

Cellular Automata Describing Non-equilibrium Fluids with Non-mixing Substances

Carlos Ramos, Fernando Carapau , and Paulo Correia

1 Introduction

Cellular automata (CA) are discrete dynamical systems introduced by Von Neumann and Ulam, in the late 1940s [1]. Since then many applications in natural sciences and mathematics were developed, for example, in fluid dynamics with lattice Boltzmann methods [2, 3], fluids in heterogeneous porous media [4], in genetics [7], dune dynamics [5], spatial pattern formation [6], and many others [8]. Cellular automata can be seen as an idealization of a physical system in which space, time, and certain physical quantities take a finite set of values. Cellular automata provide simple models of complex systems showing that collective complex behavior can emerge from the composition or interaction of simple components. Even if the local interactions are perfectly described in a direct manner, it is possible that the global behavior of a system obeys unexpected patterns. This fact makes CA suitable to model and simulate non-equilibrium systems. In the 1980s, Wolfram [9, 10] gave a classification of cellular automata which produces an intuitive way to distinguish the dynamical behavior of cellular automata in four distinct classes, accordingly: Class 1: almost every initial conditions produce an eventually fixed point behavior. Class 2: almost every initial conditions produce an eventually periodic behavior. Class 3: almost every initial conditions produce a pseudo-random behavior. Class 4: almost every initial conditions produce a complex behavior articulating regular patterns with structured non-periodic geometric patterns.

C. Ramos (✉) · F. Carapau · P. Correia

Departamento de Matemática and CIMA-UE, Universidade de Évora, Évora, Portugal
e-mail: ccr@uevora.pt; flc@uevora.pt; pcorreia@uevora.pt

A very detailed comparison of the CA methods, for practical fluid-dynamics problems, with conventional methods from numerical analysis is explained in [11], in particular, with details from a computational point of view, considering memory usage, computational time, and other characteristics.

Nevertheless, most of the interest in the use of cellular automata focuses on non-equilibrium fluids or fluids composed with different phases, in which the differential equations are hard to implement.

In [12] were introduced and discussed several techniques to explore evolutionary dynamics of the automata space, using biologically motivated concepts. In particular, specific genetic algorithms and techniques such as mutation, assembly, and recombination of CA. In that context the code rule of a CA was called the genotype, and the diverse characteristics of generic CA realizations were called phenotype. Here, these denominations are changed and adapted to the present context. The advantage of evolutionary methods is to efficiently obtain CA rules with specific characteristics. Previous work on evolutionary search over cellular automata can be found also in [13, 14].

In this paper are presented techniques for modeling systems, seen as idealized fluids, where may coexist distinct substances in diverse phases. These techniques, using cellular automata, are suitable to simulate transient, non-equilibrium behavior in fluid mechanics or other phenomena, such as fracture dynamics on heterogeneous materials.

Our main result is the development of the assembly method, introduced in [12], to determine CA code rules of increasing complex behavior. This means that the systems present an increasing number of distinct behavior and spatial-temporal patterns. A canonical process of assembling two CA rules is defined. This method allows the study of the singular perturbation of a complex fluid and the study of the interaction between two similar fluids subject to instabilities, leading to global phase transitions.

In Sect. 2 the notions and concepts used in the paper regarding cellular automata are introduced, in particular those notions from [12], such as the singular perturbation, assembly, and the canonical assembly. The basic CA rule $3E6IGSS8S$, which is used in the simulations, is also defined. In Sect. 3, the computation of the canonical assembly of the CA rule $3E6IGSS8S$, its variations, and the simulations of its perturbations are presented.

2 Preliminaries and Definitions

Some notions regarding one dimension cellular automata are here introduced. Let $\mathbb{Z}_n = \{0, 1, 2, \dots, n-1\}$, $n > 0$, be the *local state space*. Let $\phi : \mathbb{Z}_n^m \rightarrow \mathbb{Z}_n$ be a map, which determines the local dynamics of the system and is called *local map* or *CA rule*. An element in \mathbb{Z}_n^m , i.e., a word or a block of size m in the alphabet \mathbb{Z}_n , is called a *local configuration*. The map ϕ induces a block map ϕ_k , $k \in \mathbb{N}$, which transforms words in \mathbb{Z}_n , of size $m+k$, into words of size k , through

$$\phi_k : \mathbb{Z}_n^{m+k} \rightarrow \mathbb{Z}_n^k,$$

$$\phi_k(x_1 \dots x_{m+k}) := \phi(x_1 \dots x_m) \phi(x_1 \dots x_m) \dots \phi(x_1 \dots x_m).$$

There is a natural identification of ϕ_1 with ϕ . To simplify the exposition, consider m to be an odd number so that $m = 2r + 1$, for a certain natural number r . The global map is then defined by

$$\Phi : \mathbb{Z}_n^{\mathbb{I}} \rightarrow \mathbb{Z}_n^{\mathbb{I}},$$

$$\Phi(x) := (\phi(x_{[j-r, j+r]}))_{j \in \mathbb{I}},$$

where \mathbb{I} can be \mathbb{Z} , \mathbb{N} or a finite set $\mathbb{Z}_L = \{1, 2, \dots, L\}$. A *cellular automaton* is the specification of the number of local states n , the size of the local configuration m , the local map or CA rule ϕ , the configuration space or global state space \mathbb{I} , and if needed, the boundary conditions which depend naturally on \mathbb{I} and m . The time evolution of the system is given by the iteration of the map Φ , given an initial condition $x(0) = (x_i(0))_{i \in \mathbb{I}} \in \mathbb{Z}_n^{\mathbb{I}}$,

$$\begin{aligned} x(t+1) &= \Phi(x(t)), \quad t \geq 0, \\ x(0) &= (x_i(0))_{i \in \mathbb{I}} \in \mathbb{Z}_n^{\mathbb{I}}. \end{aligned}$$

