CELLULAR AUTOMATA METHOD IN STUDYING RECRYSTALLIZATION PROCESS

Vasile ANĂSTĂSOAEI, Constantin GHEORGHIES
”Dunărea de Jos” University of Galati, Faculty of Science and Environment,
111, Domnească Street, RO-800201, Galati, Romania
emails: Vasile.Anastasoaei@ugal.com, Constantin.Gheorghies@ugal.ro

ABSTRACT

A cellular automaton is a decentralized computing model providing a platform for performing complex computation with the help of only local information. The cellular automata method describes the evolution of a discrete system of variables by applying a set of deterministic (or probabilistic) rules that depend on the values of the variables as well as those in the nearby cells of a regular lattice. In this paper we present some types of cellular automata used for modelling and simulating recrystallization process.

KEYWORDS: cellular automata, modelling, recrystallization process

1. Introduction

A cellular automaton is a discrete dynamical system that consists of a regular network of finite state automata, named cells, which change their states depending on the states of their neighbors, according to a local update rule. All cells change their state simultaneously, using the same update rule. The process is repeated at discrete time steps. Cellular automata are discrete in both space and time, homogeneous in space and time, and local in their interactions. Cellular automata can have arbitrary dimensions. Space is defined on a regular array of cells that can be regarded as the nodes of a finite difference field. The state of each cell is characterized in terms of a set of generalized state variables. The actual values of these state variables are defined at each of the individual cell. The opening state of the automaton is defined by mapping the initial distribution of the values of the chosen state variables onto the cells in the network. The dynamic evolution of the automaton takes place through the application of deterministic (or probabilistic) transformation rules that act on the state of each cell. These rules determine the state of a cell as a function of its previous state and the state of the neighbours of the cell. Cellular automata work in discrete time steps. After each time interval, the values of the state variables are updated for all cells simultaneously, mapping the new values assigned to them through the transformation rule. Cellular automata were introduced by von Neumann and Ulam for the simulation of self-reproducing Turing automata and population evolution. In the last years, cellular automata have been used to simulate the microstructure evolution in materials science. In this domain a number of different aspects were addressed, namely, primary static recrystallization (Hesselbarth and Gobel, [2]; Pezzee and Dunand [3]; Marx and Raabe [4]). Although cellular automaton simulations are typically carried out at an elementary level (atoms, clusters of atoms, dislocation segments, subgrains), it should be emphasized that particularly those variants that discretize and map in a continuum space are not intrinsically calibrated by a characteristic physical length or time scale. This means that a cellular automaton simulation of continuum systems requires the definition of elementary units and transformation rules that adequately reflect the system behaviour at the level addressed.

2. Basic definitions and notations

Let d be a positive integer. We consider Z^d a d-dimensional cellular space, whose elements are called cells. Let S be a finite set of elements, called states. A *configuration* is a function $c: Z^d \rightarrow S$ that assigns a state to each cell. A d-dimensional *neighbourhood vector*, of size m, is a t-uple $N= (z_1,z_2,...,z_m)$, where each $z_i \in Z^d$, and $z_i \neq z_j$ for all $i \neq j$. The elements z_i specify the relative locations of the neighbours of each cell. A *rule* (local rule) is a function $f: S^m \rightarrow S$, that specifies the new state of each cell based on the old states of its neighbours. If the neighbours of a cell have states $s_1,s_2,\ldots,s_m$, then the new state of the cell is $f(s_1,s_2,\ldots,s_m)$. In cellular automata all cells use the same rule, and the rule is applied at all cells simultaneously.

To specify a cellular automaton one needs to specify the following items:

- the dimension *d,*
- the finite state set S,
- the neighbourhood vector $N= (z_1, z_2 \ldots , z_m)$, and
- the local rule $f{:}S^m\rightarrow S$.

Formally, we define the corresponding *cellular automaton* to be the 4-uple $A=(d,S,N,f)$.

Let $N= (z_1, z_2 \ldots, z_m)$ be a *d*-dimensional neighbourhood vector. For any cell $z \in Z^d$, we denote $N(z) = (z + z_1, z + z_2, \ldots, z + z_m)$, and for any $K \in Z^d$, we denote $N(K) = \{z + z_i | z \in K, i = 1,2,\ldots, m\}$, that is N(z) is the ordered sequence of the neighbours of cell z, while N(K) is the unordered set of neighbours of cells in K. In the two-dimensional space, the *von Neumann*- and the *Moore*- neighbourhoods are often used. In the von Neumann configuration, $N=(z_1,z_2,z_3,z_4)$, $z_1=(-1,0)$, $z_2=(0,-1)$, $z_3=(0,1)$, $z_4=(1,0)$, while in the Moore configuration, $N=(z_1,z_2,z_3,z_4, z_5,z_6,z_7,z_8)$, $z_1=(-1,-1)$, $z_2=(-1,0)$, $z_3=(-1,1)$, $z_4=(0,-1)$, $z_5=(0,1)$, $z_6=(1,-1)$, $z_7=(1,0)$, $z_8=(1,1)$ (Fig. 1).

