# Scale-Splitting Error in Complex Automata Models for Reaction-Diffusion Systems

Alfonso Caiazzo[1], Jean Luc Falcone[2],
Bastien Chopard[2], and Alfons G. Hoekstra[1]

[1] Section Computational Science, University of Amsterdam, The Netherlands
{acaiazzo,alfons}@science.uva.nl
[2] CUI Department, University of Geneva, Switzerland
{Jean-Luc.Falcone,Bastien.Chopard}@cui.unige.ch

**Abstract.** Complex Automata (CxA) have been recently proposed as a paradigm for the simulation of multiscale systems. A CxA model is constructed decomposing a multiscale process into single scale sub-models, each simulated using a Cellular Automata algorithm, interacting across the scales via appropriate coupling templates. Focusing on a reaction-diffusion system, we introduce a mathematical framework for CxA modeling. We aim at the identification of error sources in the modeling stages, investigating in particular how the errors depend upon scale separation. Theoretical error estimates will be presented and numerically validated on a simple benchmark, based on a periodic reaction-diffusion problem solved via multistep lattice Boltzmann method.

**Keywords:** Complex Automata, multiscale algorithms, reaction-diffusion, lattice Boltzmann method.

## 1 Introduction

Complex Automata (CxA) have been recently proposed [1,5,6] as a paradigm for the simulation of multiscale systems. The idea of a CxA is to model a complex multiscale process using a collection of single scale algorithm, in the form of Cellular Automata (CA), lattice Boltzmann methods, Agent Based Models, interacting across the scales via proper coupling templates [6].

To construct a CxA we identify the relevant sub-processes, defining their typical time and space scale. The concept of the Scale Separation Map (SSM) [6] helps in this modeling stage. It is defined as a two dimensional map with the Cartesian axes coding the temporal and spatial scales. Each single scale model defines a *box* on the SSM, whose leftmost and bottom edges indicate the resolution of the single scale Automaton, while the rightmost and top edges are defined by the characteristics of the single scale process (e.g. the extreme of spatial and temporal domains). The key idea of a CxA is to transform a single big box spanning several time and space scales on the SSM in a set of interconnected smaller boxes (see Fig. 1).

M. Bubak et al. (Eds.): ICCS 2008, Part II, LNCS 5102, pp. 291–300, 2008.

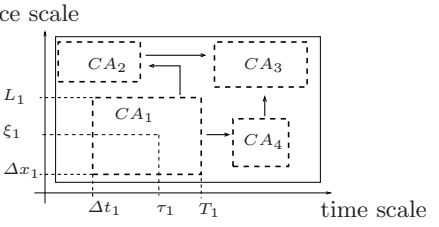


**Fig. 1.** Example of a scale separation map for a CxA model. The solid box represents a complex process spanning several space and time scales. The CxA model splits the box in $N$ smaller boxes (dashed lines, centered in the typical scales $(\tau_i, \xi_i)$ of the sub-processes $i$) interacting across the scales.

Formally, we replace the original multiscale process, identified by *state variable + update rule*, with a collection of $N$ *single scale models*, defined by

$$(\text{state variables} + \text{update rule})_i,\ i = 1, \ldots, N$$

plus a set of *coupling templates*, describing how the single scale models interact together.

Aim of this paper is to propose a formalism and a procedure to analyze CxA models. Focusing on an algorithm designed for reaction-diffusion problems, we investigate the *scale-splitting error*, i.e. the difference between the numerical solution obtained using a single *multiscale* algorithm based on single time and space discretization (equal to the ones needed to resolve the small scales) and the numerical solution obtained using the corresponding CxA model.

In section 2 we introduce shortly a lattice Boltzmann scheme for a reaction-diffusion problem, describing how a CxA model for that system can be constructed. In section 3 we investigate the error introduced in the CxA model, deriving explicit estimates for the considered benchmark, which are validated on simple numerical simulations. Finally, we draw the conclusion in section 4.

