

On the iota-delta function: a link between cellular automata and partial differential equations for modeling advection–dispersion from a constant source

Luan Carlos de S. M. Ozelim¹ ·
André Luís B. Cavalcante² · Jan M. Baetens³

Published online: 28 June 2016
© Springer Science+Business Media New York 2016

Abstract Describing complex phenomena by means of cellular automata (CAs) has shown to be a very effective approach in pure and applied sciences. Most of the applications, however, rely on multidimensional CAs. For example, lattice gas CAs and lattice Boltzmann methods are widely used to simulate fluid flow and both share features with two-dimensional CAs. One-dimensional CAs, on the other hand, seem to have been neglected for modeling physical phenomena. In the present paper, we demonstrate that some one-dimensional CAs are equivalent to a stable linear finite difference scheme used to solve advection–diffusion partial differential equations (PDEs) by relying on the so-called iota-delta representation. Consequently, this work shows an important link between continuous and discrete models in general, and PDEs and CAs more in particular.

Keywords Cellular automata · Advection–dispersion iota delta function · Advection · Diffusion

✉ Luan Carlos de S. M. Ozelim
luanoz@gmail.com

André Luís B. Cavalcante
abrasil@unb.br

Jan M. Baetens
jan.baetens@ugent.be

¹ Department of Civil Engineering, University of Brasilia, Brasília 70910-900, Brazil

² Department of Civil and Environmental Engineering, University of Brasilia, Brasília 70910-900, Brazil

³ KERMIT, Department of Mathematical Modelling, Statistics and Bioinformatics, Ghent University, Coupure links 653, 9000 Ghent, Belgium

1 Introduction

Complexity inevitably arises in nature [1]. Possible ways of addressing and understanding this are of great interest to the scientific community [2]. Moreover, as the amount of available computing power grew during the past three decades, the study of dynamical systems has intensified considerably. Amongst other things, the pioneering work by John von Neumann and Stanislaw Ulam shows that a special class of the latter, so-called cellular automata (CAs), can give rise to rich and physically justifiable behavior [1]. This was further confirmed by Wolfram [3], who established the basis for the study of one-dimensional CAs [1].

The applicability of CAs to describe physical systems quickly drew attention of scientists in the 1970s and early 1980s, culminating in the development of the HPP model [4]. Yet, this model lacked Galilean invariance, and presented statistical noise, which ultimately led to lack of isotropy [5].

Some works built upon and improved the HPP model and about 10 years later the FHP model was put forward by Frisch et al. [6]. Both the HPP and FHP models belong to the class of the so-called lattice gas CAs (LGCAs). Independently, Wolfram [7] proposed another approach to avoid some of the shortcomings of the HPP model. Still, few problems could not be overcome by relying on the FHP model (such as noise), which ultimately led to the development of the Lattice Boltzmann method [5]. In an early work, Frisch et al. [8] already had implemented Boltzmann distributions to evaluate the viscosity of LGCA schemes. The noise related to the Boolean nature of LGCAs, however, remained an issue. McNamara and Zanetti [9] presented for the first time the Lattice Boltzmann method as an autonomous numerical scheme to conduct hydrodynamical simulations. These automata-related approaches are models whose interaction rules mimic fluid-like behavior. When spatially averaging the evolved configurations, the obtained overall behavior should approach the outcome one gets by relying upon the Navier–Stokes equations.

Some insights into the usage of 1D CAs to model physical, biological and computational phenomena have been discussed by, amongst others, Wolfram [1] Ozelim et al. [10], Wuensche and Lesser [11], Chua [12–16] and Redeker et al. [17]. Still, despite the recurrent discussion on the topic [18–20], no generally valid relation between PDEs and CAs has been established. In the present paper, this issue is addressed for what concerns advection–dispersion from a constant source.

In Sect. 2, elementary cellular automata (ECAs) are briefly reviewed, while the formalism of the so-called iota-delta representation of ECAs is introduced in Sect. 3. Section 4 presents the basic notions on the considered advection–dispersion phenomenon and its description by means of a PDE and Sect. 5 shows how the iota-delta representation of ECAs can be used to establish a link between ECAs and the governing PDE.

2 Elementary cellular automata (ECAs)

A comprehensive study of ECAs can be found in [1] and [12–16]. In short, ECAs consist of a one-dimensional array of cells C_i , each of them bearing one out of two



Fig. 1 Evolution of ECA rule 30

states, namely 0 or 1. At discrete time steps a predefined transition function is applied to each of the cells synchronously to compute their states at the next time step. Essentially, the state of a given cell C_i at the next time steps $k + 1$ is determined by its own and the one of its left C_{i-1} and right C_{i+1} , neighbors at the previous time step k [1]. So, denoting the state of cell C_i at time step k as $s(C_i, k)$, we may write

$$s(C_i, k + 1) = f [s(C_{i-1}, k), s(C_i, k), s(C_{i+1}, k)] \tag{1}$$

In this way, an ECA can be characterized by a so-called rule icon (Fig. 1) and it is clear that 256 different transition functions can be listed in total. The enumeration scheme developed by Wolfram considers the outputs of the transition functions as coefficients in the decomposition of a given number base 2 to determine the corresponding rule number [1]. In the case of the rule depicted in Fig. 1, the rule number is given as $0 \times 2^7 + 0 \times 2^6 + 0 \times 2^5 + 1 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 = 30$.