The parameter $m = 2r + 1$ gives the dependence of each state, in the next time instant on the states of the neighbor cells, r cells to the left and r cells to the right. In the case \mathbb{I} is \mathbb{N} or a finite set, it is necessary to specify boundary conditions, on the left in the first case and both left and right in the second case. For convenience, $[j]_n$ denote the n -expansion of the natural number j , that is, j in base n . By convention, the number of digits in $[j]_n$ is fixed and equal to n^m . That is, if $[j]_n = j_1 \dots j_r$ then

$$j = j_1 \times n^{r-1} + j_2 \times n^{r-2} + \dots + j_{r-1} \times n^1 + j_r \times n^0.$$

On the other hand, a word $j_1 \dots j_r$, in \mathbb{Z}_n^r with $r \geq 1$, can be seen as a representation of a natural number $j \in \mathbb{N}$, in base n , denoted by $\langle j_1 \dots j_r \rangle_n \in \mathbb{N}$. With this notation

$$\begin{aligned} j_1 \dots j_r \in \mathbb{Z}_n^r &\rightarrow \langle j_1 \dots j_r \rangle_n = j \in \mathbb{N}, \\ j \in \mathbb{N} &\rightarrow [j]_n = j_1 \dots j_r \in \mathbb{Z}_n^r. \end{aligned}$$

Once fixed the value m and the configuration space \mathbb{I} (and eventually the boundary condition), a cellular automaton is completely characterized specifying a sequence $\alpha = (\alpha_1, \dots, \alpha_{n^m}) \in \mathbb{Z}_n^{n^m}$ corresponding to the sequence of the images of every local configuration under ϕ . This sequence is called *CA code rule* and is a *functional representation* of the CA, that is, a particular symbol in a certain position in the referred sequence has a functional meaning. The position j in the sequence α

gives a configuration which is the n -expansion of the integer $(j - 1)$ and the value α_j is the image of that configuration under the rule ϕ , that is, $\alpha_j = \phi [j - 1]_n$. Therefore, the CA code rule is

$$\alpha = (\phi [j - 1]_n)_{j=1}^{n^m}.$$

A more compact way to give a particular CA code rule is to use the *Wolfram numbering*. The CA code rule is seen as the expansion in base n of a certain number which when converted to decimal is designated as the Wolfram number of the CA code rule. If $n < 10$ the number of digits of the Wolfram number is less than the number of digits corresponding to the original CA code rule; therefore it is a more compact way of specifying the CA rule. An even more compact form is to use hexadecimal base (if the number of states is less than 16), or a larger base number. Since we deal with very large CA-code rules, we will use base 32-expansion to represent the CA code rules in compact way. The base 32, similarly to base 16, uses the digit set

{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V}.

As an illustrative example consider the CA code rule 01110110 which determines the local map

```

000 001 010 011 100 101 110 111
  0   1   1   1   0   1   1   0
    
```

The sequence 01110110 represents a number in binary. The corresponding number in decimal base is $110 = 0 \times 2^0 + 1 \times 2^1 + 1 \times 2^2 + 1 \times 2^3 + 0 \times 2^4 + 1 \times 2^5 + 1 \times 2^6 + 0 \times 2^7$ (note the reversed order). In hexadecimal the rule 110 is designated by 6E, and in base 32 is 3E. See Figs. 1 and 2.

To resume, a CA code rule will be a sequence $\alpha = \alpha_1 \alpha_2 \dots \alpha_{n^m} \in \mathbb{Z}_n^{n^m}$, with $n \in \mathbb{N}$. The space of the CA code rules is denoted by \mathcal{G} . The space of CA code rules which have n different symbols is denoted by \mathcal{G}_n , and the space of CA code rules which have n different symbols and with neighbor number equal to m is denoted by $\mathcal{G}_{n,m}$.

The cellular automaton which is central in this work is a 3-state rule, with $m = 3$ and $\mathbb{I} = \mathbb{Z}_L$, for a certain natural L . The CA code rule is

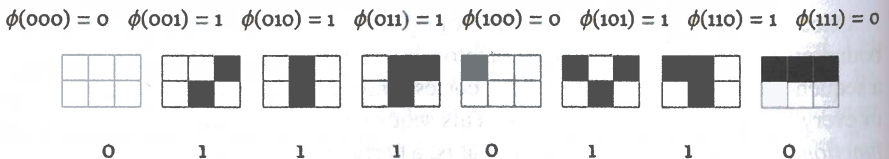
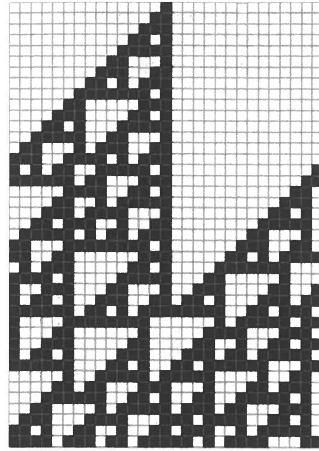


