Exploring Synthetic Mass Action Models*

Oded Maler[1], Ádám M. Halász[2], Olivier Lebeltel[1], and Ouri Maler[2]

[1] VERIMAG
CNRS and the University of Grenoble-Alpes
France
[2] Department of Mathematics
West Virginia University
USA

Abstract. In this work we propose a model that can be used to study the dynamics of mass action systems, systems consisting of a large number of individuals whose behavior is influenced by other individuals that they encounter. Our approach is rather synthetic and abstract, viewing each individual as a probabilistic automaton that can be in one of finitely many discrete states. We demonstrate the type of investigations that can be carried out on such a model using the *Populus* toolkit. In particular, we illustrate how sensitivity to initial spatial distribution can be observed in simulation.

1 Introduction

Mass action is a fundamental notion in many situations in Chemistry, Biochemistry, Population Dynamics and Social Systems [2]. In this class of phenomena, one has a large population of individuals partitioned into several types of "species", whose dynamics is specified by a set of reaction rules. Each reaction indicates the transformation that is likely to take place when individuals of specific types come into contact. For example, a rule of the form

$$A + B \;\rightarrow\; A + C$$

says that when an instance of A meets an instance of B, the latter is transformed into C. Denoting by n_A and n_B the number of instances of A and B existing at a certain moment, the likelihood of an (A, B)-encounter is proportional to $n_A \cdot n_B$. Hence the rate of change of n_B will have a *negative* contribution proportional to $n_A \cdot n_B$ and that of n_C will have the same magnitude of *positive* contribution. Combining for each of the species the negative contributions due to reactions in which it is transformed into something else with the positive contributions due to reactions that yield new instances of it, one can obtain a system of *polynomial*[1] differential/difference equations.

Hybrid systems research led in the past to interactions between several branches of Computer Science and Control that have resulted in new ways to specify and analyze

* This work was partially supported by the French ANR project Cadmidia and the NIH Grants K25 CA131558 and R01 GM104973. Part of the work done while the second author was visiting CNRS-VERIMAG. This paper is an extended version of [14].

[1] Actually *bilinear* if one assumes the probability of triple encounters to be zero, as is often done in Chemistry.

the behavior of complex dynamical systems [12, 13]. The present paper is a preliminary step in a research program, initially inspired by [5], to pull into the CS sphere of ideas, additional domains currently dominated by the culture of Applied Mathematics and Scientific Computing, most notably the modeling and simulation of chemical reactions inside the cell [1, 8, 4, 9].

Our approach is *top-down* and *synthetic* in the sense of defining a class of *general* mathematical models for such systems, inspired by common knowledge on the way chemical reactions work but still abstracting away from many problem-specific details due to Chemistry, Physics and even some Geometry. We believe that better conceptual and computational insights can be achieved on cleaner models, focused on what we view as essential features of the phenomenon, before adding the additional details associated with each concrete problem. We hope that investigating such models will eventually lead to novel ways to simulate and control mass action systems with potential applications, among others, in drug design and social engineering. These issues have been studied, of course, for many years in various contexts and diverse disciplines, [11, 3] to mention a few, but we hope, nevertheless and despite the present apparently naive beginning, to provide a fresh look at the subject.

The rest of this paper is organized as follows. In Section 2 we present the basic model of the individual agent (particle) as a *probabilistic automaton* capable of being in one out of several states, and where transition labels refer to the state of the agent it encounters at a given moment. In Section 3 we discuss several ways to embed these individual agents in a model depicting the evolution of a large ensemble of their instances. In Section 4 we describe three such aggregate models, starting with a rather standard model where state variables correspond to the relative concentrations of particle types. Such models depict the dynamics of the *average* over all behaviors and they are typically realized by ordinary differential equations (ODEs) but we prefer to work in discrete time.

The second model is based on stochastic simulation under the well-stirred assumption. The third model embeds the particles in space where they perform some type of random motion and encounters may occur when the distance between two particles becomes sufficiently small. The model thus obtained is essentially a kind of a *reaction-diffusion* model for a restricted class of reactions. In Section 5 we briefly describe the **Populus** tool kit that we developed for exploring the dynamics of such models and illustrate its functionality. In particular, we demonstrate the effects of the initial spatial distributions of certain particle types which result in deviations from the behavior predicted by a well-stirred version of the model.

