Algorithmic Renormalization for Network Dynamics

Bernard Chazelle

Abstract—The aim of this work is to give a full, elementary exposition of a recently introduced algorithmic technique for renormalizing dynamic networks. The motivation is the analysis of time-varying graphs. We begin by showing how an arbitrary sequence of graphs over a fixed set of nodes can be *parsed* so as to capture hierarchically how information propagates across the nodes. Equipped with parse trees, we are then able to analyze the dynamics of averaging-based multiagent systems. We investigate the case of diffusive influence systems and build a renormalization framework to help resolve their long-term behavior. Introduced as a generalization of the Hegselmann-Krause model of multiagent consensus, these systems allow the agents to have their own, distinct communication rules. We formulate new criteria for the asymptotic periodicity of such systems.

Index Terms—Dynamic networks, influence systems, renormalization

1 Introduction

There is by now a wide, well-established body of techniques for decomposing networks into smaller pieces [3], [8]. These include methods based on spectral partitioning, SDP relaxation, diffusion, coarsening, flows, metric embeddings, local search, etc. By comparison, the cupboard of decomposition tools for dynamic networks looks bare. Allowing edges to come and go puts basic connectivity questions in a new light and calls for a novel set of tools [11]. This need, of course, hinges on the relevance of dynamic graphs in the first place. It is easy to argue that, in practice, networks rarely come with a fixed set of edges. On the web, for example, hyperlinks are added and deleted all the time. The same is true of social networks and virtually any large graph subject to failure. The present motivation for investigating dynamic networks emanates from a specific concern, however: the dynamics of multiagent systems and, more ambitiously, the emergence of collective behavior in living systems.

Think of how fireflies synchronize their flashes, birds form V-shaped flocks, ants find shortest paths, and bacteria perform quorum sensing. The standard approach to modeling such systems is to look at the individual organisms as *agents* endowed with two kinds of rules: *communication* rules to specify which agents "listen" to which ones under what circumstances; and *action* rules to instruct the agents on what to do with the information they acquire. Communication between the agents is channeled through a dynamic network whose topology changes endogenously as the system evolves over time. Before we describe our approach to analyzing such systems, we provide a few words of intuition.

• *The author is with the Department of Computer Science, Princeton University, NJ 08540. E-mail: chazelle@cs.princeton.edu.*

Manuscript received 7 Nov. 2014; revised 16 Mar. 2015; accepted 30 Mar. 2015. Date of publication 1 Apr. 2015; date of current version 24 Apr. 2015. Recommended for acceptance by F. Fagnani.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TNSE.2015.2419133

1.1 Parsing Dynamic Networks

The analysis relies on specific methods for tracking the propagation of information at all scales. This means monitoring how often any two groups of agents of *any size* communicate with each other. This, in turn, opens the door to divide-and-conquer. To take an extreme example, suppose that the agents can be partitioned into two subsets that never exchange information. If so, each subgroup is decoupled from the other and can be analyzed separately. Somewhat trickier is the case of two groups A and B with no edges pointing from B to A. Since information runs in the reverse direction of the edges,[1] the group B is decoupled and can be analyzed on its own. The same is not true of A, however, since the group relies on information from B for its dynamics. What to do then? The key observation is that, should the system B eventually settle around a fixed point or a limit cycle, its predictable behavior will allow us to treat the group B, after a suitable period of time, as a single *block-agent*: think of it as some sort of super-agent with static or periodic behavior. In this way, the original system can now be analyzed by first working on B and then turning our attention to a *block-directional* system of $|A|+1$ agents ($+1$ because of the block-agent). Think of an Ancient Régime monarchy where the royal court B has its own internal dynamics, one that is vital to the dynamics of A yet entirely oblivious to it.

How do we choose the partition? There are many considerations at play but one of them stands out: updating A and B should be as infrequent as possible. To take a fanciful example, consider a basketball game. It might be sensible to choose one team to be A and the other one B on the grounds that the ball stays within a given team more often than not. On the other hand, this might not be true when the action is near the basket and possession switches rapidly between shooters, blockers, and rebounders. One could then imagine changing the choice of the groups A and B every now and

1. An edge pointing from me to you indicates that I am listening to you and therefore that the information flows from you to me.

then in order to keep the interactions between the two groups to a minimum. Yet to find the absolute minimum is not an option. The choice must not only mirror the flow of information across the networks but also proceed on-line: in particular, one should not have to look ahead into the future to decide how to split the agents into groups A and B.[2]

The dynamic assignment of A and B partitions the timeline $t = 0, 1, 2, \ldots$ into a sequence of consecutive intervals within which the assignment is time-invariant. Within each such interval, the intra-communication among the agents of A (or B) could be itself quite complex, thus prompting us to partition A and B. Proceeding in this way recursively produces a hierarchical decomposition of the timeline, i.e., a *parse tree* such as $\big((01)(234)\big)\big((5)(67)(8\cdots)\big)$. In this example, the timeline forms the root of the tree. The root has two children associated with the time intervals 01234 and $5678\cdots$, which themselves have respectively two and three children, with intervals 01 and 234 for one and 5, 67, and $8\cdots$ for the other. As we show below, this allows us to view the transmission of information across the agents at all relevant timescales and "renormalize" the system accordingly. The parsing procedure is a *message passing* algorithm—as are, we should point out, most spectral methods and belief propagation algorithms used for fixed networks (a word we use interchangeably with "graphs").

1.2 Temporal Renormalization

The systems are deterministic, so a given starting configuration of the agents yields a single orbit. The question is under which conditions is the orbit attracted to a limit cycle. Whereas attraction to a fixed point can often be analyzed by looking at the orbit of interest and setting up a suitable Lyapunov function for it, asymptotic periodicity does not lend itself to this kind of investigation as easily. It is often necessary to reason about the space of *all* orbits, which is, of course, a source of complication: as though trying to understand the behavior of an infinite sequence of networks was not hard enough, we must consider the space of all possible such sequences at once. How do we even encode such a structure? Each orbit gives rise to a parse tree, so the challenge is how to organize the set of all possible parse trees into a single structure: this is the role of the *coding tree*. By way of analogy, consider the set of all English sentences. Via lexicographic ordering, we can organize this set as an infinite coding tree whose paths are in bijection with the sentences. Such data structures are known in computer science as prefix, digital, or radix trees. Being associated with a sentence, each path can be parsed in accordance with the rules of English grammar. The key insight is that sentences whose paths share long prefixes will have parse trees with big overlaps: this in turn allows us to infer a hierarchical decomposition of the coding tree itself. In other words, we can interpret the coding tree as a tree whose nodes are themselves trees whose nodes are themselves trees, etc. This is a way of parsing not just individual paths, but the entire coding tree itself.

2. At the risk of belaboring the obvious, we mention that this concerns only the *analysis* of the dynamical system: the choice of A and B has no incidence on the dynamics itself.

In the case of influence systems, the coding tree is infinite and each path in it corresponds to a (chronological) sequence of communication networks. The trick is to infer from the parse trees associated with the paths a recursive decomposition of the entire coding tree itself. This can be thought of as a form of temporal renormalization not carried out in closed form but algorithmically (a remark the discussion below should help clarify). What makes the coding tree useful for the analysis is that it is not merely a combinatorial structure but a geometric one as well: indeed, a path of the coding tree corresponds to a whole set of nearby orbits that share the same sequence of networks. These look like "tubes" in spacetime $\mathbb{R}^n \times \mathbb{N}$. For example, an influence system with two agents would produce a coding tree in three dimensions that might look a bit like a real tree of the sort we encounter in nature (with the difference that distinct branches can occupy the same space). Paths are branches whose cross-sections have an area (or volume in higher dimensions). For reasons we discuss below, bounding the rate at which new branches are formed (entropic growth) and how thin they get as we move further from the root (dissipation) is the key to the analysis. Roughly speaking, bushy trees with thick branches correspond to chaotic systems. We have proven in a suitable model that a tiny random perturbation of the input sends the agents into a limit cycle almost surely [6]. We follow a different tack here and establish a more general result under plausible heuristic assumptions. To replace these assumptions by established facts appears to be a major mathematical challenge which is left as an open problem. As a side-benefit, our approach gives us a platform for working out the renormalization scheme in full, something that was not needed in [6].

We define the model of diffusive influence formally in the next section (Section 2) and discuss specific constructions as a warmup. In Section 3 we show how to parse an arbitrary sequence of networks. The section is of independent interest. By generalizing the idea of breadth-first search, this gives us a principled way of tracking the propagation of information (be it rumors, viruses, or trends) in dynamic graphs. We sketched the idea earlier [6] but we provide here the first complete treatment of the technique. In Section 4 we lay down the foundations of algorithmic renormalization and explain in Section 5 how to use the framework to mediate between the two "forces" driving the system: entropy and dissipation. Finally, we show how to analyze a diffusive influence system in Section 6 by setting up the relevant recurrence relations. It bears mentioning that the first three sections are highly general and apply to any sequence of directed graphs over a fixed set of nodes. It is only in Section 4 that the piecewise-continuity of the dynamical system is used and in Sections 5 and 6 that the "averaging" nature of diffusive influence systems is exploited. In particular, the algorithmic renormalization scheme applies to any discrete-time network-based dynamics.

2 The Model

The model is inspired by the classic *Hegselmann-Krause* (H-K) model of opinion dynamics [9]. Part of the appeal is the simplicity of its definition: Fix a real parameter $r > 0$ and initialize n agents on the real line $\mathbb{R}$. At each time step, each

agent moves to the mass center of its neighbors, in this case, any agent at distance r or less. In other words, the position $x_i \in \mathbb{R}$ of agent i becomes $x_i \leftarrow |N_i|^{-1} \sum_{j \in N_i} x_j$ at the next step, where $N_i = \{\, j : |x_i - x_j| \le r \,\}$. The updates are carried out synchronously and repeated ad infinitum. Numerical simulations suggest fast convergence. Although the typical relaxation time has yet to be pinned down, it is known that, within polynomial time, the agents end up frozen in single-point clusters at least r away from each other [1], [5], [10], [12], [13], [14], [15].

2.1 Generalized HK-Systems

There are three natural ways to extend the original Hegselmann-Krause model. One of them is to lift the agents into higher dimension instead of confining them to the real line. Regardless of the dimension, the agents will still converge to a fixed configuration in polynomial time [1]. Another modification is to replace the update rule with a weighted mass center. Assuming nonzero self-weights, convergence is still guaranteed but it might become asymptotic. To see why, consider two agents at distance less than r moving toward each other one third of the way at each step: $x_1 \leftarrow \frac{1}{3}(2x_1 + x_2)$ and $x_2 \leftarrow \frac{1}{3}(x_1 + 2x_2)$. In the phase space $\mathbb{R}^2$, any orbit is attracted exponentially fast to the line $x_1 = x_2$. The third type of extension is to redefine what it means to be a "neighbor." Despite massive empirical evidence that the system should converge to a fixed point, a change as simple as allowing a different threshold r_i for each agent i produces a dynamics that is still unresolved. On the other hand, certain minor variants are known to produce periodicity and even chaos [6]. To grasp the subtleties at play, we need a more expressive palette to work with. We begin with a slight generalization of *HK* systems (lin-DNF) and then push the generalization to its natural limit (diffusive influence systems). Though looking vastly different to the naked eye, this is an illusion: the two formulations are in fact equivalent (modulo an adjustment in the number of agents).

A *lin-DNF* is a set of linear constraints expressed as a disjunction of conjunctions, i.e., $P_1 \vee P_2 \vee \cdots$, where each P_l is of the form $Q_1 \wedge Q_2 \wedge \cdots$ and each Q_k is a halfspace $u^T x \le v$ (or $u^T x < v$), where $u, x \in \mathbb{R}^n$. We define the *communication graph* $G(x)$ by associating a node to each agent. The edges of the n-node graph $G(x)$ depend on $x = (x_1, \ldots, x_n)$: for each pair $i \neq j$, we choose a lin-DNF ϕ_{ij} and we declare (i, j) to be an edge of $G(x)$ if $\phi_{ij}(x)$ is true. A natural extension of *HK* systems is provided by the update rule: for $i = 1, \ldots, n$,

$$x_i \leftarrow \frac{1}{|N_i|} \sum_{j \in N_i} x_j \quad \text{and } N_i = \{\, j \,|\, (i,j) \in G(x) \,\}. \tag{1}$$

Note that the original *HK* system is put in lin-DNF form very simply by setting $\phi_{ij}(x)$ as

$$\big(x_j - x_i \le 0 \,\wedge\, x_i - x_j \le r\big) \vee \big(x_i - x_j \le 0 \,\wedge\, x_j - x_i \le r\big). \tag{2}$$

2.2 Diffusive Influence Systems

The definition of a diffusive influence system is identical to that of an *HK* system, with the only difference coming from the communication network, specifically the criterion used to include a pair (i, j) as an edge of $G(x)$. We equip the pair with its own first-order predicate $\phi_{ij}(x)$ and make (i, j) an edge of $G(x)$ if and only if $\phi_{ij}(x)$ is true. Recall that a first-order predicate over the reals is a logical sentence consisting of universal and existential quantifiers bound to real variables $y_1, \ldots, y_m$, along with (strict and nonstrict) polynomial inequalities from $\mathbb{Q}[x_1, \ldots, x_n, y_1, \ldots, y_m]$ tied together by Boolean connectives $\vee, \wedge$. Formulation (2) is a particularly simple instance of a first-order predicate, as it lacks quantifiers and bound variables, and uses only linear polynomials.

