Dynamic Systems

Peter Grogono

February 2005

Contents

1 Introduction **2**

2 Dynamic System Basics **2**
2.1 Classifying Systems 4
2.2 Metrics 7
2.3 Behaviour of Systems 8
2.4 Quantitative Analysis 11

3 Examples **13**
3.1 The Logistic Map 13
3.2 The Lorenz Attractor 13
3.3 Roots of Unity 14
3.4 Mandelbrot Set 15
3.5 Gait 16
3.6 Evolution 16
3.7 Cellular Automata 16

4 Boolean Networks **17**
4.1 Boolean Functions 17
4.2 Visualizing Boolean Networks 18
4.3 Scale-free Networks 19
4.4 Applications 21

5 Evaluation **22**

A Calculating Lyapunov Exponents **22**

B Fractional Dimensions **23**

"Have I understood you correctly?" he said at last. "You gave that brutal despot, that born slave master, that slavering sadist of a painmonger, you gave him a whole civilization to rule and have dominion over forever? And you tell me, moreover, of the cries of joy brought on by the repeal of a fraction of his cruel decree? Trurl, how could you have done such a thing?"

"You must be joking!" Trurl exclaimed. "Really, the whole kingdom fits into a box just three feet by two by two and a half . . . it's only a model."

"A model of what?"

Stanislaw Lem [11, page 139]

1 Introduction

These notes provide a brief and elementary introduction to dynamic systems and the terminology that surrounds them. There is no deep mathematics here but, if you are puzzled by frequent references to *attractors*, *trajectories*, *phase space*, *edge of chaos*, *Lyapunov exponents*, and other arcane terms, you may find some enlightenment herein.

The phrase “dynamic systems” includes an enormous variety of different kinds of system. If a system evolves with time, it can probably be described as a dynamic system. In many cases, reasoning with a concept so all-embracing is not productive, because few deep results hold over very wide ranges. The surprising thing about dynamic systems is that, diverse as they are, they share quite deep properties. For example, they may evolve in ways that are regular, cyclic, or chaotic (these terms will be defined more precisely later). Even more surprising is the fact that there are quite general laws describing these behaviours — even “laws of chaos”.

Since living organisms evolve[1] in time, dynamic systems theory must apply to them. Indeed, some progress is being made in applying the theory to how genomes work, how cells differentiate, and so on. These notes outline key points of the theory and indicate some possible applications of it.

2 Dynamic System Basics

A ***dynamic system*** is a system that, actually or conceptually, evolves over time. A simple pendulum is a system that actually evolves over time. The process of finding a square root using Newton’s formula is a system that conceptually evolves over time — we might imagine computing one approximation per minute, for example — but it can also be considered simply as a sequence of approximations. We will use terminology consistent with evolving in time even though that may not be the only way of viewing a system.

States A dynamic system has ***states*** S_i. As it evolves, it moves through a sequence of states

$$S_0,\ \ldots,\ S_{i-2},\ S_{i-1},\ S_i,\ S_{i+1},\ S_{i+2},\ \ldots$$

A state is represented by a value of some kind: an integer, a real number, a vector of real numbers, a matrix, etc. The value should represent the state completely but should not contain redundant information.

The set of all states is called the ***state space*** of the system. A sequence of states is called a ***trajectory*** or an ***orbit***. When a system evolves from a state S_i to a subsequent state S_{i+1}, we say that it undergoes a ***state transition***. If we view the system as a directed graph with states as nodes and transitions as edges, then a trajectory is a walk in this graph.

Informally, we may refer to states as being “close together” or “far apart”. Two states S_i and S_j are close together if

[1] “Evolve” here means simply “change” — this is not Darwinian evolution (yet).

- either there is a trajectory $S_i, \ldots, S_j$ in which there are only a few states between S_i and S_j,
- or the system may be moved from S_i to S_j by a small perturbation.

Sometimes, it is useful to be formal and to define a ***metric*** on the state space. A metric d assigns a value $d(S_i, S_j)$, called the ***distance between*** S_i ***and*** S_j, usually a positive real number, to any pair of states. §2.2 discusses metrics in more detail.