Fig. 1 Explicit CA code rule 110, in Wolfram numbering

Fig. 2 Example of a realization of the automaton rule 110



$$\alpha = 202000211011010222222101111.$$

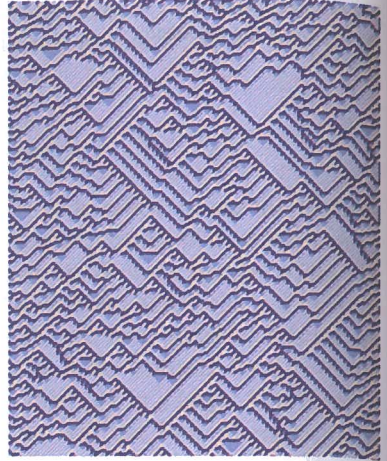
The local map in $\mathbb{Z}_3 = \{0, 1, 2\}$, is then defined by

000	001	002	010	011	012	020	021	022
↓	↓	↓	↓	↓	↓	↓	↓	↓
2	0	2	0	0	0	2	1	1
100	101	102	110	111	112	120	121	122
↓	↓	↓	↓	↓	↓	↓	↓	↓
0	1	1	0	1	0	2	2	2
200	201	202	210	211	212	220	221	222
↓	↓	↓	↓	↓	↓	↓	↓	↓
2	2	2	1	0	1	1	1	1

The corresponding natural number (in decimal base), Wolfram number, is

$$\begin{aligned}
 3786635351324 &= 2 \times 3^0 + 0 \times 3^1 + 2 \times 3^2 + 0 \times 3^3 + 0 \times 3^4 + 0 \times 3^5 \\
 &+ 2 \times 3^6 + 1 \times 3^7 + 1 \times 3^8 + 0 \times 3^9 + 1 \times 3^{10} + 1 \times 3^{11} \\
 &+ 0 \times 3^{12} + 1 \times 3^{13} + 0 \times 3^{14} + 2 \times 3^{15} + 2 \times 3^{16} + 2 \times 3^{17} \\
 &+ 2 \times 3^{18} + 2 \times 3^{19} + 2 \times 3^{20} + 1 \times 3^{21} + 0 \times 3^{22} + 1 \times 3^{23} \\
 &+ 1 \times 3^{24} + 1 \times 3^{25} + 1 \times 3^{26}.
 \end{aligned}$$

Fig. 3 Realization of CA rule 3E6IG558S, with random initial conditions



Note that the same number in base 2 is the CA code rule in reversed order, that is,

$$111101222222010110112000202_{\text{binary}} = 3786635351324_{\text{decimal}}.$$

In a more compact description, its hexadecimal representation is 371A50E151C and in 32-base is 3E6IG558S. This last representation will be chosen to refer the CA rule, since it is shorter. In Fig. 3 we present an example of a realization with initial global state given by a random vector $x_0 \in \mathbb{Z}_3^{150}$.

2.1 Singular Perturbation and Pattern Stability

A singular perturbation of the CA rule is a transformation in a single symbol of the CA code rule, and it is the simplest possible transformation defined on the rule space $\mathcal{G}_{n,m}$. This perturbation can be generated randomly or generated by a deterministic process. To give a singular perturbation, it is necessary to specify the position in the CA code rule where the mutation is to occur and how it occurs. Recall that a position j in the sequence α gives a configuration which is the n -expansion of the integer $(j-1)$, that is, $[j-1]_n$, ($00\dots00$ is the configuration of the position 1), and the value α_j is the image of the configuration under the automaton ϕ , that is, $\alpha_j = \phi[j-1]_n$, $j = 1, \dots, n^m$.

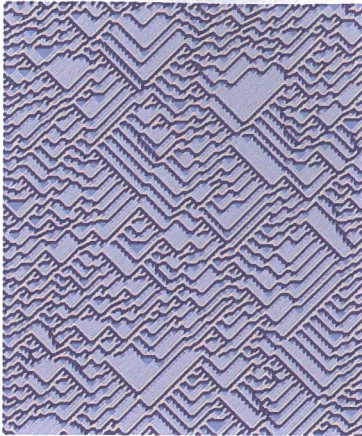
Now, consider the stability of the patterns produced by time evolution of an initial condition, with respect to singular perturbation. There are several cellular automata which are very robust under singular perturbation, regarding the geometric structure of the patterns produced, and others very sensitive. However, some CA are robust to singular perturbation in some positions and in other positions are strongly sensitive. As an example of this phenomena, see the Figs. 4, 5, 6, 7, and 8. The same CA code

position 23 $0 \rightarrow 1$

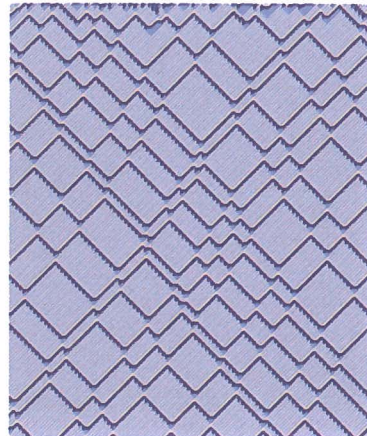


2020002110110102222221 0 1111

2020002110110102222221 1 1111



(a)



(b)

Fig. 4 (a) Realization of the CA rule 3E6IGS58S. (b) Realization of a perturbation of the CA rule 3E6IGS58S at position 23

rule 3E6IGS58S is singularly perturbed in different positions, and for each case a realization of the CA is obtained, with random initial conditions. The realizations show the similarity of some of the mutated CA codes and the drastic changes in others.

