Asymptotic Analysis of Complex Automata models for Reaction-Diffusion systems

Alfonso Caiazzo [a], Jean-Luc Falcone [c], Bastien Chopard [c],
Alfons G. Hoekstra [b]

[a] *INRIA Rocquencourt, BP105, F-78150 Le Chesnay Cedex, France*

[b] *University of Amsterdam, Section Computational Science, Kruislaan 403, 1098 SJ, Amsterdam, the Netherlands*

[c] *CUI Department, University of Geneva, Switzerland*

Abstract

Complex Automata (CxA) have been recently introduced as a paradigm to simulate multiscale multiscience systems as a collection of generalized Cellular Automata on different scales. The approach yields numerical and computational challenges and can become a powerful tool for the simulation of particular complex systems. We present a mathematical framework for CxA modeling to investigate the behavior of the model depending on scale separation and modeling choices. For a simple CxA model for a reaction-diffusion process, we define a Complex Automata model, deriving theoretical error estimates, which are numerically validated.

Key words: Complex Automata modeling, reaction-diffusion, lattice Boltzmann method, asymptotic expansion.

1 Introduction

Among the challenges in computational sciences, the field of multiscale simulation has become more and more popular in recent years. Due to the large variety of interesting problems, many approaches for multiscale systems simulations have been and are continuously developed. We focus on the recently introduced Complex Automata (CxA) paradigm [9,10].

Email addresses: `alfonso.caiazzo@inria.fr` (Alfonso Caiazzo), `Jean-Luc.Falcone@cui.unige.ch` (Jean-Luc Falcone), `Bastien.Chopar@cui.unige.ch` (Bastien Chopard), `a.g.hoekstra@uva.nl` (Alfons G. Hoekstra).

Preprint submitted to Elsevier Science *27th January 2009*

Looking at multiscale systems as an ensemble of processes happening on different temporal and spatial scales, the idea behind a Complex Automata is that a *multiscale algorithm*, designed to simulate such multiscale system, can be replaced by many *single scale models*, constructed to simulate the relevant sub-processes, choosing appropriately different resolution according to the properties of the original system. According to the original dynamics, the single scale algorithms have to be coupled across the scales using proper coupling templates. Furthermore, we constrain these single scale models to have a specific update paradigm. Namely, we consider numerical methods whose time evolution can be decomposed in a *local collision* step (when the state of the system is updated using only local information), plus a *propagation* step, when the new states are communicated through the system. Approaches such as Cellular Automata (CA), lattice Boltzmann methods (LBM), or Agent Based models (ABM) satisfy these assumption. In the context of CA, it has been shown [7] that this update paradigm is equivalent to the more popular choices. Moreover, we observe that many FD schemes can be written in the same fashion. Focusing on this class of algorithm allows us to introduce a special formalism to describe CxA modeling [2,10], to formalize classes of multiscale couplings couplings [9], and it is particularly interesting from the computational point of view, since it can results in efficient numerical schemes and it can be used to design a specific CxA simulation framework [8].

In a few words, a CxA model is a reduction of an original (*complicated but accurate*) multiscale algorithm to a collection of (*simpler but less precise*) single scale sub-algorithms. Detailed introduction of the Complex Automata approach and of related issues and perspectives can be found in [9,10]. The aim of this paper is to describe a formalism for the CxA modeling and investigate the difference between the numerical solution obtained using a single multiscale algorithm, based on fine time and space discretization, and the numerical solution obtained using a CxA model (extending the results presented in [2]. In particular, we are interested in quantifying the role of the "scale separation" in the quality of the results, in order to justify the CxA approach. A description of a general formalism for the CxA modeling is a rather broad topic and involves a large set of problems. The goal of this paper is to provide a first introduction to that formalism. We show the application in a simple yet very interesting test case of a lattice Boltzmann method (LBM) applied to a reaction-diffusion system.

As a corollary results, the presented analysis demonstrate general mathematical properties of LBM operators, which can be of interest for the LBM community itself, also uncorrelated to the CxA context.

In §2 we introduce our benchmark problem and the numerical scheme, together with the Complex Automata formulation, which results in an operator splitting (with multiple time steps) approach for the LBM. In §3 two different approaches to derive explicit error estimates will be presented. The first is based on a direct asymptotic analysis of the Complex Automata model. Since we cannot expect this technique to be available in general, we discuss also a hybrid approach, which uses asymptotic analysis and particular properties of the discrete algorithms. The theoretical expectations will be validated performing several numerical investigations, shown in §4, to investigate also the influence of the difference scales on the quality of the results. Conclusions are drawn in §5.

2 A Complex Automaton model for Reaction-Diffusion

Our benchmark is a reaction-diffusion process for a concentration field $\rho(t,x)$ described by the equation

$$\begin{aligned} &\partial_t \rho = d\partial_{xx}\rho + \kappa(\rho_\lambda - \rho),\ t \in (0, T_{end}],\ x \in (0,1] \\ &\rho(0,x) = \rho_{in}(x) \end{aligned} \tag{1}$$

with periodic boundary conditions in x-direction and initial condition ρ_{in}, where ρ_λ is a given function, corresponding to a reaction local equilibrium. With $\rho_\lambda(x) = \sin(\lambda x)$, for $\dfrac{\lambda}{2\pi} \in \mathbb{Z}$, problem (1) admits the analytical solution

$$\rho^*(t,x) = \exp(-(4\pi^2 d + \kappa)t)\sin(2\pi x) + \frac{\kappa}{d\lambda^2 + \kappa}\sin(\lambda x)\,. \tag{2}$$

Problem (1) is formulated in dimensionless units, introducing characteristic times and lengths depending on κ, d and λ. We investigate multiscale models where the (dimensionless) reaction rate κ is much larger than the (dimensionless) diffusion coefficient d, i.e. when the reaction is characterized by a typical time scale faster than the diffusion.

2.1 Lattice Boltzmann method

To solve numerically (1) we employ a lattice Boltzmann method (LBM) (see [3,6,12] and references therein for overview of LBM, while specific applications to reaction-diffusion have been discussed, for example, in [4,5,13]). To define the numerical scheme, 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 $\Delta x = h$. We focus on a D1Q2 model[1] . Namely, it is a one-dimensional model where each grid

[1] It must be mentioned that the D1Q2 is not the only possible LB model for (1), but indeed the simplest. For the sake of simplicity, being interested in the methodology

node $j \in \mathcal{G}_h$ holds a two-dimensional vector $\hat{\mathbf{f}} = \left(\hat{f}_1, \hat{f}_{-1}\right)$, representing the numerical solution for the probability densities traveling forward and backward with discrete velocities $c_i \in \{-1, 1\}$. The approximation of the concentration is given by the moment of the distribution $\hat{\mathbf{f}}$:

$$\hat{\rho} = \rho(\hat{\mathbf{f}}) = \sum_{i=1,-1} \hat{f}_i.$$

Each component of $\hat{\mathbf{f}}$ is updated according to

$$\hat{f}_i^{t_{n+1}}(j + c_i) = \hat{f}_i^{t_n}(j) + \frac{1}{\tau}\left(f_i^{eq}(\hat{\rho}^{t_n}(j)) - \hat{f}_i^{t_n}(j)\right) + \Delta t \frac{1}{2} R(\hat{\rho}^{t_n}(j)). \quad (3)$$

where $R\left(\hat{\rho}(j)\right) = \kappa(\rho_\lambda(x_j) - \hat{\rho}(j))$, $t_n = n\Delta t$, for $n \in \mathbb{N}_0$, and the time step Δt is such that [2]

$$\forall h > 0 : \frac{\Delta t}{\Delta x^2} = \text{const.} \quad (4)$$

The equilibrium distribution f_i^{eq} is a function of the moment ρ defined as

$$f_i^{eq}(\rho) = \frac{\rho}{2}, \ i \pm 1\ . \quad (5)$$

The parameter τ is chosen according to the diffusion constant in (1)

$$\tau(d, \Delta x, \Delta t) = \tfrac{1}{2} + d\frac{\Delta t}{\Delta x^2}. \quad (6)$$

Observe that τ is independent from h in virtue of (4).

2.2 Asymptotic Analysis

It can be shown [6,11] that algorithm (3) yields a second order accurate approximation of the solution of (1). We report here only the main lines of the derivation. For details, we refer to [11].