## 2 From a Multiscale Algorithm to Single Scale Models

### 2.1 Lattice Boltzmann Method for a Reaction-Diffusion Problem

We want to construct a CxA model for a reaction-diffusion process (RD) described by the equation

$$\partial_t \rho = D\partial_{xx}\rho + R(\rho),\ t \in (0, T_{end}],\ x \in (0, L] \tag{1}$$

(plus additional initial and boundary conditions).

To solve numerically (1) one can use an algorithm based on the lattice Boltzmann method (LBM) (overviews of the LBM can be found for example in [4,8], or in [2] for application to RD problems). For a chosen small parameter $h$, we

discretize the space interval with a regular grid $\mathcal{G}_h = \{0, \ldots, N_x - 1\}$ of step size $\Delta x_h = h$, and employ a D1Q2 model based on two velocities $c_i \in \{-1, 1\}$ and an update rule of the form

$$f_i^{t_n+1}(j + c_i) = f_i^{t_n}(j) + \frac{1}{\tau}(f_i^{eq}(\rho^{t_n}(j)) - f_i^{t_n}(j)) + h^2 \frac{1}{2} R(\rho^{t_n}(j)) \,. \tag{2}$$

Here $j \in \mathcal{G}_h$ is the spatial index of the node[1], while $n \in \mathbb{N}_0$ is the index which counts the time steps of length such that

$$\frac{\Delta t_h}{\Delta x_h^2} = \text{const.} \tag{3}$$

For the D1Q2 model we have

$$f_i^{eq}(\rho) = \frac{\rho}{2}, \; i = 1, -1 \,. \tag{4}$$

Algorithm (2) leads to a second order approximation of the solution of (1) [7] if the parameter $\tau$ in (2) is chosen according to the diffusion constant in (1) as

$$\tau = \frac{1}{2} + D \frac{\Delta t_h}{\Delta x_h^2} \,. \tag{5}$$

Observe that $\tau$ is independent from $h$ in virtue of (3).

In a more compact form, we can rewrite (2) as

$$\hat{f}_h^{t_{n+1}} = P_h(I_h + \Omega_{D_h}(\tau) + \Omega_{R_h})\hat{f}_h^{t_n}, \tag{6}$$

where $\hat{f}_h$ (omitting the subscript $i$) represents a $h$-grid function , i.e. a real valued function defined on a grid of step $h$:

$$\hat{f}_h : \mathcal{G}_h \to \mathbb{R}^2, \; \hat{f}_h : j \mapsto \hat{f}_h^n(j) \,.$$

Introducing the set $\mathcal{F}_h = \left\{\phi : \mathcal{G}_h \to \mathbb{R}^2\right\}$ we have $\hat{f}_h \in \mathcal{F}_h$. With this notation the subscript $h$ denotes operators acting from $\mathcal{F}_h$ to itself: $I_h$ is simply the identity on $\mathcal{F}_h$, $P_h$ acts on a grid function shifting the value on the grid according to $c_i$

$$\left(P_h \hat{f}_h\right)_i (j) = \hat{f}_{i,h}(j - c_i),$$

while $\Omega_{D_h}$ and $\Omega_{R_h}$ are the operations defined in the right hand side of (2):

$$\left(\Omega_{D_h} \hat{f}_h\right)_i (j) = \frac{1}{\tau}(f_i^{eq}(\rho(\hat{f}_h(j)) - \hat{f}_{h,i}^n(j)), \quad \left(\Omega_{R_h} \hat{f}_h\right)(j) = h^2 \frac{1}{2} R(\rho(\hat{f}_h(j)) \,.$$

Since $\Omega_{D_h}$ depends on the *non-equilibrium* part of $\hat{f}_h$ and $\Omega_{R_h}$ is a function of the moment of the distribution, i.e. of the equilibrium part, it can be shown [3] that

$$\forall \hat{f}_h : \; \Omega_{D_h} \Omega_{R_h} \hat{f}_h = 0 \,. \tag{7}$$

[1] For simplicity we assume periodic boundary conditions, considering the addition $j + c_i$ modulo $N_x$.