3 The iota-delta function representation of ECAs

In a recent paper, Ozelim et al. [21] propose a new general representation for ECAs, the so-called iota-delta representation. This was done using a homonymous function defined in [21]: the iota-delta function. The idea behind the iota-delta function is quite simple. Starting from the fact that the transition function of ECA rule 90 is given by

$$s(C_i, k + 1) = \text{mod} [s(C_{i-1}, k) + s(C_{i+1}, k); 2], \tag{2}$$

which is equivalent to the XOR operation with $s(C_i, k)$ and $s(C_{i+1}, k)$, a generalization can be conceived by introducing three constants $\alpha_j = \{0, 1\}$ for $j = 1, 2, 3$. More precisely, a more general representation of ECA transition functions is given by [21]:

$$s(C_i, k + 1) = \text{mod} [\alpha_1 s(C_{i-1}, k) + \alpha_2 s(C_i, k) + \alpha_3 s(C_{i+1}, k); 2]. \tag{3}$$

Since the constants α_j can only be 0 and 1 only a limited number of rules can be described by Eq. (3), namely ECA rules 0, 60, 90, 102, 150, 170, 204 and 240 [21]. Equation (3) also presents a “parity” problem: the odd rules, according to Wolfram’s enumeration scheme, can never be represented because in that case

$$s(C_k, i + 1) = \text{mod} [\alpha_1 0 + \alpha_2 0 + \alpha_3 0; 2] = 0; \quad \forall \alpha_j, j = 1, 2, 3. \tag{4}$$

In order to overcome this issue, the representation in Eq. (3) may be improved by considering a fourth constant [21]:

$$s(C_i, k + 1) = \text{mod} [\alpha_1 s(C_{i-1}, k) + \alpha_2 s(C_i, k) + \alpha_3 s(C_{i+1}, k) + \alpha_4; 2]. \tag{5}$$

Using the 16 possible combinations of binary coefficients, 16 ECA evolution rules can be defined [21]. In order to represent even more ECA rules, Ozelim et al. [21] propose to wrap multiple modular operations:

$$s(C_i, k + 1) = \text{mod} [\text{mod} [\alpha_1 s(C_{i-1}, k) + \alpha_2 s(C_i, k) + \alpha_3 s(C_{i+1}, k) + \alpha_4; 3]; 2]. \tag{6}$$

The constants in Eq. (6) can now take three values instead of 2, resulting in a total of 81 combinations. Using Eq. (6) it is now possible to represent 53 ECA rules, taking into account the fact that some rules are represented by different sets of constants [21]. To represent even more ECA rules, the wrapping has to be done with higher congruence moduli. This was made possible by conceiving the so-called iota-delta function $\delta : \mathbb{C} \rightarrow \{a + bi; a, b \in [0, p_m)\}$, which is defined mathematically as [21]:

$$\begin{aligned} i\delta_n^m(x) &= \text{mod}[\text{mod}[\dots \text{mod}[\text{mod}[x, p_m], p_{m-1}], \dots, p_j], n], \\ m &\geq j; m, n \in \mathbb{Z}_+; x \in \mathbb{C}; j = \pi(n) + 1, \end{aligned} \tag{7}$$

in which m and n are parameters of the iota-delta function, p_m is the m -th prime number and $\pi(n)$ stands for the prime counting function that gives the number of primes less than or equal to n .

According to Ozelim et al. [21] every ECA can be represented by:

$$s(C_i, k + 1) = i\delta_2^5 (\alpha_1 s(C_{i-1}, k) + \alpha_2 s(C_i, k) + \alpha_3 s(C_{i+1}, k) + \alpha_4), \tag{8}$$

where $\alpha_j = \{r | r \leq p_5 - 1, r \in \mathbb{Z}_+\}$ for $j = 1, \dots, 4$.

Hence, every ECA can be represented by means of a quadruple $\{\alpha_1, \alpha_2, \alpha_3, \alpha_4\}$. The list of quadruples for the 88 minimal ECA rules can be found in Table 1.