A ***perturbation*** is a forced change of the state of the system, usually to a nearby state. In general, perturbing a system on one trajectory will move it to another trajectory.

Here are two simple systems that we will use as running examples below.

Example 1: Simple Pendulum. A simple pendulum consists of a weight (called the *bob*) suspended by a rod. We assume that as the pendulum swings, the rod remains in a plane. The rod is rigid (not a string) so that, for example, the bob may be higher than the support. We will refer to the simple pendulum as SP.

One component of the state of the pendulum is the *swing angle*, θ, which is the angle between the string and the vertical. The swing angle is not sufficient to describe its state. For example, when $\theta = 0$, the pendulum might be at rest or passing through the vertical during its swing. The pair (θ, ω), where ω (or $d\theta/dt$ or $\dot{\theta}$) is the time derivative of θ, is sufficient to describe the state completely.

Although the simple pendulum does indeed seem to be a "simple" system, it can be used to demonstrate almost all of the interesting features of dynamic systems [2]. □

Example 2: Boolean Network. A Boolean network (BN) can be modelled by a directed graph $G = (V, E)$. An edge (v_i, v_j) transmits a Boolean signal (0 or 1) from node v_i to node v_j. Each node is associated with a Boolean function; the inputs of the function are the signals received on the incoming edges; and the output of the Boolean function is the state of the node. The state of the system is a Boolean string consisting of the state of each node. We will refer to a network of this kind as a BN.

A state transition in a BN is defined as follows:

1. Each node propagates its current state (0 or 1) along its out-edges.
2. Each node computes its new state from the signals it receives on its in-edges.

Figure 1 shows a simple BN and its state diagram. The function at v_1 is *and* ($\wedge$). The function at v_2 and v_3 is *or* ($\vee$). Since there are three nodes, there are $2^3 = 8$ states. We have used binary encoding to number the states, as shown in the table below. A perturbation of BN would change a 0 to a 1 or *vice versa*. □

v_1	v_2	v_3	S	v_1	v_2	v_3	S
0	0	0	0	1	0	0	4
0	0	1	1	1	0	1	5
0	1	0	2	1	1	0	6
0	1	1	3	1	1	1	7

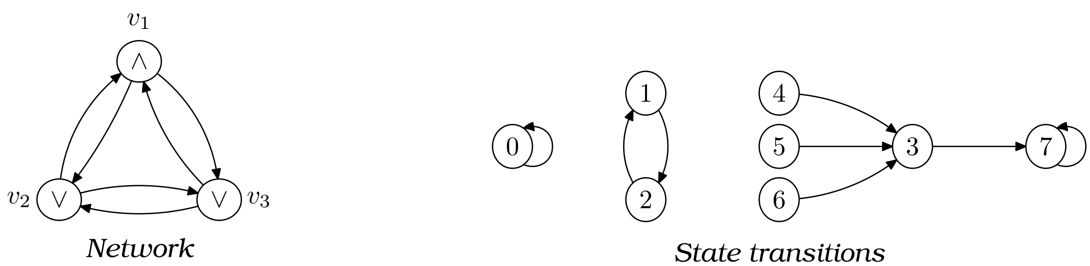

Figure 1: A Boolean network

2.1 Classifying Systems

Discrete and Continuous Systems Systems may be classified by the discreteness of their state spaces. The state space of a ***discrete system*** is a set of isolated points, usually described by a vector of integers. In many cases, the integers are restricted to $\{0, 1\}$ and the state space can be represented as a bit string. The state space of a ***continuous system*** is a continuum, usually described by a set of real numbers.

> ► SP is a continuous system. BN is a discrete system.

The interval between the state transitions of a continuous system may also be continuous or discrete. Discrete transitions are usually described by iterative formulas, such as $x' = \lambda x(1 - x)$ for the logistic map (§3.1). Continuously evolving systems such as the Lorenz attractor (§3.2) are usually described by differential equations.