2.2 Assembly of CA Code Rules

Next, it is described the assembly technique which produces CA rules obtained from two given CA rules. The assembled CA rule inherits several characteristics from the original rules; in particular, it maintains the original CA as subcases for special initial conditions. Let $\alpha = \alpha_1 \dots \alpha_{p^m} \in \mathcal{G}_{p,m}$ and $\beta = \beta_1 \dots \beta_{q^m} \in \mathcal{G}_{q,m}$ be two CA code rules with $p, q, m \in \mathbb{N}$. The alphabet underlying $\mathcal{G}_{p,m}$ is, as usual \mathbb{Z}_p , and for $\mathcal{G}_{q,m}$ is \mathbb{Z}_q . The assembly of α with β is a general procedure which gives a class of CA code rules in the space $\mathcal{G}_{n,m}$, where $n = p + q$.

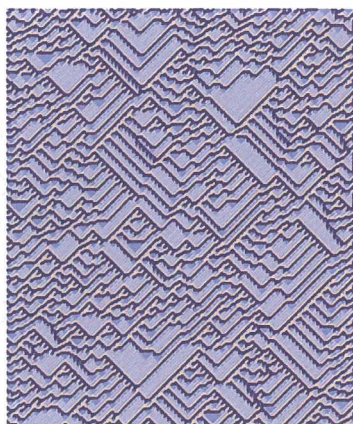
The first p symbols of \mathbb{Z}_n are reserved to codify the rule α and the last q symbols of \mathbb{Z}_n to codify the rule β , using the correspondence

position 25, 1→0

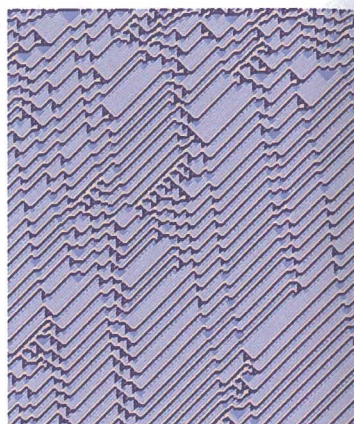


202000211011010222222101 1 11

202000211011010222222101 0 11



(a)



(b)

Fig. 5 (a) Realization of the CA rule 3E6IGS58S. (b) Realization of a perturbation of the CA rule 3E6IGS58S at position 23, with random initial conditions

$$\begin{array}{ccc}
 \mathbb{Z}_p & 0 \dots p-1 & \mathbb{Z}_q & 0 \dots q-1 \\
 \downarrow & \downarrow & \downarrow & \downarrow \\
 \mathbb{Z}_n & 0 \dots p-1 & \mathbb{Z}_n & p \dots p+q
 \end{array}
 \text{ and }$$

Denote the correspondence $\hat{\cdot} : \mathbb{Z}_q \rightarrow \{p, \dots, p+q\} \subset \mathbb{Z}_n$ and the reversed correspondence $\tilde{\cdot} : \{p, \dots, p+q\} \rightarrow \mathbb{Z}_q$. Note that $\hat{\cdot}$ can be seen as adding p to each symbol if each symbol is seen as a natural number. For each local configuration $i_1 \dots i_m \in \mathbb{Z}_p^m$, corresponding to the CA code rule $\alpha \in \mathcal{G}_{p,m}$, it is associated the same configuration (with the same symbols) in \mathbb{Z}_n^m . To each configuration $j_1 \dots j_m \in \mathbb{Z}_q^m$, corresponding to the CA code rule $\beta \in \mathcal{G}_{q,m}$ it is associated the configuration $\hat{j}_1 \dots \hat{j}_m$ in \mathbb{Z}_n^m . This gives a large number of degrees of freedom to choose the image of the local map associated with configurations which mix symbols from $\{0, \dots, p-1\}$ and $\{p, \dots, p+q\}$. This means that there are many different CA code rules arising from assembly of two specific CA code rules α, β .

Let $\phi_\alpha, \phi_\beta, \phi_\gamma$ denote the local rules for each CA code rule α, β, γ . Then γ is a CA code rule assembly of α, β if the following property is Satisfied:

$$x_1 \dots x_m \in \{0, \dots, p-1\}^m \Rightarrow \phi_\gamma(x_1 \dots x_m) = \phi_\alpha(x_1 \dots x_m),$$

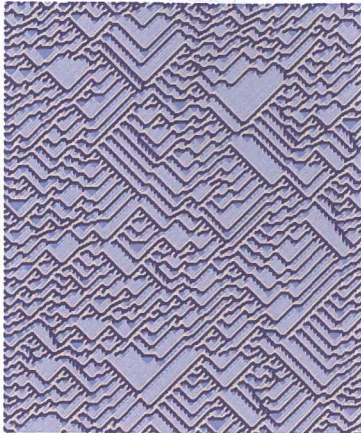
$$x_1 \dots x_m \in \{p, \dots, p+q\}^m \Rightarrow \phi_\gamma(x_1 \dots x_m) = \phi_\beta(\tilde{x}_1 \dots \tilde{x}_m).$$

position 1, 2→1



2 02000211011010222222101111

1 02000211011010222222101111



(a)



(b)

Fig. 6 (a) Realization of the CA rule 3E6IGS58S. (b) Realization of a perturbation of the CA rule 3E6IGS58S at position 23