Since different quadruples can encode the same ECA rule, an ECA rule can have more than one iota-delta function representation. For example, ECA rule 150 can be represented by $s(C_i, k + 1) = i\delta_2^5 (s(C_{i-1}, k) + s(C_i, k) + s(C_{i+1}, k) + 3)$ and $s(C_i, k + 1) = i\delta_2^3 (s(C_{i-1}, k) + s(C_i, k) + 4s(C_{i+1}, k))$. Now, one may look for a representation where for every odd rule it holds that $\alpha_4 = 1$, whereas for every even rule it holds that $\alpha_4 = 0$. This can be achieved by increasing the possible values of the other constants, which implies wrapping more congruence moduli around, such as increasing the parameter m in the iota-delta function representation. Thus, in this case, the minimal function to be used to represent every ECA is $i\delta_2^6(x)$. Table 2 presents the iota-delta representation of the 88 minimal ECA rules for which it holds that $\alpha_4 = 1$ for every odd rule, whereas for every even rule $\alpha_4 = 0$.

For example, based on Table 1, ECA rule 30 rule can be written as:

$$s(C_i, k + 1) = i\delta_2^5 (s(C_{i-1}, k) + 4s(C_i, k) + 4s(C_{i+1}, k)). \tag{9}$$

Table 1 Iota-delta representation of the 88 minimal ECA rules with $m = 5$ and $n = 2$

RN	α_1	α_2	α_3	α_4	RN	α_1	α_2	α_3	α_4	RN	α_1	α_2	α_3	α_4	RN	α_1	α_2	α_3	α_4
0	0	0	0	0	1	1	1	1	8	2	1	1	3	9	3	1	1	0	1
4	1	3	1	9	5	1	0	1	1	6	1	2	6	2	7	1	2	2	6
8	1	3	10	10	9	1	2	2	8	10	1	0	2	2	11	1	2	7	8
12	1	2	0	2	13	1	2	4	6	14	1	2	2	2	15	1	0	0	1
18	1	2	1	7	19	2	1	2	6	22	1	1	1	0	23	2	2	6	6
24	1	2	2	0	25	2	1	1	6	26	1	2	4	0	27	2	7	8	4
28	1	3	4	5	29	2	8	7	4	30	1	3	7	5	32	1	3	8	10
33	1	2	3	8	34	0	1	2	2	35	2	1	7	8	36	1	2	1	10
37	1	2	1	6	38	2	1	4	0	40	1	1	2	9	41	1	1	2	1
42	2	2	1	5	43	2	2	5	1	44	1	3	5	9	45	1	3	2	1
46	2	8	4	0	50	2	1	2	2	51	0	1	0	1	54	2	10	2	2
56	1	3	7	7	57	2	10	7	6	58	2	4	5	10	60	1	1	0	0
62	2	2	7	10	72	1	2	1	9	73	1	2	1	1	74	1	2	6	9
76	2	1	2	5	77	2	3	4	1	78	2	4	8	0	90	1	0	1	0
94	2	4	6	2	104	1	1	1	2	105	1	1	1	4	106	2	2	6	0
108	2	6	2	0	110	2	4	4	0	122	2	8	2	2	126	2	2	4	2
128	1	1	1	9	130	1	1	6	9	132	1	2	7	2	134	1	2	2	10
136	0	1	1	2	138	2	3	1	0	140	2	1	3	0	142	2	3	3	9
146	1	2	1	0	150	1	1	1	3	152	2	1	1	2	154	2	3	10	2
156	2	6	7	2	160	1	0	1	2	162	2	4	1	5	164	1	2	1	2
168	2	2	4	9	170	0	0	1	0	172	2	5	3	7	178	2	4	7	10
184	2	4	3	10	200	2	1	5	9	204	0	1	0	0	232	2	2	2	0

For what concerns the iota-delta function $\iota\delta_2^5(x)$, it can be verified that this is a periodic function with period 11, being the largest last prime in its definition and the period of the outmost modular function. Its behavior in one period is shown in Fig. 2.

In the subsequent part of this paper, the iota-delta representation will be used to link ECAs to the PDE describing the advection–dispersion phenomenon, but first we will discuss advection–dispersion phenomena and how finite difference methods (FDMs) for solving the governing equations can be retrieved from ECAs.

4 Advection–dispersion and finite differences schemes

Finite differences are used to numerically approximate PDEs by turning them into difference equations, which can then be solved to arrive at an approximate solution of the governing PDE [22]. In order to illustrate the applicability of the methodology hereby presented, the advection–dispersion equation is considered. This equation describes, for example, the solute flow in a porous medium.