Do we really need the full first-order theory of the reals? The answer is yes. Here is a simple example taken from the field of robotics [2]. The agents are represented by points in a room littered with obstacles: this means that x_i is now a point in $\mathbb{R}^3$ instead of a single real number. The graph $G(x)$ is defined as the "constrained Delaunay graph": this means that (i, j) is an edge of $G(x)$ if there exists a sphere passing through the agents i and j with no agent or obstacle protruding inside. An edge (i, j) is characterized by the truth-condition of a first-order formula with both existential and universal quantifiers. In plain English, the formula reads as follows: "*There exist a center and a radius such that, for all points p on an obstacle or at an agent, p does not lie in the corresponding ball.*" This is formally expressed as a first-order predicate over the reals: the formula contains the symbols $\exists, \forall$, Boolean connectives, and a number of bound variables, along with a set of polynomial inequalities with rational coefficients. We see that, even for a simple communication network from robotics, alternating quantifiers are already necessary.

Whereas the update rule itself (1) is kept deliberately simple, it is the high expressiveness of the language in which the communication network is specified that gives diffusive influence systems their distinctive feature. Because virtually any edge selection rule is acceptable, the definition meets the primary objective of the model, which is to allow the agents to have their own, *distinct* communication rules: indeed, any two agents can use entirely different criteria to pick their neighbors.

2.3 Equivalence of the Models

Phrased in the language of first-order logic, diffusive influence systems seem far removed from the lin-DNF formulation of generalized *HK* systems. The two definitions are in fact equivalent. Indeed, any diffusive influence system can be put in lin-DNF form after suitable transformation. This involves a number of steps including quantifier elimination, linearization, tensoring, addition of new agents, etc. See [6] for details. It might seem surprising that polynomials of arbitrary degree can thus be replaced by linear ones, but this is made possible by the "convex" nature of the map (1).

We can go further and rid the lin-DNF formulation of all its Boolean connectives. To do that, we consider the set $\mathcal{D}$ of hyperplanes formed by the linear constraints in the formulas ϕ_{ij}. By adding a variable if necessary,[3] we can always assume that these hyperplanes, called the *discontinuities*, are of the form $u^T x = 1$. They form an arrangement whose full-dimensional cells c, the *continuity pieces* of f, are each

3. For example, $x_1 = 0$ might becomes $x_1 + x_0 = 1$, with x_0 set to 1. Perturbation now involves both x_0 and x_1.

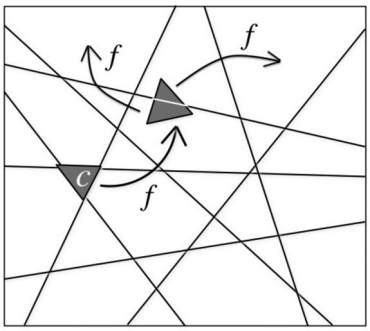

Fig. 1. A map f for $n = 2$. Note that the continuity piece c is mapped continuously but $f(c)$ is not. It is this sort of fragmentation that makes the dynamics difficult to analyze.

assigned a directed n-node graph G_c. We extend the labeling to all of $\mathbb{R}^n$ by defining $G(x)$ as the graph with no edges if x lies on a discontinuity and $G(x) = G_c$ if x lies in cell c.

For convenience (specifically, to avoid cluttering the notation), we assume that the number of hyperplanes in $\mathcal{D}$ is at most polynomial in n, so that the number of graphs is bounded by $n^{O(n)}$. After a translation along the all-one vector **1** and rescaling if necessary, we can always choose the unit cube $X \triangleq [0, 1]^n$ as the phase space, so the continuity pieces are bounded open n-dimensional polyhedra. We summarize the main features of a diffusive influence system:

Definition 2.1. *A diffusive influence system (f, X) is a piecewise-linear system specified by a map f from X to itself, $x \mapsto P(x)x$, where $P(x)$ is the incidence matrix of the communication graph $G(x)$ augmented with self-loops, with each row rescaled so as to sum up to 1. The matrix $P(x)$ is constant over the cells of a hyperplane arrangement in $\mathbb{R}^n$.*

We assume a positive diagonal to avoid spurious periodicities of little mathematical significance.[4] We define $P(x)$ as the identity matrix for any x on a discontinuity or on the boundary of the unit cube. Our discussion generalizes easily to update rules based on weighted mass centers, and we use uniform weights across each matrix row only for notational convenience.

A more substantive simplification in this paper is our assumption that the systems are *locally coupled*, meaning that each $\phi_{i,j}$ depends only on x_i, x_j and not on the other agents. Local coupling means that the presence of the edge (i, j) in $G(x)$ is indicated by the sign-condition of x with respect to discontinuities $u^T x = 1$ for which only the coefficients u_i and u_j may be nonzero. To summarize, a diffusive influence system (f, X) is specified by an arrangement of hyperplanes, where each continuity piece c (an open n-cell) is labeled by a directed graph whose edges (i, j) depend only on the projection of c onto the (x_i, x_j) plane. Fig. 1 illustrates the case of two agents.

2.4 A New Brand of Renormalization

Most dynamic regimes can be observed in low dimensions (e.g., fixed-point attraction, periodicity, quasi-periodicity, chaos, strange attractors), which is why research in dynamics has had a tendency to focus on systems with few degrees of freedom. At the opposite extreme, statistical mechanics prefers infinite-dimensional systems because of the mathematical benefits given by the thermodynamic limit. Influence systems sit in the "mesoscopic" middle: many agents, but still too few, too diverse, and too autonomous for any classical treatment. These network-based systems seek to model the emergence of collective behavior through the multiplicity of individual interactions. But how do we go about analyzing the endogenous interaction of many diverse agents?

The physical analogue would be to allow each particle in a gas to follow its own laws of physics. While this feature alone may push influence systems beyond the purview of statistical mechanics, the shared concept of *renormalization* plays a key role—just as it does in low-dimensional dynamics (e.g., the logistic map). The basic idea is to rescale a system while retaining its dynamics. In our case, scaling is with respect to both time and the number of agents. Renormalizing an influence system is to produce another one with fewer agents and a behavior resembling a coarse-grained version of the original one. If the communication graph was fixed then graph clustering techniques might suggest a way to do that: agents interacting with their neighbors in a grid, for example, can be clustered into subgrids, so that a $\sqrt{n} \times \sqrt{n}$ grid of n agents can be renormalized as a $\sqrt{n/4} \times \sqrt{n/4}$ grid of "block-agents" consisting of four agents each. Naturally, this decomposition is to be iterated recursively.

This form of block-spin renormalization, famously introduced by Kadanoff for the Ising model, works only if the interaction among the block-agents can be understood reasonably well without having to track the precise behavior of their constituent agents. In other words, a block-agent needs to be able to "hide" the internal role of its agents from the outside. The other requirement is that the coarse-grained system should look like a "blurred" version of the larger one. To achieve this, it is customary to view the coarse-graining process itself as a dynamical system mapping an Ising model to another, simpler one. The goal is then to adjust the coarse-graining parameters to home in on a fixed-point attractor and thus allow the basic behavior to be retained throughout the renormalization steps.

What is the role of time in the case of influence systems? Presumably, a block-agent has a characteristic timescale: the time it takes its own agents to settle in their long-term mode. If this time is the same for all the block-agents, then we can rescale the time axis by redefining its basic unit at every step in the coarse-graining hierarchy. To carry out this plan, of course, one needs to cope with the ever-changing topology of the communication graph. Worse, the changes being endogenous, one cannot postulate a prior on the graph distribution. Determinism is not necessarily an impediment to using tools from statistical mechanics [4] because the bifurcation analysis needs to focus on the "edge of chaos," a region replete with pseudorandomness.

In [6] we outlined an approach to network-based renormalization and derived sufficient conditions for the asymptotic periodicity of diffusive influence systems. We revisit the method here and show how to replace a certain timing

4. For example, consider the two-node cycle $G(x)$ with $P(x) = \left(\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right)$.

assumption by a general-position heuristic under which the critical region can be shown to be of measure zero—the details are given below in Section 6. This is an opportunity to develop the renormalization framework to its fullest, something our earlier timing assumption [6] allowed us to bypass. The algorithmic renormalization of dynamic networks is a general, powerful idea likely to be useful elsewhere.

2.5 Mixed Timescales

Diffusive influence systems have been shown to span the entire range of dynamic modes, from fixed-point attraction to chaos. Predicting their behavior can be undecidable [6]. It is no surprise therefore that they should exhibit periodic orbits of any length. Remarkably, long limit cycles with large basins of attraction can be manufactured as well: think of them as extremely long periodic orbits robust under perturbation. We give such a construction below: the period is a tower-of-twos of height proportional to the number of agents. This type of behavior may be pathological but it touches upon a phenomenon at the heart of influence systems: the *mixing* of timescales. If one regards attraction as a process of amnesia, then to be caught in a limit cycle means to forget one's origins. Indeed, an agent begins with an unbounded amount of information (encoded in its initial position) but, once trapped in a limit cycle, carries only a few positional bits from then on. This near-total loss of memory is what we term "amnesia."

In a diffusive influence system, some agents can "collude" to form memory devices that can hold information over periods of time much higher than their own characteristic times. These gadgets allow agents to recover their memory just as they are about to reach amnesia, which in turn allows them to extend their internal characteristic time and delay the eventual attraction to a limit cycle. This archival mechanism can be nested hierarchically to create bigger and bigger timescales. Remarkably, such constructions are robust to noise. The existence of hierarchical schemes that combine to create new dynamical regimes points to the need for time-based renormalization, the main theme of this work. The ability to create higher scales both in length and time seems a necessary component of any living system, and the relevance of the present investigation should be appreciated in this light.

THE VERY SLOW CLOCK (VSC). We describe an n-agent diffusive influence system with a limit cycle of length roughly equal to a tower-of-twos of height $n/2$. This is "two to the two to the two..." repeated $n/2$ times. The design relies on a small gadget which converts an $(n-2)$-agent system with period $p_{n-2} = k$ into an n-agent system with period $p_n \approx 2^k$. All gadgets share two static agents a, b positioned at -1 and 1, respectively; for convenience, we do not count them in n. Each gadget also has its own pair of mobile agents. Applying the construction (roughly) $n/2$ times leads to the desired result. As an inessential but convenient exception, we allow zero entries in the matrix diagonals. The n (mobile) agents are at positions $x_1, \ldots, x_n$. The update rules will automatically ensure that, for all i: $x_i \leq 0$ precisely when the time t is $-1 (\mathrm{mod}\, p_n)$; and $x_i = (-1)^i$ precisely when $t = 0 (\mathrm{mod}\, p_n)$. For $n = 2$, we use a, b to (easily) engineer the system so that (x_1, x_2) follows the periodic orbit $(-1, 1), (1, 1), (1, -1), (-1, -1)$. In general, to form $G(x)$, we extend the communication graph of the $(n-2)$-agent system by adding to it the two nodes n and $n-1$, together with the edges specified below:

i) If $x_i \geq 0$ for some $i < n-1$, then add the self-loop (x_{n-1}, x_{n-1}) and the edge (x_{n-1}, x_n). Note: agent $n-1$ moves half way toward agent n; this case is the most frequent, as it occurs whenever $t \neq -1 (\mathrm{mod}\, k)$.

ii) Otherwise, add (x_{n-1}, a); if $x_n \leq 0$ then add (x_n, b) else add (x_n, x_{n-1}). Note: in the first (resp. second) case, agent n moves to the location of agent b (resp. agent $n-1$) and agent $n-1$ moves to -1; this occurs only when $t = -1 (\mathrm{mod}\, k)$, which is typically very rare.