The local configurations for ϕ_γ with digits exclusively from $\{0, \dots, p - 1\}$ or exclusively $\{p, \dots, p + q\}$ are called *pure local configurations*; the local configurations mixing digits from $\{0, \dots, p - 1\}$ and $\{p, \dots, p + q\}$ are called *mixed local configurations*. The images under ϕ_γ of pure local configurations in $\{0, \dots, p - 1\}$ are determined by ϕ_α , and the images under ϕ_γ of pure local configurations in $\{p, \dots, p + q\}$ are determined by ϕ_β . The images under ϕ_γ of the mixed configurations are not determined by α, β . Therefore, must be as external parameters or degrees of freedom. As an example, consider the CA code rules $\alpha \in \mathcal{G}_{2,3}$, rule 18, and $\beta \in \mathcal{G}_{2,3}$, rule 110, given by

$$\alpha = 01001000 \quad \text{and} \quad \beta = 01110110.$$

The second CA code rule, β , is transformed via $0 \rightarrow \widehat{0} = 2$ and $1 \rightarrow \widehat{1} = 3$ into

$$\widehat{\beta} = 23332332.$$

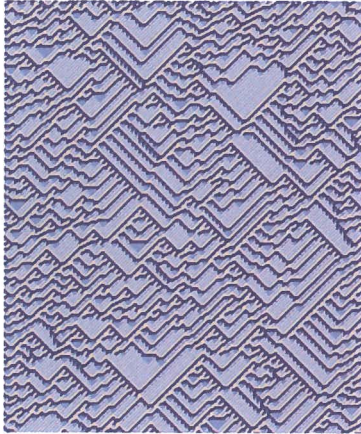
Note that the cellular automata β and $\widehat{\beta}$ are equivalent, although the symbols are distinct; therefore, the two automata are identified $\beta \longleftrightarrow \widehat{\beta}$.

position 1, 2 → 0



2 02000211011010222222101111

0 02000211011010222222101111



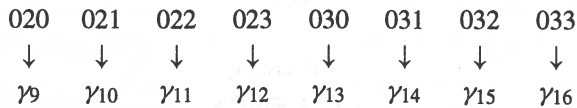
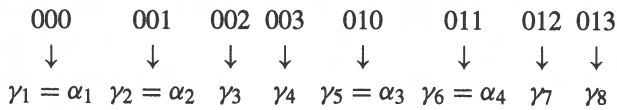
(a)



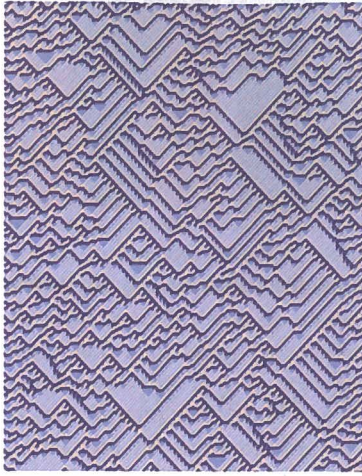
(b)

Fig. 7 (a) Realization of the CA rule 3E6IG558S. (b) Realization of a perturbation of the CA rule 3E6IG558S at position 23

Consider a CA code rule $\gamma \in \mathcal{G}_{4,3}$ obtained by assembly of α and β . Therefore, corresponds to a cellular automaton that when restricted to initial conditions (and boundary conditions) with states 0, 1 will reproduce the exact patterns of α and when restricted to initial conditions (and eventual boundary conditions) with states 3, 4 will reproduce the patterns of β (up to the transformation $0 \rightarrow 3, 1 \rightarrow 4$). A CA code rule γ satisfying this property is called the assembly of α and β . There are many different CA code rules arising from assembly. The local map in $\mathbb{Z}_4 = \{0, 1, 2, 3\}$, for a rule $\gamma \in \mathcal{G}_{4,3}$, assembly of α and β has the following structure:



2020002110110102 2 22 2 2101 1 11
 2020002110110102 1 22 0 2101 2 11



(a)



(b)

Fig. 8 (a) Realization of the CA rule 3E6IGS58S. (b) Realization of a perturbation of the CA rule 3E6IGS58S at position 23

100	101	102	103	110	111	112	113
↓	↓	↓	↓	↓	↓	↓	↓
$\gamma_{17} = \alpha_5$	$\gamma_{18} = \alpha_6$	γ_{19}	γ_{20}	$\gamma_{21} = \alpha_7$	$\gamma_{22} = \alpha_8$	γ_{23}	γ_{24}

120	121	122	123	130	131	132	133
↓	↓	↓	↓	↓	↓	↓	↓
γ_{25}	γ_{26}	γ_{27}	γ_{28}	γ_{29}	γ_{30}	γ_{31}	γ_{32}

200	201	202	203	210	211	212	213
↓	↓	↓	↓	↓	↓	↓	↓
γ_{33}	γ_{34}	γ_{35}	γ_{36}	γ_{37}	γ_{38}	γ_{39}	γ_{40}

220	221	222	223	230	231	232	233
↓	↓	↓	↓	↓	↓	↓	↓
γ_{41}	γ_{42}	$\gamma_{43} = \hat{\beta}_1$	$\gamma_{44} = \hat{\beta}_2$	γ_{45}	γ_{46}	$\gamma_{47} = \hat{\beta}_3$	$\gamma_{48} = \hat{\beta}_4$

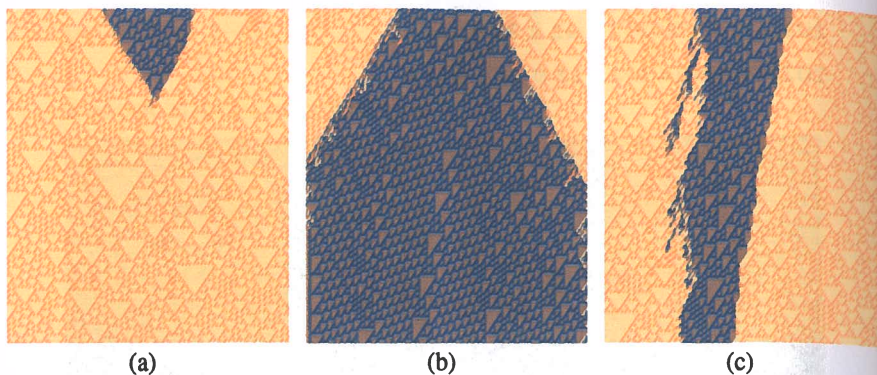


Fig. 9 Distinct assembly of rule 18 with rule 110: (a) Realization for $\gamma^{(1)}$. (b) Realization for $\gamma^{(2)}$. (c) Realization for $\gamma^{(3)}$