Table 2 Iota-delta representation of the 88 minimal ECA rules

RN	m	n	α_1	α_2	α_3	α_4	RN	m	n	α_1	α_2	α_3	α_4	RN	m	n	α_1	α_2	α_3	α_4
0	2	2	0	0	0	0	1	3	2	2	2	2	1	2	4	2	5	5	4	0
4	4	2	5	4	5	0	5	2	2	1	0	1	1	6	5	2	2	1	8	0
8	3	2	2	3	3	0	9	5	2	1	8	8	1	10	2	2	2	0	1	0
12	2	2	2	1	0	0	13	3	2	4	3	2	1	14	5	2	3	4	4	0
18	5	2	1	2	8	0	19	3	2	3	1	3	1	22	2	2	1	1	1	0
24	2	2	1	2	2	0	25	3	2	3	4	4	1	26	3	2	1	2	4	0
28	3	2	1	4	2	0	29	5	2	3	5	9	1	30	4	2	1	4	4	0
33	5	2	1	8	4	1	34	2	2	0	2	1	0	35	3	2	2	4	3	1
37	3	2	4	3	4	1	38	3	2	2	1	4	0	40	5	2	2	9	10	0
42	4	2	5	5	6	0	43	3	2	2	2	3	1	44	4	2	3	6	5	0
46	5	2	2	8	4	0	50	5	2	4	3	4	0	51	2	2	0	1	0	1
56	4	2	6	3	5	0	57	3	2	3	1	2	1	58	5	2	4	9	8	0
62	5	2	8	8	4	0	72	5	2	2	10	9	0	73	2	2	1	2	1	1
76	4	2	5	6	5	0	77	3	2	2	3	2	1	78	5	2	2	4	8	0
94	5	2	8	4	8	0	104	2	2	2	2	2	0	105	3	2	1	4	4	1
108	3	2	2	4	2	0	110	4	2	2	4	4	0	122	4	2	4	2	4	0
128	4	2	5	5	5	0	130	5	2	2	5	8	0	132	5	2	2	1	9	0
136	2	2	0	2	2	0	138	3	2	2	3	1	0	140	3	2	2	1	3	0
146	2	2	1	2	1	0	150	3	2	1	1	4	0	152	5	2	4	3	5	0
156	4	2	4	1	3	0	160	2	2	2	0	2	0	162	3	2	2	3	4	0
168	4	2	5	5	3	0	170	2	2	0	0	1	0	172	6	2	7	4	10	0
184	6	2	4	7	10	0	200	4	2	5	3	5	0	204	2	2	0	1	0	0

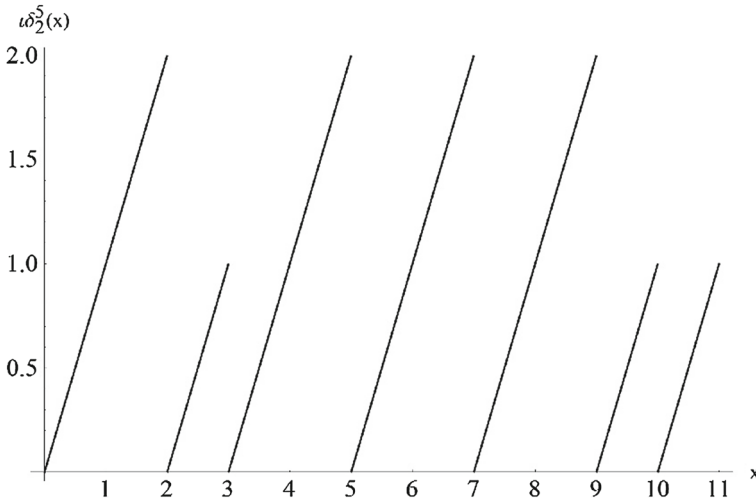


Fig. 2 The graph of $t\delta_2^5(x)$

Let the equation that describes the concentration $c(x,t)$ of a given solute flowing in a porous medium at some location in space and instance in time be given by [23]:

$$\frac{\partial c}{\partial t} = D_x \frac{\partial^2 c}{\partial x^2} - v_x \frac{\partial c}{\partial x}, \tag{10}$$

in which $D_x[L^2/T]$ is the hydrodynamic dispersivity of the medium and $v_x[L/T]$ is the mean velocity of the interstitial fluid.

Now, let us rely on the forward and backward difference approximation in time and space, respectively, for the first-order derivative, while using a central difference approximation for discretizing the second-order derivative in space [24]. Hence, by considering a rectangular mesh with grid spacings Δt and Δx Eq. (10) can be approximated by

$$\frac{c(x, t + \Delta t) - c(x, t)}{\Delta t} = D_x \frac{c(x + \Delta x, t) - 2c(x, t) + c(x - \Delta x, t)}{\Delta x^2} - v_x \frac{c(x, t) - c(x - \Delta x, t)}{\Delta x}. \tag{11}$$