The mechanism of the clock is quite simple. The n-agent system consists of a subsystem of $n-2$ agents (the "subclock") and a gadget that interacts only with the static agents a, b. There is never any edge between the gadget and the subclock: their interaction is mediated entirely through the positional constraints.

At time 0, the odd-labeled agents are at -1 and the others at 1. Rule (i) kicks in for the next $k-1$ steps, pulling agent $n-1$ toward agent n, so they end up separated by $\approx 2^{-k}$. At time $k-1$, the subclock is entirely in negative territory, so rule (ii) kicks in. If agent n (which stays always to the right of agent $n-1$) is still at a positive location, then it slides left to the position of agent $n-1$ while the latter is repositioned at -1. This is the key moment: in k steps, the system has managed to move agent n to the left by as little as $\approx 2^{-k}$. As long as $x_n > 0$, we can repeat these k steps, which means the subclock can run through roughly 2^k cycles. (The leftward slide of agent n varies a little in length at each iteration but always remains on the order of 2^{-k}.) Note that we do not rush to reset the subclock as soon as $x_n \leq 0$ but, rather, allow it to complete its natural cycle. The initial position of the agents does not require fine calibration and the astronomical period is robust under perturbation.

The recursive construction of the *Very Slow Clock* suggests how renormalization should proceed. The n-agent system VSC_n results from nonlocal-coupling between a clock VSC_{n-2} and a gadget consisting of two private agents and two shared ones. The characteristic timescale of the gadget is exponentially higher than that of the VSC with which it is coupled. Renormalizing the system is easy because it was essentially put in by hand. In general, to tease out the subparts of the dynamics that can be "factored out" is no easy task. In the end, renormalization is about dimension reduction. When the graph never changes, the dynamics can be expressed by the powers of a fixed matrix and the dimension reduction is done by breaking down the system into its eigenmodes. When the communication topology keeps changing, however, linear algebra is of little use. The dimension reduction cannot be carried out in "closed form." It would seem that it can only be performed as a step-by-step process that evolves alongside the dynamics of the system. This is what algorithmic renormalization tries to achieve. Note that the VSC is not locally coupled. In fact, to avoid its pathological (yet robust) behavior is the reason for assuming local coupling.

Theorem 2.2. *For any $n > 2$, there exists an n-agent diffusive influence system with a periodic orbit of length equal to a tower-of-twos of height proportional to n. The dynamics is robust under perturbation.*

3 Parsing Graph Sequences

The algorithmic renormalization of influence systems relies on a general procedure of independent interest: a method for *parsing* arbitrary graph sequences. Recall that the analysis of influence systems rests crucially on our ability to decompose an infinite sequence of communication graphs hierarchically. By building a tree on top of the network sequence, we are thus able to break down the original system into subsystems. This is similar to forming the parse tree of an ordinary English sentence: whereas the latter is driven by the rules of syntax, what sort of rules shall we then use to parse a sequence of communication graphs? In a nutshell, the idea is to track the propagation of information across the networks (think of a virus spreading in a population) and use the pauses (for example, due to poor connectivity) as breakpoints to guide the construction. The parsing algorithm can be seen as a grand generalization of breadth-first search for dynamics graphs.

Given a (finite or infinite) sequence $\mathcal{G} = G_0 G_1 G_2 \cdots$ of directed graphs over the same set of n labeled nodes, make the lowest-labeled node "wet" and declare wet any node pointing to a wet node in G_0. Next, switch to G_1 and call wet any node that is either already wet or points to a wet node via an edge of G_1. Water flows only to immediate neighbors in a single step and wet nodes remain wet. We use water as an anology but we could substitute rumor propagation or viral contagion. Transmission runs in the reverse direction of the edges, following the principle that information is acquired only by pointing to it. As soon as all the nodes become wet (if ever), the subsequence of graphs considered up to this point is said to form a *wave*. When this occurs, we dry up all of the n nodes and repeat the entire process, restarting from the current graph in the sequence. We begin the second wave at the same lowest-labeled node used earlier. It might happen that some nodes never get wet or that wetness propagation is interrupted for long periods of time before resuming again. What to do in such cases is at the heart of the parsing procedure, which is called the *flow tracker*.

3.1 Making Waves

We treat the sequence $\mathcal{G}$ as a word in a language over an alphabet of 2^{n^2} letters, where each letter represents an n-node graph (or, equivalently, an n-by-n 0-1 matrix). Parsing the sequence means producing a *parse tree* $T(\mathcal{G})$: this is a rooted tree whose leaves are associated with the graphs $G_0, G_1, G_2, \ldots$ from left to right and whose internal nodes reflect important transitions during the flooding. As we shall see, even when the sequence $\mathcal{G}$ is infinite, the depth of the tree remains finite. For example, if all the graphs are strongly connected (or if the graphs are connected but undirected), then[5]

$$T(\mathcal{G}) = (\,(G_0 \cdots G_{i_1})^{wave}\ (G_{i_1+1} \cdots G_{i_2})^{wave}\ \cdots)^{seq}, \quad (3)$$

where i_k is the time at which the graph becomes entirely wet for the k-time. The tree $T(\mathcal{G})$ has depth only two because it is a simple case. The superscripts indicate the nature of the nodes: the node $(\cdots)^{seq}$ is the root of the parse tree of the original sequence; its children $(\cdots)^{wave}$ correspond each to a wave and their number can be infinite; the leaves of the tree are given by G_0, G_1, etc. In this particular case where all the graphs are assumed to be strongly connected, the waves are less than n-long since every step causes at least one dry node to get wet. In general, there is no a priori bound on the wavelengths. The first two levels of any parse tree look like (3); most trees have higher depth, however, and recursive rewriting rules are needed to express them.

We now put this informal description on a more rigorous footing. Recall that $\mathcal{G}$ is an arbitrary (bounded or not) sequence of directed graphs over n labeled nodes. We denote by B the set of nodes initialized as wet. So far, we have considered only the case where B is the singleton consisting of the lowest-labeled node. A variant of the procedure will soon require wetting more than one node at the outset, hence the use of the set B. We put together a partial parse tree in the form of $(\texttt{wavemaker}\,(\mathcal{G}, B))^{seq}$, which we then proceed to refine. This is how `wavemaker` works:

wavemaker $(\mathcal{G}, B)$ *" $\mathcal{G}$ is a graph sequence and $B \subset [n]$ "*

$W_0 \leftarrow B$ and print '('
for $t = 0, 1, \ldots$
  print 'G_t'
  $W_{t+1} \leftarrow W_t \cup \{\, i \,|\, \exists j \in W_t \text{ such that } (i,j) \in G_t \,\}$
  if $|W_{t+1}| = n$ then $W_{t+1} \leftarrow B$ and print '$)^{wave}($'
print '$)^{wave}$'

We bend standard programming language conventions by allowing the last print statement to be executed even if the sequence $\mathcal{G}$, hence the for-loop, is infinite.[6] The output of `wavemaker` looks like the right-hand side of (3). A wave can be either complete or incomplete, depending on whether all the nodes get wet: a wave is *complete* if it is terminated by the successful outcome of the conditional "if $|W_{t+1}| = n$." For that reason, an infinite wave is always incomplete but the converse is not true: a finite sequence $\mathcal{G}$ might simply run out of graphs to flood all the nodes, hence leaving an incomplete final wave. If $\mathcal{G}$ is infinite, then either all the waves are finite but their number is not (as is the case with strongly connected graphs) or there are a finite number of waves but the last one is infinitely long. In this case, the partial parse tree $(\texttt{wavemaker}\,(\mathcal{G}, B))^{seq}$ is of the form:

$$(\,(G_0 \cdots G_{i_1})^{wave} \cdots (G_{i_l+1} \cdots)^{wave}\,)^{seq}. \quad (4)$$

Why isn't this the end of the story? The reason is that waves themselves need to be parsed, especially when they are very long. For example, suppose that B is a singleton and its node is always disconnected from the other ones. In that case, water never flows anywhere and `wavemaker` parses $\mathcal{G}$ as $((G_0 G_1 G_2 \cdots)^{wave})^{seq}$, which is as simple as it is useless. Clearly, parsing must be carried out even during flooding delays. We explain now how to do this.

5. We shall often represent trees as nested parenthesis systems. Here is a tree whose root has two children, with the leftmost one having two children of its own: ((()()) ( )).

6. In the odd chance that `wavemaker` prints a single spurious $()^{wave}$ at the end, we simply erase it.

3.2 Parsing a Wave

A flooding delay implies that the current "wet" set W_t stops growing: it could be momentary or permanent (there is no way to tell ahead of time). A delay occurs when the graphs G_k have no edges pointing from the dry nodes to the wet ones, i.e., from $[n] \setminus W_t$ to W_t. This motivates the definition of a *block-sequence* as any subsequence of $G_0G_1G_2\cdots$ such that, for some partition $[n] = A \cup B$ of the nodes, no edge from the subsequence points from B to A: the block-sequence is said to be of *type* $A \to B$. Let $\mathcal{G} = G_0G_1G_2\cdots$ be a wave. Recall that, in the case of a complete wave, this means that all the nodes are wet at the end but not earlier. As usual, B is set to the singleton consisting of the lowest-labeled node. By definition of a wave, `wavemaker` $(\mathcal{G}, B)$ terminates at or before the first time $|W_{t+1}| = n$. Let $t_1, t_2, \ldots$ be the times t at which $W_t \subset W_{t+1}$ (i.e., strict inclusion). These *coupling times* indicate when the water propagates. They break down the wave $\mathcal{G}$ into block-sequences $\mathcal{G}_k$ via $(\texttt{blockseqmaker}\,(\mathcal{G}, B))^{wave}$. Using superscripts to distinguish the wave nodes from the block-sequence kind, the output is of the form:

$$\begin{aligned}&(\,(\mathcal{G}_{A_0\to B_0})^{bseq}\ G_{t_1}\,(\mathcal{G}_{A_1\to B_1})^{bseq}\\ &\qquad\qquad G_{t_2}\,(\mathcal{G}_{A_2\to B_2})^{bseq}\ \ G_{t_3}\ \cdots)^{wave}.\end{aligned} \tag{5}$$

blockseqmaker $(\mathcal{G}, B)$ *"$\mathcal{G}$ is a wave and $B \subset [n]$"*

$W_0 \leftarrow B$ and print '('
for $t = 0, 1, \ldots$
  $W_{t+1} \leftarrow W_t \cup \{\, i \mid \exists j \in W_t \text{ such that } (i,j) \in G_t \,\}$
  if $W_t = W_{t+1}$ then print 'G_t'
    else print '$)^{bseq}\ G_t($'
print '$)^{bseq}$'

As was the case with `wavemaker`, we remove empty parenthesis pairs $()^{bseq}$; so, for example, in the case of strongly connected graphs, the output of `blockseqmaker` for a wave will simply be $G_{t_1}G_{t_2}\cdots$ $(t_i = i-1)$ since there are no delays. In general, the block-sequence $\mathcal{G}_{A_k\to B_k}$ is a maximal subsequence of the wave $\mathcal{G}$ that witnesses no water propagation: it is of type $A_k \to B_k$, where $A_k = W_{t_k+1}$, $B_k = [n] \setminus A_k$, and $B \subseteq A_k \subset A_{k+1}$. The initialization requires setting $t_0 = -1$ so that $A_0 = W_0 = B$. Note how the letters A and B switch roles (more on this below). Until now, B has been a singleton: parsing block-sequences, our next topic, will change all that.

3.3 Parsing a Block-Sequence

We have built the first three levels of the parse tree so far. The root is associated with the full sequence and its children with its constituent waves. The root's grandchildren denote either single-graphs (leaves of the tree) or block-sequences. Parsing the latter repeats the treatment of general sequences described above with two small but crucial differences. Let $\mathcal{G}_{A\to B}$ denote a block-sequence of type $A \to B$. First, we break it down into *block-waves* by applying `wavemaker`$(\mathcal{G}_{A\to B}, B)$. The output is of the form

$$(\,(G_0\cdots G_{i_1})^{bwave}\ (G_{i_1+1}\cdots G_{i_2})^{bwave}\ \cdots)^{bseq}. \tag{6}$$

We note that B is now wet at the outset. This contrasts with the circumstances behind the creation of a block-sequence of type $A \to B$, which feature wet A, dry B, and delay in water propagation. This radical change in wetness status explains the need for a recursive method that encapsulates wetness status by local variables hidden from the outside. It is in that sense that the renormalization is truly *algorithmic*.