300	301	302	303	310	311	312	313
↓	↓	↓	↓	↓	↓	↓	↓
γ_{49}	γ_{50}	γ_{51}	γ_{52}	γ_{53}	γ_{54}	γ_{55}	γ_{56}

320	321	322	323	330	331	332	333
↓	↓	↓	↓	↓	↓	↓	↓
γ_{57}	γ_{58}	$\gamma_{59} = \widehat{\beta}_5$	$\gamma_{60} = \widehat{\beta}_6$	γ_{61}	γ_{62}	$\gamma_{63} = \widehat{\beta}_7$	$\gamma_{64} = \widehat{\beta}_8$

In Fig. 9 are shown three realizations of distinct CA-rules in $\mathcal{G}_{4,3}$ arising from the assembly of α and β . From left to right are denoted by $\gamma^{(1)}$, $\gamma^{(2)}$, and $\gamma^{(3)}$. The initial conditions are composed by two segments with random initial conditions from $\{0, 1\}$, and the middle segment is generated randomly from $\{2, 3\}$. The difference between the rules $\gamma^{(1)}$, $\gamma^{(2)}$ and $\gamma^{(3)}$ are that for $\gamma^{(1)}$ the values of the rule for mixed local configurations are taken randomly only from $\{0, 1\}$ which means that the patterns arising from α dominate. For $\gamma^{(2)}$ the values of the rule for mixed local configurations are taken randomly only from $\{2, 3\}$ which means that the patterns arising from β dominate. Finally, for $\gamma^{(3)}$ the values of the rule for mixed local configurations are taken randomly only from $\{0, 1, 2, 3\}$ with equal probability, which means that the initial patterns arising from α and β mix and interact along the time flow.

The examples shown in Fig. 9, with $\alpha = 01001000$ and $\beta = 01110110$, correspond to

$$\gamma^{(1)} = 0112001120110101101100200111012000101000022311331001101202232132,$$

with a majority of states for mixed configurations taken randomly from $\{0, 1\}$,

$$\gamma^{(2)} = 0122001322123301103200332232312232321203022331331203321232232132,$$

with a majority of states for mixed configurations taken randomly from $\{2, 3\}$,

$$\gamma^{(3)} = 0132003130120123101100230113312000201000022313331023103232232132,$$

with an equilibrium of states for mixed configurations taken randomly with probability $1/2$ from $\{0, 1\}$ and $\{2, 3\}$.

2.3 Canonical Assembly of a CA Rule

Consider now a process to assembly two copies of the same CA rule. In this case, it is possible to turn the assembly uniquely determined, that is, not depending on externally given parameters. Therefore, this process is called *canonical assembly*. The canonical assembly can be viewed as the embedding of a particular system in a larger one containing two copies of the original system. This process is particularly important if it is necessary to model a system in non-equilibrium which is transforming and exhibiting new patterns of behavior although maintaining others. This can be achieved allowing singular perturbations after a canonical assembly, as it is seen in the next section.

Let $\alpha \in \mathcal{G}_{p,m}$ and $n = 2p$. Consider the state transformations

$$\widehat{\cdot}: \mathbb{Z}_n \rightarrow \mathbb{Z}_n \text{ and } \widetilde{\cdot}: \mathbb{Z}_n \rightarrow \mathbb{Z}_n,$$

with

$$\widehat{0} = p, \widehat{1} = p+1, \dots, \widehat{p-1} = 2p-1, \quad \widetilde{p} = p, \widetilde{p+1} = p+1, \dots, \widetilde{2p-1} = 2p-1,$$

and

$$\widetilde{0} = 0, \widetilde{1} = 1, \dots, \widetilde{p-1} = p-1, \quad \widehat{p} = 0, \widehat{p+1} = 1, \dots, \widehat{2p-1} = p-1.$$

Note that $\widehat{\mathbb{Z}}_n = \{p, p+1, \dots, 2p-1\}$ and $\widetilde{\mathbb{Z}}_n = \mathbb{Z}_p = \{0, 1, \dots, p-1\}$. Moreover, $\widetilde{\widehat{x}} = x$ and $\widehat{\widetilde{x}} = x$.

Let $N_X(i_1 i_2 \dots i_m)$ be the number of digits in $i_1 i_2 \dots i_m$ belonging to a certain subset $X \subset \mathbb{Z}_n$. Let

$$\chi(i_1 i_2 \dots i_m) = \begin{cases} 0 & \text{if } N_{\mathbb{Z}_p}(i_1 i_2 \dots i_m) > r, \\ 1 & \text{if } N_{\mathbb{Z}_p}(i_1 i_2 \dots i_m) \leq r. \end{cases}$$