In order to simplify the notation and clarify the link between the FDM and CAs, let us introduce i to index the finite difference nodes along the x axis, and k to index the nodes along the t axis. Further, let us introduce the following:

$$\frac{D_x \Delta t}{\Delta x^2} = N, \tag{12}$$

$$\frac{v_x \Delta t}{\Delta x} = C_r, \tag{13}$$

Fig. 3 Evolution of the FDM scheme applied to a constant source advection–dispersion phenomenon after two iterations

1	0	0	0	\dots
1	A	0	0	\dots
1	B	C	0	\dots
\vdots	\vdots	\vdots	\vdots	\vdots

with N and C_r being the Neumann and Courant numbers, respectively. By denoting c ($i \Delta x, k \Delta t$) as c_i^k , Eq. (11) can be reformulated as

$$c_i^{k+1} = (N + C_r) c_{i-1}^k + (1 - C_r - 2N) c_i^k + N c_{i+1}^k. \tag{14}$$

Relying upon the Courant, Fredrich and Levy condition [24], it may be concluded that the introduced FDM scheme is convergent if

$$2N + C_r \leq 1. \tag{15}$$

The similarity between Eqs. (8) and (14) is remarkable, but a clear conceptual difference is present: while the former describes the evolution of a discrete-state system, the latter describes the dynamics of a continuous-state system. The main question is how to relate both.

5 ECAs for advection–dispersion phenomena

In order to compare the dynamics evolved by ECAs to the one evolved by PDEs for one-dimensional advection–dispersion phenomenon, one first has to set the proper boundary and initial conditions. In the present paper, a semi-infinite domain is chosen. The concentration of a given substance is kept constant at the left side of the spatial domain, i.e. a so-called Dirichlet boundary condition, while it is initially 0 elsewhere. Moreover, the concentrations are scaled with respect to a certain maximum concentration, namely the one that is present at the leftmost side of the spatial domain. Consequently, the concentration at the origin is always 1, while it lies in the unit interval elsewhere. Figure 3 shows the first three configurations that are obtained by applying Eq. (14) iteratively from the imposed initial and boundary conditions. It is easy to see that the unknowns in the consecutive configurations are given by

$$\begin{cases} A = N + C_r \\ B = (N + C_r)(2 - C_r - 2N) \\ C = (N + C_r)^2 \end{cases} \tag{16}$$

It should be recalled that A, B and C are real numbers as they result from a PDE that is built upon a continuous state domain.

The next step is to link the ECA evolution from the same initial configuration as the one given at the top of Fig. 3 to the one obtained with the FDM. First of all, Eq.

(8) should be modified in such a way that it can lead to the values obtained using the FDM, i.e. the values given by system (16).

In essence, the iota-delta representation of ECAs rules enables one to mathematically relate states in previous time steps k to future time steps $k + 1$, similar to the representations proposed others (see [1] for a review on this topic), but the key advantages of the iota-delta function is that this relation is done in terms of a linear combination of the cell states in the previous time steps. It is clear from Eq. (9), for example, that the evolution function can be split up into two parts: the linear dependence on the cell states and the nonlinear wrapping function (the iota-delta function).

Equation (14), which represents a FDM for the advection–dispersion PDE, can be interpreted as an evolution rule in which the value of the cell in the next time step (concentration in this case) is given by a linear combination of cell values’ in previous time steps. This is quite similar to the iota-delta representation of the ECA rules, with the only difference that this one involves the iota-delta function “wrapping”. Thus, to relate both evolution rules, it is clear that one has to remove the nonlinear part of the iota-delta representation. In other words, one has to look for ways to get $i\delta_2^m(x) = x$ for every m .

On the basis of Fig. 2, it is clear that if the arguments of the iota-delta function are lower than 2, the identity function is retrieved. Hence, it is key to make sure that the argument of the iota-delta function is always less than 2 in order to link the FDM and ECAs. This can be achieved by two procedures: linear scaling the ECA rules by a factor S and changing the initial condition from 1 to Ω . From the definition of the iota-delta function it follows that $i\delta_2^m(x) < 2$ for every m . Thus, by making S less than 1, one guarantees that in next time steps the outputs will be less than 2, while choosing Ω lower than 1 guarantees that after the first time step, the argument will also be less than 2.

Mathematically, by linearly scaling the right-hand side of Eq. (8) by a non-zero factor S , the corresponding ECA transition function becomes

$$s(C_i, k + 1) = S i\delta_2^5 (\alpha_1 s(C_{i-1}, k) + \alpha_2 s(C_i, k) + \alpha_3 s(C_{i+1}, k) + \alpha_4). \quad (17)$$

When Eq. (17) is applied to the initial configuration in the top row of Fig. 3, the subsequent configurations still obey the constant source hypothesis. Further, as discussed previously, in order to guarantee that after the first time step the arguments of the iota-delta function will be less than 2, one has to set the concentration in the source to Ω (Fig. 4).

