spaces) involved in the process, and the update rule has been defined in order to simulate the interactions occurring during the percolation. Thus, a cell can be seen as a container, that can be *empty* or contain *water* or *soil*. Water or soil cells can also contain pesticide, expressed as an integer number of *particles*. Furthermore, each cell is divided in four parts, (as shown in Fig. 2), and the overall number of particles contained in the cell has to be evenly distributed among

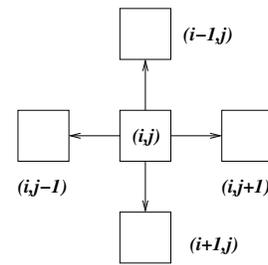


Figure 1: The von Neumann neighborhood of a cell located in a two-dimensional square lattice.

the four portions. The basic idea of this model has been applied to other similar problems (see for example Bandini and Magagnini [2001]), and can also be used to study phenomena not strictly related to percolation, as in Bandini and Simone [1996].

3 THE CELLULAR AUTOMATON

The cells of the automaton are contained in the infinite two-dimensional square lattice \mathcal{Z}^2 . We suppose that the cells are located on the nodes (i, j) , with $0 \leq i < M$ and $0 \leq j < N$, where M is the number of rows of the automaton and N is the number of columns. With $C(i, j)$ we will refer to the cell located on the node (i, j) . The neighbors of $C(i, j)$ will be thus denoted by $C(i-1, j)$, $C(i+1, j)$, $C(i, j-1)$ and $C(i, j+1)$. The state of each cell $C(i, j)$ of the automaton is defined by:

$$C(i, j) = \langle I, N, S, W, E, \mathbf{F} \rangle$$

where:

1. I is the *identity*, that can assume one of the following values: *empty*, *water*, *soil*.
2. N, S, W, E are four integer variables representing the number of pesticide particles contained in each portion of the cell.
3. $\mathbf{F} = \langle u, d, l, r \rangle$ is a 4-tuple of one-bit variables (flags), needed to simulate the movement of water in the percolation bed.

From now on, we will refer with $N_{i,j}$, $W_{i,j}$, $E_{i,j}$, and $S_{i,j}$ to the number of particles contained in the four portions, with $u_{i,j}$, $d_{i,j}$, $l_{i,j}$ and $r_{i,j}$ to the four movement flags, and with $I_{i,j}$ to the identity of the cell $C(i, j)$.

In the initial configuration, some cells of the automaton have $I = \textit{soil}$. These cells are grouped together to form grains that resemble the morphology



Figure 2: The structure of a cell of the automaton.

of actual percolation beds as shown in Fig. 3. All the other cells are empty. It is worth mentioning that this step, that is, defining an initial configuration resembling the features of real soil, has been performed with another cellular automaton, that allows the generation of percolation beds reproducing in two dimensions the morphological properties (for example, shape and size of the grains) of different types of soil, as described in Bandini et al. [2001]. Then, a given number of pesticide particles is assigned to the cells located on the surface of grains (that is, cells with $I = soil$ with at least one neighbor with $I = empty$). When the automaton is started, water is added to the percolation bed by setting to *water* the identity of some empty cells. Then, water starts to flow inside the bed and wash pesticide from the soil for effect of the update rule of the automaton, described in the following section.

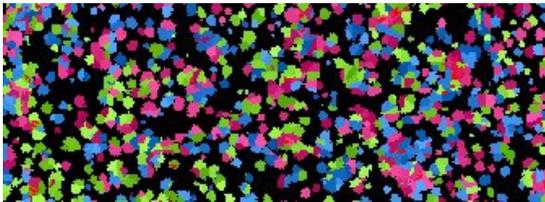


Figure 3: An initial configuration of the automaton. Gray agglomerates of cells represent the grains forming the percolation bed.

4 THE UPDATE RULE

The update rule of the automaton is divided in three separate steps: *reaction*, *balance*, and *movement*. In the first step pesticide particles are exchanged by water cells and soil cells. In the second, the particles contained in each cell after the reaction step is evenly balanced in the four portions. In the last step, water is allowed to move inside the percolation bed, transferred from cell to cell.