The ***phase space*** of the system is a particular representation of the state space (the two terms are practically synonymous). If there are N state variables, the phase space is N dimensional. For $N \leq 3$, we can draw diagrams of the phase space; for $N > 3$, we can draw projections of the phase space but it may be harder to obtain intuition about the system's behaviour.

> ► The phase space of SP can be represented as a graph with axes θ and ω. If we allow the pendulum to go "over the top", so that θ can have values less than zero and greater than 2π, the phase space is infinite ($-\infty < \theta < \infty$ and $-\infty < \omega < \infty$). Figure 2 shows some trajectories in the state space (θ, ω) of the damped simple pendulum. The attractors (see §2.3) correspond to the points $(2k\pi, 0)$ where the pendulum is pointing straight down in stable equilibrium. The attractors are separated by repellors where the pendulum is pointing straight upwards in unstable equilibrium.

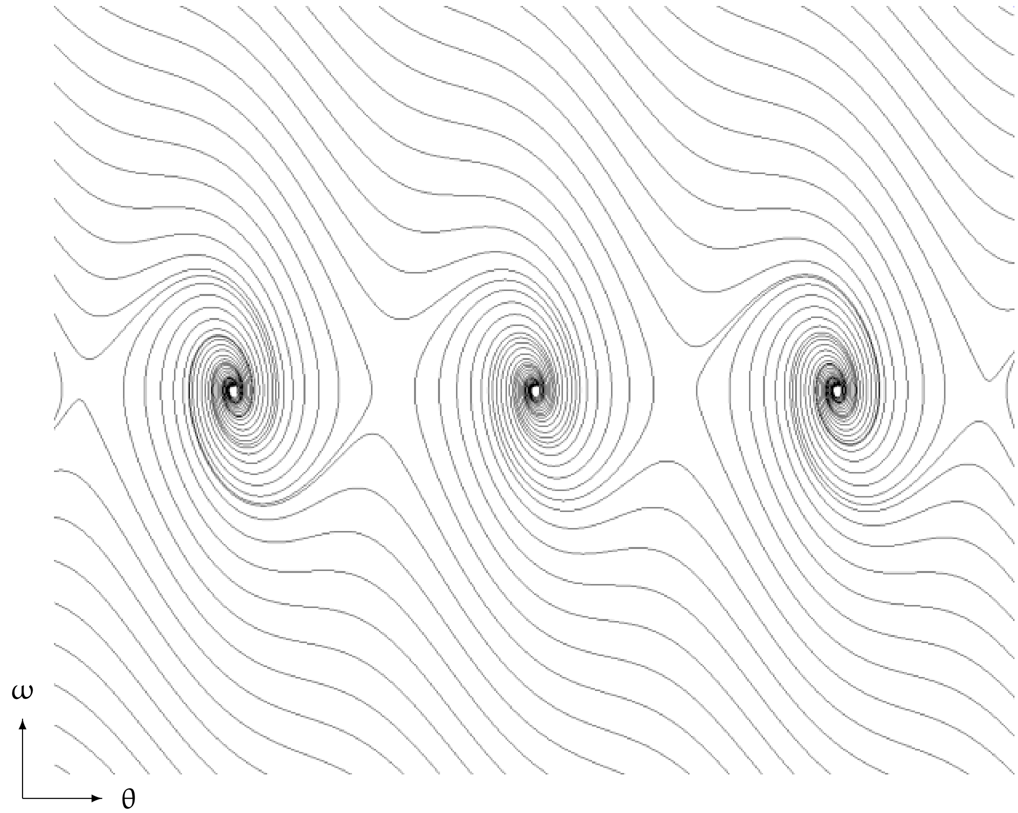

Figure 2: State space of the damped simple pendulum. The equation of motion is $\ddot{\theta}+\dot{\theta}+5\sin\theta = 0$.

Conservative and Dissipative Systems Systems may also be classified by the number of predecessors and successors a state may have. We assume that each state must have at least one successor — a system cannot simply stop evolving. A state with two or more successors is called a ***fork*** (see Figure 3). A system is ***deterministic*** if each state has exactly one successor. It is ***non-deterministic*** if a state may have more than one successor. All of the systems discussed in these notes are deterministic (and so do not have forks).