Fig. 4 Evolution of the ECA updating scheme applied to a constant source advection–dispersion phenomenon

Ω	0	0	0	...
Ω	A'	0	0	...
Ω	B'	C'	0	...
\vdots	\vdots	\vdots	\vdots	\vdots

Fig. 5 Steady-state condition that will be obtained after a sufficient number of time steps using the FDM (*left*) and ECA (*right*)

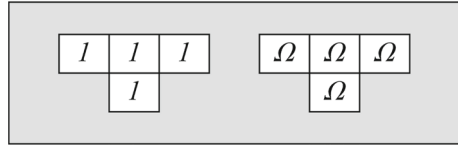


Figure 4 shows the first two configurations that are obtained by applying Eq. (17) iteratively, where one obtains for the unknowns A' , B' and C' :

$$\begin{cases} A' = S\iota\delta_2^5 (\alpha_1\Omega + \alpha_4) \\ B' = S\iota\delta_2^5 (\alpha_1\Omega + \alpha_2 S\iota\delta_2^5 (\alpha_1\Omega + \alpha_4) + \alpha_4) \\ C' = S\iota\delta_2^5 (\alpha_1 S\iota\delta_2^5 (\alpha_1\Omega + \alpha_4) + \alpha_4) \end{cases} \quad (18)$$

Starting from the imposed initial and boundary conditions, it is clear that after a large number of iterations, the cells close to the origin have values that are very close to 1 or Ω , when relying on the FDM and ECA, respectively. This is the so-called steady-state solution of the involved advection–dispersion problem. Hence, after a sufficient number of time steps, the configuration shown in Fig. 5 should be present near the origin.

Once a configuration with a sequence of 1s is obtained, it will persist in the case of the FDM because the sum of the coefficients which multiply the values of the cells in Eq. (14) is equal to 1. For the ECA, on the other hand, the following equation is obtained for the evolution from a configuration containing only Ω 's (Fig. 5):

$$\Omega = S\iota\delta_2^5 (\Omega (\alpha_1 + \alpha_2 + \alpha_3) + \alpha_4) \quad (19)$$

Reconsidering Fig. 2 and the discussions regarding the argument of the iota-delta function, it is clear that the argument of the iota-delta function in Eq. (19) has to be smaller than 2, since then it holds that $\iota\delta_2^5(x) = x$. For that reason, the following condition should hold:

$$\Omega < \frac{2 - \alpha_4}{(\alpha_1 + \alpha_2 + \alpha_3)} \quad (20)$$

Given that it is possible to express every odd rule with $\alpha_4 = 1$ and every even rule with $\alpha_4 = 0$, the right-hand side of Eq. (20) is positive for every ECA, so it reduces to

$$\Omega < \frac{1}{(\alpha_1 + \alpha_2 + \alpha_3)} \quad (21)$$

For simplicity, one can choose $\Omega = [2(\alpha_1 + \alpha_2 + \alpha_3)]^{-1}$, though any other value that satisfies Eq. (20) may be used as well. Continuing with this choice, Eq. (19) finally yields

$$S = \frac{1}{(\alpha_1 + \alpha_2 + \alpha_1)(1 + 2\alpha_4)} \quad (22)$$

Since every odd rule is related to an even one by color inversion, one may restrict attention to the even rules only, such that we may consider $\alpha_4 = 0$.

In order to link the evolutions in Figs. 2 and 3, one has to divide the states of the cells in the latter by Ω . This way, the following system of equations arises:

$$\begin{cases} 2(\alpha_1 + \alpha_2 + \alpha_3) \frac{i\delta_2^5 \left(\frac{\alpha_1}{2(\alpha_1 + \alpha_2 + \alpha_3)} \right)}{(\alpha_1 + \alpha_2 + \alpha_3)} = N + C_r \\ 2(\alpha_1 + \alpha_2 + \alpha_3) \frac{i\delta_2^5 \left(\frac{1}{(\alpha_1 + \alpha_2 + \alpha_3)} \left[\frac{\alpha_1}{2} + \alpha_2 i\delta_2^5 \left(\frac{\alpha_1}{2(\alpha_1 + \alpha_2 + \alpha_3)} \right) \right] \right)}{(\alpha_1 + \alpha_2 + \alpha_3)} = (N + C_r)(2 - C_r - 2N) \\ \frac{1}{(\alpha_1 + \alpha_2 + \alpha_3)(1 + 2\alpha_4)} i\delta_2^5 \left(\frac{\alpha_1}{(\alpha_1 + \alpha_2 + \alpha_3)(1 + 2\alpha_4)} i\delta_2^5 \left(\frac{\alpha_1}{2(\alpha_1 + \alpha_2 + \alpha_3)} + \alpha_4 \right) + \alpha_4 \right) \\ = (N + C_r)^2 \end{cases} \tag{23}$$

It is easy to see that the arguments of the iota delta functions in system (23) are all less than 2, such that they are equivalent to identity functions (Fig. 2). Thus, the solution of system (23) is:

$$\begin{cases} N = \frac{\alpha_3}{(\alpha_1 + \alpha_2 + \alpha_3)} \\ C_r = \frac{\alpha_1 - \alpha_3}{(\alpha_1 + \alpha_2 + \alpha_3)} \end{cases}, \tag{24}$$

which gives a physical meaning to ECAs rules. By choosing rules that obey system (24) and its physical constraints, an advection–dispersion phenomena characterized by a certain set of Neumann and Courant numbers may be associated with an ECA. The latter can then be used to mimic advection–dispersion phenomena at stake.