and similarly, since the operator $\Omega_{D_h}$ maps any element of $\mathcal{F}_h$ in grid function with zero moment:

$$\forall \hat{f}_h : \Omega_{R_h}(I_h + \Omega_{D_h})\hat{f}_h = \Omega_{R_h}\hat{f}_h \,. \tag{8}$$

Relation (7) allows to split the LB algorithm (6) (see also [2]) as

$$\hat{f}_h^{t_{n+1}} = P_h(I_h + \Omega_{D_h}(\tau))(I_h + \Omega_{R_h})\hat{f}_h^n = \mathcal{D}_h\mathcal{R}_h\hat{f}_h^{t_n}, \tag{9}$$

separating reaction $\mathcal{R}_h = I_h + \Omega_{R_h}$ and diffusion $\mathcal{D}_h = P_h(I_h + \Omega_{D_h}(\tau))$.

**Scale Separation Map.** The next step towards the CxA model is the scale separation map. We start with the discrete process described by equation (6), which represents in our case the multiscale algorithm. On the SSM (in this simplified case restricted to the time scale axis) it spans a range of time scales between $\Delta t_h = h^2$ and $T_{end}$ (fig. 2a). Equation (9) shows that we can split the problem in two processes. Assuming that the diffusion is characterized by a typical time scale larger than the reaction time scale (for example if $D$ is small compared to $\kappa$), we introduce a coarser time step

$$\Delta t_D = M\Delta t_h, \quad M \in \mathbb{N} \,.$$

In practice, it corresponds to execute $M$ steps of the reaction $\mathcal{R}$, up to a time $T_R = \Delta t_D$, followed by a single diffusion step.

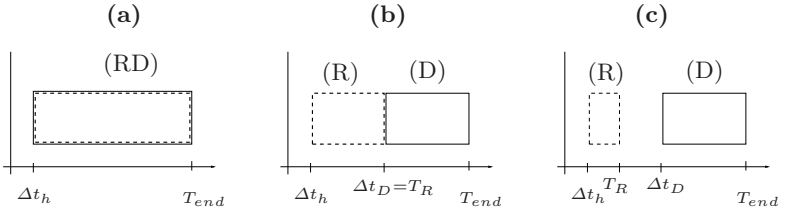


**Fig. 2.** The SSM for the reaction-diffusion problem. In **(a)** reaction (dashed line) and diffusion (solid line) are considered as a single multiscale algorithm. In **(b)** we assume to use different schemes, where the diffusion time step $\Delta t_D$ is larger than the original $\Delta t_h$. **(c)** represents the situation where the two processes are time separated, with a very fast reaction yielding an equilibrium state in a time $T_R \ll \Delta t_D$.

Back to the SSM, we have shifted to the right the lower edge of the diffusion box (figure 2b) and to the left the upper extreme of the reaction box. We have $T_R = \Delta t_D$, since the reaction is run in the whole time interval $[0, \Delta t_D]$ with time step $\Delta t_h$ and reinitialized after each diffusion step.

Observe that this is not the only possibility. In some systems, the reaction leads very quickly to an equilibrium state in a typical time smaller than the discrete time step of the diffusion. The scale map for this situation is depicted in

Fig. 2c. However, this is a rather special case and will not be discussed here. We restrict the analysis to a single species reaction-diffusion with linear reaction, focusing on the SSM in Fig. 2b, where the time scales are *not completely separated*. A more complete treatment of the different cases (including more general multiscale processes) shall be topic of an upcoming work.

**CxA Model for Reaction-Diffusion.** After coarsening the time scale of diffusion algorithm with $\Delta t_D = M\Delta t_h$, we can define a coarser space discretization selecting a new parameter $h_2$

$$\Delta x_{h_2} = h_2 = M_X h, \quad M_X \in \mathbb{N}\,. \tag{10}$$

Note that equation (2) must be modified according to (5). We restrict to two possibilities: $M_X = 1$ (time coarsened CxA) or $M_X = \sqrt{M}$ (time-space coarsened CxA, in order to preserve relation (3)).

