Influence Systems and Natural Algorithms

Bernard Chazelle*
Department of Computer Science
Princeton University
chazelle@cs.princeton.edu

— *Draft (CACM Research Highlights, to appear)* —

ABSTRACT

Algorithms offer a rich, expressive language for modelers of biological and social systems. They lay the grounds for numerical simulations and, crucially, provide a powerful framework for their analysis. Natural algorithms may reprise in the life sciences the role differential equations have long played in the physical sciences. For this to happen, however, an "algorithmic calculus" is needed. We discuss what this program entails in the context of *influence systems*.

1. INTRODUCTION

The gradual elevation of "computational thinking" within the sciences is enough to warm the heart of any computer scientist. Yet the long-awaited dawning of a new age may need to wait a little longer if we cannot move beyond the world of simulation and build a theory of natural algorithms with real analytical heft. Just as differential equations have given us the tools to explain much of the physical world, so natural algorithms will help us model the living world and make sense of it. At least this is the hope and, for now, I strongly believe, one of the most pressing challenges facing computer science.

Science or engineering?

To draw a fine line between science and engineering is a fool's errand. Unrepentant promiscuity makes a clean separation neither wise nor easy. Yet a few differences bear mentioning. If science is the study of the nature we have, then engineering is the study of the nature we want: the scientist will ask how the valley was formed; the engineer will ask how to cross it. Science is driven by curiosity and engineering by need: one is the stuff of discovery, the other of invention. The path of science therefore seems more narrow. We want our physical laws to be right and our mousetraps to be useful. But there are more ways to be useful than to be right. Engineering can "negotiate" with nature in ways science cannot. This freedom comes at a price, however. Any mousetrap is at the mercy of a better one. PageRank one day will go; the Second Law of thermodynamics never will. And so algorithms, like mousetraps, are human-designed tools: they are engineering artifacts.

But is this view entirely accurate? Perhaps search engines don't grow on trees but leaves do, and a sophisticated algorithmic formalism, *L-systems*, is there to tell us how. It is so spectacularly accurate, in fact, that the untrained eye will struggle to pick out computer-generated trees from the real thing. The algorithmic modeling of bird flocking has been no less successful. Some will grouch that evolution did not select the human eye for its capacity to spot fake trees and catch avian impostors. Ask a bird to assess your computer-animated flock, they'll snicker, and watch it cackle with derision. Perhaps, but the oohs and ahhs from CGI fans everywhere suggest these models are on to something. These are hardly isolated cases. *Natural algorithms* are quickly becoming the language of choice to model biological and social processes. They are both science *and* engineering.

It's all about language

The triumph of 21st-century physics has been, by and large, the triumph of mathematics. A few equations scattered on a single page of paper explain most of what goes on in the physical world. This miracle speaks to the organizing principles of the universe: symmetry, invariance, and regularity—precisely the stuff on which mathematics feasts. Alas, not all of science is this tidy. Instead of identical particles subject to the same forces, biology and economics feature autonomous agents, each one with its own idea of what laws to obey. It is a long way, scientifically speaking, from planets orbiting the sun in orderly fashion to unruly slime molds farming bacterial crops. Gone are the symmetry, invariance, and clockwork regularity of astronomy: what we have is, well, sludge. But the sludge follows a logic that has its own language, the language of *natural algorithms*.

The point of departure with classical mathematics is indeed linguistic. While differential equations are the native idiom of electromagnetism, no one believes that cancer has

*This work was supported in part by NSF grants CCF-0832797, CCF-0963825, and CCF-1016250.

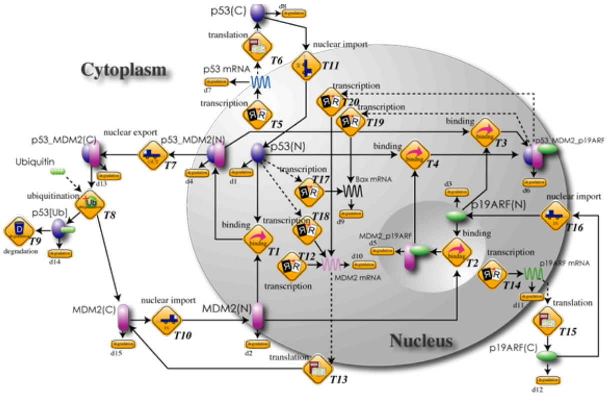

Figure 1: A protein and its natural algorithm.

its own "Maxwell's equations." Yet it may well have its own *natural algorithm*. The circuit in Fig.1 attempts to explain, in algorithmic terms, how the protein p53 keeps cancer at bay. The chain of causal links, some deterministic, others stochastic, cannot be expressed merely in the language of differential equations. It is not just the diversity of factors at play (genetic, infectious, environmental, etc); nor is it their heterogeneous modes of interaction. It is also the need for a narrative of collective behavior that can be expressed at different levels of abstraction: first-principles; phenomenological; systems-level; etc. The issue is not realism but *tractability*. This point is crucial. It could well be that PDEs will continue to provide the most accurate descriptions of bioprocesses, just as Newtonian mechanics still gives the most principled model of an ideal gas at equilibrium. But it was the genius of Boltzmann and others to realize that the thermodynamics model of Carnot differed from a Newtonian framework only by being a tractable, coarse-grained level of abstraction of the same processes. Likewise, the promise of agent-based natural algorithms is a *tractable level of abstraction* for reasoning about complex systems characterized by heterogeneity, agency, and spacetime scale diversity.

Beyond simulation

Decades of work in programming languages have produced an advanced theory of abstraction. Ongoing work on reactive systems is attempting to transfer some of this technology to biology [8]. Building on decades of progress in automata theory, temporal logic, and process algebra, the goal has been to build a modeling framework for biological systems that integrate the concepts of concurrency, interaction, refinement, composition, encapsulation, modularity, asynchrony, stochasticity, causality, etc. With the right specifications in place, the hope is that established programming language tools, such as type theory, model checking, abstract interpretation, and the pi-calculus can aid in verifying temporal properties of biosystems. The idea is to reach beyond numerical simulation to provide proven certificates about collective behavior.

Such an approach, however, can only be as powerful as the theory of natural algorithms behind it. To illustrate this point, consider classifying all possible sequences $\mathbf{x}$, $P\mathbf{x}$, $P^2\mathbf{x}$, $P^3\mathbf{x}$, etc, where $\mathbf{x}$ is a vector and P is a fixed stochastic matrix. Simulation, machine learning, and verification techniques can help, but no genuine understanding of the process can be achieved without Perron-Frobenius theory. Likewise, natural algorithms need not only computers but also a theory.