2 Individuals

We consider population-preserving mass action systems where new individuals are not born and existing ones neither die nor aggregate into compound entities: they only change their internal *state*. A particle can be in one of finitely-many states and its (probabilistic) dynamics depicts what happens to it every time instant, either spontaneously

or upon encountering another particle. The object specifying a particle is a probabilistic automaton:[2]

Definition 1 (Probabilistic Automaton). *A probabilistic automaton is a triple* $\mathcal{A} = (Q, \Sigma, \delta)$ *where* Q *is a finite set of states,* Σ *is a finite input alphabet and* $\delta : Q \times \Sigma \times Q \to \mathbb{R}$ *is a probabilistic transition function such that for every* $q \in Q$ *and* $a \in \Sigma$*,*

$$\sum_{q' \in Q} \delta(q, a, q') = 1.$$

In our model $Q = \{q_1, \ldots, q_n\}$ is the set of particle types and each instance of the automaton is always in one of those. The input alphabet is $Q \cup \{\perp\}$ intended to denote the type of *another* particle encountered by the automaton and with the special symbol $\perp$ indicating a non-encounter. An entry $\delta(q_1, q_2, q_3)$ specifies the probability that an agent of type q_1 converts to type q_3 given that it encounters an agent of type q_2. Likewise $\delta(q_1, \perp, q_3)$ is the probability of converting into q_3 spontaneously without meeting anybody. We use the notation $q_1 \xrightarrow{q_2} q_3$ for an actual invocation of the rule, that is, drawing an element of Q according to probability $\delta(q_1, q_2, .)$ and obtaining q_3 as an outcome.

Table 1 depicts a 3-species probabilistic automaton. Looking at the diagonal of the $\perp$ matrix we can observe that the three species are rather stable in isolation. On the other hand, they may influence each other significantly upon encounter. For instance, q_3 transforms q_1 to q_3 with probability 0.3 while q_1 transforms q_2 to q_3 and q_3 to q_1 with probabilities 0.4 and 0.7, respectively.

δ	$\perp$			q_1			q_2			q_3		
	q_1	q_2	q_3	q_1	q_2	q_3	q_1	q_2	q_3	q_1	q_2	q_3
q_1	0.9	0.1	0.0	1.0	0.0	0.0	0.7	0.2	0.1	0.7	0.0	0.3
q_2	0.1	0.8	0.1	0.0	0.6	0.4	0.0	1.0	0.0	0.1	0.9	0.0
q_3	0.0	0.0	1.0	0.7	0.0	0.3	0.3	0.4	0.3	0.0	0.0	1.0

Table 1. A 3-species probabilistic automaton.

Our models are *synchronous* with respect to time: time evolves in fixed-size steps and at every step each particle detects whether it encounters another (and of what type) and takes the appropriate transition. The definition of when an agent meets another depends, as we shall see, on additional assumptions on the global aggregate model.
Remark: It is worth noting that we restrict ourselves to reaction rules which are "locally causal" in the following sense: when an (A, B)-encounter takes place at time t, the state of A at time $t + 1$ is does not depend on the state of B at $t + 1$ and vice versa: states of particles at time $t+1$ depend only on states at t. Compared to more general probabilistic

[2] A probabilistic automaton [15] is a Markov chain with an input alphabet where each input symbol induces a different transition matrix. It is also known as a Markov Decision Process (MDP) in some circles.

rewrite rules that can specify the outcome of an (A, B)-encounter, our formalism can express rules which are products of simple rules. For instance, in a general rule like

$$A + B \to A_1 + B_1\ (p_{11}) \mid A_1 + B_2\ (p_{12}) \mid A_2 + B_1\ (p_{21}) \mid A_2 + B_2\ (p_{22})$$

the probabilities of the four outcomes should sum up to 1 while in our formulation they should satisfy the additional condition $p_{11}/p_{12} = p_{21}/p_{22}$. This restriction is not crucial for our approach but it simplifies some calculations.

3 Aggregation Styles

Consider now a set S consisting of m individuals put together, each being modeled as an automaton. The set of all possible global configurations of the system (micro-states in Physpeak) is the set Q^S of all functions from S to Q. This is an enormous state space of size n^m. A very useful and commonly-used abstraction is the *counting abstraction* obtained by considering two micro-states equivalent if they agree on the number of particles of each type, regardless of their particular identity. The equivalence classes of this relation form an abstracted state-space P of macro-states (also known as particle number representation) each being an n-dimensional vector:

$$P = \{(X_1, \ldots, X_n) : \forall i\ 0 \leq X_i \leq m \wedge \sum_{i=1}^{n} X_i = m\}.$$

Models that track the evolution of an ensemble of particles are often viewed as dynamical systems over this abstract state-space.