The block-waves constitute the children of the node associated with the block-sequence $\mathcal{G}_{A\to B}$. Let $\mathcal{H}$ denote any one of the subsequences of the form $G_{i_k+1}\cdots G_{i_{k+1}}$ $(i_0 = -1)$. To extend the parse tree further, we replace in (6) each occurrence of the block-wave $(\mathcal{H})^{bwave}$ by $(\texttt{blockseqmaker}\,(\mathcal{H}, B))^{bwave}$. Introducing new notation, we parse each block-wave as

$$\begin{aligned}&(\,(\mathcal{G}_{A\|B})^{dec}\ G_{t_1}\,(\mathcal{G}_{A_1\to B_1})^{bseq}\\ &\qquad\qquad G_{t_2}\,(\mathcal{G}_{A_2\to B_2})^{bseq}\ \ G_{t_3}\ \cdots)^{bwave}.\end{aligned} \tag{7}$$

The parsing of a block-wave differs from that of a wave (5) in two ways:

- First, we notice the difference in the first term $\mathcal{G}_{A\|B}$. As we scan through the sequence $\mathcal{H}$, we may have to wait for a while before some edge actually joins A to B. Since there is no edge from B to A, the two sets are decoupled until then (hence the superscript *dec*). In other words, A and B form a cut in all the graphs in the sequence $\mathcal{G}_{A\|B}$: such "decoupled" sequences are parsed as a single leaf in the tree.
- The second difference is more subtle. When parsing an ordinary wave, recall that the block-sequence $\mathcal{G}_{A_k\to B_k}$ satisfies $A_k = W_{t_k+1}$. In the case of a block-wave, instead we have $A_k = W_{t_k+1} \setminus B$ (and, as usual, $B_k = [n] \setminus A_k$). Note that it is still the case that $A_k \subset A_{k+1}$. This assignment of A_k satisfies the main requirement of a block-sequence of type $A_k \to B_k$, namely the absence of any edge from B_k to A_k.

We had to amend the old invariant $A_k = W_{t_k+1}$ because of inductive soundness. Here is an example to illustrate why: Let $G_0 = a \to b \quad c$; $G_1 = a \to b \to c$; $G_2 = a \leftarrow b \to c$; $G_3 = a \quad b \to c$; and $G_4 = G_1$. In this notation, G_0 has a single edge (a, b). The block-wave $G_0\cdots G_4$ of type $\{a, b\} \to \{c\}$ is parsed as

$$(\,(\mathcal{G}_{\{a,b\}\|\{c\}})^{\text{dec}}\ G_1\,(\mathcal{G}_{\{b\}\to\{a,c\}})^{bseq}\ \ G_4\,)^{bwave}, \tag{8}$$

where $\mathcal{G}_{\{b\}\to\{a,c\}} = G_2G_3$. The reason why setting $A_k = W_{t_k+1}$ is a bad idea is that the block-sequence G_2G_3 would then be interpreted as $\mathcal{G}_{\{b,c\}\to\{a\}}$, which would have the effect of making a block-sequence of type $\{b, c\} \to \{a\}$ a child of another one of type $\{a, b\} \to \{c\}$. To ensure the soundness of the recursion, we need the cardinality of the A-set to drop by at least by 1 as we go from from parent to child: this property is ensured by setting $A_k = W_{t_k+1} \setminus B$ since this implies that $A_k \subset A$.

There is a subtlety in the recursion (for the curious reader only). The arrival of G_4 brings in the edge (a, b), which is incompatible with a block-sequence of type $\{b\} \to \{a, c\}$. It is therefore no surprise to see $\mathcal{G}_{\{b\}\to\{a,c\}}$ in (8) terminate and give way to the single-graph G_4. Actually this is not the reason for the termination of the block-sequence. Indeed, if G_4 were of the form $b \to a \to c$, the block-sequence would still terminate regardless of its continuing compatibility with the type $\{b\} \to \{a, c\}$. In other words, the rule is *not* to parse the

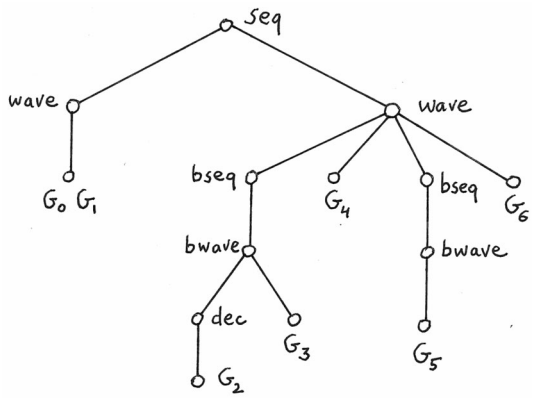

Fig. 2. The parse tree for the graph sequence $G_0 \cdots G_6$, where $G_0 = G_1 = G_4 = G_6 = a \leftarrow b \leftarrow c$, $G_3 = G_5 = a \rightarrow b \rightarrow c$, and $G_2 = a \quad b \rightarrow c$, with $B = \{a\}$.

current sequence $\mathcal{G}_{\{b\}\to\{a,c\}}$ as long as one can. The true cause of termination is the growth of W_t: both a and b are now wet. This shows that to parse a node requires knowing the relevant parameters of its ancestors.

3.4 Rewriting Rules

It is convenient to think of the flow tracker (the name for the entire parsing algorithm) as the application of certain productions of the sort found in context-free grammars: *seq*, *wave*, *bseq*, *bwave* are used as nonterminal symbols and *sg*, *dec* as terminals; here, *sg* is shorthand for "single-graph." Via the formulas (3, 5, 6, 7), the flow tracker yields the following productions (Fig. 2):

$$\begin{cases} seq & \Longrightarrow & (\, wave, wave, \,) \\ wave & \Longrightarrow & (\, bseq, sg, bseq, sg, \dots \,) \\ bseq & \Longrightarrow & (\, bwave, bwave, \,) \\ bwave & \Longrightarrow & (\, dec, sg, bseq, sg, bseq, sg, \dots \,). \end{cases} \tag{9}$$

How sensible is it to make $\mathcal{G}_{A\|B}$ a leaf of the parse tree? On the face of it, not very much. Indeed, suppose that $\mathcal{G}$ is a long, complicated sequence with a rich parse tree and V is its node set. Form a new sequence $\mathcal{G}'$ by adding an isolated node a (with no edge to or from V). Further, suppose that a is given the lowest label so that it defines the starting singleton B. The flow tracker will parse $\mathcal{G}'$ as the single-path tree:

$$T(\mathcal{G}') = (\,(\,(\mathcal{G}_{\{a\}\to V})^{bseq}\,)^{wave}\,)^{seq},$$

where $\mathcal{G}_{\{a\}\to V} = (\,(\,\mathcal{G}_{a\|V})^{dec}\,)^{bwave}$. Adding a single node hides the richness of $T(\mathcal{G})$ by producing a completely trivial parse tree. Of course, it is quite possible that picking another node for B besides a would produce a more interesting parse tree. But optimizing the selection of the starting singleton is out of the scope of our discussion. As for the parallel treatment of A and B in $\mathcal{G}_{A\|B}$, this is something for the renormalization of the dynamical system itself to handle. The reason we delegate this task is that A and B may not operate along the same time scale. Indeed, in a block-sequence of type $A \rightarrow B$, the set B, unlike A, can be handled separately over the entire length of the sequence. It follows from our previous discussion that the parse tree has depth at most roughly $2n$, as a typical downward path reads:

seq, *wave*, *bseq*, *bwave*, *bseq*, *bwave* $\cdots$ *bseq*, *bwave*, *dec*/*sg*.

The number of waves or block-waves that are the children of a given parent can be unbounded, however. In the case of diffusive influence systems, large degrees express structural properties of the dynamics and play a key role in the analysis. We conclude our discussion of graph-sequence parsing with a simple example of the flow tracker in action. Let $G_0 = G_1 = G_4 = G_6 = a \leftarrow b \leftarrow c$, $G_3 = G_5 = a \rightarrow b \rightarrow c$, and $G_2 = a \quad b \rightarrow c$. Setting $\mathcal{G} = G_0 \cdots G_6$ and choosing a to form the initial singleton B, we have

$$T(\mathcal{G}) = (\,(G_0 G_1)^{wave}\big(((G_2)^{dec}\ G_3)^{bwave}\,)^{bseq} \\ G_4\big((G_5)^{bwave}\big)^{bseq}\ G_6\,\big)^{wave}\,)^{seq}.$$

4 Algorithmic Renormalization

Let (f, X) be a system as in Definition 2.1. The *coding tree* $\mathcal{T}_f(X)$ of (f, X) captures both its geometry and its symbolic dynamics [6]. We defined it informally as a combinatorial object tracing the graph sequences of the orbits and we made a passing remark about the geometric information it encodes. We formalize these ideas now. The levels of the coding tree correspond to the times $t = 0, 1, 2$, etc. Combinatorially, its paths encode the entire symbolic dynamics of the system by listing in chronological order all the possible communication graph sequences formed by the orbits. In addition, nodes carry geometric information about the corresponding orbits. Each node v of the coding tree is associated with two cells U_v, V_v, where t_v is the depth of node v: the cell U_v is a continuity piece of f^{t_v}, i.e., a maximal connected subset of $X = [0,1]^n$ over which f^{t_v} is continuous; and $V_v = f^{t_v}(U_v)$ (which is not necessarily n-dimensional). The coding tree is defined inductively by starting with the root, $U_{\text{root}} = V_{\text{root}} = X$, and attaching a child w to a node v for each one of the cells c into which the discontinuities of f break V_v (note that there could be a single c). We then set $V_w = f(c)$ and $U_w = U_v \cap f^{-t_v}(c)$.[7] We define $V_t = \bigcup_{t_v = t} V_v$ and note that V_t includes all the points reachable in t steps. Any point reachable in $t+1$ steps can also be reached in t steps (just start at the second step); therefore,

$$V_{t+1} \subseteq V_t\,. \tag{10}$$

The *nesting time* $\nu = \nu(\mathcal{T}_f)$ is the smallest t such that V_t intersects no discontinuity. This number cannot be bounded if the system is chaotic; on the other hand, a finite nesting time implies asymptotic periodicity. To see why, observe that, for any v at depth ν, $f^k(f(U_v))$ does not intersect any discontinuity for any $k \le \nu$; therefore, $f(U_v) \subseteq U_w$ for some node w ($t_w = \nu$). It follows that the symbolic dynamics can be modeled by paths in the functional graph whose nodes are labeled by v ($t_v = \nu$) and (v, w) is an edge if $f(U_v) \subseteq U_w$.[8] Asymptotic periodicity follows from classic Markov chain theory: the powers of a stochastic matrix with a positive diagonal always converge to a fixed matrix. It appears therefore that the key question is to pin down the conditions under which the nesting time remains bounded. A node v is

7. Recall that f is the identity along the discontinuities so there is no need to define children for them. This causes a node to become a leaf if V_v falls entirely within one of the discontinuities, which can be handled separately and hence assumed away.

8. A graph is functional if exactly one edge points out of any node (possibly to itself). In any such graph, infinite paths end up in cycles.

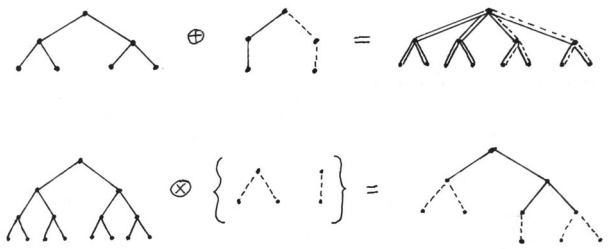

Fig. 3. The direct sum $\oplus$ and the direct product $\otimes$ in action.

called *shallow* if $t_v \leq v$: only those nodes can witness an intersection between their cell V_v and a discontinuity. Observe also that the period (plus preperiod) of any limit cycle is at most equal to the number of shallow nodes: the reason is that all eventually periodic orbits trace a path in the functional graph, whose number of nodes is itself bounded by the number of shallow nodes.

Being infinite, the coding tree has no "size" to speak of. It has a growth rate, however, which is captured by the *word-entropy* $h(\mathcal{T}_f)$: it is defined as the logarithm (to the base 2) of the number of shallow nodes. Since any node of the coding tree has at most $O(|\mathcal{D}|^n) = n^{O(n)}$ children,[9] the word-entropy is at most $O(v\, n \log n)$: crucially, it can be much smaller. Intuitively, attraction to a limit cycle hinges on keeping the word-entropy low relative to the dissipation rate of the system (the evolution rate of its contracting parts). In physical terms, this means that the loss of energy must outpace the increase in entropy. Although the system is deterministic, its behavior in the critical region (between periodic and chaotic) is essentially "random" and lends itself naturally to the language of statistical physics [7]: this is the concept of deterministic chaos familiar to dynamicists.