Algorithms from nature

If living processes are powered by the "software" of nature, then natural selection is the ultimate code optimizer. With time and numbers on their side—billions of years and 10^{30} living specimens—bacteria have had ample opportunity to perfect their natural algorithm. No wonder computer scientists are turning to biology for algorithmic insight: neural nets and DNA computing, of course, but also ant colony optimization [3], shortest path algorithms in slime molds [2]; maximal independent sets in fly brain development [1], etc. Consensus, synchronization, and fault tolerance are concepts central to both biology and distributed computing [14, 17]. The trade of ideas promises to be flowing both ways. This article focuses on the outbound direction: how algorithmic ideas can enrich our understanding of nature.

2. INFLUENCE SYSTEMS

A bad, fanciful script will make the perfect stage-setter. One fateful morning, you stumble out of bed and into your kitchen only to discover, crawling on the floor, a swarm of insects milling around. Soon your dismay gives way to curiosity, as you watch the critters engage in a peculiar choreography. Each insect seems to be choosing a set of neighbors (living or inert) and move either toward or away from them. From what you can tell, the ants pick the five closest termites; the termites select the nearest soil pellets; the ladybugs pick the two ants closest to the powdered sugar that is not in the vicinity of any termite; etc. Each insect seems equipped with its own selection procedure to decide how to pick neighbors based on their species and the geometry of the scene. Once the selection is made, each agent invokes a second procedure, this time to move to a location determined entirely by the identities and locations of its neighbors.[1] To model this type of multiagent dynamics, we introduce *influence systems*, a brand of networks that perpetually rewire themselves.

Definition and examples

An *influence system* is specified by two functions f and $\mathcal{G}$: it is a discrete-time dynamical system,[2] $\mathbf{x} \mapsto f(\mathbf{x})$ in $(\mathbb{R}^d)^n$, where n is the number of agents, d is the dimension of the ambient space ($d = 2$ in the example above), and each "coordinate" x_i of the state $\mathbf{x} = (x_1, \ldots, x_n)$ is a d-tuple encoding the location of *agent* i in $\mathbb{R}^d$. With any state $\mathbf{x}$ comes a directed "communication" graph, $\mathcal{G}(\mathbf{x})$, with one

[1] Some living systems (eg, ants, termites) exchange information by *stigmergy*: instead of communicating directly with signals, they leave traces such as pheromones in the environment, which others then use as cues to coordinate their collective work. Although lacking autonomy, inert components can still be modeled as agents in an influence system.

[2] A dynamical system generates an orbit $\mathbf{x}, f(\mathbf{x}), f^2(\mathbf{x}), \ldots$, for any $\mathbf{x}$: the goal is to understand the geometry of these orbits.

node per agent. Each coordinate function f_i of the map $f = (f_1, \ldots, f_n)$ takes as input the neighbors of agent i in $\mathcal{G}(\mathbf{x})$, together with their locations, and outputs the new location $f_i(\mathbf{x})$ of agent i in $\mathbb{R}^d$. The philosophy behind influence systems is that the flow of information across the communication network captures the heart of the system. By distinguishing f from $\mathcal{G}$, the model separates the syntactic (who talks to whom?) from the semantic (who does what?) Recursive graph algorithms are thus expected to occupy center stage in the analysis, itself a novelty in the field of dynamics.

Both f and $\mathcal{G}$ are evaluated by a deterministic or randomized algorithm. An influence system is called *diffusive* if f keeps each agent within the convex hull of its neighbors. Diffusive systems never escape to infinity and always make consensus (all x_i being equal) a fixed point. The system is said to be *bidirectional* if the communication graph always remains undirected.

- *HK systems:* In this popular model of social dynamics [9], x_i is a real number denoting the "opinion" of agent i. That agent is linked to j if and only if $|x_i - x_j| \leq r$. The procedure for f instructs i to move to the mass center of its neighbors. This is the prototypical example of a bidirectional diffusive influence system.

- *Sync:* An instance of Kuramoto synchronization, this diffusive influence system links each of n fireflies to the other fireflies whose flashes it can spot. Every critter has its own flashing oscillator, which becomes coupled with those of its neighbors. The function f specifies how the fireflies adjust their flashings in reaction to the graph-induced couplings [3, 23]. A similar model has been applied to circadian neurons, chirping crickets, microwave oscillators, yeast cell suspensions, and pacemaker cells in the heart [25].

- *Swarming:* The agents may be fish or birds and their states indicating their positions and velocities ($d = 6$). The communication graph links every agent to some of its nearest neighbors. The function f instructs each agent to align its velocity with that of its neighbors, to move toward their center of gravity, and to fly away from its perilously close neighbors [20, 24]. Noise can be added at each step by subjecting the agents' headings to a small random rotation.

- *Chemotaxis:* Some organisms can sense food gradients and direct their motion accordingly. In the case of bacterial chemotaxis, the stimuli are so weak that the organisms are reduced to performing a random walk with a drift toward higher food concentrations. Influence systems can model these processes with the use of both motile and inert agents. Chemotaxis is usually treated as an asocial process (single agents interacting only with the environment). It has been observed, however, that schooling behavior can facilitate gradient climbing for fish, a case where living in groups enhances foraging ability [19]. Influence systems can model both social and asocial behaviors.

Other examples of influence systems include the Ising model, neural nets, Bayesian social learning, protein-protein interaction networks, etc.

How expressive are influence systems?

If agent i is viewed as a computing device, then the d-tuple x_i is its memory. The system is Markovian in that all facts about the past with bearing on the future need to be encoded in $\mathbf{x}$. The communication graph allows the function f to be local if so desired. The procedure $\mathcal{G}$ itself might be local even when appearance suggests otherwise: for example, to identify your nearest neighbor in a crowd requires only local computation, even though mathematically the function is global, requiring knowledge of all the positions. Since the emergence of macroscopic phenomena from simple local rules is the raison d'être of influence systems, one expects to see computational restrictions placed on the agents. Yet, even with deceptively simple rules, influence systems can pack a great amount of behavioral complexity:

(*i*) *Learning, competition, hierarchy:* Agents can implement game-theoretic strategies in competitive environments (e.g., pursuit-evasion games) and learn to cooperate in groups (e.g., quorum sensing). They can self-improve, elect leaders, and stratify into dominance hierarchies.

(*ii*) *Coarse-graining:* Flocks are clusters of birds that maintain a certain amount of communicative cohesion over a period of time. We can view them as "super-agents" and seek the rules governing interaction among them. Iterating in this fashion creates a coarse-graining of the original system somewhat akin to Kadanoff's block-spin renormalization in statistical mechanics. Coarse-graining can also be accomplished through time scaling (see §5).

(*iii*) *Asynchrony and uncertainty:* In the presence of delayed or asynchronous communication, agents can use their memory to implement a clock for the purpose of time stamping. Influence systems can also model uncertainty by granting agents access to approximations of their neighbors' states.