Introducing the vector of small parameter $H = (h, h_2)$, the CxA can be formally described with the state variable $\hat{f}_H = \left(\hat{f}_{1,h}, \hat{f}_{2,h_2}\right)$, whose components denote the state after reaction $\hat{f}_{1,h}$ and after diffusion $\hat{f}_{2,h_2}$ and evolve according to

$$\begin{array}{ll} (\mathrm{CA}_R) & (\mathrm{CA}_D) \\ \hat{f}^{t_{1,0}=t_{2,n_2}}_{1,h_1} = f_1^{init}\left(\hat{f}^{t_{2,n_2}}_{2,h_2}\right), & \hat{f}^{0}_{2,h_2} = \hat{f}^{init}_{2,h_2}(\rho_0), \\ \hat{f}^{t_{1,n_1+1}}_{1,h_1} = \mathcal{R}_{1,h_1}\hat{f}^{1,t_{n_1}}_{1,h_1},\, n_1=0,\ldots,M-1 & \hat{f}^{t_{2,n_2+1}}_{2,h_2} = \mathcal{D}_{2,h_2}\Pi_{h_2,h}\hat{f}^{t_{2,n_2}+M\Delta t_R}_{1,h_1}. \end{array} \tag{11}$$

$\hat{f}^{init}_{2,h_2}$ being the initial condition.

Using different discretizations, we define also a projection $\Pi_{h_2,h}$ from the grid $\mathcal{G}_h$ to $\mathcal{G}_{h_2}$ and a new operator $\mathcal{D}_{h_2}$. In fact, it depends on the relaxation time $\tau$, which must be modified according to (5).

With the terminology introduced in [6] (11) is a CxA composed of two Automata coupled through the following templates. (i) $CA_R$ is coupled to $CA_D$ through the initial conditions, since the output value of a single $CA_D$ iteration is used to define $CA_R$ initial conditions. (ii) $CA_D$ is coupled to $CA_R$ through the collision operator, since the output value of the reaction (after $M$ iterations of $CA_R$) is used to compute $CA_D$ collision operator.

This example is a special case where the two processes act on the same variable, and it is possible to write the algorithm only depending on $\hat{f}_{2,h_2}$:

$$\begin{array}{l} \hat{f}^{t_{2,n_2+1}}_{2,h} = \mathcal{D}_{2,h}(\tau_{(M)})\mathcal{R}^M_{2,h}\hat{f}^{t_{2,n_2}}_{2,h}, \\ \hat{f}^{0}_{2,h} = \hat{f}^{init}_{2,h}(\rho_0)\,. \end{array} \tag{12}$$

**General Formalism.** From a more general point of view, we can formalize the Complex Automata modeling technique starting with an algorithm in the form

$$f_h^{t_{n+1}} = \Phi_h f_h^{t_n} \tag{13}$$

defined on *fine* space and time scales, identified by a discretization parameter $h$. Equation (13) describes the evolution of a single multiscale algorithm for a complex process where the numerical solution $f_h^{t_n}$ denotes the state of the system at the $n$-th iteration and $\Phi_h$ is the update rule. As before, we introduce the spatial grid $\mathcal{G}_h$ so that

$$f_h^{t_n} \in \mathcal{F}_h = \{\phi : \mathcal{G}(h) \to \mathcal{R}^2\}, \quad \Phi_h : \mathcal{F}_h \to \mathcal{F}_h \,. \tag{14}$$

Constructing a CxA we replace algorithm (13) with several simpler single scale Cellular Automata, described by the state variable $\hat{f}_H = \left(\hat{f}_{1,h_1}, \ldots, \hat{f}_{R,h_R}\right)$ (where $H = (h_1, \ldots, h_R)$) and an update rules for each component of $\hat{f}_H$:

$$\Phi_H = (\Phi_{1,h_1}, \ldots, \Phi_{R,h_R}) \,. \tag{15}$$

The numerical solution of the CxA model $\hat{f}_H$ belongs to a space $\mathcal{F}_H^{\mathrm{CxA}} \subset \mathcal{F}_1 \times \ldots \mathcal{F}_R$ (since the state spaces can be shared by different Automata), and $\Phi_H : \mathcal{F}_H \to \mathcal{F}_H$. We can also introduce a general *projection* operator $\Pi$ which defines the way we *coarsen* our process and *lift* operator $\Lambda$ which describes an approximate reconstruction of the fine grid solution starting from the CxA one:

$$\Pi_{Hh},\ \Pi_{h_r,h} : \mathcal{F}_h \to \mathcal{F}_{h_r}, r = 1, \ldots, R, \quad \Lambda_{hH} : \mathcal{F}_H \to \mathcal{F}_h \,.$$

The relevant spaces and operators introduced in the formalism in the particular case of reaction-diffusion can be represented with the diagram

$$\begin{array}{ccc} f_h \in \mathcal{F}_h & \xrightarrow{\Pi_{Hh}=(\Pi_{h_1,h},\Pi_{h_2,h})} & f_H = \left(\hat{f}_{1,h_1}, \hat{f}_{2,h_2}\right) \in \mathcal{F}_H \\ \Big\downarrow {\scriptstyle \Phi_h^M=(\mathcal{D}_h\mathcal{R}_h)^M} & & \Big\downarrow {\scriptstyle \Phi_H=\left(\mathcal{R}_{1,h}^M,\mathcal{D}_{2,h_2}\right)} \\ f_h \in \mathcal{F}_h & \xleftarrow{\Lambda_{hH}} & f_H \in \mathcal{F}_H \end{array}$$

where a single step of the update rule $\Phi_H$ corresponds to $M$ steps of the $\Phi_h$, since $\Delta t_D = M \Delta t_R$. In this example example, if $h_1 = h_2 = h$, the components of the projection $\Pi_{Hh}$ are equal to the identity on $\mathcal{F}_h$. If $h_1 \neq h_2$, $\Pi_{h_2,h}$ can be a sampling of the numerical solution on a coarser grid, and $\Lambda_{h,h_2}$ an interpolation routine.

## 3 The Scale-Splitting Error

The idea of the CxA model is to replace

($\mathbf{A}_h$): the original (complex) multiscale algorithm depending on a discretization parameter $h$,

with

(**CxA**): a collection of coupled single scales algorithms.

This yields an improvement of the performance which is paid by a possible loss of precision. In analyzing the CxA modeling procedure, we are interested in quantifying $E^{A_h,\mathrm{CxA}}$, expressing the difference between the numerical solutions of $(\mathbf{A}_h)$ and $(\mathbf{CxA})$, which we call *scale-splitting* error or CxA-*splitting* error.

This quantity can be related to the loss of accuracy. Calling $E^{\mathrm{CxA},EX}$ the error of the CxA model with respect to the exact solution and $E^{A_h,EX}$ the absolute error of the model $(\mathbf{A}_h)$, we can write

$$\left\|E^{\mathrm{CxA},EX}\right\| \leq \left\|E^{A_h,EX}\right\| + \left\|E^{A_h,\mathrm{CxA}}\right\|. \tag{16}$$

### 3.1 Error Estimates for the Reaction-Diffusion Model

For the CxA model of (RD), since the algorithm is designed to approximate the variable $\rho$, we can define the scale-splitting error at time iteration $t_N$ as

$$E^{\mathrm{A,CxA}}(\rho; M, M_X, t_N) = \left\|\rho\left(\Pi_{h_2,h}\hat{f}_h^{t_N}\right) - \rho\left(\hat{f}_{2,h_2}^{t_N}\right)\right\| = \left\|\rho\left(\Pi_{Hh}\hat{f}_h^{t_N} - \hat{f}_{2,h_2}^{t_N}\right)\right\| \tag{17}$$

representing the difference between $\rho(\hat{f}_h)$, i.e. the numerical solution of the fine-grid algorithm (9) and $\rho(\hat{f}_{2,h_2})$, i.e. the output of the CxA model (11) after both reaction and diffusion operators have been applied.