Since the numerical solution is defined on a grid of spacing h, we will explicitly denote the dependence of $\hat{\mathbf{f}}$ on the particular discretization using a subscript h. However, to have simpler notations, this will not be used together with the index $i \in \{1, -1\}$, denoting the element of $\hat{\mathbf{f}}_h$, unless required in the context to avoid confusion.

of analysis, we stick to this case. Other realizations of LBM can be analogously treated.

[2] This relation, called in literature *diffusive scaling* is a necessary condition [6,11] to preserve accuracy properties of the scheme.

In order to understand the behavior of the numerical scheme (3) for small h, we look for an approximation of the numerical solution $\hat{\mathbf{f}}_h^{t_n}(j)$ starting with an *ansatz*

$$F_h^{t_n}(j) = \mathbf{f}^{(0)}(t_n, x_j) + h\mathbf{f}^{(1)}(t_n, x_j) + h^2\mathbf{f}^{(2)}(t_n, x_j) + \dots \tag{7}$$

where $\mathbf{f}^{(k)}(t, x)$ are *smooth* and *h-independent* coefficients.

As next, we observe that the LBM (3) can be equivalently rewritten (moving on the right hand side all the terms) as

$$L_{hi}(\hat{\mathbf{f}}_h) = 0\,, \tag{8}$$

i.e. as a h-depending equation, of which $\hat{\mathbf{f}}_h$ is a solution. For any function in the form (7), we can now define the *residue* as

$$\begin{aligned} L_{hi}(F_h^{t_n}(j)) = F_h^{t_{n+1}}(j + c_i) - F_h^{t_n}(j) - \frac{1}{\tau}\left(f_i^{eq}(\rho(F_h^{t_n}(j))) - F_h^{t_n}(j)\right) \\ - \frac{h^2}{2}R(\rho(F_h^{t_n}(j))). \end{aligned} \tag{9}$$

Driven by (8), we look for an expansion (7) for which the residue stays small. Using the hypotheses of smoothness, Taylor expanding L_h around (t_n, x_j), and sorting the different orders in h, we obtain a set of equations which we can be recursively solved, allowing to define the coefficients $\mathbf{f}^{(k)}$ in order to obtain a residue of high order in h. The general expression holds [11]

$$f_i^{(k)} = \frac{\rho_k}{2} - \tau\left(c_i\partial_x f_i^{(k-1)} + \left(\tfrac{1}{2}\partial_x^2 + \partial_t\right) f_i^{(k-2)} + \left(\frac{1}{6}\partial_x^3 + \partial_t dx\right) c_i f_i^{(k-3)} + \dots\right), \tag{10}$$

where ρ_k, for $k \geq 0$, are smooth functions to be specified step by step.

For $k = 0, 1, 2$ we obtain from (10)

$$\begin{aligned} f_i^{(0)} &= \tfrac{1}{2}\rho_0, \\ f_i^{(1)} &= \tfrac{1}{2}\rho_1 - \tau\tfrac{1}{2}c_i\partial_x\rho_0, \\ f_i^{(2)} &= \tfrac{1}{2}\rho_2 - \tau\tfrac{1}{2}\left(\partial_t\rho_0 - \left(\tau - \tfrac{1}{2}\right)\partial_x^2\rho_0 + c_i\partial_x\rho_1\right). \end{aligned} \tag{11}$$

Simple algebra shows that the resulting expansion F_h cancels the residue (9) up to order h^3 if ρ_0 is a solution of (1), taking $\rho_1 = 0$, and for any smooth function ρ_2. Thus, the previous 11 can be simplified as

$$f_i^{(0)} = \tfrac{1}{2}\rho_0, \quad f_i^{(1)} = -\tau\tfrac{1}{2}c_i\partial_x\rho_0, \quad f_i^{(2)} = \tfrac{1}{2}\rho_2. \tag{12}$$

yielding the the expansion

$$F_{hi} = \rho_0 + h\left(-\tau\tfrac{1}{2}c_i\partial_x\rho_0\right) + h^2\frac{\rho_2}{2}, \tag{13}$$

which we will call *prediction*, such that

$$L_h(F_h^{t_n}(j)) = O(h^3).$$

Assuming that the numerical solution $\hat{\mathbf{f}}_h$ is approximated by F_h up to order h^3 we conclude

$$\hat{\rho} = \rho(F_h) + O(h^3) = \rho_0 + h^2\rho_2 + O(h^3),$$

i.e. the numerical scheme (3) yields a second order accurate approximation of the solution of (1) [11,13].

The function ρ_2 represents the leading order term of the error. More details about this term can be derived requiring the prediction to cancel further orders in the residue (explicit results will be presented in §3).

2.3 Operator Splitting Approach for LBM

Formally, the variable $\hat{\mathbf{f}}_h$ is a h-grid function, i.e. a real-valued map from a discrete grid of step h:

$$\hat{\mathbf{f}}_h : \mathcal{G}_h \to \mathbb{R}^2,\ \hat{f}_h : j \mapsto \hat{\mathbf{f}}_h(j) = (\hat{f}_{h1}(j), \hat{f}_{h-1}(j)).$$

Introducing the set $\mathcal{F}_h = \{\phi : \mathcal{G}_h \to \mathbb{R}^2\}$ we have $\hat{\mathbf{f}}_h \in \mathcal{F}_h$.

Moreover, using the subscript h for the operators acting from $\mathcal{F}_h$ to itself, we can introduce a *propagation* operator P_h, which acts on a grid function shifting the value on the grid according to c_i:

$$\forall \hat{\mathbf{f}} \in \mathcal{F}_h :\ \left(P_h \hat{\mathbf{f}}\right)(j) = \begin{bmatrix} \hat{f}_1(j-1) \\ \hat{f}_{-1}(j+1) \end{bmatrix}$$

and diffusion and reaction operators, Ω_{D_h} and (resp.) Ω_{R_h} according to the right hand side of (3):

$$\forall \hat{\mathbf{f}} \in \mathcal{F}_h : \quad \begin{aligned} \left(\Omega_{D_h}(\Delta t)\hat{\mathbf{f}}\right) &= \frac{1}{\tau(d,h,\Delta d)} \begin{bmatrix} f_1^{eq}(\rho(\hat{\mathbf{f}})) - \hat{f}_1 \\ f_{-1}^{eq}(\rho(\hat{\mathbf{f}})) - \hat{f}_{-1} \end{bmatrix} \\ \left(\Omega_{R_h}(\Delta t)\hat{\mathbf{f}}\right) &= \frac{\Delta t}{2} \begin{bmatrix} R\left(\rho(\hat{\mathbf{f}})\right) \\ R\left(\rho(\hat{\mathbf{f}})\right) \end{bmatrix} \end{aligned} \qquad (14)$$

(where, in the last definition, the two components are equal).

Definitions (14) stress that reaction and diffusion operators depends on the discretization, not only concerning the domain of definition (subscript h, but also through the particular expressions). In the following part, the argument Δt will be omitted when equal to h^2, unless particularly required by the context.

Now we can rewrite (3) in the compact form

$$\hat{\mathbf{f}}_h^{t_{n+1}} = P_h(I_h + \Omega_{D_h} + \Omega_{R_h})\hat{\mathbf{f}}_h^{t_n} = \Phi_h \mathbf{f}_h^{t_n}, \tag{15}$$

introducing the update rule $\Phi_h = P_h(I_h + \Omega_{D_h} + \Omega_{R_h})$.