If each state has at most one predecessor, the system is called ***conservative***. In a conservative, deterministic system, every state has one predecessor and one successor. Consequently, its trajectories contain no forks or joins and consist either of closed loops or infinite chains. In principle, information about a single state is sufficient to trace the entire history of the system, both before and after the given state.

If a state may have more than one predecessor, it is called a ***join*** (see Figure 3) and the system is called ***dissipative***. Two or more trajectories of a dissipative system may merge together to form a single trajectory.

A ***closed dissipative system*** receives no energy from the environment and will eventually stop (strictly: converge to a point attractor). An ***open dissipative system*** receives energy from the environment and may continue to run indefinitely. The earth is an open dissipative system because it receives energy from the sun and radiates energy (at longer wavelengths) into space.

A system may have states for which there are no predecessors. These states are sometimes called

Garden of Eden states but we will use the shorter term ***initial states***.

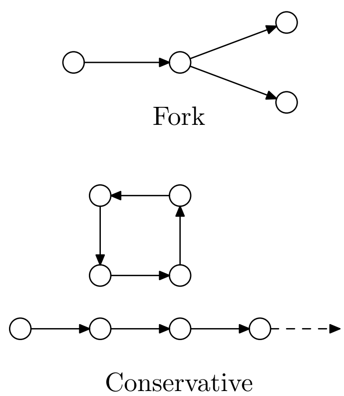

Figure 3: System fragments.

Figure 3 shows a fork, a join, and two small systems. The conservative system has a limit cycle with four states and one initial state followed by an infinite sequence of states (of which only the first four are shown). The dissipative system has two initial states and one limit cycle with three states.

Conservation of Area Suppose that the phase space of a continuous system has two dimensions. Consider two trajectories, T_1 and T_2, in the phase space over two intervals of equal durations Δt. The intervals define two areas, as shown in Figure 4. If the system is conservative, these areas will be the same. If it is dissipative, as in Figure 4, they will be different. Systems with more state variables have a similar property, but it is harder to visualize. There is an obvious parallel between diminishing areas in a continuous system and trajectories joining in a discrete system.

An important property of dissipative systems is that they are ***memoryless***. Since trajectories may join, different initial conditions may lead to identical outcomes. In the limiting case of a dissipative system with a single attractor, the initial conditions are completely irrelevant, because the system will always evolve to the attractor.

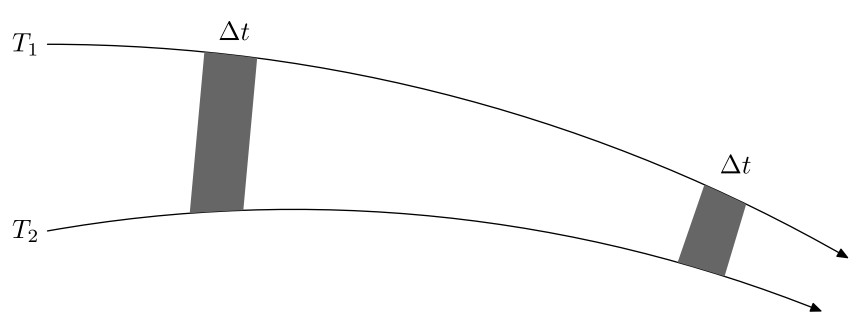

Figure 4: Area reduction in a dissipative system

▶ SP is a conservative system if there is no friction: if the pendulum is set swinging, it will swing indefinitely at the same amplitude. If there is friction, then SP is dissipative. At the end of the motion, the pendulum points straight down and does not "remember" its initial state. This example explains the choice of terms "conservative" and "dissipative": if there is no friction, the pendulum *conserves* energy but, if there is friction, the pendulum *dissipates* energy.

▶ BN is clearly dissipative with initial states $\{4,5,6\}$. The energy analogy doesn't work here but BN is dissipative by definition because the state space contains joins.

