Exploring the Dynamics of Mass Action Systems

Oded Maler[*]
CNRS-VERIMAG
University of Grenoble
Oded.Maler@imag.fr

Ádám M. Halász[†]
Department of Mathematics
West Virginia University
halasz@math.wvu.edu

Olivier Lebeltel
CNRS-VERIMAG
University of Grenoble
Olivier.Lebeltel@imag.fr

Ouri Maler
Grenoble
ouri.maler@gmail.com

1 Introduction

Mass action is a fundamental notion in many situations in Chemistry, Biochemistry, Population Dynamics and Social Systems [1]. 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 $A+B \quad \rightarrow \quad A+C$ says that each time an instance of A meets an instance of B, the latter is transformed into a 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 typically obtains a system of *polynomial*[1] differential/difference equations.

The goal of the research program sketched in this paper (initially inspired by [3]) is to build a class of *synthetic* mathematical models for such systems, admitting some nice and clean properties which will reflect essential and fundamental aspects of mass action behavior while at the same time abstract away from accidental real-life details due to Chemistry, Physics and even some Geometry. Introducing such details at this preliminary stage would obscure the essence and render the analysis more complex. These models will then be subject to various investigations by analytical, simulation-based and other methods to explore their dynamics and discover the principles that govern their behavior. Such investigations may eventually lead to novel ways to 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, [7, 2] to mention a few, but we hope, nevertheless, to provide a fresh look on 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. We then discuss several ways to embed these individual agents in a model depicting the evolution of a large ensemble

[*] Supported by the ANR projects Syne2Arti, Eqinocs and Cadmidia.

[†] Supported by NIHGrant K25 CA131558. Work done while the author was visiting CNRS-VERIMAG.

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

Submitted to:
HSB 2013

of their instances. In Section 3 we describe three such aggregate models. We start with a rather standard model where state variables correspond to the relative concentrations of agent types. Such models depict the dynamics of the *average* over all behaviors and they are traditionally ODEs but we prefer to work in discrete time to simplify the notation. The second model is based on stochastic simulation under the well-stirred assumption with no modeling of *space*, which is introduced in the third model where particles wander in space in some kind of random motion and a reaction takes place when the distance between two particles becomes sufficiently small. The model thus obtained is essentially a kind of a *reaction-diffusion* model. In Section 4 we briefly describe the **Populus** tool kit that we developed for exploring the dynamics of such models and illustrate its functionality by demonstrating some effects of the initial spatial distributions of some particles that lead to deviation from the predictions of a well-stirred model.

2 Individual Models and Aggregation Styles

We consider mass action systems where new individuals are not born and existing ones neither die nor aggregate into compound entities: they only change their *state*.

2.1 Individuals

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 $\mathscr{A} = (Q, \Sigma, \delta)$ 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. Intuitively, $\delta(q_1, q_2, q_3)$ represents the probability that an agent of type q_1 converts to type q_3 when it encounters an agent of type q_2. Likewise $\delta(q_1, \perp, q_3)$ is the probability of becoming q_3 spontaneously without meeting anybody. Table 1 depicts a 3-species probabilistic automaton. 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.

In general 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 interpretation of when an agent meets another depends, as we shall see, on additional assumptions on the global aggregate model. It is worth noting that we restrict ourselves here to reaction rules which are "causal" in the following sense: when an (A, B)-encounter takes place, the influence of

[2] A probabilistic automaton [8] is a Markov chain with an input alphabet where each input symbol induces a different transition matrix. It is called sometimes a Markov Decision Process (MDP) but we prefer to reserve this term for a strategy synthesis problem in a game where the alphabet denotes the controller's action against a stochastic adversary. In the present model there is no "decision" associated with the input as it is an external (to the particle) influence on the dynamics.

A on B and the influence of B on A are *independent*. Hence not all types of probabilistic rewrite rules of the form $A+B \to A_1+B_1\ (p_1) \mid A_2+B_2\ (p_2) \mid \cdots \mid A_k+B_k\ (p_k)$ can be realized, only those that are products of simple rules. This restriction is not crucial for our approach but it simplifies some calculations.

δ	$\perp$			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

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

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

Table 1: A 3-species probabilistic automaton, and the average dynamical system derived for the sparse situation $\alpha = 0.1$ and for the dense situation $\alpha = 0.9$. Starting from initial state $x = (0.4, 0.3, 0.3)$ the first system converges to the state $(0.366, 0.195, 0.437)$ while the second converges to $(0.939, 0.027, 0.033)$.