Influence systems are agent-based, a concept well worth a detour. Consider the diffusion of pollen particles suspended in water. A *Eulerian* approach to this process seeks a differential equation for the concentration $c(x, t)$ of particles at any point x and time t. There are no agents, just density functions evolving over time [18]. An alternative approach, called *Lagrangian*, would track the movement of all the individual particles and water molecules by appealing to Newton's laws. Given the sheer number of agents, this line of attack crashes against a wall of intractability. One way around it is to pick a single imaginary self-propelled agent and have it jiggle about randomly in a Brownian motion. This agent models a typical pollen particle, typical in the "ergodic" sense that its time evolution mimics the space distribution of countless particles caught on film in a snapshot. Time scaling plays a key role: our pollen particles indeed can be observed only on a time scale far larger than the molecular bumps causing the jiggling. Luckily, Brownian motion is scale-free. As we shall see in §5, the ability to express a dynamical process at different scales is an important feature of influence systems.

The strength of the Eulerian approach is its privileged access to an advanced theory of calculus. Its weakness derives from two commitments: global behavior is implied by *infinitesimal* changes; and every point is subject to *identical* laws. While largely true in physics, these assumptions break down in the living world, where diversity, heterogenity, and autonomy prevail. Alas, the Lagrangian answer, agent-based modeling, itself suffers from a crippling handicap: the lack of a theory of natural algorithms.

A preview

This article confines itself to influence systems. It examines the diffusive case in §3–5 and a classic model of bird flocking in §6.[3] It begins with a surprising observation: bidirectional diffusive systems always converge. This is best seen via an analytical device known as the *total s-energy*, the topic of the next section. The general case requires the use of an algorithmic calculus, which we sketch in §5.

General diffusive influence systems are Turing-complete, yet the mildest perturbation ruins it all to create periodic behavior. This is disconcerting. Influence systems model how people change opinions over time as a result of human interaction and knowledge acquisition. Instead of walking their way toward enlightenment, people are doomed to recycle the same opinions in the same order in perpetuity... At least that's what the math says.

3. THE s-ENERGY

Let $(P_t)_{t\geq 0}$ be an infinite sequence of n-by-n stochastic matrices; stochastic means that the entries are nonnegative and the rows sum up to 1. Leaving aside influence systems momentarily, we make no assumption about the sequence, not even that it is produced endogenously. We ask a simple question: under what conditions does $P_{<t} := P_{t-1}\cdots P_0$ converge as $t \to \infty$? Certainly not if

$$P_t = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}.$$

The problem here is lack of *self-confidence*: the two agents in the system $\mathbf{x} \mapsto P_t\,\mathbf{x}$ trust their neighbors too blindly. So let's assume that the diagonal entries of P_t are positive. Alas, this still does not do the trick: the matrices

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 2/3 & 1/3 \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 2/3 & 0 & 1/3 \end{pmatrix}$$

grant the agents self-confidence, yet composing them in alternation exchanges the vectors $(0, 1, 1/4)$ and $(0, 1, 3/4)$ endlessly. The oscillation is caused by the lack of *bidirectionality*: indeed, agents 1 and 2 never link to agent 3 in the communication graph. The fix is to require the graphs to be undirected. With both self-confidence and bidirectionality (ie, mutual self-confidence) in place, surprise, the sequence $P_{<t}$ *always* converges [11, 13, 16]. (Interestingly, this is *not* true of forward products $P_0 \cdots P_t$ in general.) With nothing keeping $(P_t)_{t\geq 0}$ from "stalling" by featuring arbitrarily long repeats of the identity matrix, bounding the convergence rate is impossible. Yet an analytical device, the *total s-energy* [5], allows us to do just that for influence systems with "$P_{<t}$ style" dynamics. The trick is to show that they cannot stall too long without dying off.

Preliminaries

Fix small $\rho > 0$ and let $(P_t)_{t\geq 0}$ be a sequence of stochastic matrices such that $\rho \leq (P_t)_{ii} \leq 1-\rho$ and $(P_t)_{ij} > 0 \Rightarrow (P_t)_{ji} > 0$. Let $\mathcal{G}_t$ be the (undirected) graph whoses edges are the positive entries in P_t. With $\mathbf{x}(t+1) = P_t\mathbf{x}(t)$ and $\mathbf{x}(0) = \mathbf{x} \in [0,1]^n$, the *total s-energy* is defined as:

$$E(s) = \sum_{t\geq 0} \; \sum_{(i,j)\in \mathcal{G}_t} |x_i(t) - x_j(t)|^s \,. \tag{1}$$

Being a generalized Dirichlet series, the s-energy can be inverted and constitutes a lossless encoding of the edge lengths. Why this unusual choice? Because, as with the most famous Dirichlet series, the Riemann zeta function $\sum n^{-s}$, the system's underlying structure is multiplicative: indeed, just as n is a product of primes, $x_i(t) - x_j(t)$ is a product of the form $v^T P_{t-1}\cdots P_0\,\mathbf{x}$. Let $E_n(s)$ denote the maximum value of $E(s)$ over all $\mathbf{x} \in [0,1]^n$.

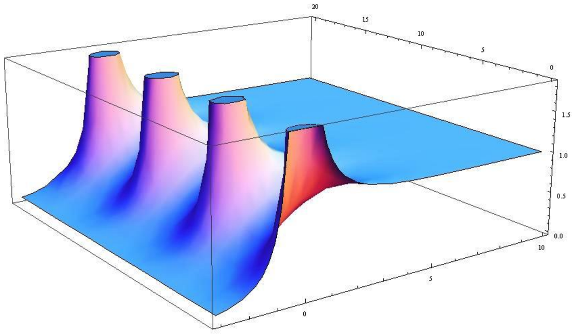

Figure 2: The analytic continuation of $|E_2(s)|$.

The sequence formed by $(P_t)_{t\geq 0}$ is called *reversible* if $\mathcal{G}_t$ is connected and there is a probability distribution $(\pi_1, \ldots, \pi_n)$ such that $\pi_i(P_t)_{ij} = \pi_j(P_t)_{ji}$ for any t; see detailed definition in [5]. This gives us a way to weight the agents so that their mass center never moves. The notion generalizes the concept of reversible Markov chains, with which it shares some of the benefits, including faster convergence to equilibrium.