For our purposes we classify models according to two features: 1) Individual vs. average dynamics; 2) Spatially-extended vs. non-spatial (well-stirred) dynamics. For the first point, let us recall the trivial but important fact that we have a non-deterministic system where being in a given micro-state, each particle tosses one or more coins, properly biased according to the states of the other particles, so as to determine its next state. To illustrate, consider a rule which transforms a particle type A into B with probability p. Starting with m instances of A, there will be m coin tosses each with probability p leading to some number close to $m \cdot p$ indicating how many A's convert into $B's$. Each individual run will yield a different number (and a different sequence of subsequent numbers) but on the average (over all runs) the number of A's will be reduced in the first step from m to $m \cdot (1 - p)$.

Individualistic models, those used in stochastic simulation algorithms (SSA), generate such runs, one at a time. On the other hand, "deterministic" ODE models compute at every step the average number of particles for each type where this average is taken in parallel over all individual runs . For well-behaving systems, the relationship between this averaged trajectory and individual runs is of great similarity: the evolution in actual runs will appear as fluctuating around the evolution of the average. On the other hand, when we deal with more complex systems where, for example, trajectories can switch into two or more distinct and well-separated equilibria, the behavior of the average is less informative, especially when the number of molecules is small. There is a

whole research thread, starting with [6], that feeds on this important distinction (see for example, [16, 10] for further discussions).

The other issue is whether and how one accounts for the distribution of particles in space. Ignoring the spatial coordinates of particles, the probability of a particular type of encounter depends only on the total number of particles of each type, which is equivalent to the well-stirred assumption: all instances of each particle type are distributed uniformly in space and hence all particles will see the same proportion of other particles in their neighborhood. This is very convenient computationally because we can work directly on the abstract state-space of particle counts. On the other hand, in spatially extended models each particle is endowed with a location which changes quasi-randomly and what it encounters along its moving neighborhood determines the interactions it is likely to participate in. Such a particle will be exposed to what happens locally along its trajectory rather than to the global number of particles. Note that embedding particles in an Euclidean-like space is just one possibility and one can think or putting them in more abstract graph-based space where distance between locations is defined by the length of shortest path, as is common in social models.

4 Implemented Aggregate Models

We will now describe in some detail the derivation of three models: average dynamics, individual well-stirred dynamics and spatially-extended dynamics. All our models are in discrete time which will hopefully make them more accessible to those for whom the language of integrals is not native. For the others, note that our model corresponds to a fixed time-step simulation.

4.1 Average Well-Stirred Dynamics

To develop the average dynamics under the well-stirred assumption we normalize the global macro-state of the system, a vector $X = (X_1, \ldots, X_n)$, into $x = (x_1, \ldots, x_n)$ with $x_i = X_i/m$ and hence $\sum x_i = 1$ (population fractions). Let α, $0 \leq \alpha \leq 1$ be a density parameter which determines the probability of encountering another particle in one step. The evolution in this state space over time is the outcome of playing the following protocol at every time step. First, $\alpha \cdot m$ of the particles on average encounter others and hence follow a binary reaction rule while the remaining $(1-\alpha) \cdot m$ particles do not interact and hence follow the solitary transition function. The dynamics is of the general form[3]

$$x' = x + \Delta(x),$$

where for each variable, the additive change can be written as

$$\Delta(x_k) = (1-\alpha)\Delta_1(x_k) + \alpha\Delta_2(x_k)$$

[3] We export the primed variable notation from program verification where x stands for $x[t]$ and x' denotes $x[t+1]$.

where

$$\Delta_1(x_k) = \sum_{i=1}^{n}(x_i \cdot \delta(q_i, \perp, q_k) - x_k \cdot \delta(q_k, \perp, q_i))$$

$$\Delta_2(x_k) = \sum_{i=1}^{n}\sum_{j=1}^{n}(x_i x_j \cdot \delta(q_i, q_j, q_k) - x_k x_i \cdot \delta(q_k, q_i, q_j))$$

Here, Δ_1 and Δ_2 are the expected net contributions to x_k by the solitary (resp. binary) reactions, each summing up the transformations of other agents into type k minus the transformation of type k into other types. Thus, we obtain a discrete-time bilinear dynamical system, which is linear when $\alpha = 0$.