Observe that in this compact notation ρ can be seen as a particular operator as well:

$$\rho : \mathcal{F}_h \to \mathbb{R}, \quad \rho(\mathbf{f}_h) = f_{h1} + f_{h-1}.$$

Lemma. *According to definitions* (14)*, it can be shown that*

$$\forall \hat{\mathbf{f}}_h \in \mathcal{F}_h \; : \; \Omega_{D_h}\Omega_{R_h}(\hat{\mathbf{f}}_h) = 0. \tag{16}$$

Proof. It follows observing that $f^{eq}(\Omega_{R_h}(\hat{\mathbf{f}}_h)) = \Omega_{R_h}(\hat{\mathbf{f}}_h)$ and $\Omega_{D_h}(\hat{\mathbf{f}}_h))$ is proportional to $\hat{\mathbf{f}}_h - f^{eq}(\rho(\hat{\mathbf{f}}_h))$ (i.e. the operator Ω_{R_h} maps any grid function onto the kernel of Ω_{D_h}. □

In virtue of (16), the LB algorithm (3) can be equivalently rewritten as

$$\hat{\mathbf{f}}_h^{t_{n+1}} = P_h(I_h + \Omega_{D_h})(I_h + \Omega_{R_h})\hat{\mathbf{f}}_h^{t_n} = \mathcal{D}_h \mathcal{R}_h \hat{\mathbf{f}}_h^{t_n}. \tag{17}$$

In other words, the LBM can be naturally split in a reaction $\mathcal{R}_h(\Delta t) = I_h + \Omega_{R_h}(\Delta t)$ and a diffusion $\mathcal{D}_h(\Delta t) = P_h(I_h + \Omega_{D_h}(\Delta t))$ part. The previous results justify formally basic operator splitting techniques for the reaction-diffusion LBM (proposed, for example, in [1]). In our case, the formulation (17) represents the starting point of the Complex Automata approach.

2.4 CxA Formalism and Scale Separation Map

We consider the LBM (17) as the original *multiscale process*, to which we apply a Complex Automata formulation. The parameter h determines spatial and temporal scales, and in this case it is strictly related to the faster process, since it must be chosen according to the smaller time scale. Assuming that the two processes involved in (17) are characterized by different typical time scales (informally, d small compared to κ in (1)), we define a Complex Automaton composed of two separate algorithms, $\mathcal{C}_1$ (reaction) and $\mathcal{C}_2$ (diffusion). We remark that both $\mathcal{C}_1$ and $\mathcal{C}_2$ satisfy the collision + propagation update rule, being formally copies of the original LBM (15).

In practice, we coarsen the time scale of the diffusion, choosing $\Delta t_1 = \Delta t = h^2$, $\Delta t_2 = M\Delta t_1 = Mh^2$, for some $M \in \mathbb{N}_0$ (note that the time step of the reaction coincide with the time step of the original algorithm (17), which was in fact depending on the reaction time scale).

An important tool to define and graphically visualize a Complex Automaton is the Scale Separation Map (SSM), which consists of a Cartesian plane whose axes represents temporal and spatial scales [9,10]. On this map, the multiscale algorithm and the processes defining the CxA can be represented as *boxes*, defined by their discrete resolution in time and space, and by the upper bound of their domain of definition.

The SSM for the considered system is shown in figure 1. The box representing the LBM (17) is defined by $(\Delta t, \Delta x)$ (lower-left corner) and $(1, T_{\text{end}})$ (upper-right corner). The model $\mathcal{C}_1$ is represented by a box, whose lower-left corner coincides with the original one. For the model $\mathcal{C}_2$, coarsened in time, the edge denoting the time step has been moved to the right. Unlike the box for the original algorithm, once reaction and diffusion are separated, the boxes for these sub-processes are not completely defined, and the specification depends mainly on modeling choices, dictated by the particular physical problem. In particular, the time step Δt_2 is function of the typical diffusion time, while the time T_1 (upper bound for reaction temporal domain) might be in principle defined at any point of the temporal axis, within the original interval.

Figure 1 shows two particular realizations, which are more relevant in practical applications. In figure 1b $T_1 = \Delta t_2$: the reaction model is iterated in an interval $[t_0, t_0 + \Delta t_2]$, with a time step Δt_1, and re-initialized after an iteration of the diffusion model, which is iterated between $[0, T_{\text{end}}]$, with time step Δt_2. Given that $\Delta t_2 = M\Delta t_1$, this corresponds in practice to execute M steps of the reaction $\mathcal{C}_1$ within each iteration of $\mathcal{C}_2$. The CxA model for this problem can be seen as an operator splitting approach with multiple time steps.

Figure 1c sketches a case of complete scale-separation. This is the situation when reaction leads very quickly to an equilibrium state, in a typical time which is even smaller than the discrete time step of the diffusion. In this case, the time T_1 can be taken less than Δt_2, assuming that further iterating the model will not lead to significantly different results. This is a rather particular case, happening if $\kappa \gg d$ and for special structure of equation (1). In what follows, we will focus on the situation of figure 1b.

To formally define the CxA model, we introduce the parameters $H = (\delta_1, \delta_2)$, where each $\delta_i = (h_i, \Delta t_i)$ represents a short notation for a discretization.

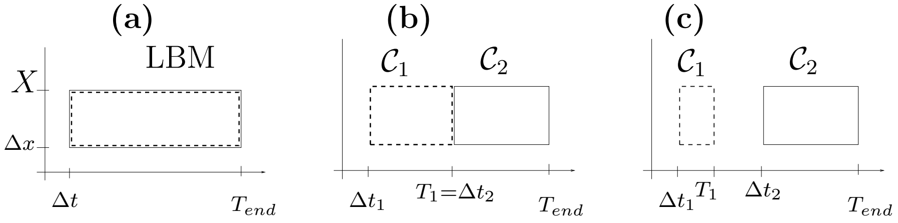

Figure 1. SSM for the reaction-diffusion LBM. In **(a)** reaction (dashed line) and diffusion (solid line) are considered as a single multiscale algorithm (17). In **(b)** we assume to use different schemes, where the diffusion time step Δt_2 is larger than the original Δt (while keeping $\Delta t_1 = \Delta t$). Figure **(c)** represents the situation where the two processes are time separated, with a very fast reaction yielding an equilibrium state in a time $T_1 \ll \Delta t_2$.

In the case discussed above, and depicted in the SSM in figure 1b, $h_1 = h_2 = h$ (the spatial grid is not changed), and the time step of the diffusion is such that

$$\Delta t_2 = Mh^2. \tag{18}$$

More in general, we can take into account spatial coarsening choosing

$$h_2 = M_X h,\ \Delta t_2 = Mh^2 \tag{19}$$

for $M_X, M \in \mathbb{N}$. We remark that both M and M_X are free parameters. However, the choice $M = M_X^2$ allows to keep constant the relaxation parameter τ, according to (6).

We describe the state of the CxA using the variable $\hat{\mathbf{f}}_H = \left(\hat{\mathbf{f}}_{1,h_1}, \hat{\mathbf{f}}_{2,h_2}\right)$, whose components corresponds to the LB variables after reaction and (resp.) diffusion.

Dealing with functions on different spatial grids (for example if $h_2 \neq h$), the space $\mathcal{F}_H^{\mathrm{CxA}}$ and $\mathcal{F}_h$ are connected via non trivial projection Π_{Hh} (*fine-to-coarse*) and lift Λ_{hH} (*coarse-to-fine*) operators, such that

$$\begin{tikzcd}
\hat{\mathbf{f}}_h \in \mathcal{F}_h \arrow[r, "\Pi_{Hh}=(\Pi_{h_1,h},\Pi_{h_2,h})"] \arrow[d, "\Phi_h=(\mathcal{D}_h\mathcal{R}_h)"'] & \hat{\mathbf{f}}_H = \left(\hat{\mathbf{f}}_{1,h_1}, \hat{\mathbf{f}}_{2,h_2}\right) \in \mathcal{F}_H \arrow[d, "\Phi_H=\left(\mathcal{R}_{h_1}(\Delta t_1),\mathcal{D}_{h_2}(\Delta t_2)\right)"] \\
\hat{\mathbf{f}}_h \in \mathcal{F}_h & \hat{\mathbf{f}}_H \in \mathcal{F}_H \arrow[l, "\Lambda_{hH}"']
\end{tikzcd}$$

Example. Using a coarser grid for the diffusion, the projection $\Pi_{h_2,h}$ could be a sampling of the fine solution taken on the coarser grid or an average of the fine-grid points close to a coarse grid node. The lift Λ_{hH} involves in general an interpolation to reconstruct the function $\hat{\mathbf{f}}_h$ on the fine grid, based on the results of reaction (which operates on the same discretization) and diffusion (which operates on a coarser grid). The above diagram generalizes, in the context of CxA, procedures which are standardly used in multiscale problems. We remark that a similar approach can be used to describe projection and lifting between different time discretizations.