Observing that $\hat{f}_h$ is the solution of (2) and $\hat{f}_{2,h_2}$ is obtained from (11) we can rewrite (17) as

$$= \left\|\rho\left(\Pi_{h_2,h}\left(\mathcal{D}_h\mathcal{R}_h\right)^M \hat{f}_h^{t_N - M\Delta t_h} - \mathcal{D}_{2,h_2}\Pi_{h_2,h}\mathcal{R}_h^M \Lambda_{h,h_2}\hat{f}_{2,h_2}^{t_N - M\Delta t_h}\right)\right\| \tag{18}$$

For simplicity, let us assume that the two solutions coincide at the previous time and that we can write[2]

$$\hat{f}_h^{t_N - M\Delta t_h} = \Lambda_{h,h_2}\hat{f}_{2,h_2}^{t_N - M\Delta t_h}$$

(for example if $t_N - M\Delta t_h$ corresponds to the initial time). Since the reaction is local, we have

$$\forall M > 0 : \mathcal{R}_h^M \Lambda_{h,h_2} = \Lambda_{h,h_2}\mathcal{R}_{2,h_2}^M .$$

Additionally, note that

$$\Pi_{h_2,h}\Lambda_{h_2,h}\hat{f}_{2,h_2} = \hat{f}_{2,h_2}$$

(projecting after reconstructing gives the same function). Hence, we conclude that the distance between the numerical solutions can be estimated by measuring the distance between the algorithms

[2] In general, it holds

$$\hat{f}_h^{t_N - M\Delta t_h)} = \Lambda_{hH}\hat{f}_H^{t_N - \Delta t_2} + \epsilon(M_X, M, t_N - \Delta t_2) + \epsilon_{\Lambda,\Pi}(H,h)$$

where $\epsilon(M_X, M, t)$ is bounded by $E(M_X, M, t)$ and $\epsilon_{\Lambda,\Pi}(H,h)$ depend on the accuracy of projection and lift operations. The derivation of the estimate for this case is not reported in this short communication [3].

$$
\begin{aligned}
&E^{A,\mathrm{CxA}}(M_X, M) := \left\| \rho \left( \Pi_{h_2,h} \left( \mathcal{D}_h \mathcal{R}_h \right)^M \Lambda_{h,h_2} - \mathcal{D}_{2,h_2} \mathcal{R}_{2,h_2}^M \right) \right\| \leq \\
&\left\| \rho \left( \Pi_{h_2,h} \left[ \left( \mathcal{D}_h \mathcal{R}_h \right)^M - \mathcal{D}_h^M \mathcal{R}_h^M \right] \Lambda_{h,h_2} \right) \right\| + \\
&\left\| \rho \left( \left( \Pi_{Hh} \mathcal{D}_h^M - \mathcal{D}_{2,h_2} \Pi_{h_2,h} \right) \mathcal{R}_h^M \Lambda_{h,h_2} \right) \right\| = E^{(1)} + E^{(2)} .
\end{aligned} \tag{19}
$$

The contribution $E^{(1)}$ depends on the difference $\left( \mathcal{D}_h \mathcal{R}_h \right)^M - \mathcal{D}_h^M \mathcal{R}_h^M$, which can be estimated as a function of $[\mathcal{D}_h, \mathcal{R}_h] = \mathcal{D}_h \mathcal{R}_h - \mathcal{R}_h \mathcal{D}_h$, i.e. the commutator of the operators $\mathcal{R}_h$ and $\mathcal{D}_h$. For example, if $M = 2$ we have

$$
\left( \mathcal{D}_h \mathcal{R}_h \right)^2 = \mathcal{D}_h \mathcal{R}_h \mathcal{D}_h \mathcal{R}_h = \mathcal{D}_h^2 \mathcal{R}_h^2 - \mathcal{D}_h [\mathcal{D}_h, \mathcal{R}_h] \mathcal{R}_h .
$$

An argument based on asymptotic analysis [3,7], assuming that the numerical solution (which is a $h$-grid function) can be approximated by a smooth function evaluated on the grid points, i.e.