4.1 Reaction

Reaction takes place as follows: for each cell, the amount of pesticide contained in each portion is balanced with the amount of pesticide contained in the adjacent portions of the neighboring cells. For example, if cell $C(i, j)$ contains $p = N_{i,j}$ particles in its northern portion, and its northern neighbor $C(i-1, j)$ contains $q = S_{i-1,j}$ particles in its southern portion, at the end of the reaction each portion will contain $\lfloor (p + q)/2 \rfloor$ particles. Since the number of particles must be integer, the possible particle corresponding to the remainder of the division is assigned randomly to one of the portions. This rule describes how the distribution of concentration solubles follows an equilibrium search law, corresponding to the mechanism of the physical diffusion equation considering the concentration diffusion. The same rule is applied simultaneously to the other three portions (eastern, southern and western) of the cell and the corresponding neighbors. Figure 4 shows two neighboring portions before and after the reaction step. The reaction rule is applied when two adjacent cells contain water, or when one contains water and the other soil. No reaction takes place between two soil cells, in order to keep pesticide particles only in cells located on the surface of grains. Likewise, no reaction is allowed when one of the portions is empty, since particles cannot be contained by an empty cell.

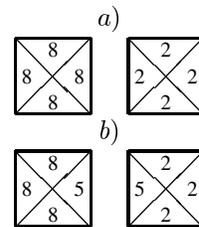


Figure 4: Two cells before (a) and after (b) the reaction step. Numbers represent the particles contained in each portion of the cells.

4.2 Balance

This rule makes the concentration of solubles homogeneous within each element involved in the reactions described in the previous section. Once the reaction step has been performed simultaneously by all the cells of the automaton, the particles contained in each cell are balanced in the four portions. That is, given $N_{i,j}$, $S_{i,j}$, $E_{i,j}$, and $W_{i,j}$ (the number of particles in the four portions of cell

$C(i, j)$ after reaction), at the end of the balance step the number of particles in each portion will be $\lfloor (N_{i,j} + W_{i,j} + S_{i,j} + E_{i,j}) / 4 \rfloor$. Once again, the particles corresponding to the remainder of the division are assigned randomly to the four portions. Figure 5 shows two cells before and after the this step.

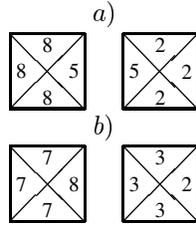


Figure 5: Two cells before (a) and after (b) the balance step. Numbers represent the particles contained in each portion of the cells.

4.3 Movement

In the *movement* step, water is moved from cell to cell inside the percolation bed simulating the effect of gravity. Basically, water always tries to move downwards (that is, to the southern neighbor of the cell). Otherwise, (if the southern neighbor is not empty) water tries to move laterally. If both the lateral neighbors are not empty, water cannot move.

Had the water moved only downwards, the rule would have been quite simple: exchange the parameters of cell $C(i, j)$ containing water with the parameters of the empty neighbor $C(i + 1, j)$ (all set to zero, since an empty cell cannot contain particles). Trouble comes from lateral movement: that is, the same empty cell could be the destination chosen by water coming from more than one cell (at most three, the northern neighbor and the two lateral neighbors). Therefore, another rule must be added, deciding which one of the different water cells is allowed to transfer its contents to the empty cell, while the others keep their state unchanged. For this reason, the movement step has been split into three sub-steps, employing the *movement flags* defined in the previous section.

Given a cell $C(i, j)$, with $I_{i,j} = \textit{water}$, the first sub-step is defined as follows:

1. If $I_{i+1,j} = \textit{empty}$, $d_{i,j} = 1$. That is, if the southern neighbor is empty, water tries to

move downwards, and the corresponding flag is set.

2. Else, if exactly one of the lateral neighbors is empty, set the corresponding flag ($l_{i,j}$ or $r_{i,j}$).
3. Else, if *both* the lateral neighbors are empty, choose one of the two directions at random, and set the corresponding flag.
4. Else, water cannot leave the cell: no flag is set.

At the end of the first sub-step, each cell containing water has therefore signaled the intention to transfer its content to an empty cell by setting the flag corresponding to the chosen direction. In the second sub-step, empty cells examine the flags of their neighbors, and decide which one can transfer the water. This step is performed by setting to one the movement flag of the empty cell in the direction corresponding to the position of the succeeding neighbor. Given a cell $C(i, j)$, with $I_{i,j} = \textit{empty}$, the choice is based on the following criterion:

1. If $d_{i-1,j} = 1$, then $u_{i,j} = 1$. That is, water trying to move downwards always succeeds, and the empty cell sets its upper flag.
2. Else, if exactly one of the lateral neighbors has the movement flag set ($r_{i,j-1}$ or $l_{i,j+1}$), it wins, and the corresponding flag (respectively, $l_{i,j}$ or $r_{i,j}$) is set.
3. Else, if *both* the lateral neighbors have the movement flag set ($r_{i,j-1}$ and $l_{i,j+1}$), the cell chooses randomly the winner and sets its own corresponding flag ($l_{i,j}$ or $r_{i,j}$).
4. Else, none of the neighbors is trying to transfer its water; no flag is set.