4.1 Parsing a Dynamical System

Every path of the coding tree corresponds to an infinite string $\mathcal{G} = G_0 G_1 G_2 \cdots$ of n-node directed graphs, where G_i is the graph associated with the continuity piece c_i. The set of paths is in bijection with the refinement of the iterated pullback, i.e., the language $\{c_0 c_1 c_2 \cdots \mid \bigcap_{t \geq 0} f^{-t}(c_t) \neq \emptyset\}$, which we explained how to parse in Section 3. In this way, every infinite path of the coding tree can be parsed according to $T(\mathcal{G})$. Parsing "renormalizes" the orbits of the system with respect to time, but this is not what we want. The objective is to renormalize the system itself, not individual orbits; for this, we need to "parse" the coding tree itself. We do this by combining together the individual parse trees of the infinite paths to form a single parse tree for $\mathcal{T}_f(X)$.

In the language of compiler theory, the parsing algorithm is of type LR, meaning that it can be executed by reading the string from left to right without backtracking. For this reason, the parse trees of paths with long common prefixes must share large subtrees. The productions in (9) specify exactly one rewriting rule for each left-hand side term, so all the parse trees are identical up to the number of terms inside the parentheses and the coupling times. This allows us to express the coding tree recursively via three tensor-like operations (Fig. 3):

(a) Direct sum. When the context is clear, we use A either to refer to a set of m agents or to denote the corresponding phase space $[0,1]^m$; same with the set B of $n-m$ agents. The *direct sum* $\mathcal{T}_f(A) \oplus \mathcal{T}_g(B)$ models the evolution in $A \times B$ of two decoupled systems (f, A) and (g, B). A path in the direct sum is of the form $(u_0, v_0), (u_1, v_1), \ldots$, where u_i and v_i $(i \geq 0)$ form paths in $\mathcal{T}_f(A)$ and $\mathcal{T}_g(B)$, respectively. A node w is formed by a pair (u, v) of nodes from each tree and $U_w = U_u \times U_v$, $V_w = V_u \times V_v$.

(b) Direct product. Let (f, X) and (g, X) be two systems over the same phase space $X = [0,1]^n$. We choose an arbitrary set of nodes in $\mathcal{T}_f(X)$ and prune the subtrees rooted at them. This creates new leaves in the coding tree, which we call *absorbing*.[10] Next, we attach $\mathcal{T}_g(V_v)$ to the absorbing leaves v. The reason we use V_v (defined here with respect to f) and not X as argument for $\mathcal{T}_g$ is to make the glueing seamless. The resulting tree is called the *direct product* $\mathcal{T}_f(X) \otimes \mathcal{T}_g(X)$. The operation is not commutative. In fact the two operands play very different roles: on the left, the tree $\mathcal{T}_f(X)$ gets pruned; on the right, a copy of the tree $\mathcal{T}_g(X)$ gets glued to every absorbing node, each copy cropped in a different way. Technically, we should denote the operation $\mathcal{T}_f(X) \otimes \{\mathcal{T}_g(X)\}$ because we may attach different coding trees at the absorbing leaves. We omit the parentheses to simplify the notation.

(c) Lift. A system is called *block-directional* of type $A \to B$ if no edges in any of the communication graphs point from B to A. Its coding tree is not a direct sum because, although B evolves independently, the agents of A are coupled with those of B. This one-way coupling is expressed by the *lift* $\mathcal{T}_f(A \nearrow B)$. The arrow highlights both the dependency of A on B and the fact that the coding tree of the whole system is a lift of $\mathcal{T}_f(B)$ from $\mathbb{R}^{|B|} \times \mathbb{N}$ into $\mathbb{R}^n \times \mathbb{N}$.

(d) Nesting time and word-entropy. The nesting time of a direct sum is at most the bigger of the two nesting times. The word-entropy of a direct sum/product is subadditive, even when it is infinite:

$$h(\mathcal{T}_f\{\oplus, \otimes\}\mathcal{T}_g) \leq h(\mathcal{T}_f) + h(\mathcal{T}_g). \tag{11}$$

4.2 The Renormalized Coding Tree

Translating (3, 5, 6, 7) in the tensor language of coding trees gives us the following rewriting rules. We omit the subscript f to avoid cluttering the notation

$$\left\{\begin{array}{lll} 1.\ \mathcal{T}(X) & \Longrightarrow \bigotimes_k \mathcal{T}(X)^{wave} \\ 2.\ \mathcal{T}(X)^{\text{wave}} & \Longrightarrow \\ & \bigotimes_{k<n} \left\{ \mathcal{T}(A_k \nearrow B_k)^{bseq} \otimes \mathcal{T}(G_{t_{k+1}})^{sg} \right\} \\ 3.\ \mathcal{T}(A \nearrow B)^{\text{bseq}} & \Longrightarrow \quad \bigotimes_k \mathcal{T}(A_k \nearrow B_k)^{bwave} \\ 4.\ \mathcal{T}(A \nearrow B)^{\text{bwave}} & \Longrightarrow \\ & (\mathcal{T}(A) \oplus \mathcal{T}(B))^{dec} \otimes \mathcal{T}(G_{t_1})^{sg} \otimes \\ & \bigotimes_{k<n} \left\{\mathcal{T}(A_k \nearrow B_k)^{bseq} \otimes \mathcal{T}(G_{t_{k+1}})^{sg}\right\}. \end{array}\right. \tag{12}$$

The notation borrows from the theory of programming languages, which makes it concise but nonstandard, so a few

9. Recall that the set $\mathcal{D}$ of discontinuities is assumed to be of polynomial size for simplicity.

10. The terminology "absorbing" is by analogy with the absorbing states of a Markov chain.

words of explanation are in order. Recall that water starts flowing from agent 1 to the nodes of G_0 that point to it, and then proceeds in this manner in G_1, G_2, etc. The propagation of the water determines how to break up the graph sequence into waves as shown in (3). In other words, any path of $\mathcal{T}(X)$ from the root forms a sequence of waves. Let us call "absorbing" the node corresponding to the end of the first wave along each such path. Pruning the subtrees rooted at the absorbing nodes leaves us with a coding tree which we denote by $\mathcal{T}(X)^{wave-1}$ as a reminder that it encodes all first waves. In this way, $\mathcal{T}(X) = \mathcal{T}(X)^{wave-1} \otimes \mathcal{T}(X)$. Repeating this process for each attached tree yields:

$$\mathcal{T}(X) = \bigotimes_{k=1}^{\ell} \mathcal{T}(X)^{wave-k}.$$

Note that ℓ is a variable and not a fixed parameter: it counts the number of waves along each path, meaning that ℓ can be finite as well as infinite. Dropping all the subscripts that can be inferred from the context leads us to 12.1, which we express as a rewriting rule. In rule 12.2, both A_k and B_k involve fewer than n agents, so k ranges from 0 to $n-2$ or less. The arguments X, A_k, B_k, etc, are used as a reminder of the sets of agents involved. In rule 12.4, the index k extends from 1 to less than $|A|$. The induction is sound because $|A_k| < |A| < n$. We write $k < n$ not to specify the precise range but to indicate whether there is an a priori bound on k or not.

The paths of a coding tree can be infinite but they can all be parsed by trees of linear depth. The rewriting rules (12) give us a quick-and-dirty way to bound the nesting time in terms of the number N of (pre-nesting) waves and block-waves. Let $\nu(n)$ and $\nu(m,n)$ be upper bounds on the nesting time for, respectively, a general n-agent system and an n-agent block-directional system of type $A \to B$ with $|A| \leq m$. We derive from (12) the recurrence

$$\begin{cases} \nu(n) \leq N(n(\nu(n-1,n)+1)) \\ \nu(n-1,n) \leq N(\nu(n-1) + n\nu(n-2,n)). \end{cases}$$

It follows that $\nu(n) \leq (Nn)^{O(n)}$. To bound N and the word-entropy requires a closer look at the branching structure of the coding trees.

5 Pseudorandomness and Dissipation

The discussion so far applies to any piecewise-linear network-based system. For that reason, our emphasis has been purely on *syntactic* considerations such as the flow of information across the networks. We now turn our attention to diffusive systems and exploit the averaging nature of the updates. In other words, we expand our investigation from the communication of information to its actual processing. The dynamics of a diffusive influence system features a clash between two "forces." One of them is *entropic* and a source of pseudorandomness: when a cell V_v bumps into a discontinuity, it is broken apart and its pieces are mapped to random-like locations. This is not always the case, however, and since chaos hinges on this entropic explosion, the process bears close examination. To counter this entropic effect, we have the dissipation of energy provided by the stochastic matrices. These linear maps are contractive along all the eigendirections except the principal ones (with eigenvalue 1). To appreciate how this complicates matters, an illustration might help.

Picture a balloon bouncing on a lined, corrugated surface. Imagine that the balloon has a tiny hole and deflates slowly at each bounce. The probability of bouncing right across a line will decrease over time. This illustrates the case of a cell V_v decreasing in volume as v goes down a path of the coding tree. Hitting a discontinuity (the balloon falling across a line) results in the splitting of V_v, which can often be described as a (pseudorandom) branching process. A high enough deflation rate might be able to overcome the splitting rate, with the production of new pieces slowing down over time and the branching process dying out: this is how limit cycles are produced. The difficulty is that the balloon does *not* contract along the principal direction(s); furthermore these directions can change and span spaces of varying dimension. Because of the non-commutativity of the matrices, the system does not have coherent eigenmodes. The true picture, therefore, is not that a round balloon deflating over time but, rather, of a balloon turning into a football, then into a sausage, etc. While our earlier intuition had no trouble with a shrinking balloon hitting a line with decreasing frequency, this new picture of footballs and sausages is more difficult to grasp. If you throw a sausage on the ground, its thickness has little effect on its probability of crossing a fixed line (think of Buffon's needle): only the length matters. But if this length does not decrease, then how can the branching process die out? To answer this question, which is at the heart of the renormalization process, we provide a dynamic classification of the agents into dominant and subdominant groups.

5.1 Dominance Structure

We begin with the simple case of a fixed communication graph G. The standard decomposition of the corresponding Markov chain partitions the nodes into essential and inessential classes. For completeness, we briefly review this process. The strongly connected components of G partition the node set into subsets, which, if contracted into single nodes, are seen to be joined together by edges so as to form an acyclic graph.[11] The sinks of this graph form the classes of *dominant* agents: no path in G can exit from a dominant class. The other agents are called *subdominant*.

For example, Fig. 4a features a nine-node graph G with three dominant classes: $\{1,2,4\}, \{5,7,9\}, \{6\}$. With a single graph, the system evolves as $P^t x$, where P is the stochastic matrix associated with G. Because of the positive diagonal, each dominant class, being strongly connected, converges to an attracting fixed point (more on which below) and the number of such classes represent the long-term rank of the system (i.e., its long-term "dimensionality"). The subdominant agents 3 and 8 are attracted to some convex combination of the dominant agents (Fig. 4b). The system is block-directional of type $subdominant \to dominant$, which in this running example is $\{3,8\} \to \{1,2,4,5,6,7,9\}$. In our discussion, the agents lie on the real line. In the figure, however,

11. The property that two nodes are joined by paths in both directions forms an equivalence relation. The strongly connected components of the graph are the equivalence classes.

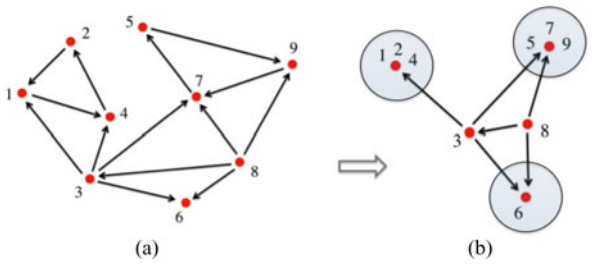

Fig. 4. A one-graph system evolving to a fixed-point attractor, with its three dominant classes highlighted.

we have placed them in the plane for visual convenience (the same ideas work in higher dimension, anyway).

Suppose now that the communication graph changes with time but that the dominance structure does not. Furthermore, suppose that none of the three dominant classes can communicate with one another. The dominant agents end up frozen in place and only the two subdominant agents can move. Asymptotic periodicity means that each of the agents 3 and 8 approaches a cyclic trajectory, not necessarily with the same period. Crucially, the subdominant agents can never leave the convex hull of the dominant ones once they are inside it (which will eventually happen). We wish to derive sufficient conditions under which, over an infinite time horizon, the dominance structure eventually settles, the dominant agents converge to fixed-point attractors, and the subdominant ones reach limit cycles.