Bounds

The s-energy measures the total length of all the edges for $s = 1$ and counts their number for $s = 0$; the latter is usually infinite, so it is sensible to ask how big $E_n(s)$ can be for $0 < s \leq 1$. On the lower bound front, we have $E_n(1) = \Omega(1/\rho)^{\lfloor n/2 \rfloor}$ and $E_n(s) = s^{1-n}(1/\rho)^{\Omega(n)}$, for any n large enough, $s \leq s_0$, and any fixed $s_0 < 1$. Of course, the s-energy is useful mostly for its upper bounds:

$$E_n(s) \leq \begin{cases} (1/\rho)^{O(n)} & \text{for } \; s = 1; \\ s^{1-n}(1/\rho)^{n^2+O(1)} & \text{for } \; 0 < s < 1. \end{cases} \tag{2}$$

For reversible sequences and any $0 < s \leq 1$,

$$n^{-2}E_n(s) \leq E_n^D(s) \leq \frac{2n}{s}\Big(\frac{2n}{\rho}\Big)^{s/2+1}, \tag{3}$$

where $E_n^D(s) = \sum_{t\geq 0} \operatorname{diam}^s\{x_1(t), \ldots, x_n(t)\}$. This is essentially optimal. Fix an arbitrarily small $\varepsilon > 0$. A step

[3] Unless noted otherwise, the results discussed are from: [5] for §3; [6] for §4,5; and [4] for §6.

t is called *trivial* if $|x_i(t) - x_j(t)| < \varepsilon$ for each $(i,j) \in \mathcal{G}_t$. The maximum number C_ε of nontrivial steps is bounded by $\varepsilon^{-s} E_n(s)$; hence,

$$C_\varepsilon(n) \leq \min\{ \tfrac{1}{\varepsilon} (\tfrac{1}{\rho})^{O(n)}, (\log \tfrac{1}{\varepsilon})^{n-1} (\tfrac{1}{\rho})^{n^2+O(1)} \}, \qquad (4)$$

which is optimal if ε is not too small. Convergence in the reversible case is polynomial: if $\varepsilon < \rho/n$, then $\|\mathbf{x}(t) - \pi^T \mathbf{x}\|_2 \leq \varepsilon$, for $t = O(\rho^{-1} n^2 |\log \varepsilon|)$. This bound is optimal. In particular, we can specialize it to the case of random walks in undirected graphs and retrieve the usual mixing times.

Applications

(A) *HK systems* track opinion polarization in a population. In the *bounded-confidence* version, the agents consist of n points in $\mathbb{R}^d$. At each step, each agent moves to the mass center of the agents within distance r. This bidirectional influence system converges in $n^{O(n)}$ time. The bound is known to be polynomial for $d = 1$ [15].

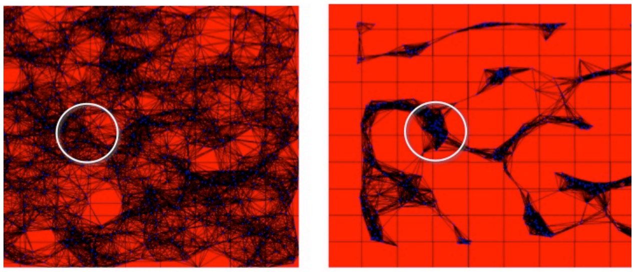

Figure 3: The initial graph $\mathcal{G}_0$ and a few steps later.

(B) *Truth-seeking systems* differ from the bounded-confidence model by assuming a "cognitive division of labor" [10]. We fix one agent, the truth, and keep the $n-1$ others mobile. A "truth seeker" is a mobile agent that is joined to the truth in every $\mathcal{G}_t$. All the other mobile agents are "ignorant," meaning that they never join to the truth through an edge, although they might indirectly communicate with it along a path. Any two mobile agents are joined in G_t whenever their distance is less than r. Assuming that r and the initial configuration are encoded as $O(n)$-bit rationals, the system converges in $n^{O(n)}$ time. All the truth seekers lie within a tiny ball centered at the truth (doubly-exponential small). Ignorant agents either lie in that ball or are frozen in place forever.

Why the s-energy?

Let $\texttt{conv}P$ denote the convex hull of the points formed by the rows of the matrix P. We have the "Russian doll" nesting structure:

$$\texttt{conv}P_{<t} \subseteq \texttt{conv}P_{<t-1} \subseteq \cdots \subseteq \texttt{conv}P_0 \subset \mathbb{R}^n.$$

The literature on stochastic matrices features a variety of "coefficients of ergodicity" [21] to help us measure how quickly the Russian dolls deflate: eigenvalues, joint spectral radius, width, diameter, volume, etc. By seeking progress at each step, however, these methods cannot cope with stalling. The total s-energy gets around this by providing a *global* deflation measure (Fig.4). The s-energy is controlled by a single parameter s, which we can adjust at will to get the most out of the inequality $C_\varepsilon \leq \varepsilon^{-s} E(s)$, typically choosing s so that $(dE/ds)_{|s} = E \ln \varepsilon$. Proving (2) for $s = 1$ entails a delicate credit amortization argument, a classical algorithmic technique apparently new to the field of dynamics. For brevity, we discuss only the case $s < 1$. This will also give us a chance to introduce a concept fundamental to the theory of diffusive influence systems: the *flow tracker.* Think of it as breadth-first search in a dynamic graph. A little imagery will help. Pick agent 1 and dip it in water, keeping all the other agents dry. Whenever an edge of $\mathcal{G}_t$ links a dry agent to a wet one, the dry one gets wet. As soon as all the agents become wet (if ever), dry them all except agent 1; repeat.

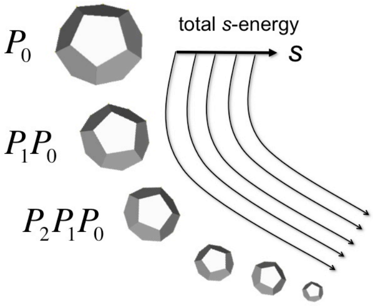

Figure 4: The deflating matrix polytope.

Flow tracker (5)

```
[1]  $t_0 \leftarrow 0$.
[2]  Repeat forever:
     [2.1]  $W_{t_0} \leftarrow \{1\}$.
     [2.2]  For $t = t_0, t_0 + 1, \ldots, \infty$:
                $W_{t+1} \leftarrow W_t \cup \{ i \mid \exists (i,j) \in \mathcal{G}_t \ \& \ j \in W_t \}$.
     [2.3]  If $|W_\infty| = n$ then $t_0 \leftarrow \min\{ t > t_0 : |W_t| = n \}$
            else stop.
```

Let W_t denote the set of wet agents at time t, which always includes agent 1. The assignments of t_0 in step [2.3] divide the timeline into *epochs*, time intervals during which either all agents become wet or, failing that, the flow tracker comes to a halt (breaking out of the repeat loop at "stop"). Take the first epoch: it is itself divided into subintervals by the *coupling times* $t_1 < \cdots < t_\ell$, with $W_{t_k} \subset W_{t_k+1}$. If $\|W_t\|$ denotes the length of the smallest interval enclosing W_t, it can be shown by induction that $\|W_{t_k+1}\| \leq 1 - \rho^k$. It then follows that $E_1(s) = 0$ and, for $n \geq 2$,

$$E_n(s) \leq 2n E_{n-1}(s) + (1 - \rho^n)^s E_n(s) + n^3,$$

which implies (2) for $s < 1$.