2.2 Aggregation Styles

Consider now a set S consisting of m individuals put together, each being modeled as an automaton. A global configuration of such a system should specify, at least, the state of each particle, resulting in the enormous state space Q^S consisting of n^m states (micro-states in Physpeak). A very useful and commonly-used abstraction is the *counting abstraction* obtained by considering two micro-state 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 abstract state-space of macro-states (also known as particle count representation) $P \subseteq S^Q$ consisting of n-dimensional vectors:

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

The formulation of a model that tracks the evolution of an ensemble of particles can be done in different styles. For our purposes we classify models according to two features: 1) Individual vs. average dynamics and 2) Spatially-extended vs. well-stirred dynamics. These two features are related but not identical.

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, that is, stochastic simulation algorithms (SSA), generate such runs, one at the time. On the other hand, "deterministic" ODE models compute et every step the average number of particles for each type where this average is taken (in parallel) over all 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 not so informative. There is a whole research thread, starting with [4], that feeds on this important distinction (see [9, 6] for further discussions).

The other issue is whether and how one models the distribution of particles in *space*. Ignoring the spatial coordinates of particles, the probability of a particular transition being taken depends only on the total *numbers* 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. On the other hand, in spatially extended models each particle is endowed with a location which changes quasi-randomly and what it encounters in its moving neighborhood determines the interactions it is likely to participate in.

3 Implemented Aggregate Models

In the sequel we 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 of ODEs.

3.1 Average Well-Stirred Dynamics

To develop the average dynamics under the well-stirred assumption.[3] 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 $\sum x_i=1$ (relative concentration). Let α, $0\leq\alpha\leq 1$ be a density parameter which determines the probability of bumping into another particle in one time step. The evolution in this state space over time is the outcome of playing the following protocol at every time step. First, αS of the particles bump into others and hence follow a binary reaction rule while the remaining $(1-\alpha)S$ particles do not interact and hence follow the solitary transition function. We will derive the dynamics, which is of the general form[4] $x'=x+\Delta(x)$. For each variable, the additive change can be written as

$$\begin{aligned}
\Delta(x_k) &= (1-\alpha)\Delta_1(x_k)+\alpha\Delta_2(x_k)\\
\Delta_1(x_k) &= \sum_{i=1}^{n}(x_i\cdot\delta(x_i,\perp,x_k)-x_k\cdot\delta(x_k,\perp,x_i))\\
\Delta_2(x_k) &= \sum_{i=1}^{n}\sum_{j=1}^{n}(x_ix_j\cdot\delta(x_i,x_j,x_k)-x_kx_i\cdot\delta(x_k,x_i,x_j))
\end{aligned}$$

[3] Using PDEs one can sometimes derive average models under *non-uniform* distributions of particles in space but most chemical reaction models employ the well-stirred assumption.

[4] We use the primed notation where x stands for $x[t]$ and x' denotes $x[t+1]$.

Here, Δ_1 and Δ_2 are the expected 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 $\alpha = 0$, see example in Table 1. As already mentioned, this deterministic dynamics tracks the evolution of the *average* concentration of particles over all individual runs.

3.2 Individual Well-Stirred Dynamics

The second model, whose average behavior is captured by the previous one, generates individual behaviors without spatial information. A micro-state of the systems 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.

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 $\alpha$
  **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 algorithm can most likely be made more efficient by working directly on macro-states and drawing the increments of each particle type using a kind of binomial distribution that sums up the multiple coin tosses. Similar ideas underlie the τ-leaping algorithm of [5].

3.3 Individual Spatial Dynamics

Our third model does take space into account by representing each particle as (g,q,y) with y being it spatial coordinates, currently 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* so that when a particle crosses the boundary of the rectangle it reappears on the other side as if it was a torus. 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. If the particle has several neighbors[5] we compute the outcome of all those possible interactions and choose among them

[5] Which turns out not to be negligible with the parameters we have chosen so far which are unlike commonly-used models where the average distance between particles is orders of magnitude larger than the interaction radius r.

randomly.