Thus, at the end of the second sub-step, water can move to the empty cell indicated by its flag if the empty cell has the flag on its direction set. The rule has been defined in order to give the precedence to water moving downwards, while water trying to move laterally may have tough luck. The actual movement of water takes place in the third sub-step. Given a cell $C(i, j)$ containing water trying to move to a neighboring empty cell:

1. If $d_{i,j} = 1$ and $u_{i+1,j} = 1$ then $C(i + 1, j) = C(i, j)$ (all the parameters are copied to the empty cell) and $C(i, j) = \textit{EMPTY}$ (the cell becomes empty and all its parameters are set to zero).

2. Else, if $l_{i,j} = 1$ and $r_{i,j-1} = 1$ then $C(i, j - 1) = C(i, j)$ and $C(i, j) = EMPTY$.
3. Else, if $r_{i,j} = 1$ and $l_{i,j+1} = 1$ then $C(i, j + 1) = C(i, j)$ and $C(i, j) = EMPTY$.
4. Else, the cell cannot transfer its water, and its state remains unchanged.

5 THE SIMULATION

To start a simulation, we need two further parameters: the *water saturation constant* and the *soil saturation constant*, representing, respectively, the maximum number of pesticide particles that can be contained by a water portion and by a soil portion. These parameters change according to the chemical properties of the pesticide and the soil employed in the simulation. Adding the saturation constants changes slightly the reaction rule, while the other rules remain the same. For instance, we show how the rule changes given two reacting portions $E_{i,j}$ and $W_{i,j+1}$. Let p and q be the number of particles contained respectively in the two portions; let $SC_{i,j}$ and $SC_{i,j+1}$ be the saturation constants of the two portions (corresponding to their content). Let us suppose that $\lfloor (p + q)/2 \rfloor > SC_{i,j}$, that is, the number of particles resulting from the reaction rule is greater than the saturation constant of one of the portions (notice that it cannot be greater than both the saturation constants). Then, the amount of particles resulting from the reaction step defined before changes in this way:

$$\begin{aligned} E_{i,j} &= SC_{i,j} \\ W_{i,j+1} &= (p + q) - SC_{i,j} \end{aligned}$$

This ensures that after the reaction step neither portion contains more particles than its saturation constant. The possible exceeding particles are simply assigned to the other portion. The higher is the water saturation constant, the more pesticide can leach from the soil, that is, it is more soluble in water.

As mentioned before, the simulation starts with some cells of the automaton set to *soil* and containing pesticide particles, while the remaining cells are empty. Then, to start the simulation and during the simulation itself water is added to the automaton by setting to *water* the identity of some empty cells. For example, in order to simulate heavy rain, at each update step the empty cells located in the topmost row of the automaton are filled with water, or, if rain is not that heavy, the above operation can be performed every k steps, or at random (water is added

at each step with probability p). Moreover, to reproduce a flood, all the empty cells can have their identity set to *water* at the beginning of the simulation, and water can be added to the automaton before each update step according to one of the rules described above. Water is also allowed to leave the percolation bed. After each update step, the cells in the bottom row of the automaton are emptied, that is, if their identity is *water*, it is set to *empty*, and their parameters are set to 0. During this operation, we also compute the overall number of pesticide particles that were contained by the cells that have been emptied. In this way, we obtain the amount of pesticide that leaves the percolation bed at each update step, that is, the amount of pesticide leached from the bed that reached the groundwater. Simulations can be stopped after a given number of steps, or when the percolation bed is empty. That is, we stop adding water to the bed and wait until all the water has left it. However, water can form puddles when it cannot move, either downwards or laterally. The steps performed during a simulation can be summarized as follows:

1. Set up of the percolation bed and of the saturation constants;
2. Add water (optional);
3. Update:
 - (a) Reaction, modified to take into account the saturation constants;
 - (b) Balance;
 - (c) Movement:
 - i. Movement 1;
 - ii. Movement 2;
 - iii. Movement 3;
4. Remove water from the bottom row;
5. Update of the simulation statistics;
6. Go to 2.