Coming back to the CxA model for (1), we denote with t_{1,n_1} and t_{2,n_2} the current simulation times (where $t_{m,n} = n\Delta t_m$, depending on the iteration index n and the size time step Δt_m, for $m = 1, 2$) of reaction and diffusion algorithms. The state of the CxA is updated according to

$$\begin{array}{ll} \mathcal{C}_1,\ t_1 \in [t_0, t_0 + \Delta t_2] & \mathcal{C}_2,\ t_2 \in [0, T_{\text{end}}] \\ \text{set } t_0 = t_{2,n_2},\ \hat{\mathbf{f}}^{t_0}_{1,h_1} = \Lambda_{h_1,h_2}\hat{\mathbf{f}}^{t_{2,n_2}}_{2,h_2}, & \hat{\mathbf{f}}^{0}_{2,h_2} = \hat{\mathbf{f}}^{init}_{2,h_2}(\rho_{in}), \\ \hat{\mathbf{f}}^{t_{1,n_1+1}}_{1,h_1} = \mathcal{R}_{h_1}(\Delta t_1)\hat{\mathbf{f}}^{1,t_{1,n_1}}_{1,h_1},\ _{n_1 \le M-1} & \hat{\mathbf{f}}^{t_{2,n_2+1}}_{2,h_2} = \mathcal{D}_{h_2}(\Delta t_2)\Pi_{h_2,h_1}\hat{\mathbf{f}}^{t_{2,n_2}+M\Delta t_1}_{1,h_1}. \end{array} \tag{20}$$

A few comments are necessary to explain the formalism introduced in (20). The left side describe the algorithm $\mathcal{C}_1$, which is coupled to $\mathcal{C}_2$ through the initial condition (by setting at the initial time $t_{1,0} = t_{2,n_2}$ (the current simulation time of $\mathcal{C}_2$) the initial condition equal to the results after a time iteration of $\mathcal{C}_2$. Then, M steps according to an update rule depending only on the reaction process are performed. On the right, the diffusion part is coupled to the reaction through the update rule, since the new state of $\hat{\mathbf{f}}_{2,h_2}$ is computed starting from the output state of $\mathcal{C}_1$. With $\hat{\mathbf{f}}^{init}_{2,h_2}(\rho_{in})$ we denoted the original initial condition, function of the initial concentration in (1).

The operators Λ_{h_1,h_2} and Π_{h_2,h_1} have been used in equation (20) to handle functions on different discrete spaces. Note also that in (20) we have introduced the operators $\mathcal{R}_{h_1}$ and $\mathcal{D}_{h_2}$, which depend on the different discretizations. The case $h_2 = h_1 = h$ yields $\mathcal{R}_{h_1} = \mathcal{R}_h$. However, $\Delta t_2 \neq \Delta t_h$, yields $\mathcal{D}_{h_2}(\Delta t_2) \neq \mathcal{D}_h$. Namely, the diffusion operator depends on τ, which must be modified according to (6) as

$$\tau_{(M)} := \tfrac{1}{2} + Md. \tag{21}$$

Finally, observe that this situation is a special case where the two processes act on the same variable, and it is possible to write the algorithm only depending on $\hat{\mathbf{f}}_{2,h}$ (with $h_2 = h$):

$$\begin{aligned} &\hat{\mathbf{f}}^{t_{2,n_2+1}}_{2,h} = \mathcal{D}_h(\Delta t_2)\left(\mathcal{R}_h(\Delta t)\right)^M \hat{\mathbf{f}}^{t_{2,n_2}}_{2,h}, \\ &\hat{\mathbf{f}}^{0}_{2,h} = \hat{\mathbf{f}}^{init}_{2,h}(\rho_{in}). \end{aligned} \tag{22}$$

This formulation will be useful analyzing the error contributions.

3 Error Analysis and Investigations

A CxA model for a multiscale process replaces a single multiscale algorithm $(\mathbf{A}_h)$, related to a discretization h, with a collection of coupled simpler algorithms $(\mathbf{CxA}_H)$. The simplification in terms of complexity yields an improvement of the performance [10], which is paid by a possible loss of precision.

Focusing on the benchmark (1) in this section we describe possible analysis approaches to quantify the *scale-splitting* error [2], i.e. the difference between the numerical results of the algorithm $(\mathbf{A}_h)$, defined by (17), and the formal $(\mathbf{CxA}_H)$ (20).

A formal definition of the scale-splitting error depends in general on the particular problem and on the specific quantities of interests. In this case, since our algorithm is designed to approximate the variable ρ, we define the scale-splitting error at time iteration t_N taking the difference in the moment of $\hat{\mathbf{f}}$, after both reaction and diffusion have been executed, between $\rho(\hat{\mathbf{f}}_h)$ (numerical solution of the fine-grid algorithm (17)) and $\rho(\hat{\mathbf{f}}_{2,h_2})$ (the output of the CxA model (20)):

$$E^{\mathrm{A}\to\mathrm{CxA}}(t_N) = \left\| \rho\left(\Pi_{h_2,h}\hat{\mathbf{f}}_h^{t_N}\right) - \rho\left(\hat{\mathbf{f}}_{2,h_2}^{t_N}\right) \right\| \tag{23}$$

(evaluating the difference on the coarser spatial grid), where $E^{\mathrm{A}\to\mathrm{CxA}}$ depends on the simulation parameters (in particular M, M_X). The scale-splitting error is purely a measure of a difference between discrete systems (the two algorithms). In case they are derived from an underlying physical process, as (1), $E^{\mathrm{A}\to\mathrm{CxA}}$ is a direct measure of the loss of accuracy. In fact, calling $E^{\mathrm{CxA},EX}(\rho)$ the error of the $(\mathbf{CxA}_H)$ model with respect to the exact solution of (1) and $E^{\mathrm{A},EX}(\rho)$ the error of the model $(\mathbf{A}_h)$, we can write

$$\left\|E^{\mathrm{A}\to\mathrm{CxA}}\right\| \leq \left\|E^{\mathrm{A},EX}(\rho)\right\| + \left\|E^{\mathrm{A},\mathrm{CxA}}(\rho)\right\|. \tag{24}$$

In the following part, we present two possible methodologies to investigate and estimate the scale-splitting error. The first method follows the asymptotic expansion technique, generalizing the conclusions presented in §2.2 to the CxA model (22). Since we cannot expect a full analysis to be available or feasible in general, we present a second error investigation approach which combines basic results of the analysis with the difference between the grid operators present in (17) and (20), and appears a more general technique to investigate *single-domain* CxA models.

We focus on the case $h_2 = h$ (only time coarsening). A general result based on the same approach for the case of CxA coarsened in space will be given at the end of the section. In this particular problem, the asymptotic expansion can be fully used to derive a prediction of the numerical solution of (20). To estimate the difference in the numerical results, we will then compare the CxA prediction with the fine scale prediction (13).

Considering the algorithm

$$\hat{\mathbf{f}}_h^{t_{n+1}} = \mathcal{D}_h(\Delta t_2)\left(\mathcal{R}_h(\Delta t)\right)^M \hat{\mathbf{f}}_h^{t_n}\,, \tag{25}$$

we aim to find a prediction using the ansatz

$$F_{(M),h}^{t_n}(j) = \mathbf{f}^{(0)}(t_n, x_j) + h\mathbf{f}^{(1)}(t_n, x_j) + \dots\,, \tag{26}$$

where $x_j = hj$, $t_n = Mnh^2$. First of all we observe that the reaction operator can be written as

$$\mathcal{R}_h(\Delta t) = \left(I_h - \frac{h^2}{2}\kappa(\mathbf{1} - \hat{\rho_\lambda})\right) \tag{27}$$

(recalling that $\Delta t = h^2$), introducing the matrix $\mathbf{1} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$ and denoting with $\hat{\rho_\lambda}$ a constant grid function such that $\hat{\rho_\lambda}(j) = \rho_\lambda(x_j)$. Observe that, for all $\mathbf{f}_h \in \mathcal{F}_h$,

$$\mathbf{1}\mathbf{f}_h = [\rho(\mathbf{f}_h), \rho(\mathbf{f}_h)]\,, \quad \rho\left(\tfrac{1}{2}\mathbf{1}f_h\right) = \rho(\mathbf{f}_h).$$