4. DIFFUSIVE INFLUENCE SYSTEMS

Recall that a diffusive influence system $\mathbf{x} \mapsto f(\mathbf{x})$ in $\mathbb{R}^{dn}$ requires that the agents move within the convex hull of their neighbors. For convenience, we set $d = 1$, so $f(\mathbf{x})$ is encoded as an n-by-n stochastic matrix $P(\mathbf{x})$. We further assume that the system is piecewise-linear in the sense of (Sontag [22]). While, in practice, f is likely to be a "simple" function, our main result needs no such assumption. Given

$\mathbf{x}$, we need only assume that $P(\mathbf{x})$ is specified by a first-order sentence in the theory of the reals.[4] By algebraic lifting and Collins's cylindrical decomposition, we can linearize the constraints. To specify the model, therefore, it suffices to fix an arrangement of hyperplanes in $\mathbb{R}^n$ and define the *switching partition* (*SP*) as its set of n-dimensional cells, the *atoms* (Fig.5). The matrix $P(\mathbf{x}) = P_c$ depends only on the atom c that contains $\mathbf{x}$. We assume self-confidence but not mutual confidence, ie, positive diagonal entries but not necessarily bidirectionality. (Typically, the correspondence $c \mapsto P_c$ is implicit and need not be encoded directly.)

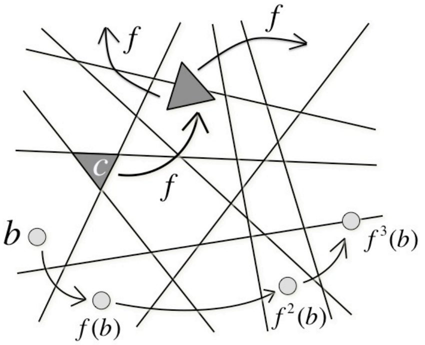

Figure 5: The atom c of the *SP* maps via f to a cell intersecting two atoms.

Diffusive systems can be as expressive as general piecewise-linear systems: they can be chaotic and even Turing-complete. The surprise is that perturbations wipe out their computational power: diffusive systems are robustly predictable; in fact, they are clocks in disguise. This dichotomy requires a subtle bifurcation analysis, which we sketch in the next section.

Theorem 1. [6] *Given any initial state, the orbit of a diffusive influence system is attracted exponentially fast to a limit cycle almost surely under an arbitrarily small perturbation. The period and preperiod are bounded by a polynomial in the reciprocal of the failure probability. In the bidirectional case, the system is attracted to a fixed point in time $n^{O(n)}|\log \varepsilon|$, where n is the number of agents and ε is the distance to the fixed point.*

The result says that, given any starting point, a random perturbation of the *SP* ensures that the orbit eventually approaches a fixed periodic orbit exponentially fast: for example, $(-1)^t + 2^{-t}$. The number of limit cycles is infinite but, if we measure distinctness the right way (ie, by factoring out foliations), there are actually only a finite number of them.

As in statistical mechanics, the system's complexity arises from the tension between two opposing forces: one, caused by the map's discontinuities, is "entropic" and leads to chaos; the other one, related to the Lyapunov exponents, is "energetic" and pulls the system toward an attracting manifold within which the dynamics is periodic. The goal is to show that, outside a vanishingly small region, entropy always loses. What does it mean? If, unlike in Fig.5, the iterated image of any ball b never intersected the *SP* hyperplanes, it would merrily bounce around until eventually periodicity kicked in. In the figure, $f^3(b)$ does not oblige and splits into two smaller bodies. Both will bounce around until possibly splitting again and so on. If this branching process gets out of control, chaos will ensue. To squelch this entropic process and induce periodicity, we have the stochasticity of the matrices to credit: it causes the ball b to shrink (at least along directions not parallel to the eigenvector $\mathbf{1}$) and thus dissipate a form of "energy." Entropy vs energy: which one will win? For entropy to lose out, the ball b must avoid splitting too frequently. This can be expressed by an (infinite) system of linear inequalities. Feasibility hinges on a type of *matrix rigidity* question: in this case, given a certain matrix, how many rows must be removed before we can express the first column as a linear combinations of the others? Periodicity requires that this number be high. The matrix in question is extracted from the system's stochastic matrices and the *SP* equations, hence is highly structured.

5. AN ALGORITHMIC CALCULUS

Left to its own devices, the s-energy can produce convergence rates only in favorable conditions, such as those found in the bounded-confidence model. To see the difficulty, consider a 3-agent system with agents 1 and 2 moving toward each other: $x_1 \to (2x_1{+}x_2)/3$ and $x_2 \to (x_1{+}2x_2)/3$. Starting at positions -1 and 1, agents 1 and 2 move to positions $\pm 3^{-t}$ at time t. Imagine now having a third agent starting at position $0.9 < x_3 < 1$ and programmed to join with agent 1 when their distance falls below 1: this will cause a graph switch in $\mathcal{G}_t$ for $t = \Omega(|\log(1-x_3)|)$, thus dashing any hope of a uniform bound on the time to reach equilibrium. The solution is to perturb the system. Another, more serious reason for doing so is to avoid chaos and Turing universality.

Perturbing a classical algorithm affects only a finite number of tests, hence creates a bounded number of probabilistic conditions to satisfy. Not so with natural algorithms. Perturbing each *SP* hyperplane $a_0 + \sum a_i x_i = 0$ into $a_0 + \sum a_i x_i = \delta$, for some random δ, produces an infinite number of probabilistic conditions since it affects *all* of their iterated images. Union bounds thus involve infinite series; since the geometry of bad perturbations matters greatly, nonprobabilistic structural parameters must be thrown in as well.

Perturbations

We begin with a little trick, which is to thicken the discontinuities. This is a purely analytical device with no effect on the dynamics. Fix $\varepsilon > 0$ and define the *margin*

$$\mathcal{R}_\varepsilon = \bigcup_{SP} \Big\{ \mathbf{x} \in \mathbb{R}^n \,:\, |\, a_0 + a_1 x_1 + \cdots + a_n x_n + \delta | \leq \varepsilon \Big\},$$

over all the *SP* hyperplanes. We define the label $\ell(\mathbf{x}) = \min\{\, t \geq 0 \,|\, f^t(\mathbf{x}) \in \mathcal{R}_\varepsilon \,\}$ for any $\mathbf{x} \in [0,1]^n$; by convexity, we can limit the phase space to the unit cube. The point $\mathbf{x}$ is said to *vanish* at time $\ell(\mathbf{x})$ if its label is finite. The subset of $[0,1]^n$ not vanishing before time t is denoted by $\mathcal{S}_t$: obviously, $\mathcal{S}_0 = [0,1]^n$; and, for $t > 0$,

$$\mathcal{S}_t = [0,1]^n \setminus \bigcup_{k=0}^{t-1} f^{-k}(\mathcal{R}_\varepsilon)\,.$$

[4] This is the language of geometry and algebra with statements specified by any number of quantifiers and polynomial (in)equalities. It was shown to be decidable by Tarski.