It is interesting to check the stability of the ECA scheme by means of Eq. (15). This way, the CFL condition implies

$$\begin{aligned} 2N + C_r &= \frac{2\alpha_3}{(\alpha_1 + \alpha_2 + \alpha_3)} + \frac{\alpha_1 - \alpha_3}{(\alpha_1 + \alpha_2 + \alpha_3)} \\ &= \frac{\alpha_1 + \alpha_3}{(\alpha_1 + \alpha_2 + \alpha_3)} \leq 1, \quad \forall \alpha_j, \quad j = 1, 2, 3. \end{aligned} \tag{25}$$

This shows that the ECA schemes are always stable, in addition to the fact that they give the same results as the FDM. The latter is illustrated in Fig. 6 for ECA rule 62 that mimics an advection–dispersion phenomenon for which $N = C_r = 0.2$.

For ECA rule 62, we infer from Table 2 that $\{\alpha_1, \alpha_2, \alpha_3, \alpha_4\} = \{8, 8, 4, 0\}$. Figure 6a shows the configuration evolved by the corresponding iota-delta function starting from a single black cell at the LHS of the spatial domain. The next step involves a linear scaling of the transition function by S , where the latter can be found using Eq. (22), and yields $S = 0.05$. Figure 6b depicts the application of the scaled transition function starting from the same initial and boundary conditions as in Fig. 6a. At this point, we have to choose a value for Ω that fulfills the condition given by Eq. (21). Here, we choose $\Omega = 0.025$ and Fig. 6c shows the corresponding evolution. Finally, all cell

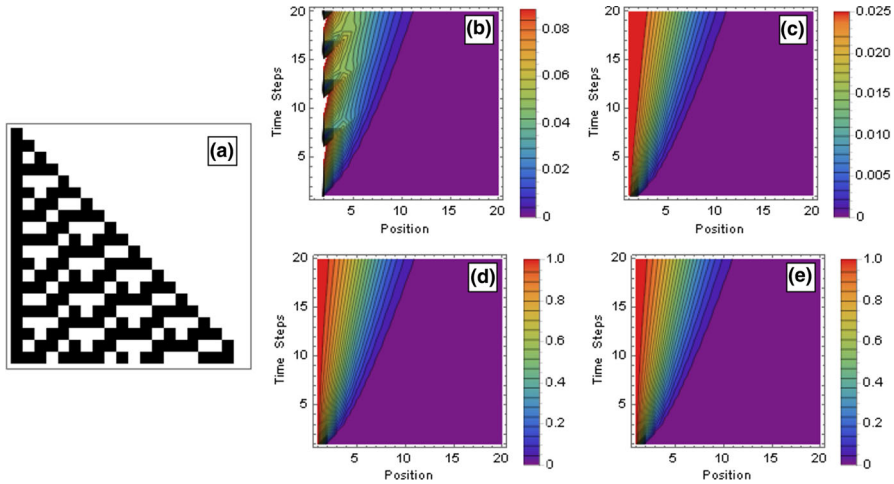


Fig. 6 Example of the proposed methodology: **a** space–time diagram of ECA rule 62; **b** after linear scaling of the transition function by S ; **c** after changing the initial and boundary values to Ω ; **d** after dividing the cell states by Ω and **e** corresponding FDM results for $N = C_r = 0.2$

states are scaled back to meet the FDM the initial and boundary conditions for which the FDM is implemented, which is achieved by dividing all cell states $\Omega = 0.025$ (Fig. 6d). Comparing Fig. 6d with Fig. 6e that depicts the evolution obtained using the governing FDM scheme with $N = C_r = 0.2$ (cfr. Eq. (24)) it is clear that the obtained space–time diagrams are the same.

The results presented here are easily generalized for larger FDM stencils. This is done by taking the dependency of $s(C_i, k + 1)$ with respect to more cells in Eq. (3).

Not only is the present methodology extendable to larger FDM stencils, but also to other partial differential equations. It is known that FDM schemes approximate derivatives by differences. These differences take into account the values of the dependent variables of the PDE in different spatial positions and/or time steps.