In virtue of these special properties, we obtain the following operator equality:

$$\begin{aligned}\left(I_h - \frac{h^2}{2}\kappa(\mathbf{1} - \hat{\rho_\lambda})\right)^M &= I_h + \sum_{m=1}^{M}\begin{pmatrix} M \\ m \end{pmatrix}(-1)^m\left(\frac{h^2\kappa}{2}\right)^m(\mathbf{1} - \hat{\rho_\lambda}) = \\ &= I_h + \left(1 - \left(1 - \frac{\kappa h^2}{2}\right)^M\right)(\mathbf{1} - \hat{\rho_\lambda})\,,\end{aligned} \tag{28}$$

which allows to rewrite (25) in a form analogous to the original (3):

$$\begin{aligned}\hat{f}_i^{t_{n+1}}(j + c_i) = \hat{f}_i^{t_n}(j) + \frac{1}{\tau_{(M)}}\left(f_i^{eq}(\rho(\hat{f}_h^{t_n}(j))) - \hat{f}_i^{t_n}(j)\right) \\ + \frac{Mh^2}{2}(-\kappa_{(M),h})(\rho(\hat{\mathbf{f}}^{t_n}(j)) - \rho_\lambda(x_j))\end{aligned} \tag{29}$$

(using also that $\Delta t_2 = Mh^2$), and

$$\kappa_{(M),h} = \frac{2}{Mh^2}\left(1 - \left(1 - \frac{\kappa h^2}{2}\right)^M\right). \tag{30}$$

For all h, we have $\kappa_{(1),h} = \kappa$. Moreover, the following Taylor expansion around $h = 0$ holds

$$\kappa_{(M),h} = \kappa + h^2\left(\frac{1-M}{4}\kappa^2\right) + O(h^3) \tag{31}$$

As before, we insert the expansion (26) into (29), deriving an expansion of the resulting residue. The main differences respect to the previous case (§2.2) are

- a time step Mh^2 must be taken into account, i.e. $t_n = Mnh^2$,
- the different relaxation parameter $\tau_{(M)}$,
- an h-dependent reaction constant expanded according to (31).

3.1.1 The M-dependent prediction

Omitting the details of the computations (based on the analogous of §2.2), we obtain the following expressions for the coefficients up to order $k = 3$:

$f_i^{(0)} = \frac{1}{2}\rho_0\,, \quad f_i^{(1)} = -\tau_{(M)}\frac{1}{2}c_i\partial_x\rho_0\,, \quad f_i^{(2)} = \frac{1}{2}\rho_{(M),2}$
(analogous to (12)),

$$\begin{aligned} f_i^{(3)} = &-\frac{\tau_{(M)}}{2}\left(-\left(M^2d^2 - \frac{1}{12}\right)c_i\partial_x^3\rho_0 - \kappa M\left(\tfrac{1}{2} - dM\right)c_i\partial_x(\rho_\lambda - \rho_0)\right) \\ &- \frac{\tau_{(M)}}{2}c_i\partial_x\rho_{(M),2}\,. \end{aligned} \tag{32}$$

Where ρ_0 solves (1) (and it is therefore independent from M). The derivation of coefficients of order h^3 is needed to specify $\rho_{(M),2}$ (the leading order of the error) which turns out to be solution of the following equation

$$\begin{aligned} \partial_t\rho_{(M),2} =& d\partial_x^2\rho_{(M),2} + \kappa\rho_{(M),2} \\ &+ C_1M^2d^3\partial_x^4\rho_0 + C_2M^2d^2\kappa\partial_x^2(\rho_0 - \rho_\lambda) + M\kappa^2(\rho_0 - \rho_\lambda) \\ &+ C_4M\kappa^2\rho_0 \end{aligned} \tag{33}$$

with bounded constants $C_1, \ldots, C_4$ (depending on $\tau_{(M)}$). The first three terms depending on ρ_0 come from the different relaxation time, while the fourth is due to the expansion of the reaction rate $\kappa_{(M),h}$.

By setting $M = 1$ in (29)-(33), the last result can be extended also to the classical LBM (3). Hence, we can evaluate the scale-splitting error as the difference $\rho_{2,(M)} - \rho_{2,(1)}$, which can be estimated using equation (33) (involving the time derivative of $\rho_{(M),2}$):

$$\rho\left(F_{(M),h} - F_{(1),h}\right) = O\left(M^2 d^3 + M^2 d^2 \kappa^2 \|\partial^2(\rho - \rho_\lambda)\| + M\kappa^2\right). \qquad (34)$$

Remarks. (i) The predictions differ already in order h. However, taking the moment $\rho(F_h)$ the first order in h (as well as the order h^3) vanishes, for any value of M. (ii) Estimate (34) contains both quadratic and linear terms in M. In particular conditions (depending on κ, d, $|\rho - \rho_\lambda|$) the linear growth can be more relevant than the quadratic one.

3.1.2 Estimates for Space-Coarsened CxA

A similar argument can be used also in case we use a coarser grid for the diffusion algorithm. Repeating the analysis, the expression of the coefficients (32) does not change (according to the hypothesis of h-independence). For the case $M_X^2 = M$ (Δt is coarsened, but τ remains unchanged), recalling that the first order coefficients do not give contribution to the moment, we obtain that the difference in the moment ρ of the expansions on different grid appears in the second order:

$$\rho\left(F_{(M,M_X),h} - F_{(1),h}\right) = O\left(h^2 M\right). \qquad (35)$$

3.2 Algebra of Discrete Operators

Without performing a full analysis, interesting estimates for the scale-splitting error can be derived based on the properties of the grid operators involved in ($\mathbf{A}_h$) and ($\mathbf{CxA}_H$).

Let us consider the CxA evolution written in the form (22). Since $\hat{f}_h$ is the solution of (17), we can rewrite (23) as

$$\begin{aligned} & E^{\mathrm{A}\to\mathrm{CxA}}(t_N) = \\ & \left\|\rho\left(\Pi_{h_2,h}\left(\mathcal{D}_h\mathcal{R}_h\right)^M \hat{\mathbf{f}}_h^{t_N - M\Delta t} - \mathcal{D}_{h_2}(\Delta t_2)\Pi_{h_2,h}\mathcal{R}_h^M \Lambda_{h,h_2}\hat{\mathbf{f}}_{2,h_2}^{t_N - \Delta t_2}\right)\right\| \end{aligned} \qquad (36)$$

Using $\Lambda_{hH}\hat{\mathbf{f}}_H = \Lambda_{h,h_2}\hat{\mathbf{f}}_{2,h_2}$ (a lift operator which only depends on the grid h_2), and remarking that $t_N - M\Delta t = t_N - \Delta t_2$, the latter can be decomposed as

$$\begin{aligned} & E^{\mathrm{A}\to\mathrm{CxA}}(t_N) \leq \\ & \leq \left\|\rho\left(\Pi_{h_2,h}\left(\mathcal{D}_h\mathcal{R}_h\right)^M \hat{\mathbf{f}}_h^{t_N - M\Delta t}\right) - \rho\left(\Pi_{h_2,h}\left(\mathcal{D}_h\mathcal{R}_h\right)^M \Lambda_{h,h_2}\hat{\mathbf{f}}_{2,h_2}^{t_N - \Delta t_2}\right)\right\| \\ & + \left\|\rho\left(\Pi_{h_2,h}\left(\mathcal{D}_h\mathcal{R}_h\right)^M \Lambda_{h,h_2}\hat{\mathbf{f}}_{2,h_2}^{t_N - \Delta t_2} - \mathcal{D}_{h_2}(\Delta t_2)\Pi_{h_2,h}\mathcal{R}_h^M \Lambda_{h,h_2}\hat{\mathbf{f}}_{2,h_2}^{t_N - \Delta t_2}\right)\right\| \end{aligned} \qquad (37)$$

The first part corresponds to the difference in the results of the algorithm $(\mathbf{A}_h)$ and another algorithm obtained starting with a coarse grid function $\hat{\mathbf{f}}_{2,h_2}^{t_N-\Delta t_2}$, lifting it to the fine grid and applying the same evolution rule as $(\mathbf{A}_h)$. If the involved operators are stable, the difference will remain small if the difference in the initial condition is small, i.e.

$$\begin{aligned}\left\|\rho\left(\Pi_{h_2,h}\left(\mathcal{D}_h\mathcal{R}_h\right)^M\hat{\mathbf{f}}_h^{t_N-M\Delta t}\right)-\rho\left(\Pi_{h_2,h}\left(\mathcal{D}_h\mathcal{R}_h\right)^M\Lambda_{h,h_2}\hat{\mathbf{f}}_{2,h_2}^{t_N-\Delta t_2}\right)\right\|\\ \leq C_{h,h_2}(\Lambda,\Pi,\mathcal{D},\mathcal{R})E^{\mathrm{A}\to\mathrm{CxA}}(t_N-\Delta t_2)\,,\end{aligned}\tag{38}$$

where $C_{h,h_2}(\Lambda,\Pi,\mathcal{D},\mathcal{R})$ depends on the RD operators and on the accuracy of the projection and lift operators, and $\hat{\mathbf{f}}_{2,h_2}^{t_N-\Delta t_2}$ is the scale-splitting error at the previous iteration.