Each cell of $\mathcal{S}_{t+1}$ lies within a cell of $\mathcal{S}_t$ (one of $\#\mathcal{S}_t$ open n-cells). The limit set $\mathcal{S}_\infty = \bigcap_{t\geq 0} \mathcal{S}_t$ collects the points that never vanish. Why the thickening? Because it makes the cell decomposition of the nonvanishing region finite: in fact, for any t, $\#\mathcal{S}_t = (n/\varepsilon)^{O(n)}$. The system is *nesting at t* if no cell c of $\mathcal{S}_t$ contains more than one cell of $\mathcal{S}_{t+1}$. The minimum value of t is called the *nesting time* ν of the system.

We can show that the system is nesting at *any* time $t \geq \nu$. This result has remarkable consequences. For one thing, it immediately bounds the period of any limit cycle. Consider the directed graph with one node per cell c of $\mathcal{S}_\nu$ and an edge from c to the unique cell of $\mathcal{S}_\nu$ (if it exists) that intersects $f(c)$: the graph defines a *functional* regular language (sofic shift), meaning that each node has exactly one outgoing edge (possibly a self-loop), so any infinite path ends up in a cycle. Periodicity follows. The difficulty is to show that $\mathcal{S}_\nu$ covers most of $[0,1]^n$; in fact, even to show that it is nonempty takes work. The previous discussion hints at the tree structure of the orbits, an idea we need to develop further.

The coding tree

This infinite rooted tree $\mathcal{T}$ encodes into one geometric object the set of all orbits. It is embedded in $[0,1]^n \times \mathbb{R}_{\geq 0}$, with the last dimension representing time. The atoms are redefined as the n-dimensional cells outside the margin. Each child v of the root is associated with an atom U_v. The *phase tube* (U_v, V_v) of each child v is the "time cylinder" whose cross-sections at times 0 and 1 are U_v and $V_v = f(U_v)$, respectively. The tree is built recursively by subdividing V_v into the cells c formed by its intersection with the atoms, and attaching a new child w for each c: we set $V_w = f(c)$ and $U_w = U_v \cap f^{-t_v}(c)$, where t_v is the depth of v (Fig.6). We denote by P_w the matrix of the map's restriction to c. The phase tube (U_v, V_v) consists of all the cylinders whose cross-sections at $t = 0, \ldots, t_v$ are, respectively, $U_v, f(U_v), \ldots, f^{t_v}(U_v) = V_v$. If V_v intersects the margin, we add a *vanishing* leaf below v. Intuitively, $\mathcal{T}$ divides up the atoms into maximal regions over which the iterated map is linear.

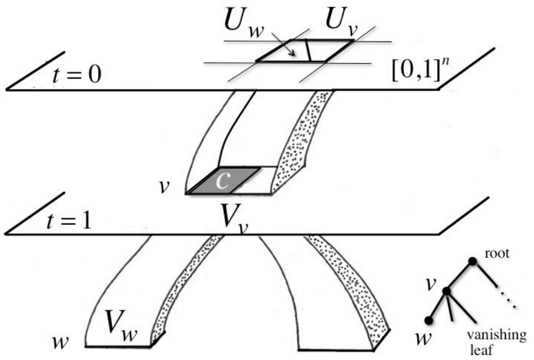

Figure 6: A phase tube (U_w, V_w) of length two.

Let $ww'w'' \cdots$ denote the upward, t_w-node path from w to the root (but excluding the root). Using the notation $P_{\leq w} = P_w P_{w'} P_{w''} \cdots$, we have the identities $V_w = P_{\leq w} U_w$ and $\mathcal{S}_k = \bigcup_w \{ U_w \,|\, t_w = k \}$. Labeling each node w by the unique atom that contains the cell c above allows us to interpret any path as a word of atom labels and define the language $L(\mathcal{T})$ of all such words. The coding tree is the system's Rosetta stone, from which everything of interest can be read. To do that, we need to define a few parameters:

- The *nesting time* $\nu = \nu(\mathcal{T})$ is the minimum depth at which any node has at most one nonvanishing child. A node v is *shallow* if $t_v \leq \nu$.
- The *word-entropy* $h(\mathcal{T})$ captures the word-length growth of the language $L(\mathcal{T})$: it is defined as the logarithm of the number of shallow nodes; $\#\mathcal{S}_\nu \leq 2^{h(\mathcal{T})}$.
- The *period* $p(\mathcal{T})$ is the maximum (prime) period of any word in $L(\mathcal{T})$.
- The *attraction rate* θ_α is the maximum, over all cells c of $\mathcal{S}_\nu$, of $\min\theta$ such that $\| f^t(\mathbf{x}) - \Pi_{\,t (\text{mod}\, p)}\, \mathbf{x} \|_\infty \leq \alpha$, for all $\mathbf{x} \in c$ and $\theta \leq t \leq \ell(\mathbf{x})$, for $\Pi_{t,k} = \Pi_{t,k}(c)$.

The arborator

This algorithm assembles the coding tree by glueing smaller pieces together. It relies on a few primitives that we now describe. The direct sum and direct product are tensor-like operations the arborator uses to attach coding trees together. The primitives `absorb` and `renorm` respectively prune and compress trees. We compile a dictionary to keep track of the tree's parameters (nesting time, period, etc) as we build it up one piece at a time.

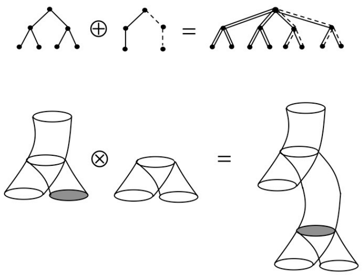

Figure 7: The two tensor operations.