5.2 Attraction Rate

We go back to the case of a fixed graph and show how to bound the convergence rate. In anticipation of our treatment of dynamic networks, however, we do it by using our water propagation mechanism, instead. Let $B_1, \ldots, B_r$ be the dominant classes, and let $n_i = |B_i|$ and $m = n - (n_1 + \cdots + n_r)$. The system is block-directional of type $A \to B$, where $B = B_1 \cup \cdots \cup B_r$ and $A = [n] \setminus B$. By abuse of terminology, the stochastic matrix P is of the form

$$P = \begin{pmatrix} A & C \\ \mathbf{0} & B \end{pmatrix}.$$

(a) Limit of A^t. Our next observation provides the key link between water propagation and dissipation: it is simple and crucial. Every time water propagates to new agents, something shrinks: in the case of B, it is the length of the smallest interval enclosing the wet agents; in the case of A, it is memory about itself. We explain. Initialize the agents of A at 1 and those of B at 0. Since G models a block-directional system of type $A \to B$, the agents of B will never be able to leave 0. We assume that $m > 0$. Because G is fixed, there is no flooding delay, so the *dec* and *bseq* sequences in (7) are empty, and formulas (6, 7) give us

$$\big(\underbrace{A \cdots A}_{\leq m}\big)^{bwave} \big(\underbrace{A \cdots A}_{\leq m}\big)^{bwave} \cdots$$

Consider a single block-wave and let λ_t be the length of the smallest interval enclosing the wet agents (which includes those in B) at time t, with $\lambda_0 = 0$. We denote by R the complement of this interval in $[0, 1]$. Note that a dry agent becomes wet as soon as it leaves 1. For λ_t to increase, a wet agent must move into R: it could be one currently wet that slides to the right into R or a dry agent at 1 moving left into R and becoming wet. In both cases, the agent moves to the mass center of its neighbors (including itself), so R can shrink by at most a factor of n; in other words

$$1 - \lambda_{t+1} \geq \tfrac{1}{n}\big(1 - \lambda_t\big).$$

Since water propagates at each step, it follows that $\lambda_t \leq 1 - n^{-n}$ at the end of the first block-wave. In other words, the smallest interval enclosing wet agents can grow but only up to length $1 - n^{-n}$. When the block-wave terminates, the agents are then wet, which means that all n agents fit strictly within $[0, 1]$. In other words, each block-wave shrinks the smallest interval enclosing the whole system by at least a factor of $1 - n^{-n}$; hence $\lambda_t \leq 2^{-\gamma t}$, for some fixed $\gamma \geq \Omega(n^{-n})$. Since the placement of the agents of A after t steps is given by the coordinates of $A^t\mathbf{1}$, it follows that

$$\|A^t\mathbf{1}\|_\infty \leq 2^{-\gamma t}. \tag{13}$$

The previous argument relies on the absence of any flooding delay. Indeed, a delay of θ steps might allow R to shrink by a factor exponential in θ (if wet agents point to dry ones but not the other way around), which can lead to the sort of crawling behavior in evidence in the *Very Slow Clock* of Section 2.5.

(b) Limit of B^t. To bound the convergence rate of B^t proceeds along similar lines. The matrix B is the block diagonal matrix $(B_1, \ldots, B_r)$. Each B_i can be treated separately, so we might as well assume that $r = 1$. Since B is strongly connected, formulas (3, 5) simplify into[12]

$$\big(\underbrace{B \cdots B}_{<n-m}\big)^{wave} \big(\underbrace{B \cdots B}_{<n-m}\big)^{wave} \cdots$$

Initialize all the agents B at 0, except for the kth one, which is placed at 1. As before, we define λ_t as the length of the smallest interval enclosing the wet agents. Removing this interval from $[0, 1]$ leaves us with two intervals L, R. The previous argument shows that neither L nor R can shrink by more than a factor of n in a single step. Since B is strongly connected, water propagates at each time step, so $\lambda_t \leq 1 - n^{-n}$ at the end of the first wave ($t < n$). The placement of the B-agents is given by the kth column of B^t. It follows that, for some fixed probability distribution vector $z \in \mathbb{R}^n$,

$$\big\| B^t - \mathbf{1}z^T \big\|_{\max} \leq 2^{-\gamma t}. \tag{14}$$

(c) Limit of P^t. With the estimates on the powers of A and B given by (13, 14), it is now routine to bound

$$P^t = \begin{pmatrix} A^t & C_t \\ \mathbf{0} & B^t \end{pmatrix},$$

where $C_1 = C$ and $C_{t+1} = AC_t + CB^t$. By (13), the matrix $I - A$ is nonsingular, hence the elementary identity

12. By abuse of notation, we write B for its induced subgraph.

$$C_t = (I - A^t)(I - A)^{-1} C \mathbf{1} z^T + \sum_{k=0}^{t-1} A^{t-k-1} C (B^k - \mathbf{1} z^T).$$

By (13, 14), any matrix entry in the kth summand is bounded by $2^{-\gamma(t-k-1)-\gamma k}$ in absolute value. The max-norm of the matrix sum itself is therefore at most $t2^{-\gamma(t-1)}$ entry-wise. The entries of $(I - A)^{-1}$ are at most $\sum_t 2^{-\gamma t} = O(n^n)$. By rescaling γ, we reach the following conclusion: there exists $\gamma \geq \Omega(n^{-n})$ such that, for any $t > 1/\gamma$, $\| P^t - \mathbf{P} \|_{\max} \leq 2^{-\gamma t}$, where

$$\mathbf{P} = \begin{pmatrix} \mathbf{0} & (I - A)^{-1} C \mathbf{1} z^T \\ \mathbf{0} & \mathbf{1} z^T \end{pmatrix}. \tag{15}$$

If the number r of dominant classes is larger than 1, then $\mathbf{1}z^T$ needs to be replaced by a rank-r stochastic matrix of the form $\mathrm{diag}(\mathbf{1}\mathrm{z}_1^{\mathrm{T}}, \ldots, \mathbf{1}\mathrm{z}_{\mathrm{r}}^{\mathrm{T}})$, where z_i is a probability distribution vector in $\mathbb{R}^{|B_i|}$. The limit cycles of directional systems of type $A \to B$ are described by (time-invariant) matrices such as P, with one difference: all the agents in A are subdominant but not all of B needs to be dominant. As long as the powers of B converge at the rate given above, however, P^t will also converge accordingly.

6 The Analysis

Recall that our diffusive influence systems are assumed to be locally coupled (see Section 2.3). As in [6], we assume a *snap rule*: the status (in or out) of (i, j) as an edge of $G(x)$ is constant over the slab $|x_i - x_j| < \varepsilon_0$; in other words, the edge cannot vanish and reappear incessantly as the agents i and j get infinitesimally close to each other.[13] The idea of renormalization is to prove that a certain dynamical behavior is recursively preserved as we move up the hierarchy of subsystems. In this case we prove inductively that each one of the systems on the right-hand side of the rules in (12) satisfies the following properties:

- There exists a region E, called the *exclusion zone*, such that the coding tree $\mathcal{T}(X \backslash E)$ has a nesting time and a word-entropy bounded by v and h respectively, where $X = [0, 1]^n$. The exclusion zone is the union of a set of *δ-slabs*, which are regions of $\mathbb{R}^n$ of the form $\{x : |u^T x - 1| \leq \delta\}$, for $\|u\|_2 = n^{O(1)}$ and δ at most a small positive constant. (These bounds can be adjusted liberally.) Only shallow nodes contribute δ-slabs and at most $|\mathcal{D}|$ each. Given a node v of the coding tree, let $P_{\leq v}$ denote the stochastic matrix encoding the linear restriction of f^{t_v} to U_v. The node is said to contribute to the exclusion zone if the latter includes one (or several) δ-slabs of the form $\{x : |u^T P_{\leq v} x - 1| \leq \delta\}$, where $u^T x = 1$ is a discontinuity of $\mathcal{D}$. Because $\|P_{\leq v}^T u\|_2 = n^{O(1)}$, the polynomiality condition on the coefficients of δ-slabs is preserved under any of the tensor operations, since these only require updating $P_{\leq v}$. The total number of δ-slabs in the exclusion zone is bounded by $|\mathcal{D}|2^{h(\mathcal{T})}$. For a δ-slab to intersect the unit cube X, the vector $P_{\leq v}^T u$ cannot be too short: indeed, the width is $2\delta / \|P_{\leq v}^T u\|_2 = O(\delta\sqrt{n})$, so the slab spans a volume of $O(\delta n^n)$ within X; hence
$$\mathrm{Vol}\,(X \backslash E) \geq 1 - n^{O(n)} \delta 2^{h(\mathcal{T})}. \tag{16}$$
- Every orbit starting in $X \backslash E$ (and not hitting an absorbing node) is a limit cycle (possibly of period 1). There is a single infinite path descending from any nonshallow node v. If p (resp. q) is its corresponding period (preperiod), then $p + q \leq 2^h$ and $P_{\leq v} = P_{\leq w} P^{(t_v - t_w)/p}$, where w is an ancestor of v of depth between q and $p + q$ (assuming v deep enough); there are p matrices P associated with the given path, each one the product of p of the original stochastic matrices associated with the linear restrictions of f. We assume a uniform lower bound γ for the values of $\gamma = \gamma(P)$ in (15) over all subsystems of any type with a given number of agents.

We analyze the effect of the four rules (12) on the coding tree. By convention, v, h (resp. v', h') denote the parameters for the right-hand (resp. left-hand) side of the rule under consideration.

6.1 Sequence to Block-Sequence

Rule 12.1 rewrites $\mathcal{T}(X)^{seq}$ as the iterated direct product $\bigotimes_k \mathcal{T}(X)^{wave}$. We show that, upon completion of the sth wave (should it exist), all the agents are covered by an interval of length $2^{-s/n^{O(1)}}$. The argument is a variant of the one we used in Section 5.1 to prove (14). The number of direct products in rule 12.2 is less than n and every single-graph $G_{t_{k+1}}$ signifies the propagation of water to dry agents. Let λ_k be the length of the smallest interval enclosing A_k at time t_k, i.e., at the formation of the block-directional sytem of type $A_k \to B_k$. One difference with Section 5.1 is that, by the time A_k gives way to A_{k+1} at time t_{k+1}, its enclosing interval might have grown to cover almost all of $[0, 1]$. This might happen if B_k has agents at 0 and 1, for example, and A_k has edges pointing to them.[14] Obviously, the worst case features all the agents B_k located at 0 or 1. Neither L nor R can shrink by more than a factor of n in a single step, so the length of neither one can fall below ε_0/n prior to t_{k+1} (note that it can be smaller than ε_0/n to begin with: it just cannot become so). It follows that

$$1 - \lambda_{k+1} \geq \tfrac{1}{n} \min\{1 - \lambda_k, \varepsilon_0\} \geq \varepsilon_0 n^{-n} \geq n^{-O(n)},$$

which proves that all the agents are covered by an interval of length $2^{-s/n^{O(n)}}$ after s waves. Once all the agents lie within an interval of length ε_0, the snap rule freezes the communication graph and, by (15), the system is attracted to a fixed point at a rate of $2^{-t/n^{O(n)}}$. The system is then

13. We can choose ε_0 to be arbitrarily small but, for convenience, we set $\varepsilon_0 = 2^{-O(n)}$. The snap rule is needed for our main result: indeed, without it, the systems can be chaotic even under perturbation [6]. In fact, without it, any piecewise-linear system can be simulated by a diffusive influence system: in other words, the matrices need not even be stochastic and averaging becomes essentially irrelevant. We note that the snap rule is automatically implied by *metrical systems*, where discontinuities are of the form $x_i - x_j = u$.