Taking the particle automaton of Table 1 and deriving the dynamics for the sparse situation where $\alpha = 0.1$, we obtain

$$\begin{aligned} x_1' &= x_1 - 0.09x_1 + 0.09x_2 - 0.06x_1x_2 + 0.08x_1x_3 + 0.08x_2x_3 \\ x_2' &= x_2 + 0.09x_1 - 0.18x_2 - 0.04x_1x_2 + 0.06x_2x_3 \\ x_3' &= x_3 + 0.09x_2 + 0.1x_1x_2 - 0.08x_1x_3 - 0.14x_2x_3 \end{aligned} \tag{1}$$

Starting from initial state $x = (0.4, 0.3, 0.3)$ and following the dynamics, the system converges to the state $(0.366, 0.195, 0.437)$. The same individual model, in a dense situation characterized by $\alpha = 0.9$, yields

$$\begin{aligned} x_1' &= x_1 - 0.01x_1 + 0.01x_2 - 0.54x_1x_2 + 0.72x_1x_3 + 0.72x_2x_3 \\ x_2' &= x_2 + 0.01x_1 - 0.02x_2 - 0.36x_1x_2 + 0.54x_2x_3 \\ x_3' &= x_3 + 0.01x_2 + 0.9x_1x_2 - 0.72x_1x_3 - 1.26x_2x_3 \end{aligned} \tag{2}$$

This system, started in the same initial state $x = (0.4, 0.3, 0.3)$, converges to state $(0.939, 0.027, 0.033)$. Let us point out once more that this deterministic dynamics tracks the evolution of the average population fraction of particles over all individual runs.

4.2 Individual Well-Stirred Dynamics

The second model, whose average behavior is captured to some extent by the previous one, generates individual behaviors without spatial information following Algorithm 1. A micro-state of the system is represented as a set L of particles, each denoted as (g, q) where g is the particle identifier and q is its current state. The update round of the algorithm consists of repeatedly choosing a particle g and deciding probabilistically whether it interacts with another particle. If this is the case, another particle g' is randomly selected, and they both undergo their respective probabilistic binary reactions. Otherwise, g is subject to a unary reaction. Reacting particles are removed from the list and the process terminates when the list becomes empty. Other variants of this procedure are mentioned in the next section.

Algorithm 1 (Individual Well-Stirred Dynamics)
Input*: A list L of particles and states*
Output *A list L' representing the next micro-state*

$L' := \emptyset$
repeat
  draw *a random particle* $(g, q) \in L$*;* $L := L - \{(g, q)\}$
  draw *binary/solitary with probability* α
  if *solitary* **then**
    apply solitary rule $q \xrightarrow{\perp} q'$
    $L' := L' \cup \{(g, q')\}$
  else
    draw *a random particle* $(g', q') \in L$*;* $L := L - \{(g', q')\}$
    apply binary rules $q \xrightarrow{q'} q''$ *and* $q' \xrightarrow{q} q'''$
    $L' := L' \cup \{(g, q''), (g', q''')\}$
  endif
until $L = \emptyset$

After each update round, particle types are counted to create macro-states. The current description and implementation of the algorithm is at the level of micro-states and a more efficient stochastic simulation algorithm working directly on macro-states is discussed in Section 6.

4.3 Individual Spatial Dynamics

In our third model the particles are embedded in space, each particle represented as (g, q, y) with y being it spatial coordinates ranging over a bounded rectangle. The next state is computed in two phases that correspond to diffusion and reaction. First, each particle is displaced by a vector of random direction and magnitude (bounded by a constant s). For mathematical convenience reasons we use periodic boundary conditions that treat the rectangle as a torus: when a particle crosses the boundary of the rectangle it reappears on the other side.

Then for each particle we compute its set of neighbors N, those residing in a ball of a pre-specified *interaction radius* r, typically in the same order of magnitude as s. When N is empty the particle undergoes a unary reaction, otherwise it interacts with a randomly chosen particle in N, as describe in Algorithm 2. Other variants of the algorithm may differ by not taking the complementary transition $q' \xrightarrow{q} q'''$, and by not removing g and g' from L (such variations apply also to the well-stirred model). We have observed empirically that these variations did not influence model behavior significantly. We should point out that the parameters we used so far rendered the situation rather dense with the typical distance between particles comparable to the interaction radius, frequently resulting in particles having multiple neighbors.

Algorithm 2 (Individual Spatial Dynamics)
Input*: A list L of particles and states including planar coordinates*
Output *A list L' representing the next micro-state*

$L' := \emptyset$
foreach *particle* $(g, q, y) \in L$
  draw *randomly* $h \in [0, s]$ *and* $\theta \in [0, 2\pi]$
  $y := y + (h, \theta)$
endfor
repeat
  draw $(g, q, y) \in L$
  $L := L - \{(g, q, y)\}$
  $N := \{(g', q', y') \in L : d(y, y') < r\}$
  if $N = \emptyset$ **then**
    apply solitary rule $q \xrightarrow{\perp} q'$
    $L' := L' \cup \{(g, q', y)\}$
  else
    draw $(g', q', y') \in N$
    $L := L - \{(g', q', y')\}$
    apply binary rules $q \xrightarrow{q'} q''$ *and* $q' \xrightarrow{q} q'''$
    $L' := L' \cup \{(g, q'', y), (g', q''', y')\}$
  endif
until $L = \emptyset$