Direct sum. The coding tree $\mathcal{T} = \mathcal{T}_1 \oplus \mathcal{T}_2$ models two independent systems of size n_1 and n_2. The phase space of the direct sum is of dimension $n = n_1 + n_2$. A path $w_0, w_1, \ldots$ of $\mathcal{T}$ is a pairing of paths in the constituent trees: the node w_t is of the form (u_t, v_t), where u_t (resp. v_t) is a node of $\mathcal{T}_1$ (resp. $\mathcal{T}_2$) at depth t (Fig.7). The direct sum is commutative and associative: in matrix notation, $P_w = P_u \oplus P_v$. The following relations are part of a *dictionary* that we compile to help us monitor how the coding tree's parameters evolve throughout the assembly:

$$\begin{cases} \nu(\mathcal{T}) \leq \max_i \nu(\mathcal{T}_i); \quad p(\mathcal{T}) \leq \prod_i p(\mathcal{T}_i) \\ \theta_\alpha(\mathcal{T}) \leq \max_i \theta_\alpha(\mathcal{T}_i); \quad h(\mathcal{T}) \leq h(\mathcal{T}_1) + h(\mathcal{T}_2) + 1. \end{cases}$$

Direct product. The tree $\mathcal{T} = \mathcal{T}_1 \otimes \mathcal{T}_2$ models the concatenation of two systems. The direct product is associative but not commutative. Before we get into the formalism, a few words of intuition. Consider two systems S_1 and S_1, governed by different dynamics yet evolving in the same phase space. Given an arbitrary region Λ in it, we define the hybrid system S with the dynamics of S_2 over Λ and S_1 elsewhere. Suppose we had complete knowledge of the coding tree $\mathcal{T}_i$ for each S_i $(i = 1, 2)$. Could we then combine them in some ways to assemble the coding tree $\mathcal{T}$ of S? To answer this question, we follow a three-step approach:

(i) Identify all the nodes v of $\mathcal{T}_1$ with $V_v \cap \Lambda \neq \emptyset$ and prune the subtrees below them. Because their identification will be achieved by the flow tracker through water propagation, we call the newly formed leaves *wet*. This process is called *absorption* by analogy with the absorbing states of a Markov chain.

(ii) Attach copies of $\mathcal{T}_2$ to the wet leaves; the trees must be *cropped* so that the joining phase tubes fit together.

(iii) Iterate and glue $\mathcal{T}_1$ and $\mathcal{T}_2$ in alternation, as orbits move back and forth in and out of Λ.

Returning now to the general case, we form the direct product $\mathcal{T} = \mathcal{T}_1 \otimes \mathcal{T}_2$ by, first, selecting nodes of $\mathcal{T}_1$ for absorption, which means removing the subtrees they root and marking the nodes as *wet*. Next, we attach copies of $\mathcal{T}_2$, properly cropped, to the wet leaves of $\texttt{absorb}(\mathcal{T}_1)$. With α_a denoting a parameter at least $\varepsilon n^{-O(1)}$, we add the following relations to the dictionary:

$$\begin{cases} \nu(\texttt{absorb}(\mathcal{T}_1)) \leq \max\{\, \nu(\mathcal{T}_1),\, \theta_{\alpha_a}(\mathcal{T}_1) + p(\mathcal{T}_1)\,\} \\ \nu(\mathcal{T}) \leq \nu(\texttt{absorb}(\mathcal{T}_1)) + \nu(\mathcal{T}_2) \\ \theta_\alpha(\mathcal{T}) \leq \max\{\, \theta_\alpha(\mathcal{T}_1),\, \nu(\texttt{absorb}(\mathcal{T}_1)) + \theta_\alpha(\mathcal{T}_2)\,\} \\ p(\mathcal{T}) \leq \max\{\, p(\mathcal{T}_1),\, p(\mathcal{T}_2)\,\}, \text{and} \end{cases}$$

$$h(\mathcal{T}) \leq (n+2)h(\mathcal{T}_1) + h(\mathcal{T}_2) + (n+1)\log\theta_{\alpha_a}(\mathcal{T}_1) + O(n\log n).$$

Renormalization

As the arborator assembles the coding tree, the dictionary allows us to keep track of the tree's structural parameters. Who tells the arborator which direct sums and products to perform and what wet nodes to select for absorption? Answer: the flow tracker. Although defined for undirected graphs in (5), it works just the same for directed graphs. Note that water flows in the reverse direction of the edges. This divides up the time horizon into epochs, themselves broken up into intervals by the coupling times. Over each interval, the system is *block-directional*, meaning that the agents can be partitioned into two groups A and B, so that, written in adjacency-matrix form, the communication graph looks like:

$$\mathcal{G}(\mathbf{x}) = \begin{pmatrix} \mathcal{G}_{AA}(\mathbf{x}) & \mathcal{G}_{AB}(\mathbf{x}) \\ \mathbf{0} & \mathcal{G}_{BB}(\mathbf{x}) \end{pmatrix}. \tag{6}$$

In other words, in a block-directional system, no B-agent ever links to an A-agent. We modify the flow tracker in (5) to ensure that the B-agents are kept perpetually wet. The groups A and B are analyzed recursively. The flow tracker instructs the arborator on how to express the entire system as a recursive assembly of subsystems. Treating a whole subtree recursively and squeezing it into a single node is called *renormalization*. Here is how it works:

Let $\mathcal{T}_{m\,\to\,n-m}$ denote the coding tree of a block-directional system consisting of m (resp. $n-m$) A-agents (resp. B-agents). The arrow indicates that no B-agent can ever link to an A-agent. We use the notation $\mathcal{T}_{m\,\|\,n-m}$ to indicate decoupling. The coupling time t_k is immediately followed by a renormalization phase of the form $\mathcal{T}_{w_k\,\to\,n-w_k}$, where $w_k = |W_{t_k+1}| - n + m$ is the *renormalization scale* ($k = 1, \ldots, \ell-1$). Thus, any path of the coding tree can be renormalized as[5]

$$\mathcal{T}_{m\,\to\,n-m} \Longrightarrow \underline{\mathcal{T}_{m\,\|\,n-m}}_{\,|\,t_1} \otimes \Big\{ \bigotimes_{k=1}^{\ell-1} \underline{\mathcal{T}_{w_k\,\to\,n-w_k}}_{\,|\,t_{k+1}-t_k-1} \Big\} \otimes \mathcal{T}_{m\,\to\,n-m}\,. \tag{7}$$

The subscripts indicate the lengths of the (underlined) renormalized subsystems. Varying δ may change the coding tree. To get around this difficulty, we add δ as an extra (nonmoving) agent and define the *global coding tree* $\mathcal{T}^\Delta$ with phase space $[0,1]^n \times \Delta$, for a tiny interval Δ centered at O.

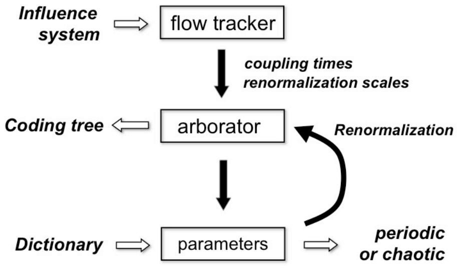

Figure 8: The algorithmic calculus.