$$
\hat{f}_h(n, j) \approx f(t_n, x_j), \tag{20}
$$

can be used to show (also in virtue of the properties (7)-(8)) that

$$
[\mathcal{D}_h, \mathcal{R}_h] \in O \left( h^3 \partial_x \rho \kappa \right) . \tag{21}
$$

By counting the number of times the commutator appears in the difference $\left( \mathcal{D}_h \mathcal{R}_h \right)^M - \mathcal{D}_h^M \mathcal{R}_h^M$, we have

$$
\left( \mathcal{D}_h \mathcal{R}_h \right)^M - \mathcal{D}_h^M \mathcal{R}_h^M \in O \left( M^2 h^3 \partial_x \rho \kappa \right) . \tag{22}
$$

The part $E_H^{(2)}$ derives from the coarsening of the original lattice Boltzmann algorithm. Assuming (20) for both the coarse- and the fine-grid solutions we obtain

$$
\rho \left( \Pi_{h_2,h} \mathcal{D}_h^M - \mathcal{D}_{2,h_2} \Pi_{h_2 h} \right) = O(M_X^2 h^2 + M^2 D^3 h^2) . \tag{23}
$$

We are interested in how the error depends on the coarsening in time and space, fixed a fine discretization parameter $h$ and for particular values of $\kappa$ and $D$ which regulate the typical time scales of the system. In conclusion, (22)-(23) yield

$$
\begin{aligned}
&\text{if } M_X = 1: \quad E^{A,\mathrm{CxA}}(M) = O \left( M^2 h^3 \partial_x \rho \kappa \right) + O(M^2 D^3 h^2), \\
&\text{if } M_X = 1, M = M_X^2: \quad E^{A,\mathrm{CxA}}(M) = O \left( M h^2 \right) .
\end{aligned} \tag{24}
$$

### 3.2 Numerical Results

We consider the benchmark problem (1) with $D = 0.1$, $\kappa = 3$ and initial value $\rho(0, x) = \sin(2\pi x)$, for which we have the analytical solution

$$
\rho_{EX}(t, x) = \sin(2\pi x) \exp \left( \left( R - D 4\pi^2 \right) t \right) . \tag{25}
$$

From the simulation results $\hat{f}_h$ of algorithm (2) and $\hat{f}_{2,h_2}$ of model (11), we evaluate (17) as

$$E(M_X, M; t_N) = \left\| \Pi_{h_2,h}\rho(\hat{f}_h) - \rho(\hat{f}_{2,h_2}) \right\| = \\ = \max_{t_N} \frac{1}{N_x(h_2)} \left\{ \sum_{j\in\mathcal{G}_{h_2}} \left\| \Pi_{h_2,h}\rho(\hat{f}_h^{t_N})(j) - \rho(\hat{f}_{2,h_2}^{t_N}(j)) \right\| \right\}, \tag{26}$$

Using (25), we compare also the scale-splitting error with the quantity

$$E^{A_h,EX} = \|\rho(\hat{f}_h) - \rho_{EX}\|_h = \max_n \frac{1}{N_x(h)} \left\{ \sum_{j\in\mathcal{G}_h} \left\| \rho(\hat{f}_h^{t_n}(j)) - \rho_{EX}(t_n, x_j)) \right\| \right\}, \tag{27}$$

i.e. the error of the original fully fine-discretized algorithm (6) (evaluated with an opportune norm on the fine-grid space, according to the norm chosen in (26)). Fig. 3 shows the results of scale-splitting error investigation, choosing different values of $M$ and comparing the cases $M_X = 1$ and $M = M_X^2$.