The connection between this model, embedded in a rectangle of area w, and the non-spatial ones can be made via the estimation of the density factor α. The probability of a particle g not interacting with another particle g' is the probability of g' being outside an interaction ball, that is, $\beta = (\pi r^2)/w$, and the odds of g not interacting with any of the other $m - 1$ particles is $(1 - \beta)^{m-1}$. After the reaction the number of remaining particles is either $m - 1$ or $m - 2$ and a good estimation of the average probability to interact is:

$$\alpha \approx 1 - \frac{\sum_{i=1}^{m-1}(1-\beta)^i}{m} = 1 - \frac{1 - (1-\beta)^m)}{m\beta}.$$

5 The Populus Toolkit: Preliminary Experiments

We developed a prototype tool called **Populus**, written in Java and Swing, for exploring such dynamics. The input to the tool is a particle automaton (Definition 1, Tab. 1) along with additional parameters such as the dimensions of the rectangle where particles live, the geometric step size s, the interaction radius r and the initial number of each particle type, possibly restricted to some sub-rectangles. The tool simulates the three models, average well-stirred, individual well-stirred and spatially-extended, plots the evolution of particle counts over time and animates the spatial evolution. We illustrate below the type of exploration made available by the tool.

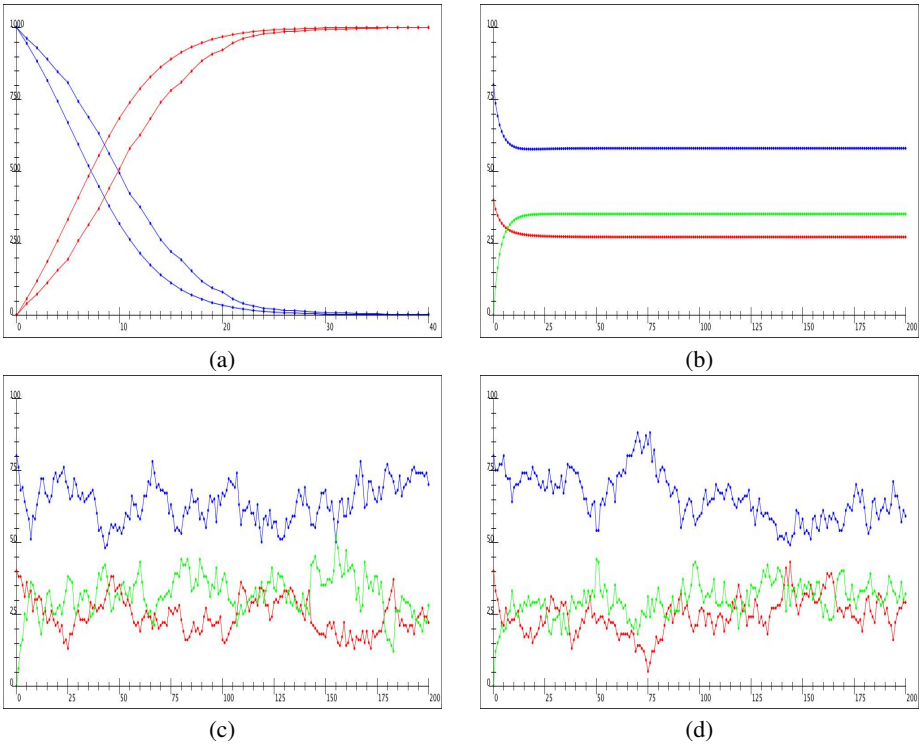

Fig. 1. (a) The evolution of a simple system where A is eventually transformed to B. The smoother curves depict the average while the other curve shows an individual trajectory. (b-d) A 3-species system where the average trajectory (b) stabilizes rapidly while individual well-stirred (c) and spatial (d) trajectories fluctuate.

5.1 Average and Individual Dynamics

Fig. 1 compares the behaviors of the averaged model (deterministic) and the individual model (stochastic simulation) for two systems: a simple one where the two models exhibit similar behaviors and a more complex one where the average behavior stabilizes rapidly while individual trajectories, well-stirred and spatial, fluctuate.