6. BIRD FLOCKING

We briefly discuss a classic instance of a nondiffusive influence system, bird flocking, and report the results from [4]. The alignment model [7, 12, 24] we use is a trimmed-down version of Reynolds's original model [20]. In this influence system, $d = 6$ and each bird i is specified by its position z_i and velocity v_i. The undirected communication graph $\mathcal{G}(\mathbf{x})$ joins any two birds within a certain fixed distance of each other. Reordering the coordinates of the $6n$-dimensional state vector $\mathbf{x}$ as $(\mathbf{z}, \mathbf{v})$, we specify the dynamics as

$$\mathbf{x} \xmapsto{f} P(\mathbf{x}) := \begin{pmatrix} \mathbf{I}_n & \mathbf{I}_n \\ \mathbf{0} & Q(\mathbf{x}) \end{pmatrix} \mathbf{x}\,,$$

where $Q(\mathbf{x})$ is the n-by-n stochastic matrix of a (lazy) random walk on the graph $\mathcal{G}(\mathbf{x})$. The matrix P is $2n$-by-$2n$, so each entry is multiplied not by a single coordinate in $\mathbf{x}$ but by a 3-tuple; in other words, $P(x)$ acts not on $\mathbb{R}^{2n}$ but on $(\mathbb{R}^3)^{2n}$. Although the velocities can be inferred from the positions, they need to be included in the phase space to keep the system Markovian.

The system always converges in the following sense: after an initial period when the behavior can be fairly arbitrary, the birds aggregate in flocks and from that point on can only merge together. If we wait long enough, the flocks will eventually stabilize. The communication graph will remain forever fixed and the flocks will each move at a constant speed and never meet again. The fragmentation period is at most singly-exponential, but the convergence time is an exotic function which is more a reflection of the model than

[5] We leave out height-one trees needed to glue the renormalized subtrees.

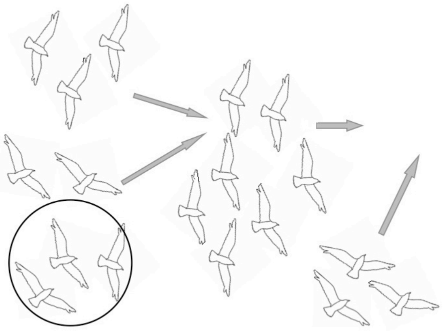

Figure 9: The bird at the center of the circle is influenced by its two neighbors in it.

of the flocking process per se. It is expressed by a tower-of-twos of logarithmic height; the exotic part is that this bound is actually tight! Even though the result can accommodate a decaying amount of noise, a more robust result awaits a realistic noisy model analysis.

7. REFERENCES

[1] Afek, Y., Alon, N., Barad, O., Hornstein, E., Barkai, N., Bar-Joseph, Z. *A biological solution to a fundamental distributed computing problem*, Science 331 (2011), 183–185.

[2] Bonifaci, V., Mehlhorn, K., Varma, G. *Physarum can compute shortest paths*, Proc. 23rd Annual ACM-SIAM Symposium on Discrete Algorithms (2012), 233–240.

[3] Camazine, S., Deneubourg, J.L., Franks, N., Sneyd, J., Bonabeau, E., Theraulaz, G. Self-Organization in Biological Systems (2001), Princeton University Press.

[4] Chazelle, B. *The convergence of bird flocking*, arXiv:0905.4241v1, 2009. Prelim. versions in Proc. SIAM SODA 2009, with improvement in Proc. ACM SoCG 2010.

[5] Chazelle, B. *The total s-energy of a multiagent system*, SIAM J. Control Optim. 49 (2011), 1680–1706.

[6] Chazelle, B. *The Dynamics of Influence Systems*, arXiv:1204.3946v1, April 2012.

[7] Cucker, F., Smale, S. *Emergent behavior in flocks*, IEEE Trans. Automatic Control 52 (2007), 852–862.

[8] Fisher, J., Harel, D., Henzinger, T.A. *Biology as Reactivity*, Communications of the ACM (2011), 54, 72–82.

[9] Hegselmann, R., Krause, U. *Opinion dynamics and bounded confidence models, analysis, and simulation*, J. Artificial Societies and Social Simulation 5, 3 (2002).

[10] Hegselmann R, Krause U. *Truth and cognitive division of labor: first steps towards a computer aided social epistemology*, J. Artificial Societies and Social Simulation 9 (2006).

[11] Hendrickx, J.M., Blondel, V.D. *Convergence of different linear and non-linear Vicsek models*, Proc. 17th International Symposium on Mathematical Theory of Networks and Systems (MTNS2006), Kyoto (Japan), July 2006, 1229–1240.

[12] Jadbabaie, A., Lin, J., Morse, A.S. *Coordination of groups of mobile autonomous agents using nearest neighbor rules*, IEEE Trans. Automatic Control 48 (2003), 988–1001.

[13] Lorenz, J. *A stabilization theorem for dynamics of continuous opinions*, Physica A: Statistical Mechanics and its Applications 355 (2005), 217–223.

[14] Lynch, N.A. Distributed Algorithms, San Francisco, CA: Morgan Kaufmann Publishers Inc., 1996.

[15] Martinez, S., Bullo, F., Cortés, J., Frazzoli, E. *On synchronous robotic networks – Part II: Time complexity of rendezvous and deployment algorithms*, IEEE Transactions on Automatic Control 52 (2007), 2214–2226.

[16] Moreau, L. *Stability of multiagent systems with time-dependent communication links*, IEEE Transactions on Automatic Control 50 (2005), 169–182.

[17] Navlakha, S., Bar-Joseph, Z. *Algorithms in nature: the convergence of systems biology and computational thinking*, Mol. Syst. Biol. (2011), 7: 546.

[18] Okubo, A., Levin, S.A. Diffusion and Ecological Problems, Springer, 2nd ed, 2002.

[19] Parrish, J.K., Hamner, W.M. *Animal Groups in Three Dimensions*, Cambridge University Press, 1997.

[20] Reynolds, C.W. *Flocks, herds, and schools: A distributed behavioral model*, Computer Graphics 21 (1987), 25–34.

[21] Seneta, E. Non-Negative Matrices and Markov Chains, Springer, 2nd ed., 2006.

[22] Sontag, E.D. *Nonlinear regulation: the piecewise linear approach*, IEEE Trans. Automat. Control 26 (1981), 346–358.

[23] Strogatz, S.H. *From Kuramoto to Crawford: exploring the onset of synchronization in populations of coupled oscillators*, Physica D 143 (2000), 1–20.

[24] Vicsek, T., Czirók, A., Ben-Jacob, E., Cohen, I., Shochet, O. *Novel type of phase transition in a system of self-driven particles*, Physical Review Letters 75 (1995), 1226–1229.

[25] Winfree, A.T. *Biological rhythms and the behavior of populations of coupled oscillators*, J. Theoret. Bio. 16, 1 (1967), 15–42.