6 EXPERIMENTAL EVALUATION

The amount of pesticide that leaches from soil and reaches groundwater is determined by the chemical properties of the substance as well as the chemical and structural characteristics of the soil. The likelihood of a pesticide to be washed away depends largely on its solubility, that is, its ability to dissolve in water. In our model, this corresponds to setting a high value for the water saturation constant, with respect to the soil saturation constant. Furthermore,

the results of the percolation process is also influenced by the properties of the soil. Sand (made of large and regular grains) permits fast movement of percolating water, and lowers the opportunity for dissolved chemicals to be absorbed, while clay (smaller and more irregular grains) tends to hold water and the chemicals dissolved into it. Also, the latter has more surface area on which pesticides can be absorbed, when compared to sandy soil. The sandier is the soil (that is, the larger are the grains composing the soil), the greater is the risk of a pesticide to reach groundwater. We performed different simulations keeping the same values for the saturation constants, and changing the initial configuration of the automaton, that is, the shape and the size of the grains forming the percolation bed.

The results we obtained reflect qualitatively the outcomes of real experiments: with larger grains, the number of pesticide particles leached was higher than the number obtained with smaller grains. It is worth mentioning that we obtained this result *before* knowing the results of real experiments, that is, we did not adapt our model in order to reproduce them. Another factor that influences the process is the amount of organic matter contained in the soil. Chemicals contained in pesticides tend to bind to the organic matter, and are less likely to be released into water. Therefore, percolation beds with higher organic matter content are less dangerous. We did not include this aspect into our model directly, but it could be easily reproduced once again by setting appropriate saturation constants, or with little changes in the reaction rule. Further details on the experiments performed with the model can be found in Bandini et al. [1998].

7 CONCLUSIONS

We presented a model based on CA for the simulation of pesticide percolation in the soil. The model, rather than replacing real experiments, can provide advice on which are the most dangerous scenarios (i.e. type of soil, weather conditions, and so on) for the application of a given pesticide, so to give directions for the planning of experimental campaigns. There are some other features that could be easily added in order to extend the model. For example, pesticides tend to degrade with time, becoming harmless for the environment. Disappearance from soil can also be a consequence of photo-decomposition, hydrolysis, oxidation or other chemical reactions. These aspects could be reproduced by defining an aging function that removes old particles after a given (possibly random)

number of update steps. Finally, we think that some attention should also be paid to the initial configuration of the automaton. In our simulations, we simply distributed pesticide particles uniformly on the surface of cells. We think that a different distribution, reflecting more in detail how pesticides are absorbed by soil, could increase the feasibility of our model.

REFERENCES

- Bandini, S., G. Azimonti, and G. Pavesi. Cellular automata for percolation processes. Technical Report DE9 - PM14, ESPRIT HPCN PST Activity CAPP, 1998.
- Bandini, S. and M. Magagnini. Parallel processing simulation of dynamic properties of filled rubber compounds based on cellular automata. *Parallel Computing*, 27(5), 2001.
- Bandini, S., G. Mauri, and G. Pavesi. Parallel generation of percolation beds based on stochastic cellular automata. In *Proceedings of Pact '01*, volume 2127 of *Lecture Notes in Computer Science*, Novosibirsk, Russia, 2001. Springer Verlag.
- Bandini, S. and C. Simone. Reaction diffusion computational models. In *Proceedings of the 3rd Systems Science European Congress*, Rome, 1996.
- Butler, P. Monitoring pesticide pollution. *BioScience*, 19(10), 1969.
- Dab, D. and J. Boon. Cellular automata to reaction-diffusion systems. *Springer Proceedings in Physics*, 46, 1990.
- Gilliom, R., J. Barbash, D. Kolpin, and S. Larson. Testing water quality for pesticide pollution. *Environmental Science and Technology*, 33, April 1999.
- Gutowitz, H. *Cellular Automata: Theory and Experiment*. MIT Press, Cambridge, 1991.
- Sahimi, M. *Applications of Percolation Theory*. Taylor and Francis, London, 1994.
- Stauffer, D. and A. Aharony. *Introduction to Percolation Theory*. Taylor and Francis, London, 1992.
- Vanclouster, M., J. Boesten, M. Trevisan, C. Brown, E. Capri, O. Eklo, B. Gottesbüren, V. Gouy, and A. van der Linden. A european test of pesticide-leaching models: methodology and major recommendations. *Agricultural Water Management*, 44:1-19, 2000.
- Wolfram, S. *Cellular Automata and Complexity*. Addison-Wesley, 1994.