Recall that $r = m/2 - 1$. The condition above simply determines if the number of digits in $i_1 i_2 \dots i_m$ belonging to \mathbb{Z}_p is larger than the number of digits in $\{p, p+1, \dots, n-1\}$, with $n = 2p$. Then the canonical assembly of α produces a CA code rule $\gamma = (\gamma_k)_{k=1, \dots, n^m}$ with

$$\gamma_{\langle i_1 \dots i_m \rangle_n} = \begin{cases} \alpha_{\langle \tilde{i}_1 \dots \tilde{i}_m \rangle_p} & \text{if } \chi(i_1 i_2 \dots i_m) = 0, \\ \beta_{\langle \tilde{i}_1 \dots \tilde{i}_m \rangle_p} & \text{if } \chi(i_1 i_2 \dots i_m) = 1. \end{cases}$$

where $\beta = \hat{\alpha} = (\hat{\alpha}_k)_{k=1, \dots, p^m}$. Recall that $\langle i_1 \dots i_m \rangle_n$ is the position number associated to the local configuration $i_1 \dots i_m$ in base n . The number $\langle \tilde{i}_1 \dots \tilde{i}_m \rangle_p$ is the position number associated with the local configuration $\tilde{i}_1 \dots \tilde{i}_m$ in base p , since the transformation $\tilde{}$ sends \mathbb{Z}_n to \mathbb{Z}_p . The canonical assembly produces a CA rule which in practical terms, reproduces two copies of the same CA with the duplication of the number of states.

3 Case Study: Rule 3E6IGS58S

The CA rule 3E6IGS58S (see Fig. 3) is seen as an idealized fluid where two substances which do not mix easily and two different phases of one of the substances are in unstable equilibrium. The states 0, 1 (lighter colors) are seen as the same substance in a different phase, and the state 2 (darker color) is a different substance. This phenomenon reflects on the persistency of the local state 2 in refined geometric structures and on the interaction between states 0, 1 which interchanges in a complex way.

Through the general process of the canonical assembly applied to the CA rule 3E6IGS58S, it is obtained a CA rule which models a system composed of two fluids of the same type. Moreover, the perturbation of the CA rule leads to complex behavior, where the realizations of the CA rule present the patterns of the original fluids and the patterns arising from the perturbation. In particular, the singular perturbations considered in Sect. 2.1, for the individual fluid, are applied.

3.1 Canonical Assembly of Rule 3E6IGS58S

Let $p = 3$, consider the 3-state CA rule 3E6IGS58S, given by

$$\alpha = 20200021101101022222101111,$$

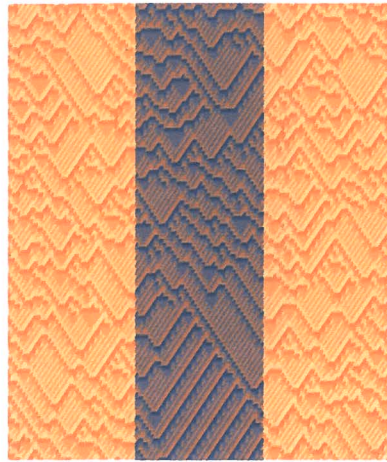
and let

$$\beta = \hat{\alpha} = 53533354434434355555434444,$$

which is obtained from α adding 3 to each symbol, as explained in the assembly section. Now, consider the canonical assembly of α , with

$$\hat{0} = 2, \hat{1} = 3, \hat{2} = 5, \hat{3} = 3, \hat{4} = 4, \hat{5} = 5,$$

Fig. 10 Realization for the canonical assembly of the rule with structured initial conditions: three segments randomly generated from {0, 1, 2}, from {3, 4, 5}, and again from {0, 1, 2}. Random initial conditions



and

$$\tilde{0} = 0, \tilde{1} = 1, \tilde{2} = 2, \tilde{3} = 0, \tilde{4} = 1, \tilde{5} = 2.$$

The CA code rule is, represented in base 32,

```
85980E44A81JS1KVBGVUQ5KDRJ7UP0JUBPL9CBFFBDIPFH1669SA00CU2
...MLT0B3AI26QSATJCN6LO7PKRTSIC2QFB180IQJTCQAFIUC6CEOSMSRI.
```

In Fig. 10 is shown a realization of the canonical assembly of the rule, exhibiting the coexistence of the two fluids in similar regimes.

3.2 Perturbations of the Canonical Assembly

In Fig. 11 it is shown the singular perturbations of the canonical assembly of the CA rule *3E6IGS58S* in which there is a singular perturbation in the local configurations: 333, 5 \mapsto 4, 544, 3 \mapsto 4, 554, 4 \mapsto 3, showing the coexistence of the patterns of the original fluid and the patterns arising from the perturbed CA, from Sect. 2.1, Figs. 4, 5, and 6.

Finally, consider the assembly of two CA code rules with randomly chosen values for the mixed configurations. In this case, the original patterns are maintained, as long as the initial condition is restricted to pure local configuration states. If the initial conditions mix states, then there is a complex interaction between the two fluids and the patterns arising from the original CA.

This method produces CA rules which have unstable equilibrium between the two coexisting fluids as is shown in Fig. 12.

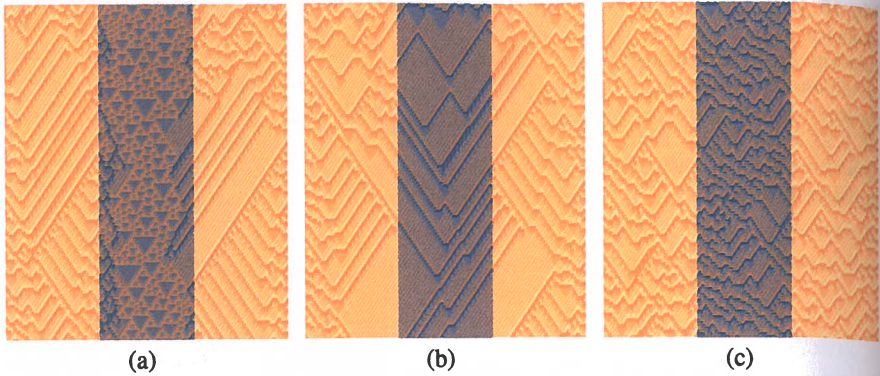


Fig. 11 (a) Perturbation of the canonical assembly of rule on the local configuration 333 as in the figure (in this case local configuration 000). (b) Perturbation of the canonical assembly of rule on the local configuration 544 as in the figure (in this case local configuration 211). (c) Perturbation of the canonical assembly of rule on the local configuration 554 as in the figure (in this case local configuration 221)