Observing that in case of *local reaction*, it holds

$$\mathcal{R}_h^M\Lambda_{h,h_2}=\Lambda_{h,h_2}\mathcal{R}_{h_2}^M,\tag{39}$$

and that $\Pi_{h_2,h}\Lambda_{h,h_2}=I_{h_2}$ (projecting after lifting gives the same function), the second part of (37) can be simplified using

$$\mathcal{D}_{h_2}(\Delta t_2)\Pi_{h_2,h}\mathcal{R}_h^M\Lambda_{h,h_2}=\mathcal{D}_{h_2}(\Delta t_2)\mathcal{R}_{h_2}^M\,.\tag{40}$$

Hence,

$$\begin{aligned}&E^{\mathrm{A}\to\mathrm{CxA}}(t_N)\leq\\ &\left\|\rho\left(\Pi_{h_2,h}\left(\mathcal{D}_h\mathcal{R}_h\right)^M\Lambda_{h,h_2}-\mathcal{D}_{h_2}(\Delta t_2)\mathcal{R}_h^M\right)\right\|\left\|f_{2,h_2}^{t_N-\Delta t_2}\right\|+\\ &+C_{h,h_2}(\Lambda,\Pi,\mathcal{D},\mathcal{R})E^{\mathrm{A}\to\mathrm{CxA}}(t_N-\Delta t_2)\,.\end{aligned}\tag{41}$$

Let us focus on the case of pure temporal coarsening, i.e. $h_2=h$, $M_X=1$. The space-coarsened CxA involves more complicate algebra depending also on the choice of Π and Λ. As for the previous §3.1, this case will be partially discussed in a separate paragraph. Applying recursively the methodology described above to the error term $E^{\mathrm{A}\to\mathrm{CxA}}(t_N-\Delta t_2)$ in equation (41), one obtain that the distance between the numerical solutions can be estimated of the order of

$$\begin{aligned}E_{\mathrm{op}}^{\mathrm{A}\to\mathrm{CxA}}(M):=&\left\|\rho\left(\left(\mathcal{D}_h\mathcal{R}_h\right)^M-\mathcal{D}_h(\Delta t_2)\mathcal{R}_h^M\right)\right\|\leq\\ \leq&\left\|\rho\left(\left(\mathcal{D}_h\mathcal{R}_h\right)^M-\mathcal{D}_h^M\mathcal{R}_h^M\right)\right\|+\left\|\rho\left(\left(\mathcal{D}_h^M-\mathcal{D}_h(\Delta t_2)\right)\mathcal{R}_h^M\right)\right\|=\\ =&\,E^{(1)}(M)+E^{(2)}(M).\end{aligned}\tag{42}$$

Two contributions appear in the last expression. $E^{(1)}(M)$, is due to the commutation of the operators $\mathcal{D}_h$ and $\mathcal{R}_h$ on the same discrete grid, while $E^{(2)}(M)$ is related to the usage of a LBM with a different time scale for solving the diffusion process.

The difference $(\mathcal{D}_h\mathcal{R}_h)^M - \mathcal{D}_h^M\mathcal{R}_h^M$, 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 $\mathcal{D}_h$. The following results will be useful:

$$\Omega_{R_h}(I_h + \Omega_{D_h}) = \Omega_{R_h}. \tag{43}$$

Proof. It follows from the fact that Ω_{D_h} depends linearly on the difference $f^{eq}(\rho(\mathbf{f})) - \mathbf{f}$, i.e. for all $\mathbf{f}_h \in \mathcal{F}_h$

$$\rho(\Omega_{D_h}\mathbf{f}_h) = 0$$

(in other words, $\mathcal{D}_h$ conserves the moment). Therefore,

$$\rho(I_h + \Omega_{D_h})\mathbf{f}_h = \rho(\mathbf{f}_h) + \rho(\Omega_{D_h}\mathbf{f}_h) = \rho(\mathbf{f}_h),$$

which yields (43), since Ω_{R_h} depends on the moment of $\mathbf{f}$. □

In case of linear reaction, with $\rho_\lambda = 0$, (43) reduces to $\Omega_{R_h}\Omega_{D_h} = 0$.

Property (43) can be considered a complementary of the previously proved (16), here reported:

$$\forall \mathbf{f}_h \in \mathcal{F}_h : \Omega_{D_h}\Omega_{R_h}\mathbf{f}_h = 0.$$

To derive qualitative estimates from (42), we apply the different discrete operators to an ansatz

$$F_h = \mathbf{f}^{(0)} + h\mathbf{f}^{(1)} + h^2\mathbf{f}^{(2)} \tag{44}$$

with smooth coefficients. The hypothesis of smoothness allows us to simplify the expressions of operators, deriving explicit estimates which can be generalized to the numerical solution using an argument based on the asymptotic expansion technique, similarly to what has been presented in §2.2.

Lemma. For any smooth ansatz F_h it holds

$$[\mathcal{D}_h, \mathcal{R}_h](F_h) \in O\left(h^3\kappa\partial_x\rho(F_h)\right). \tag{45}$$

Proof. In virtue of (16) and (43) the operators $I_h + \Omega_{R_h}$ and $I_h + \Omega_{D_h}$ commute. Therefore

$$\begin{aligned}[\mathcal{D}_h, \mathcal{R}_h] &= P_h(I_h + \Omega_{D_h})(I_h + \Omega_{R_h}) - (I_h + \Omega_{R_h})P_h(I_h + \Omega_{D_h}) = && (46)\\ &= [P_h(I_h + \Omega_{R_h}) - (I_h + \Omega_{R_h})P_h](I_h + \Omega_{D_h}). && (47)\end{aligned}$$

Due to the linearity of P_h:

$$
\begin{aligned}
[\mathcal{D}_h, \mathcal{R}_h] &= (P_h\Omega_{R_h} - \Omega_{R_h}P_h)(I_h + \Omega_{D_h}) = \\
&= P_h\Omega_{R_h}(I_h + \Omega_{D_h}) - \Omega_{R_h}P_h(I_h + \Omega_{D_h}) = \\
&= P_h\Omega_{R_h} - \Omega_{R_h}P_h(I_h + \Omega_{D_h}),
\end{aligned} \tag{48}
$$

i.e. (47) depends on the commutator of P_h and Ω_{R_h}.

Computing the difference (48) on an ansatz F_h, the first term gives

$$
[P_h\Omega_{R_h}(F_h)]_i(j) = P_h\left[\frac{h^2}{2}R\left(\rho(F_h(j))\right)\right]_i = \frac{h^2}{2}R\left(\rho(F_h(j - c_i))\right). \tag{49}
$$

Using $F_{i,h}^{neq} = F_{i,h} - f_i^{eq}(\rho(F_h)) = F_{i,h} - \frac{1}{2}\rho(F_h)$, the second contribution reads

$$
\begin{aligned}
&[\Omega_{R_h}P_h(I_h + \Omega_{D_h})(F_h)]_i\,(j) = \left[\Omega_{R_h}P_h\left(F_{i,h}(j) - \frac{1}{\tau}F_{i,h}^{neq}(j)\right)\right]_i = \\
&= \Omega_{R_h}\left(F_{i,h}(j - c_i) - \frac{1}{\tau}F_{i,h}^{neq}(j - c_i)\right) = \\
&= \frac{h^2}{2}\left(\left(1 - \frac{1}{\tau}\right)\sum_i F_{i,h}(j - c_i) + \sum_i \tfrac{1}{2}\rho(F_h(j - c_i))\right).
\end{aligned} \tag{50}
$$

Using the smoothness of the coefficients of (44), terms involving grid shifting $j - c_i$ can be expanded around x_j. From (49) we have

$$
\begin{aligned}
[P_h\Omega_{R_h}(F_h)]_i &= \frac{h^2}{2}\left(R\left(\rho(F_h) - hc_i\partial_x\rho(F_h) + \frac{h^2}{2}\partial_x^2\rho(F_h) + O(h^3)\right)\right) = \\
&\frac{h^2}{2}(-\kappa)(\rho^{(0)}(F_h) - \rho_\lambda) + \frac{h^3}{2}(-\kappa)(c_i\partial_x\rho^{(0)}(F_h) + \rho^{(1)}(F_h)) \\
&+ O(h^4)\,,
\end{aligned} \tag{51}
$$

where we have used $\rho^{(0)}(F_h) = f_1^{(0)} + f_{-1}^{(0)}$, denoting the moment of the coefficients of order zero.