14. The *Very Slow Clock* builds on this idea.

essentially of rank 1 but, of course, the high likelihood of incomplete waves can increase the rank (and the period) by creating several dominant classes. By repeated applications of subadditivity (11), $h' \leq (\#\texttt{waves})\mathrm{h}$; hence, by $\varepsilon_0 \geq 2^{-O(n)}$, $h' \leq n^{O(n)}h$. Rule 12.2 expresses $\mathcal{T}(X)^{wave}$ via fewer than n direct products whose factors are themselves products with a single-graph coding tree, so using primes to denote the trees formed by application of the two rules 12.1 and 12.2, we get

$$\nu' \leq n^{O(n)}\nu \quad \text{and} \quad h' \leq n^{O(n)}h\,. \tag{17}$$

(a) Bidirectional systems. Before we move on to the analysis of the last two rules, it is helpful to build some intuition by resolving the *bidirectional* case. This is the version of the model where the communication graph $G(x)$ is undirected: every edge (i,j) comes with (j,i). Such systems are known to converge [10], [12], [14]. We show how the renormalization framework leads to a bound on the relaxation time. (We are not aware of any other method for achieving this result.) The parsing rules in (12) reduce to these two:

$$\begin{cases} \mathcal{T}(X) \implies \bigotimes_k \mathcal{T}(X)^{wave} \\ \mathcal{T}(X)^{wave} \Longrightarrow \bigotimes_{k<n} \Big\{ \big(\mathcal{T}(A_k) \oplus \mathcal{T}(B_k)\big) \otimes \mathcal{T}(G_{t_{k+1}})^{sg} \Big\}. \end{cases}$$

Let ν, h denote upper bounds on the nesting time and word-entropy of A_k and B_k. We can show inductively that the period is 1 (fixed-point attraction) so that, along any given path of the coding tree of the direct sum, a node v of depth $t_v > \nu$ is such that $P_{\leq v} = P_{\leq w}\, P^{t_v - \nu}$ for some node w of depth ν. The matrix P is of the form

$$P = \begin{pmatrix} A_w & \mathbf{0} \\ \mathbf{0} & B_w \end{pmatrix},$$

with both of A_w and B_w playing the role of B in (15).[15] This implies the existence of an idempotent matrix $\mathbf{P}$ such that $\|P^l - \mathbf{P}\|_{\max} \leq 2^{-\gamma l}$, for $\gamma \geq 1/n^{O(n)}$ and any $l \geq 0$. (Better bounds can be found but they are not needed here.) How much higher than ν the nesting time of the direct sum can be depends on how deep a node v can be such that V_v intersects a discontinuity $u^T x = 1$ involving agents from both A_k and B_k. This occurrence implies that $u^T P_{\leq v}\, x = 1$ for some $x \in U_v$; hence,

$$|u^T P_{\leq w}\, \mathbf{P} x - 1| \leq \|u\|_1 2^{-\gamma(t_v - \nu)} \leq 2^{-\gamma(t_v-\nu)+O(\log n)}.$$

To make this into a δ-slab, we set $t_v \geq \nu + |\log \delta\,|n^{bn}$ for constant b large enough. The slab does not depend on v but on its path, so adding it to the exclusion zone guarantees the absence of absorptions deeper than t_v. Accounting for all the direct sums sets an upper bound of $n\nu + n^{O(n)}|\log\delta\,|$ on the nesting time of $\mathcal{T}(X)^{wave}$. Using subscripts to indicate the number of agents, by (17),

$$\nu(\mathcal{T}_n) \leq n^{O(n)}(\nu(\mathcal{T}_{n-1}) + |\log\delta\,|) \leq n^{O(n^2)}|\log\delta|. \tag{18}$$

15. Both A_w and B_w can have several dominant classes.

By subadditivity (11), a conservative upper bound on the word-entropy of $\mathcal{T}(X)^{wave}$ is $n(2h(\mathcal{T}_{n-1}) + \log\nu(\mathcal{T}_n))$; hence

$$h(\mathcal{T}_n) \leq n^{O(n)}\big(h(\mathcal{T}_{n-1}) + \log|\log\delta\,|\big) \leq n^{O(n^2)}\log|\log\delta\,|.$$

By (16),

$$\mathrm{Vol}\,(\mathrm{X}\backslash\mathrm{E}) \geq 1 - \mathrm{n}^{\mathrm{O(n)}}\delta 2^{\mathrm{h}(\mathcal{T}_\mathrm{n})} \geq 1 - \delta|\log\delta\,|^{\mathrm{n}^{\mathrm{O(n^2)}}} > 1 - \sqrt{\delta}\,,$$

for $\delta > 0$ small enough. This proves that the tiniest random perturbation of the starting configuration—obtained by, say, shifting each agent randomly left or right by a constant, but arbitrarily small amount—will take the system to a fixed-point attractor with probability close to 1. By (18) and the convergence rate of single-graph systems, the system is at a distance ε away from its attractor after a number of steps equal to $n^{O(n^2)}|\log\delta\varepsilon\,|$. The dependency on δ (and, of course, ε) cannot be avoided. Indeed, there is no uniform bound on the convergence rate over the entire phase space $[0,1]^n$. The following result requires the snap rule stated at the beginning of this section but holds even in the absence of local coupling.

Theorem 6.1. *With probability arbitrarily close to one, a perturbed bidirectional diffusive influence system is attracted to a fixed point. Specifically, with probability at least* $1-\delta$*, the system is at distance* ε *of the fixed point after* $c_n|\log\delta\varepsilon\,|$ *steps, where* c_n *depends on the number* n *of agents.*[16]

(b) Entropy versus energy. We return to the case of general diffusive influence systems with no assumption of bidirectionality. The nesting time ν tells us how deep we have to go down the coding tree for the dynamics to stabilize. The average degree μ of a shallow node can be defined by $2^h = \mu^\nu$, so that, as δ goes to 0, one would expect of a periodic system $(f, X\backslash E)$ that $\mu = 2^{h/\nu}$ should tend to 1. The average degree measures the tension between the entropic forces captured by h and the energy dissipation expressed by ν via the water propagation. The coding tree can branch at a maximum rate of roughly $|\mathcal{D}|^n$ per node. For the system to be attracting, the rate must be asymptotically equal to 1. It was fairly easy to achieve this without heuristic assumptions in the bidirectional case. We have shown that it is possible in the general case [6] provided that we assume a certain timing mechanism to prevent the re-entry of long-vanished edges. In the absence of such conditions, the critical region in parameter space between chaos attraction remains mysterious. In the next section, we sketch minimal heuristic assumptions to ensure asymptotic periodicity.

6.2 Block-Sequence to Block-Sequence

Rule 12.3 rewrites $\mathcal{T}(A \nearrow B)^{bseq}$ as $\bigotimes_k \mathcal{T}(A_k \nearrow B_k)^{bwave}$. This is our main focus in this section, with a brief mention of rule 12.4 at the end. Fix a path in the coding tree and let $v = v(s)$ the first node (if at all) after the first s block-waves. By definition,

16. We showed in [6] how to improve the time bound via the s-energy [5].

$$P_{\leq v} = \begin{pmatrix} A_s & C_s \\ \mathbf{0} & B_s \end{pmatrix}.$$

Repeating the argument from Section 5.1 used for the limit of A^t, we find that

$$\|A_s \mathbf{1}\|_\infty \leq 2^{-\gamma' s}, \tag{19}$$

for some $\gamma' > 0$. Note that we need not assume that B consists only of dominant agents. We used a fairly technical argument to bound γ' in [6] under some mechanism to control the reappearance of long-absent edges. We pursue a simpler, more general approach here.

(a) The intuition. Recall that the word-entropy measures how likely a typical shallow node v sees its cell V_v intersect a discontinuity and split accordingly. The best way to bound the growth of the word-entropy, therefore, is to show that the cells V_v shrink and hence split with diminishing frequency as t_v grows. The system is not globally contractive, however, and the diameter of V_v, far from shrinking, might actually grow. Indeed, consider the two-agent system with the single graph $a \to b$: the iterates of the cell $[0, 0.1] \times [0, 1]$ converge to the segment $[(0, 0), (1, 1)]$, so that the area vanishes while the diameter grows by roughly a factor of $\sqrt{2}$. In this example, the cell thins out in the horizontal direction but stays unchanged along the vertical (i.e., the dominant) one. The solution is first to factor out the dominant agents and then restore their dynamics in a neighborhood of the periodic points via coarse-graining. This can also be interpreted as a foliation of the system.

For a mechanical analogy, think of the B-agents as forming the frame of a box-spring. First, we consider a fixed frame and study the vibrations of the springs subject to an impulse:[17] the network of springs (the A-agents) may see its topology change over time but the frame itself remains rigid. In a second stage, we allow the frame to be deformed under its own internal forces. The dynamics of the frame itself is decoupled from the springs (just as B is decoupled from A in a block-directional sytem of type $A \to B$). This sort of quotient operation is precisely what algorithmic renormalization aims for. We flesh out this intuition below.

(b) Freezing the B-agents. We begin with the case of a fixed "box-spring frame." The phase space becomes $[0, 1]^m \times \{x_B\}$, where x_B is now viewed as a parameter in $[0, 1]^{n-m}$. Let $\mathcal{T}_{\leq s}$ denote the coding tree of the first s block-waves in rule 12.3 and let v be a node deep enough that at least s block-waves occur before time t_v. Because of the approximation on A_s in (19), the projection of V_v onto $[0, 1]^m$ is contained in an m-dimensional cube of side-length at most $2^{-\gamma' s}$. If V_v intersects a discontinuity $u_A^T x_A + u_B^T x_B = 1$, it then follows that

$$\left| \left(u_A^T C_s + u_B^T B_s\right) x_B - 1 \right| \leq 2^{-\gamma' s + O(\log n)}, \tag{20}$$

for some $x = (x_A, x_B)$. Fix an arbitrarily large threshold s_0 and observe that $\sigma_0 \triangleq 2^{-\gamma' s_0}$ is an upper bound on the side-length of the cube enclosing V_v for any v deeper than the s_0th block-wave. We model the children of v as the outgrowth of a branching process whose reproduction rate (the average node degree) is at most $n^2 \sigma_0 |\mathcal{D}|$.

Here is a quick heuristic justification. We begin with our earlier observation (10) that the union V_t of the cells V_v for a given depth $t_v = t$ forms a nested sequence as t grows. In the absence of any process biasing the orbits towards the discontinuities, a random point from V_t should not be significantly closer to a discontinuity than if it were random within X itself. Thus, if a typical cell $f(V_v)$ ends up being thrown randomly within V_t, one would expect it to intersect a discontinuity with probability that depends on the size of its enclosing cube. We show how to derive the reproduction rate when V_t is roughly X. (The argument is scalable so it can be extended to the case where V_t is much smaller than X.) If a point is at distance $\sqrt{n}\,\sigma_0$ from a hyperplane, it is possible to move it to the other side by changing a suitable coordinate by at most $n\sigma_0$ (easy proof omitted). The estimate of $n^2 \sigma_0 |\mathcal{D}|$ follows from a union bound on the n coordinates and the discontinuities. This heuristic validation does not hold in the chaotic construction given in [6], which is precisely why we need the snap rule. As was shown there, when the B-agents are frozen, however, the reproduction rate can be provably bounded.

The coding tree $\mathcal{T}_{\leq s}$ is renormalized by rule 12.3 as a tree of block-wave trees: the latter's absorbing nodes are at most ν away from the root. Past the s_0th block-wave, the probability that a given node v has its cell V_v is split by a discontinuity is at most $n^2 \sigma_0 |\mathcal{D}|$. This creates a reproduction rate of $\mu_v = 1 + \sigma_0 n^{O(n)}$ for a given node v and $\mu \leq \mu_v^\nu$ for an entire whole block-wave tree. Assuming that σ_0 is sufficiently smaller than $1/\nu n^{O(n)}$,

$$\mu \leq 1 + \sigma_0 \nu n^{O(n)}. \tag{21}$$

We enforce nesting after s block-waves by adding to the exclusion zone a number of δ-slabs no greater than $|\mathcal{D}| 2^{h(\mathcal{T}_{\leq s})}$: for this, we need to ensure that $s \geq s_0 + (|\log \delta| + b \log n)/\gamma'$, for constant b large enough, so that δ dominates the right-hand side in (20). Note that x_B is considered a parameter here, so the slabs do not split the cells V_v per se: they simply exclude certain positions of x_B, i.e., certain configurations of the fixed box-spring frame. By subadditivity,

$$h(\mathcal{T}_{\leq s}) \leq (s_0 + 1)h + (s - s_0) \log \mu.$$

We artificially added 1 to s_0 to account for the fact that each of the coding trees for $\mathcal{T}(A \nearrow B)$ between block-waves s_0 and s needs its own exclusion zone: this slight overestimate of the word-entropy has the benefit of keeping $|\mathcal{D}| 2^{h(\mathcal{T}_{\leq s})}$ as a valid upper bound on the number of slabs needed for the exclusion zone. Using primes to refer to the left-hand side of rule 12.3, the word-entropy can be bounded as follows:

$$h' \leq (s_0 + 1)h + \frac{1}{\gamma'} \left(|\log \delta| + O(\log n)\right) \log \mu\,. \tag{22}$$

(c) Coarse-graining. We turn to the case of the dynamic "box-spring frame." The previous analysis was premised on the assumption that the B-agents were frozen once and for all. Treating them as variables in $\mathbb{R}^{n-m}$ may violate the

17. For the analogy to be accurate, one must think of the springs as being one-way—in flagrant violation of Newtonian mechanics...

assumption that deep nodes rarely witness branching. By Crofton's Lemma (Buffon's needle), a random positioning of a cell V_v will hit a discontinuity with high probability if the diameter of the cell is large. All we can argue is that the volume decreases as the depth of v grows, but it is the diameter that matters, not the volume! The solution is to coarse-grain the phase space for B by subdividing it into tiny cubes and then treating each cube as a single point.