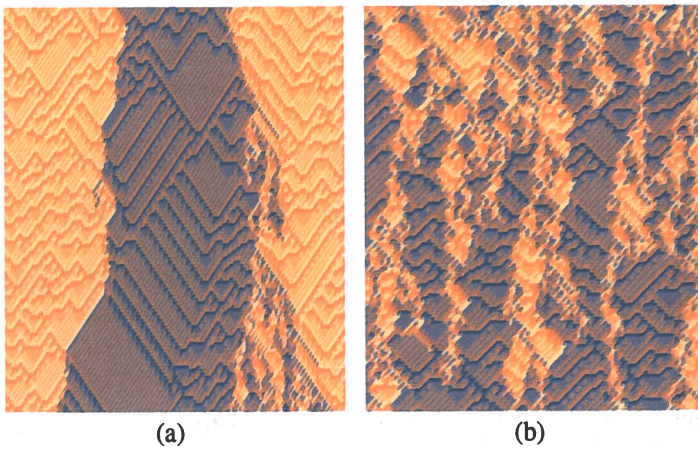


Fig. 12 (a) Realization for γ_4 , with structured initial conditions: three segments randomly generated from $\{0, 1, 2\}$, from $\{3, 4, 5\}$, and again from $\{0, 1, 2\}$. (b) Realization for γ_5 , random initial conditions

4 Conclusions and Further Developments

In the present paper techniques are developed for modeling idealized fluids where coexist distinct substances in different phases. These techniques, based on cellular automata, are appropriate to simulate transient and non-equilibrium behavior in fluid mechanics. The main result is the development of the canonical assembly method which allows the determination of CA code rules with complex behavior, obtained from given initial CA rules. The systems subject to assembly present an increasing

number of distinct behavior and spatial-temporal patterns, maintaining, for certain initial conditions, the original patterns. Several families of CA, associated with idealized fluid substances, are considered. The canonical assembly method allows the study of small perturbations of a complex fluid and the study of the interaction between two similar fluids subject to instabilities. It is clear that the instabilities depend on the particular rules. The considered rules are sensitive to certain singular perturbations and not to other. The systematic study of the perturbations of these rules will be considered in future work, aiming a complete classification of its behavior.

Acknowledgments This work has been partially supported by Centro de Investigação em Matemática e Aplicações da Universidade de Évora (CIMA-UE), through the grand UIDB/04-674/2020 of FCT-Fundação para a Ciência e a Tecnologia, Portugal.

References

1. J. von Neumann, *Theory of Self-reproducing Automata* (University of Illinois Press, Urbana, 1966)
2. B. Chopard, A. Masselot, Cellular automata and lattice Boltzmann methods: a new approach to computational fluid dynamics and particle transport. *Future Gener. Comput. Syst.* **16**, 249–257 (1999)
3. G. Doolen, *Lattice Gas Methods For Partial Differential Equations* (CRC Press, Boca Raton, 2019)
4. D. Kristanto, W. Paradhita, Simulation study of fluid flow and estimation of a heterogeneous porous media properties using lattice gas automata method. *J. Pet. Geotherm. Technol.* **1**(2), 71–82 (2020)
5. C. Narteau, D. Zhang, O. Rozier, P. Claudin, Setting the length and time scales of a cellular automaton dune model from the analysis of superimposed bed forms. *J. Geophys. Res.* **114**, F03006 (2009)
6. T. Suzudo, Spatial pattern formation in asynchronous cellular automata with mass conservation. *Phys. A* **343**, 185–200 (2004)
7. G.Ch. Sirakoulisa, I. Karafyllidisb, Ch. Mizasa, V. Mardirisa, A. Thanailakisb, Ph. Tsalidesb, A cellular automaton model for the study of DNA sequence evolution. *Comput. Biol. Med.* **33**, 439–453 (2003)
8. T. Bossomaier, D. Green, *Complex Systems* (Cambridge University Press, Cambridge, 2000)
9. S. Wolfram, Statistical mechanics of cellular automata. *Rev. Mod. Phys.* **55**(3), 601–644 (1983)
10. S. Wolfram, *Cellular Automata and Complexity* (Addison-Wesley, New York, 1994)
11. H. Abarbanel et.al, *Cellular Automata and Parallel Processing for Practical Fluid-Dynamics Problems*, Report JSR-86-303 (1990)
12. C. Ramos, M. Riera, *Evolutionary Dynamics and the Generation of Cellular Automata*. Iteration theory (ECIT '08), Grazer Math. Ber., 354 (Institut für Mathematik, Karl-Franzens-Universität Graz, Graz, 2009), pp. 219–236
13. M. Mitchell, P. Hraber, J. Crutchfield, Revisiting the edge of chaos: evolving Cellular automata to perform computations. *Complex Syst.* **7**, 89–130 (1993)
14. M. Mitchell, P. Hraber, J. Crutchfield, Evolving Cellular automata to perform computations: mechanisms and impediments. *Phys. D* **75**, 361–391 (1994)