5.2 Spatial and Well-stirred

To demonstrate the difference between spatial and non-spatial models we simulated a system with 5 species, A to E. Particle type A does not change in any transition except when it meets B and converts into C in certainty: $\delta(A, B, C) = 1$ and $\delta(A, \cdot, A) = 1$ otherwise. Particle type B is unstable and each step it transforms into D with probability 0.5: $\delta(B, \cdot, B) = \delta(B, \cdot, D) = 0.5$. Consequently it is destined to disappear after some time. Particle type C is fully stable: $\delta(C, \cdot, C) = 1$. It converts D, which is stable under all other interactions, into E: $\delta(D, C, E) = 1$ and $\delta(D, \cdot, D) = 1$, otherwise.

Particle type E is almost stable and it transforms back into D with probability 0.01: $\delta(E, \cdot, E) = 0.99$ and $\delta(E, \cdot, D) = 0.01$.

The logic behind this example is the following: there is a transient phase until B disappears completely, leaving behind a number of C's equal to the number of (A, B) encounters that have occurred. The number of the non-converted A's remains constant thereafter. Then, there are transformations of D to E that depend on the number of C's, and interaction-independent transformations in the opposite direction. Such a system is naturally sensitive to the initial spatial distribution of A and B that will determine how many C's will be eventually produced. We simulated the model starting from an initial state where there are 50 instances of A, 50 instances of B and 200 instances of D. We used a 20×20 square over which D was distributed uniformly under three scenarios that differ in the initial distribution of A and B in space:

- Scenario 1: A and B are distributed uniformly all over space;
- Scenario 2: A and B are concentrated initially in a unit square in the middle;
- Scenario 3: A and B are concentrated inside distinct unit squares far apart from each other.

The results of some runs for these scenarios are shown in Fig. 2. As a first observation, the behavior of the average model (a) and a well-stirred stochastic simulation (b) are quite similar. The number of C's produced in these models with $\alpha = 0.9$ is around 13. The results of the spatial simulation of scenario 1 (c) are quite similar. In the two runs of scenario 2 (d,e), due to the proximity of A and B there is a burst of (A, B) encounters at the beginning leading to 30 and 36 instances of C and to higher levels of E than under the well-stirred assumption. Finally, in scenario 3 (f) all B's disappear before meeting an A and hence no C nor E is produced.

6 Conclusions and Future Work

We presented a framework for studying abstract mass action dynamics based on a finite-state automaton model of the individual agent. We demonstrated how questions such as the sensitivity of some reactions to initial spatial distribution can be investigated using such models. The design philosophy underlying our framework is that all potential actions and reactions that change the state of the particle are written inside its individual model. Then the invocation of these rules by instances of the population depends on the assumptions underlying the aggregate model, its state update algorithm and the particular global state of the system.

The current framework can be improved and extended in several directions that we mention below, some being currently under investigation.

Non Mutual Interactions and Abstract Geometries Our model assumed that when particles g and g' interact, both of them undergo reactions. In many situations such as epidemiology or social systems, the influence may work only in one way. As mentioned, this involves only minor changes in Algorithms 1 and 2 and may even simplify their probabilistic analysis. Such applications may need other notions of distance and neighborhood that reflect, for example, the separation distance between two individuals.