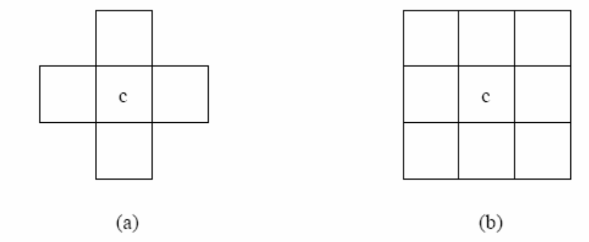

Fig. 1*. The (a) von Neumann and (b) Moore neighbors of the c cell*

The local interaction of neighbouring sites in a cellular automaton is specified through a set of deterministic or stochastic transformation rules. The value of an arbitrary state variable s assigned to a particular site at a time t ($t=t_0+\Delta t$) is determined by its present state (t_0) or its last few states (t_0, $t_0-\Delta t$, etc.) and the state of its neighbors. For example, in a one-dimensional cellular automaton case, considering the last two time steps for the evolution of the cellular automaton, we can write formally

$$s_j^{t_0+\Delta t} = f(s_{j-1}^{t_0-\Delta t}, s_j^{t_0-\Delta t}, s_{j+1}^{t_0-\Delta t}, s_{j-1}^{t_0}, s_j^{t_0}, s_{j+1}^{t_0})$$

where $s_j^{t_0}$ indicates the value of the variable at a time t_0 at the node *j*.

The positions *j+1* and *j-1* indicate the nodes in the immediate neighborhood of position *j*.

The function *f* specifies the set of transformation rules, such as provided by standard finite difference algorithms. Due to the discretization of space, the type of neighboring affects the local transformation rates. In the case of a higher-dimensional cellular automaton simulation, with independent variables described by rectangular coordinates, the predictions become dependent on the shape of cells. In the two-dimensional case, with von Neuman configuration, we have $x(i,j)=f(x(i-1,j),x(i,j-1),x(i,j+1),x(i+1,j))$, while with the Moore configuration, we have $x(i,j)=f(x(i-1,j-1),x(i-1,j),x(i-1,j+1), x(i,j-1),x(i,j+1) x(i+1,j-1),,x(i+1,j), x(i+1,j+1))$.

Transforming these somewhat abstract rules and properties associated with general automata into a materials-related simulation concept consists in mapping the values of relevant state variables onto the cells of a cellular automaton grid which reflects the independent spatial coordinates, and using the approximate local finite difference solutions of the underlying partial differential equations of the model addressed as local transformation rules.

For the Moore configuration, which allows one to introduce a certain medium-range interaction among the sites, the equation of the state can be writen as:

$$s_j^{t_0+\Delta t} = f(s_{j-2}^{t_0-\Delta t}, s_{j-1}^{t_0-\Delta t}, s_j^{t_0-\Delta t}, s_{j+1}^{t_0-\Delta t}, s_{j+2}^{t_0-\Delta t}, s_{j-2}^{t_0}, s_{j-1}^{t_0}, s_j^{t_0}, s_{j+1}^{t_0}, s_{j+2}^{t_0})$$

The state of a cellular automaton is completely specified by the transformation rules and by the values of the state variables at each site. Even for very simple automata there exists an enormous variety of possible transformation rules. For a one-dimensional binary cellular automaton with von Neumann neighboring, each node must assume one of two possible states, $s_j=0$ or $s_j=1$. From the state equation

$$s_j^{t_0+\Delta t} = f(s_{j-1}^{t_0}, s_j^{t_0}, s_{j+1}^{t_0})$$

we can define 2^8 possible deterministic or probabilistic transformation rules *f*.

One of them is

if $\left(s_{j-1}^{t_0} = 0, s_j^{t_0} = 0, s_{j+1}^{t_0} = 0\right)$

then $s_j^{t_0+\Delta t} = 0$ $(0,0,0) \rightarrow 0$

if $\left(s_{j-1}^{t_0} = 0, s_j^{t_0} = 0, s_{j+1}^{t_0} = 1\right)$

then $s_j^{t_0+\Delta t} = 1$ $(0,0,1) \rightarrow 1$

if $\left(s_{j-1}^{t_0} = 0, s_j^{t_0} = 1, s_{j+1}^{t_0} = 0\right)$

then $s_j^{t_0+\Delta t} = 0$ $(0,1,0) \rightarrow 0$

if $\left(s_{j-1}^{t_0} = 0, s_j^{t_0} = 1, s_{j+1}^{t_0} = 1\right)$

then $s_j^{t_0+\Delta t} = 1$ (0,1,1) → 1

if $\left(s_{j-1}^{t_0} = 1, s_j^{t_0} = 0, s_{j+1}^{t_0} = 0\right)$

then $s_j^{t_0+\Delta t} = 1$ (1,0,0) → 1

if $\left(s_{j-1}^{t_0} = 1, s_j^{t_0} = 0, s_{j+1}^{t_0} = 1\right)$

then $s_j^{t_0+\Delta t} = 0$ (1,0,1) → 0

if $\left(s_{j-1}^{t_0} = 1, s_j^{t_0} = 1, s_{j+1}^{t_0} = 0\right)$

then $s_j^{t_0+\Delta t} = 0$ (1,1,0) → 1

if $\left(s_{j-1}^{t_0} = 1, s_j^{t_0} = 1, s_{j+1}^{t_0} = 1\right)$

then $s_j^{t_0+\Delta t} = 1$ (1,1,1) → 1

In general, the number of rules can be calculated by $|S|^m$.