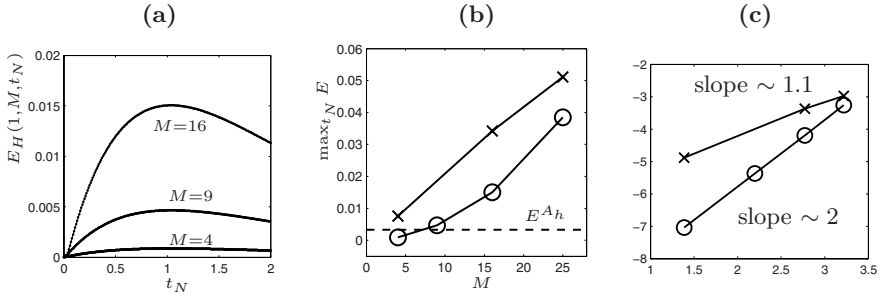


**Fig. 3. (a)**: CxA-splitting error versus time for $M = 4, 9, 16$, fixing $M_X = 1$. **(b)**: Maximum values of the CxA-splitting error for $M_X = 1$ ($\circ$) and $M_X = \sqrt{M}$ ($\times$) over a complete simulation, as a function of $M$. The dashed line shows the error $\|E_h^{A_h,EX}\|$ of the fully fine-discretized algorithm ($\Delta x_h = h$, $\Delta t_h = h^2$) with respect to the exact solution (25). **(c)**: Order plot (fig. (b) in double logarithmic scale) of maximum CxA-splitting error versus $M$. The approximate slopes confirm the behavior predicted by (24).

In particular, in fig. 3b we compare the maximum scale-splitting error in time for different values of $M$ also with the error $E^{A_h,EX}$ defined in (27). It shows that for small $M$ the splitting error is of the same order of the discretization error of the original lattice Boltzmann algorithm (6). In this cases, the simplification obtained with the CxA model does not affect the quality of the results.

The order plot in fig. 3c confirms estimates (24). In fact, the splitting error increases linearly in $M$, while for $M_X = 1$ the increment is quadratic. As a consequence, for a moderate range of $M$, $M_X = 1$ can produce quantitatively better results.

## 4 Conclusions and Outlook

We introduced a formalism to describe Complex Automata modeling. In particular we investigated the scale-splitting error, i.e. the modeling error introduced by replacing a fully fine-discretized problem with multiple Cellular Automata on different scales. Restricting to a lattice Boltzmann scheme for reaction-diffusion problems, we have derived explicit estimates for the splitting error verifying the expectation on simple numerical simulations.

The investigation of the scale-splitting error represents the basis of a theoretical foundation of the Complex Automata simulation technique. In a future work we will discuss generalizations of the concepts presented here and of the estimates derived in this particular example to more complicate systems and more general CxA models.

**Acknowledgments.** This research is supported by the European Commission, through the COAST project [1] (EU-FP6-IST-FET Contract 033664).

## References

1. The COAST project, `http://www.complex-automata.org`
2. Alemani, D., Chopard, B., Galceran, J., Buffle, J.: LBGK method coupled to time splitting technique for solving reaction-diffusion processes in complex systems. Phys. Chem. Chem. Phys. 7, 1–11 (2005)
3. Caiazzo, A., Falcone, J.L., Hoekstra, A.G., Chopard, B. Asymptotic Analysis of Complex Automata models for Reaction-Diffusion systems. (in preparation, 2008)
4. Chopard, B., Droz, M.: Cellular Automata Modelling of Physical Systems. Cambridge University Press, Cambridge (1998)
5. Hoekstra, A.G., Chopard, B., Lawford, P., Hose, R., Krafczyk, M., Bernsdorf., J.: Introducing Complex Automata for Modelling Multi-Scale Complex Systems. In: Proceedings of European Complex Systems Conference 2006 (CD), European Complex Systems Society, Oxford (2006)
6. Hoekstra, A.G., Lorenz, E., Falcone, J.-L., Chopard, B.: Towards a Complex Automata Formalism for Multi-Scale Modeling. (accepted for publication) International Journal for Multiscale Computational Engineering 5, 491–502 (2007)
7. Junk, M., Klar, A., Luo, L.-S.: Asymptotic analysis of the lattice Boltzmann Equation. Journal Comp. Phys. 210, 676–704 (2005)
8. Succi, S.: The Lattice Boltzmann Equation for Fluid Dynamics and Beyond. Oxford University Press, Oxford (2001)