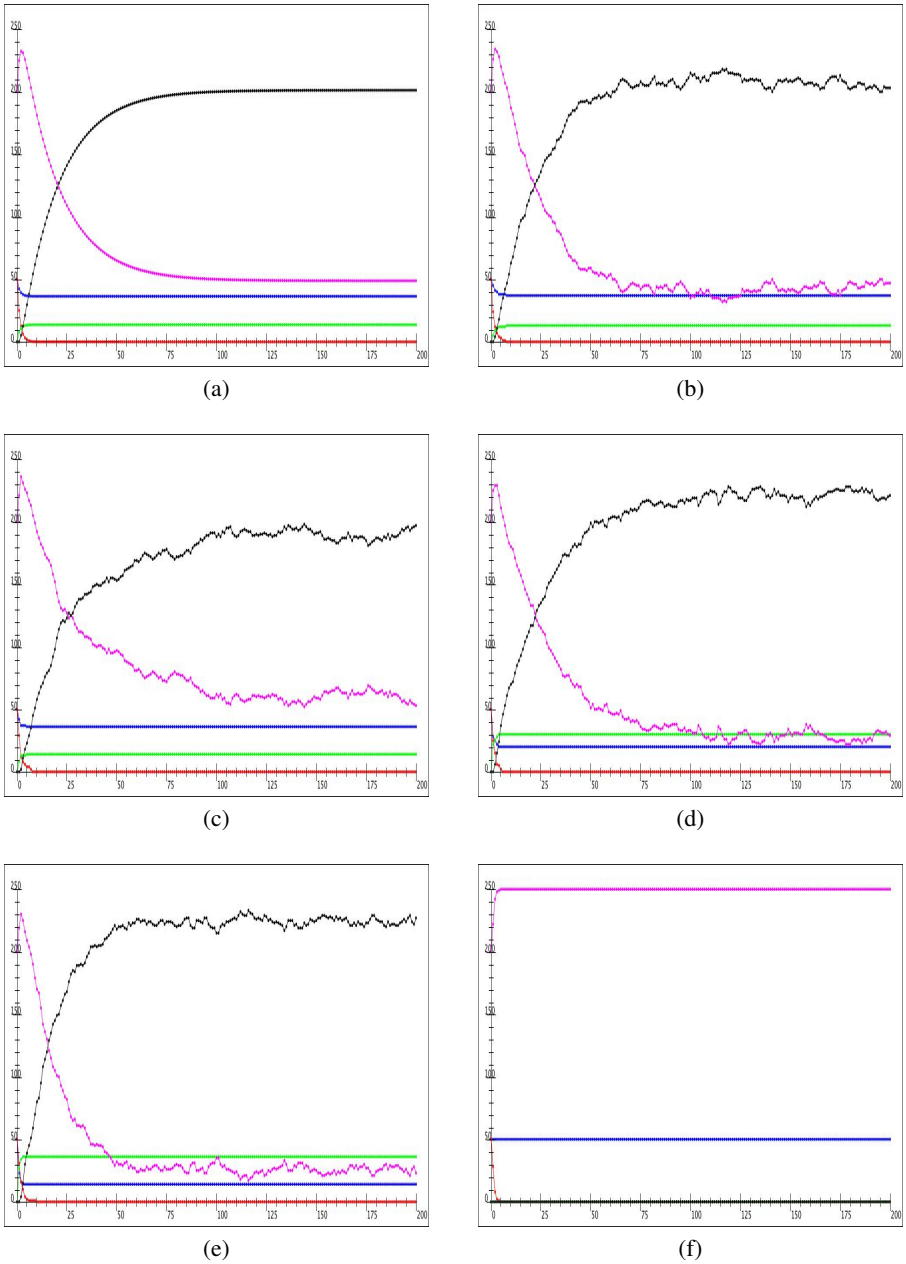

Fig. 2. The behavior of the 5-species system: (a) average well-stirred model; (b) an individual well-stirred run; (c) a run of the spatial model scenario 1; (d,e) two runs of scenario 2; (f) a run of scenario 3.

Stochastic Simulation based on Macro-States Algorithm 1 for well-stirred stochastic simulation works at the micro-state level, iterating over all particles and updating them one by one, resulting in $O(m)$ complexity. This fact might limit its applicability when some species exist in very large numbers. We are currently working on alternative update protocols where the determination of the set of particles that undergo binary (and unary) reactions, as well as its partitioning into pairs, are done in one preliminary step preceding the reaction step. Based on this scheme it might be easier to derive a more efficient simulation algorithm that works directly on macro-states that we sketch below (similar ideas underlie the τ-leaping algorithm of [7]). The update rule for such an algorithm will have the form[4] $x' = x + \Delta(x)$ where $\Delta(x)$ is a random variable over the space of increment vectors which depends on the current macro-state. An increment vector is $\Delta = (\Delta_1, \ldots, \Delta_n)$ such that $\Delta_i \in [0, 1]$ for every i and $\sum_i \Delta_i = 0$. The derivation of increment probabilities involves several steps. Assuming α of the particles participate in binary reactions, one needs to derive a probability over vectors $u = (u_1, \ldots, u_m)$ satisfying $\sum_i u_i = \alpha$, with u_i indicating how many of those α particles are of type q_i. For each u we need further to compute a probability over the different ways to partition it into pairs of particles, that is, a probability over vectors of the form $(v_{11}, \ldots, v_{nn})$ satisfying $\sum_i \sum_j v_{ij} = \alpha/2$, with v_{ij} being the number of pairs (q_i, q_j) that react together. This will give us probabilities over all the encounter types that, together with the rules of the automaton, can be used to derive probabilities over the increments.