By imagining that the values of the dependent variables of PDEs are states in a grid, one can build a CA which will mimic the dependency on surrounding cells of the FDM scheme (even non-linear ones). This, together with the iota-delta formalism hereby proposed, will lead to a correspondence between PDEs and CAs.

It is worth noticing that the ideas presented above only apply to explicit FDM schemes. Implicit schemes still need to be investigated. The following proposition is established by the authors based on the discussions presented in this paper:

Proposition 1 *Every PDE which can be approximated by an explicit FDM scheme will have a correspondent CA representation by means of the iota-delta formalism.*

6 Conclusions

The usage of ECAs to describe physical phenomena has received much attention, but up to this day there is often no clear link between PDEs that are often used to describe

such phenomena and their discrete counterparts. In this paper it is shown how an ECA can be constructed that mimics the advection–convection from a constant source and that leads to results that are comparable with those obtained using FDMs.

References

1. Wolfram S (2002) *A new kind of science*. Wolfram Media, Champaign, IL
2. Wolfram S (1988) Complex systems theory. In: Proceedings of the founding workshops of the Santa Fe Institute, Addison-Wesley, *Emerging Syntheses in Science*, pp 183–189
3. Wolfram S (1984) Cellular automata as models of complexity. *Nature* 311:419–424
4. Hardy J, Pomeau Y, de Pazzis O (1973) Time evolution of a two-dimensional model system. I. Invariant states and time correlation functions. *J Math Phys* 14(12):1746–1759
5. Succi S (2001) *The lattice Boltzmann equation, for fluid dynamics and beyond*. Oxford Science Publications, Oxford
6. Frisch U, Hasslacher B, Pomeau Y (1986) Lattice-gas automata for the Navier–Stokes equation. *Phys Rev Lett* 56(14):1505–1508
7. Wolfram S (1986) Cellular automaton fluids I: basic theory. *J State Phys* 43(3/4):471–526
8. Frisch U, d’Humières D, Hasslacher B, Lallemand P, Pomeau Y, Rivet J-P (1987) Lattice gas hydrodynamics in two and three dimensions. *Complex Syst* 1:649–707
9. McNamara G, Zanetti G (1988) Use of the Boltzmann equation to simulate lattice-gas automata. *Phys Rev Lett* 61:2332–2335
10. de Ozelim LCSM, de Cavalcante ALB, de Borges LPF (2013) Continuum versus discrete: a physically interpretable general rule for cellular automata by means of modular arithmetic. *Complex Syst* 22(1):75–99
11. Wuensche A, Lesser MJ (1992) *The global dynamics of cellular automata: an Atlas of basin of attraction fields of one-dimensional cellular automata*. Santa Fe Institute Studies in the Sciences of Complexity, Addison-Wesley, Reading, MA p 250
12. Chua L (2006) *A nonlinear dynamics perspective of Wolfram’s new kind of science*, vol 1. World Scientific Publishing Company, Singapore
13. Chua L (2007) *A nonlinear dynamics perspective of Wolfram’s new kind of science*, vol 2. World Scientific Publishing Company, Singapore
14. Chua L (2009) *A nonlinear dynamics perspective of Wolfram’s new kind of science*, vol 3. World Scientific Publishing Company, Singapore
15. Chua L (2011) *A nonlinear dynamics perspective of Wolfram’s new kind of science*, vol 4. World Scientific Publishing Company, Singapore
16. Chua L (2012) *A nonlinear dynamics perspective of Wolfram’s new kind of science*, vol 5. World Scientific Publishing Company, Singapore
17. Redeker M, Adamatzky A, Martínez GJ (2013) Expressiveness of elementary cellular automata. *Int J Mod Phys C* 24(3):1350010. doi:[10.1142/S0129183113500101](https://doi.org/10.1142/S0129183113500101)
18. Omohundro S (1984) Modelling cellular automata with partial differential equations. *Physica D* 10:128–134
19. Chopard B, Droz M (1998) *Cellular automata modeling of physical systems*. Cambridge University Press, Collection Al’ea Saclay
20. Kunishima W, Nishiyama A, Tanaka H, Tokihiro T (2004) Differential equations for creating complex cellular automaton patterns. *J Phys Soc Jpn* 73:2033–2036
21. de Ozelim LCSM, Cavalcante ALB, de Borges LPF (2013) On the iota-delta function: universality in cellular automata’s representation. *Complex Syst* 21(4):1–12
22. LeVeque RJ (2007) *Finite difference methods for ordinary and partial differential equations*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA
23. Najafi HS, Hajinezhad H (2008) Solving one-dimensional advection–dispersion with reaction using some finite-difference methods. *Appl Math Sci* 2(53):2611–2618
24. Datta BN (1995) *Numerical linear algebra and applications*. Brooks/Cole Publishing Company, Belmont, CA