The general algorithm of a cellular automaton has the form:

```
Input data:
iT – number of iteration allowed to perform when computing the solution
C – number of initial configurations used for rule evaluation

Initialization:
a) For each cell i {
    1. initialize rule table of cell i;
    2. $s_i=0;$}
b) count=0; initial configuration counter
Iteration:
while not done do:
    c) generate a random initial configuration
    d) run the algorithm on initial configuration for iT iteration;
    e) for each cell i
        1. if cell i is in the final state then $s_i=s_i+1$
    f) c=c+1
    g) if c (mod C)=0 then
    for each cell i
        1. compute $ns_i(c)$ (number of neighbors)
        2. if $ns_i(c)=0$ then don’t change the rule of cell i
        3. else if $ns_i(c)=1$ then replace the rule of cell i with the corresponding neighboring rule.
        4. else if $ns_i(c)=2$ then replace the rule of cell i with the crossover of the two corresponding neighboring rules.
        5. else if $ns_i(c)>2$ then replace the rule of the cell i with the crossover of the two randomly chosen neighboring rules.
        6. $s_i=0$.
```

3. Cellular automata in recrystallization process

Transforming the abstract rules of a cellular automaton into a materials-related concept consists of mapping the values of relevant state variables onto the points of the cellular automaton and using the local finite difference formulations of the partial differential equations of the underlying model as local transformation rule. The particular versatility of the cellular automatom approach for the simulation of recrystallization process is due to its flexibility in considering a large variety of state variables and transformation laws. Cellular automaton simulations are carried out at an elementary level using atoms, cluster of atoms, small crystalline elements as underlying units. To model the recrystallization process, the most used was the two-dimensional cellular automata. As neighbors model (related as local environment), was used the von Neuman environment and the Moore environment (Fig. 1).

In [2] it is presented a straightforward application of cellular automata to recrystalization. It is focused on capturing three phenomena:

h) nucleation of grains
i) growth of grains
j) the slowing of growth owing to the impingement of grains.

The model used contains:

1. the geometry of the cells: a two-dimensional square network of cells,
2. the number and the kind of states a cell can possess: two states per site, recrystallized or not recrystallized,
3. the definition of the neighborhood of a cell: von Neuman and Moore (Figure 1.),
4. the rules that determine the state of each cell in the next time step: rules for nucleation of new grains, growth of grains, and the impingement of grains.

The initial configuration is with all sites set to zero (all cells correspond to no recrystallized state), and a number N_i of „embryos” were placed in system by assigning non-zero values to randomly selected nodes on the network. The strategy is to assign to each embryo a label to identify the grain that arose from that embryo. The rule to describe growth: if A≥1 at time step *t*, then the central site would be considered recrystallized at time step *t+1* and take on the identity of the grain that extends into its neighborhood. Here, A is defined as the sum of recristallized neghbors of the central node, based on the selected neighborhood. At each time step, the creation of another set of N_i embryos is attempted.

The N_d cells are chosen at random and if a cell was not recrystallized, an is created an embryo, with a sequential number. Thus all growing grains have their own unique identification number.

4. Conclusions

We have presented a general model for cellular automata, and a general automaton algorithm. The implementation of this algorithm depends on the dimension of the network of cells, the model of neighbourhood, and the number of initial configurations. The presented model and algorithm are prepared in order to be applied in the modelling of recrystalization process, and it is important to define these rules, corresponding to the considered phenomena.

References

[1]. Chopard B., Droz M. - *Cellular automata modelling of Physical Systems,* Cambridge University Press, (1998)

[2]. Hesselbarth H. W., Göbel I. R. - *Simulation of recrystallization by cellular automata,* Acta Metallurgica et Materialia, 39, (1991), 2135-2143

[3]. Pezee C. E., Dunand D. C. - *The impingement effect of an inert, immobile second phase on the recrystallization of a matrix* , Acta Metall. 42 (1994)

[4]. Marx V., Raabe D., Gottstein G. - *Simulation of the influence of recovery on the texture development in cold rolled BCC-alloys during annealing.* Proceedings 16th RISØ Int. Sympos. on Mat. Science: Materials: Microstructural and Crystallographic Aspects of Recrystallization, RISO Nat. Lab, Roskilde, Denmark eds. N Hansen, D Juul Jensen, YL Liu, B Ralph, (1995) 461-466.

[5]. Dierk Raabe - *Cellular automata in Materials Science with particular reference to recrystallization simulation,* Annual Review of Materials Research, 32, 2002, 53-76

[6]. Wolfram S. (ed.) - *Theory and applications of cellular automata,* Singapore: World Scientific, (1986).