More Efficient Spatial Simulation The complexity of our current naive implementation of the spatial simulation algorithm is $O(m^2)$ as we need to compute the neighbors of each particle by scanning L. This complexity can be reduced to the necessary minimum by employing spatial data structures that can reduce the number of candidates for neighborhood that are checked. More radical performance improvements can be achieved by changing the semantics. Rather than following each particle in space we can use a more coarse-grained simulation in the spirit of the finite-element method. It consists in partitioning space into boxes and assuming each box to be well-stirred, hence represented by its (local) macro-state. The diffusion phase can be realized by particles flowing among neighboring boxes in rates proportional to their macro-state gradients. The reaction phase will consist in applying Algorithm 1 inside each box. When the size of the boxes tends to zero we obtain the current spatial algorithm and when it is the whole W we have the well-stirred algorithm. Using box sizes situated between these two extremes one can define and explore models that represent the whole spectrum between the well-stirred and not stirred assumptions. Such algorithms will be more efficient than Algorithm 2 and will allow us to perform simulations with a richer range of ratios between density, velocity and interaction radius.

Enriching the Model In the longer run we will consider more substantial extensions of the model of the individual. So far the movement of particles in the spatial model assumes the same speed for all particle types. This assumption seems to be relaxable

[4] We write the algorithm using the normalized state notation x but the combinatorial calculation underlying the derivation of probabilities will be based on the particle count X.

without much effort. Let us also note that movement in our model is very abstract, not influenced by local densities of different particle types and it remains to be seen how such aspects can be incorporated while avoiding full-fledged kinetic simulation.

Finally, we adhere so far to population-preserving reactions where particles only change their state but do not combine together to form new entities. Given that the creation of new entities and structures is primordial for chemical and biological systems, we should look at this feature of our modeling framework as a limitation that should some day be removed. More generally, one can observe a tension between two types of models. Models of the first type are more realistic and faithful to one or more concrete physical phenomenon. They are, however full of details that may hide the forest and yield cumbersome simulation procedures. Models of the second type are cleaner and consequently are more amenable to systematic study that may even lead to some mathematical statements and general principles, at the risk of taking too much distance from any reality. We hope to make the right choices in the future and this may depend on the type and granularity of the real-life phenomena we want our models to capture.

Acknowledgment: We thank Eric Fanchon for many useful comments.

References

1. Steven S Andrews and Dennis Bray. Stochastic simulation of chemical reactions with spatial resolution and single molecule detail. *Physical biology*, 1(3):137, 2004.
2. Philip Ball. *Critical mass: How one thing leads to another*. Macmillan, 2004.
3. Luca Bortolussi and Jane Hillston. Checking individual agent behaviours in markov population models by fluid approximation. In *Formal Methods for Dynamical Systems*, pages 113–149. Springer, 2013.
4. Kevin Burrage, Pamela M Burrage, André Leier, Tatiana Marquez-Lago, and Dan V Nicolau Jr. Stochastic simulation for spatial modelling of dynamic processes in a living cell. In *Design and Analysis of Biomolecular Circuits*, pages 43–62. Springer, 2011.
5. Luca Cardelli. Artificial biochemistry. In *Algorithmic Bioprocesses*, pages 429–462. Springer, 2009.
6. Daniel T Gillespie. Exact stochastic simulation of coupled chemical reactions. *The Journal of Physical Chemistry*, 81(25):2340–2361, 1977.
7. Daniel T Gillespie. Approximate accelerated stochastic simulation of chemically reacting systems. *The Journal of Chemical Physics*, 115:1716, 2001.
8. Daniel T Gillespie. Stochastic simulation of chemical kinetics. *Annu. Rev. Phys. Chem.*, 58:35–55, 2007.
9. Adám M Halász, Meghan McCabe Pryor, Bridget S. Wilson, and Jeremy S. Edwards. Spatiotemporal modeling of membrane receptors. 2015. under review.
10. A Agung Julius, Ádám Halász, Mahmut Selman Sakar, Harvey Rubin, Vijay Kumar, and George J Pappas. Stochastic modeling and control of biological systems: The Lactose regulation system of Escherichia Coli. *IEEE Transactions on Automatic Control*, 53:51–65, 2008.
11. J-Y Le Boudec, David McDonald, and Jochen Mundinger. A generic mean field convergence result for systems of interacting objects. In *QEST*, pages 3–18. IEEE, 2007.
12. Oded Maler. Control from computer science. *Annual Reviews in Control*, 26(2):175–187, 2002.
13. Oded Maler. On under-determined dynamical systems. In *EMSOFT*, pages 89–96. ACM, 2011.

14. Oded Maler, Adám M Halász, Olivier Lebeltel, and Ouri Maler. Exploring the dynamics of mass action systems. *EPTCS*, 125:84–91, 2013.
15. Azaria Paz. *Introduction to Probabilistic Automata*. Academic Press, 1971.
16. Michael S Samoilov and Adam P Arkin. Deviant effects in molecular reaction pathways. *Nature biotechnology*, 24(10):1235–1240, 2006.