To evaluate the second part (50), we introduce the first order moment

$$
\phi(F_h) = \sum_i c_iF_{i,h}.
$$

Considerations based on the asymptotic expansion (see §2.2) allow to conclude that the leading order of F_h is proportional to the moment, and thus $\phi(F_h)$ is also proportional to $\partial_x\rho(F_h)$ (in leading orders):

$$
\phi(F_h) = h\partial_x\rho^{(0)}(F_h) + O(h^2).
$$

Hence, we have

$$\begin{aligned}&\left[\Omega_{R,h} P_h (I_h + \Omega_{D,h})(F_h)\right]_i (j) = \\ &= -\frac{h^2}{2}(-\kappa)(\rho^{(0)}(F_h) - \rho_\lambda) + \frac{h^3}{2}(-\kappa)\left(1 - \frac{1}{\tau}\right)\partial_x \phi^{(1)}(F_h) \\ &+ \frac{h^3}{2}(-\kappa)\rho^{(1)}(F_h) + O(h^4).\end{aligned} \tag{52}$$

Finally, (45) follows summing (49) and (50). □

Lemma. For any ansatz F_h (44)

$$\left\| \left[(\mathcal{D}_h \mathcal{R}_h)^M - \mathcal{D}_h^M \mathcal{R}_h^M \right] (F_h) \right\| \leq M(M-1) \left\| [\mathcal{D}_h, \mathcal{R}_h] \right\| \in O(M(M-1)\kappa h^3) \tag{53}$$

Proof. It follows counting the number of times the commutator appears in the difference $(\mathcal{D}_h \mathcal{R}_h)^M - \mathcal{D}_h^M \mathcal{R}_h^M$ (using a simple induction-based argument), and applying (45). □

From the proves it can be seen that the (53) is a quite rough estimate. However, it will be enough for our purposes of understanding the behavior of the error.

Time Coarsening Error

The part $E^{(2)}$ derives from the coarsening of the diffusion part in the original lattice Boltzmann algorithm. To analyze this contribution we use the asymptotic expansion to compute predictions of the numerical solutions of $\mathcal{D}_h(\Delta t)^M$ and $\mathcal{D}_h(\Delta t_2)$. evaluating the difference afterward.

Many of the results we proved earlier can be used again. First of all, since $\mathcal{D}_h(\Delta t)^M$ corresponds to the classical LBM diffusion operator applied M times, it is enough to consider the analysis for $M = 1$ and evaluate the prediction coefficients according to the time step $\Delta t_2 = Mh^2 = M\Delta t$.

We obtain the coefficients

$$\begin{aligned}&f_i^{(0)} = \tfrac{1}{2}\rho_0 \,, f_i^{(1)} = -\tau_{(M)} \tfrac{1}{2} c_i \partial_x \rho_0 \,, f_i^{(2)} = \tfrac{1}{2}\rho_{(M),2} \,, \\ &f_i^{(3)} = -\frac{\tau_{(M)}}{2}\left(-\left(M^2 d^2 - \frac{1}{12}\right) c_i \partial_x^3 \rho_0\right) - \frac{\tau_{(M)}}{2} c_i \partial_x \rho_{(M),2}\end{aligned} \tag{54}$$

(where $\tau_{(M)}$ depends on M as in (21)) analogous to (32), without reaction terms.

The function ρ_0 is such that $\partial_t \rho_0 = d\partial_x^2 \rho_0$, while $\rho_{(M),2}$ solves the following diffusion problem

$$\partial_t \rho_{(M),2} = d\partial_x^2 \rho_{(M),2} + C_1 M^2 d^3 \partial_x^4 \rho_0 \tag{55}$$

where C_1 is a bounded constant, depending on the parameters of the algorithm. Estimating the contribution to the scale splitting error can be taking the moment of the predictions for $M = 1$ and $M \neq 1$:

$$\rho\left(F_{(M),h} - F_{(1),h}\right) = h^2(\rho_{(M),2} - \rho_{(1),2}) + O(h^3),$$

from (55) we can conclude (similarly to what has been done in §3.1)

Lemma. For any $F_h \in \mathcal{F}_h$ in the form (7), we have

$$\rho\left(\mathcal{D}_h(\Delta t)^M - \mathcal{D}_h(M\Delta t)\right) F_h \in O(h^2 M^2 d^3). \tag{56}$$

Scale-splitting error as function of M: In conclusion, (53)-(56) yield a qualitative expectation of the error as a function of M, for a fixed h:

$$E_{\text{op}}^{\text{A}\to\text{CxA}}(M) \in O\left(M^2\kappa + M^2 d^3\right) . \tag{57}$$

Respect to the results of the full analysis performed in §3.1, equation (57) shows some differences. The linear term in M, obtained in (34) as a consequence of the modified reaction rate, is not present in (57). Also, (57) overestimates the components depending on M^2, since the dependence of the quadratic part on the difference $\rho - \rho_\lambda$ is missing.

4 Numerical tests

We consider problem (1) defined before, with the exact solution (2). By selecting different values of the parameters in (1) we can tune the relevance of different time scales. Additionally, we introduce the dimensionless parameter

$$\sigma = \frac{\kappa}{\lambda^2 d}$$

to "measure" the scale separation of the simulation.

We are primarily interested in the difference between the simulation results $\hat{f}_h$ of algorithm (3) and $\hat{f}_{2,h_2}$ of (20). After performing both simulations, we evaluate the scale-splitting error (depending on the simulation parameters) as

$$E_{\text{max}}^{\text{A}\to\text{CxA}}(h, D, \kappa, M) = \max_N \left\{ \frac{1}{N_x(h)} \left\| \rho\left(\hat{\mathbf{f}}_h^{t_N}\right) - \rho\left(\hat{\mathbf{f}}_{2,h}^{t_N}\right)\right\|_1 \right\} . \tag{58}$$

The error (58) will be also compared with

$$E^{A_h,EX} = \max_n \left\{ \frac{1}{N_x(h)} \left\| \rho\left(\hat{\mathbf{f}}_h^{t_n}\right) - \rho^*(t_n,\cdot))\right\|_1 \right\} , \tag{59}$$

i.e. the error of the original fully fine-discretized algorithm (17) (evaluated according to the norm chosen in (58)).

In fig. 2 we show the results for the scale-splitting error as a function of M, in several simulations fixing h and κ, and using different values of d (thus varying σ). As explained before, this is dictated by the fact that the grid size is strictly related to the reaction rate for stability reasons. In general, CxA models yield better results when the scale are more separated (for larger values of σ). Fig. 2a focuses on the range of moderate M, while 2b shows results for $M > 100$. We observe a super-linear increase of the error, becoming linear when M grows. This effect can be explained looking at the results of the asymptotic analysis (34). A possible reason is that the M^2-part of the error depends also from the difference between ρ and the reactive equilibrium ρ_λ, and becomes smaller than the part depending on $M\kappa^2$ if M and κ are large enough. Observe that the transition from quadratic to linear slope appears earlier, for smaller values of σ (κ small compared to d), while the slope is quadratic in the case $\sigma = 5$.

Concerning the derivation in §3.2, it is possible that, due to its simplicity, (57) overestimates the errors.