Algorithm 2 (Individual Spatial Dynamics)

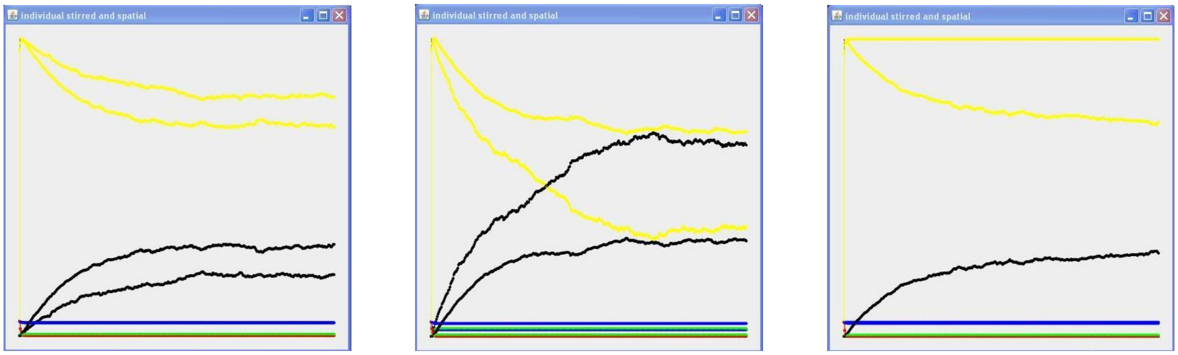
```
  Input: A list L of particles and states including planar coordinates
  Output A list $L'$ representing the next micro-state

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

The connection between this model, embedded in a rectangle of area W, and the non-spatial ones is made via the computation of the density factor α. The probability of a particle *not* interacting with another particle is its probability to be outside its interaction ball, that is, $1 - \frac{\pi r^2}{W}$, and the odds of not interacting with any of the other $m-1$ particles is $(1 - \frac{\pi r^2}{W})^{m-1}$ and hence $\alpha = 1 - (1 - \frac{\pi r^2}{W})^{m-1}$.

4 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 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, plotting the evolution of particle counts over time as well as animating the spatial evolution.

To demonstrate the difference between spatial and non-spatial models we simulated a system with 5 species, A, B, C, D and E to E. A and B are initially present in small quantities, 50 each, while D has 1000 instances. When A and B meet, A is transformed into an active and stable agent C which converts D's to E's . Since E is also rather stable, the emergence of C will eventually convert a large number of D's to E's. However, B is very unstable and each step it may change with probability 0.5. Hence the initial spatial distribution of A and B may influence the evolution significantly. We simulated the corresponding spatial model on a 20×20 square, starting from three different initial micro-states in all



(a) (b) (c)

Figure 1: The evolution of the 5-species system where A's and B's are initially (a) distributed uniformly in space; (b) close to each other and (c) remote from each other. The plot depict the spatial and non-spatial models with the black curve indicating the growth of E.

of which D is distributed uniformly over all the square: (a) A and B are distributed uniformly all over space; (b) Both A and B are concentrated in a unit square in the middle; (c) A and B are concentrated inside distinct unit squares far apart from each other. The results are plotted in Figure 1. As a first observation, in scenario (a) the spatial model converts D to E slower than the well-stirred one, despite the well-stirred initial condition. It is too early to speculate about the reasons but it might be that with our parameters where reaction is not slower than diffusion, a C particle converts the D's in its neighborhood and hence meets less of them than what their global concentration would imply. In scenario (b), due to the proximity of A and B there is a burst of C's at the beginning and the spatial model progresses faster than the well-stirred. Finally, when A and B are initially far apart, no C and hence no E are produced, unlike the prediction of the well-stirred model. In all those experiments the behavior of the well-stirred model was close to that of the average model.

References

[1] Philip Ball. *Critical mass: How one thing leads to another*. Macmillan, 2004.

[2] 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.

[3] Luca Cardelli. Artificial biochemistry. In *Algorithmic Bioprocesses*, pages 429–462. Springer, 2009.

[4] Daniel T Gillespie. Exact stochastic simulation of coupled chemical reactions. *The Journal of Physical Chemistry*, 81(25):2340–2361, 1977.

[5] Daniel T Gillespie. Approximate accelerated stochastic simulation of chemically reacting systems. *The Journal of Chemical Physics*, 115:1716, 2001.

[6] 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.

[7] 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.

[8] Azaria Paz. *Introduction to Probabilistic Automata*. Academic Press, 1971.

[9] Michael S Samoilov and Adam P Arkin. Deviant effects in molecular reaction pathways. *Nature biotechnology*, 24(10):1235–1240, 2006.