We subdivide $[0,1]^{n-m}$ into cubes of side-length σ_0 and restrict x_B to one of them, denoted c_B. Consider a path of $\mathcal{T}(B)$ and let p denote its period (taking multiples if necessary to ensure that $\nu + p$ is at least the preperiod). We denote by $B_{\le v}$ the stochastic matrix encoding the linear restriction of f^{t_v} to the space of B-agents. If $t_v > \nu + p$ then, by our induction hypothesis (since $|B| < n$),

$$B_{\le v} = B_{\le w} B_w^{(t_v - t_w)/p},$$

where $t_w \le \nu + p$ and B_w is one of p matrices associated with the periodic orbit; furthermore, there exists an idempotent matrix $\mathbf{B}_w$ such that $\|B_w^l - \mathbf{B}_w\| \max \le 2^{-\gamma l}$. Of course, this still holds if we switch our point of view and consider a node v of $\mathcal{T}(X_{|c_B})$ of depth $t_v > \nu + p$, where the notation $X_{|c_B}$ indicates that the phase space is still X but $U_{\text{root}} = [0,1]^m \times c_B$. If v is the first node after s block-waves then, by (19), for any $x \in U_v$,

$$\left\| f^{t_v}(x) - \begin{pmatrix} C_s \\ B_{\le w}\mathbf{B}_w \end{pmatrix} x_B \right\|_\infty \le 2^{-\gamma' s} + n 2^{-(t_v - \nu - p)\gamma/p}. \quad (23)$$

By our assumption that x_B can vary by at most σ_0 in each coordinate and $x = (x_A, x_B)$, the cell V_v is enclosed within a cube of side-length σ_1, where

$$\sigma_1 \le \sigma_0 + 2^{1-\gamma' s} + n 2^{1-(t_v - \nu - p)\gamma/p}. \quad (24)$$

We update (21) to estimate the new reproduction rate on the assumption that $s > s_0$ and σ_1 is sufficiently smaller than $1/\nu n^{O(n)}$:

$$\mu \le 1 + \sigma_1 \nu n^{O(n)}. \quad (25)$$

If V_v intersects the discontinuity $u_A^T x_A + u_B^T x_B = 1$, then, by (23) and $\|u\|_2 = n^{O(1)}$,

$$\begin{aligned} &\left| \left(u_A^T C_s + u_B^T B_{\le w} \mathbf{B}_w\right) x_B - 1 \right| \\ &\quad \le 2^{-\gamma' s + O(\log n)} + 2^{-(t_v - \nu - p)\gamma/p + O(\log n)} \le \delta, \end{aligned} \quad (26)$$

with the last inequality ensuring that the constraints fit within δ-slabs. Observe that the characteristic timescale is $1/\gamma'$ (measured in block-waves) for $\mathcal{T}(A \nearrow B)$ and p/γ for $\mathcal{T}(B)$. After a suitably large number s of block-waves ($s > s_0$), we add to the exclusion zone the relevant δ-slabs for each node at the depth corresponding to the end of the sth block-wave.

To see why this causes nesting, we examine the coding tree for B first. The added slabs are cylinders, with their bases in c_B, which carve the cells U_v into subcells that are "essentially" invariant for all times in the relevant residue class modulo the corresponding period. The qualifier refers to the fact that the orbit converges toward a fixed point at a rate of $2^{-\gamma}$ per cycle. For t large enough so that the second exponential term in (26) is sufficiently smaller than δ, the orbits of the B-agents might still hit the slabs but not cross their mid-hyperplane. From that point on, we can thus factor out the B-agents by pretending that they are fixed and, from there, infer nesting. Adding the contribution to the word-entropy of the grid decomposition of the space of B-agents, we update (22) as

$$h' \le (n-m)|\log \sigma_0| + (s_0 + 1)h + (s - s_0)\log \mu.$$

We set $s = s_0 + \frac{b}{\gamma'} \log \frac{n}{\delta}$ and $t_v = \nu + \frac{bp}{\gamma} \log \frac{n}{\delta}$, for a constant b large enough (reused generically to alleviate the notation). These assignments satisfy (26). We will always choose δ smaller than σ_0. By (24) and the definition of σ_0 as $2^{-\gamma' s_0}$, this implies that $\sigma_1 \le 2^{2-\gamma' s_0}$ and, by (25), $\log \mu \le 2^{-\gamma' s_0} \nu n^{O(n)}$. This upper bound is much less than 1 if we set

$$s_0 = \frac{1}{\gamma'} \log\Big(\frac{n^{bn} \nu}{\gamma'} |\log \delta| \Big).$$

It follows that $\nu' \le s\nu \le (s_0 + \frac{b}{\gamma'} \log \frac{n}{\delta})\nu$ and

$$h' \le (s_0 + 1)(h + n\gamma') + 1.$$

By subadditivity, rule 12.4 adds factors of at most n to these bounds. Using primes to refer to the parameters of $\mathcal{T}(A \nearrow B)^{bseq}$ and unprimed notation to refer to any of the trees in the right-hand side of rule 12.4, we use the inequalities $\delta < \sigma_0$ and $\nu \le 2^h$ to derive (conservatively):

$$\begin{cases} \nu' \le O(n/\gamma')^3 |\log \delta| \nu \\ h' \le O(n/\gamma')^3 (\log|\log \delta|) h^2. \end{cases} \quad (27)$$

6.3 Putting It All Together

We are now in a position to bound the volume of the exclusion zone E. We denote by h_n the maximum word-entropy of $\mathcal{T}(X \backslash E)$ for any n-agent system. We reserve the notation $h_{m,n}$ for the biggest of h_m, h_{n-m}, and the maximum word-entropy of any n-agent bidirectional system of type $A \to B$ with $|A| \le m$. By (17, 27), $h_n \le n^{O(n)} h_{n-1,n}$, and, for $0 < m < n$,

$$h_{m,n} \le \max\Big\{ h_m, h_{n-m}, O(n/\gamma')^3 (\log|\log \delta|) h_{m-1,n}^2 \Big\}.$$

It follows that $h_n \le \left(\gamma'^{-1} \log|\log \delta|\right)^{2^{O(n)}}$. Because δ appears as a double (and not single) logarithm in the upper bound, by (16),

$$\text{Vol}\,(X \backslash E) \ge 1 - n^{O(n)} \delta 2^{h_n} < 1 - \sqrt{\delta},$$

for δ small enough. This allows us to claim asymptotic periodicity under perturbation provided that we have local coupling, the snap rule, and the branching process with bounded reproduction rate discussed in Section 6.2.

Theorem 6.2 *Under the heuristic assumptions just mentioned, with probability arbitrarily close to one, perturbing the initial state of a diffusive influence system produces an orbit that is attracted to a limit cycle.*

This result makes no bidirectionality assumption. It comes with strings attached to it, however: notably pseudo-random discontinuity splitting and uniform bounds on the convergence rates. Our intuition that such heuristic assumptions are valid is backed by abundant empirical evidence. That said, one should not underestimate the challenge of justifying them mathematically. We were able to do it in [6] with a single enforceable (i.e., non-heuristic) assumption in the model and this already required quite a bit of technical work. To remove all assumptions seems a formidable endeavor. Unlike the study of traditional algorithms, the investigation of such *natural algorithms* (i.e., dynamical systems with algorithmic descriptions) can be extremely difficult even for very small input sizes: indeed, fewer than a dozen agents are sufficient to create dynamics so complex it eludes the current mathematical state of the art.

An avenue of research that seems more accessible is to develop a notion of *fractional* wetness. In our framework, an agent node is dry or wet but nothing inbetween: its status is a binary predicate that indicates whether information has been received but not how much. Since the dissipation rate depends on the quantity of information transmitted, it might be useful to delay the completion of the waves until the wetness of each agent has reached a certain threshold. How to develop a "belief propagation" method of message passing for parsing the graph sequences using a variant of the flow tracker is a promising line of attack that warrants close scrutiny.

Acknowledgments

The author wish to thank the anonymous referees for many helpful comments and suggestions. Bernard Chazelle is the corresponding author. This work was supported in part by the US National Science Foundation (NSF) grants CCF-0963825 and CCF-1420112. Part of it was done when the author was an Addie and Harold Broitman member at the Institute for Advanced Study, Princeton, NJ.

References

[1] A. Bhattacharyya, M. Braverman, B. Chazelle, and H. L. Nguyen, "On the convergence of the Hegselmann-Krause system," in *Proc. 4th Conf. Innovations Theoretical Comput. Sci.*, 2013, pp. 61–66.

[2] F. Bullo, J. Cortés, and S. Martinez, *Distributed Control of Robotic Networks*, Applied Mathematics Series. Princeton, NJ, USA: Princeton Univ. Press, 2009.

[3] A. Buluc, H. Meyerhenke, I. Safro, P. Sanders, and C. Schulz, "Recent advances in graph partitioning," Preprint arXiv:1311.3144, 2015.

[4] C. Castellano, S. Fortunato, and V. Loreto, "Statistical physics of social dynamics," *Rev. Mod. Phys.* vol. 81, pp. 591–646, 2009.

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

[6] B. Chazelle, "The dynamics of influence systems," in *Proc. 53rd IEEE Annu. Found. Comput. Sci.*, 2012, pp. 311–320. (To appear in J. SIAM Comput.)

[7] B. Chazelle, "An algorithmic approach to collective behavior," *J. Statist. Phys.* vol. 158, 2015, pp. 514–548.

[8] S. Fortunato, "Community detection in graphs," *Phys. Rep.*, vol. 486, pp. 75–174, 2010.

[9] R. Hegselmann and U. Krause, "Opinion dynamics and bounded confidence models, analysis, and simulation," *J. Artif. Soc. Soc. Simulation*, vol. 5, no. 3, pp. 1–24, 2002.

[10] J. M. Hendrickx and V. D. Blondel, "Convergence of different linear and non-linear Vicsek models," in *Proc. 17th Int. Symp. Math. Theory Netw. Syst.*, Kyoto, Japan, Jul. 2006, pp. 1229–1240.

[11] D. Kempe, J. M. Kleinberg, and A. Kumar, "Connectivity and inference problems for temporal networks," *J. Comput. Syst. Sci.*, vol. 64, pp. 820–842, 2002.

[12] J. Lorenz, "A stabilization theorem for dynamics of continuous opinions," *Physica A: Statist. Mech. Appl.*, vol. 355, pp. 217–223, 2005.

[13] S. Martínez, F. Bullo, J. Cortés, and E. Frazzoli, "On synchronous robotic networks Part ii: Time complexity of rendezvous and deployment algorithms," *IEEE Trans. Autom. Control*, vol. 52, no. 12, pp. 2214–2226, Dec. 2007.

[14] L. Moreau, "Stability of multiagent systems with time-dependent communication links," *Trans. Autom. Control*, vol. 50, no. 2, pp. 169–182, Feb. 2005.

[15] B. Touri and A. Nedić, "Discrete-time opinion dynamics," in *Proc. 45th IEEE Asilomar Conf. Signals, Syst. Comput.*, 2011, pp. 1172–1176.

Bernard Chazelle received the PhD degree in computer science from Yale University in 1980. He is the Eugene Higgins professor of computer science at Princeton University, where he has been on the faculty since 1986. He is currently a member of the Institute for Advanced Study in Princeton. He has held research and faculty positions at Collège de France, Carnegie-Mellon University, Brown University, Ecole Polytechnique, Ecole normale supérieure, University of Paris, INRIA, Xerox Parc, DEC SRC, and NEC Research, where he was a fellow for several years. The author of the book, *The Discrepancy Method*, and the winner of three Best Paper Awards from SIAM, he is a fellow of the American Academy of Arts and Sciences, the European Academy of Sciences, and the Association for Computing Machinery.