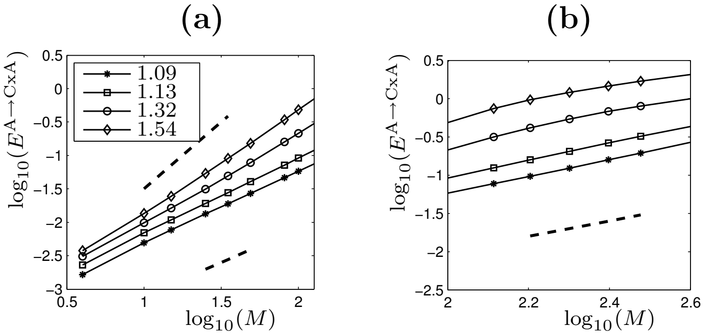

Figure 2. Scale-splitting error (58) as a function of M for a time-coarsened CxA, in double logarithmic scale. The different curves represent different values of σ: $\sigma = 0.5$ (diamonds), $\sigma = 1$ (circles), $\sigma = 2$ (squares), $\sigma = 5$ (stars). Simulation parameters: $h = 0.02$, $\lambda = 4\pi$, $\kappa = 10$, $D \in \{0.05, 0.1, 0.25, 0.5\}$. **(a)**: $1 < M < 100$. The dashed lines shows reference slopes 1 (bottom) and 2 (top). The approximated slopes (based on linear least squares fitting) are listed on the top left corner. **(b)**: $M > 100$.

Results of a further test to link together scale separation and scale-splitting error are shown in figure 3b-c. Namely, for each simulation drawn in figure 2, we select the first M such that the scale splitting error is below a certain threshold error $\bar{E}(h, H)$. These values M_{th} are plotted then as function of σ, validating the idea that larger scale separation (i.e. larger σ) allows more efficient CxA formulations.

A further argument to support the analysis is shown in figure 3c, where we draw the scale-splitting error as a function of M for a particular triple (h, κ, d), including also the results of a spatial-coarsened CxA and the error of the original LBM (17). The error for $M_X = \sqrt{M}$ increases linearly, also for small M (see (35)). Moreover, we observe that for a moderate range of M, the scale-splitting error is of the same magnitude of the discretization error. In other words the CxA model does not spoil, quantitatively, the accuracy of the original algorithm.

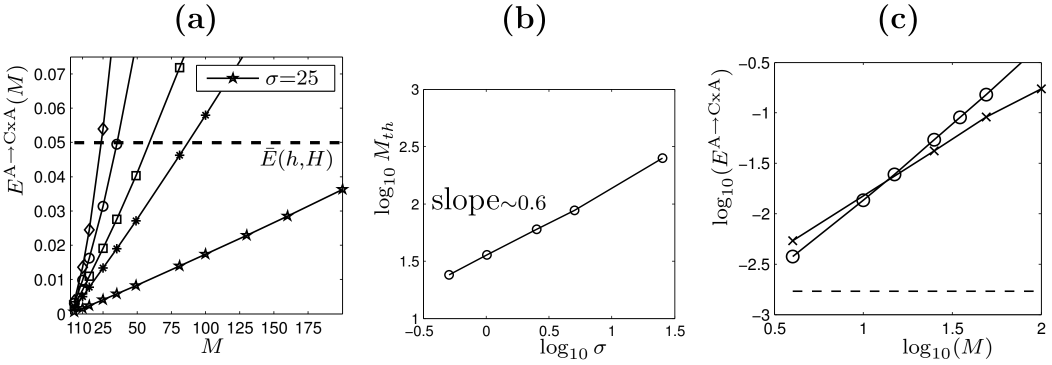

Figure 3. (**a**): Error as function of M, including a threshold error $\bar{E}(h, H) = 0.05$ (results with $\sigma = 25$ are also shown). (**b**): Values of M_{th} such that the scale-splitting error equates a threshold error $\bar{E}(h, H)$, versus the measure of scale separation $\sigma = \kappa \left(\lambda^2 D\right)^{-1}$ (in double logarithmic scale). (**c**): Error (58) for $M_X = 1$ ($\circ$) and $M_X = \sqrt{M}$ ($\times$) over a complete simulation, as a function of M. The dashed line is the error $\|E_h^{A_h,EX}\|$ of the original (fine-discretized) algorithm (17) ($\Delta x = h$, $\Delta t = h^2$) with respect to the exact solution (2).

5 Conclusions and Outlook

We have proposed a formal description of Complex Automata modeling and related error analysis. The approach has been validated on a simple reaction-diffusion benchmark, investigating theoretically and experimentally the scale-splitting error using

- a full asymptotic analysis, which yielded detailed information on the behavior of the scheme, but needs also more assumptions,
- an approach combining analysis and properties of discrete operators, more general, which in the relevant cases provides interesting estimates
- numerical simulations, comparing the results of a *multiscale algorithm* with a Complex Automata model

As a side result, the presented operator analysis leads to two interesting properties, (16) and (43), which holds for LBM applied to reaction-diffusion cases, and are relevant also independently from the CxA methodology.

The application of the analysis to CxA for more general multiscale models is a topic of ongoing research. Although we focused on a particular problem, we believe that the proposed formalism represents an important intermediate result for single-domain CxA, and sets the basis to construct a wider framework to analyze CxA models and of different type of multiscale coupling templates.

Acknowledgment This research is supported by the European Commission, through the COAST project (`www.complex-automata.org`, EU-FP6-IST-FET Contract 033664) .

References

[1] D. Alemani, B. Chopard, J. Galceran, J. Buffle. LBGK method coupled to time splitting technique for solving reaction-diffusion processes in complex systems. Phys. Chem. Chem. Phys., 7 (2005), 1–11.

[2] A. Caiazzo, J.L. Falcone, B. Chopard, A.G. Hoekstra. Error investigations in Complex Automata models for reaction-diffusion problems. Proceedings of Conference on Cellular Automata for Research and Industry, 2008, Lecture Notes in Computer Science, vol.5191, H. Umeo et al. (Eds), Springer-Verlag Berlin Heidelberg 2008, pp 260–267.

[3] S. Chen, G. D. Doolen. Lattice Boltzmann method for fluid flows. Annual Review of Fluid Mechanics (1998) pp. 329-364.

[4] D. Dab, J.-P. Boon, Y.-X. Li. Lattice-gas automata for coupled reaction-diffusion equations. Phys. Rev. Lett., 66 (1991), pp. 2535-2538.

[5] S. P. Dawson, S. Chen, and G. D. Doolen. Lattice Boltzmann computations for reaction-diffusion equations. J. Chem. Phys., 98 (1993), pp. 1514-1523.

[6] B. Chopard, M.Droz. Cellular Automata Modelling of Physical Systems. Cambridge University Press, Cambridge, 1998.

[7] B. Chopard, J.L. Falcone, A.G. Hoekstra, A. Caiazzo. On the collision-propagation and gather-update formulations of a Cellular Automata rule. Proceedings of Conference on Cellular Automata for Research and Industry, 2008, Lecture Notes in Computer Science, vol.5191, H. Umeo et al. (Eds), Springer-Verlag Berlin Heidelberg 2008, pp 144–151.

[8] Hegewald, J., Krafczyk, M., Tölke J., Hoekstra, A.G. and Chopard, B.: An Agent-Based Coupling Platform for Complex Automata. Proceedings of 8th ICCS, Lecture Notes in Computer Science, vol. 5102, Springer-Verlag Berlin Heidelberg 2008, pp 227–233. Software sources and documentation available at `www.coast-dscl.berlios.de`.

[9] A.G. Hoekstra, E. Lorenz, J.L. Falcone, B. Chopard. Towards a Complex Automata Formalism for Multi-Scale Modeling. Int. J. Mult. Comp. Eng., 5 (2007), 491–502.

[10] A.G. Hoekstra, J.L. Falcone, A. Caiazzo, B. Chopard. Multiscale Modeling with Cellular Automata: the Complex Automata Approach. Proceedings of Conference on Cellular Automata for Research and Industry, 2008, Lecture Notes in Computer Science, vol.5191, H. Umeo et al. (Eds), Springer-Verlag Berlin Heidelberg 2008, pp 192–199.

[11] M. Junk, A. Klar, L.-S. Luo. Asymptotic analysis of the lattice Boltzmann Equation. J. Comp. Phys., 210 (2005), 676–704.

[12] Succi, S. The Lattice Boltzmann Equation for Fluid Dynamics and Beyond. Oxford University Press, Oxford, 2001.

[13] P. Van Leemput, C. Vandekerckhove, W. Vanroose, D. Roose. Accuracy of hybrid lattice Boltzmann-finite difference schemes for reaction-diffusion systems. (SIAM) Multiscale Modeling and Simulation, 6(3) (2007), p. 838–857.