2.2 Metrics

A ***metric*** provides a measure of distance between states.

Continuous Systems A frequently used and convenient metric is Euclidean distance. The state is a tuple of real values. For state variables $\mathbf{x} = (x_1, x_2, \ldots, x_n)$:

$$d(\mathbf{x}, \mathbf{x}') = \sqrt{(x_1 - x_1')^2 + (x_2 - x_2')^2 + \cdots + (x_n - x_n')^2}$$

Sometimes, more intuitive results are obtained by giving different weights w_i to the variables, giving

$$d(\mathbf{x}, \mathbf{x}') = \sqrt{w_1^2(x_1 - x_1')^2 + w_2^2(x_2 - x_2')^2 + \cdots + w_n^2(x_n - x_n')^2}$$

▶ We show on page 10 that SP without friction satisfies

$$k^2\,\theta^2 + \omega^2 = k^2\,\theta_0^2$$

for swings with small amplitude θ_0. A suitable metric for the state space $\mathbf{x} = (\theta, \omega)$ would be

$$d(\mathbf{x}, \mathbf{x}') = \sqrt{k^2(\theta - \theta')^2 + (\omega - \omega')^2}.$$

With this metric, the phase space trajectories are circles with radius $k\,\theta_0$.

Discrete Systems The usual metric for a discrete system is the Hamming distance between bit strings. The ***Hamming distance*** $H(b, b')$ is defined as the number of positions at which the strings b and b' have a different value or, equivalently, the number of 1-bits in $b \otimes b'$ where $\otimes$ represents the exclusive-or operator applied to corresponding bits of the two strings.[2]

The ***normalized Hamming distance*** $\bar{H}(b, b')$ is the Hamming distance divided by the number of bits in the string. Clearly, $0 \le \bar{H}(b, b') \le 1$.

There are two problems with Hamming distance. The first one is that there not be a close relation between Hamming distance between states S and S' and the number of state transitions between S and S'. Perturbing a binary state b means changing a small number of its bits to give b'; $H(b, b')$ is small but the effect of the change on the system may be small or large.

[2] H and $\bar{H}$ are not standard notations; I have invented them for these notes.

▶ In the BN shown in Figure 1 there is a transition $4 \mapsto 3$ but states 4 and 3 are at opposite sides of Hamming space ($d(4,3) = H(100,011) = 3$ and $\bar{H}(100,011) = 1$). On the other hand, there is no path from 0 to 4 even though these states are separated by one bit in Hamming space ($d(0,4) = H(000,100) = 1$ and $\bar{H}(000,100) = 1/3$). This is a property of BNs rather than Hamming distance: a state transition in a BN may change an arbitrary number of bits in the state.

The second problem is a problem of spaces with many dimensions, not just Hamming spaces: they are hard to visualize. We can think of the states of a 3-bit Hamming space as the eight corners of a cube; edges of the cube represent state transitions. But this model doesn't work well (at least for our 3D brains) for more than 3 bits.

The distribution of distances in Hamming space is also hard to visualize. For an N-bit space, the distribution is binomial of order N. (That is, the number of pairs of states separated by d is the coefficient of x^d in the expansion of $(1+x)^N$.) For large values of N, the distribution is approximately normal. If two states S_1 and S_2 are picked at random, there is a high probability that $d(S_1, S_2) \approx N/2$ and a low probability that $d(S_1, S_2) \approx 0$ or $d(S_1, S_2) \approx N$.

2.3 Behaviour of Systems

The evolution of a system started in a particular state can often be divided into two parts: a ***transient response*** followed by a ***steady state*** response. The steady state response is normally used to characterize the system. It may be:

Stationary: the system evolves to a particular state and remains in that state from then on.

Oscillating: the system cycles through a fixed set of states (limit cycle).

Chaotic: the system moves through states without any apparent organization.

There are two points to note about chaotic responses.

- Chaotic behaviour is not caused by imprecise computation. There are equations (called "stiff") that are hard to solve because small calculation errors lead to quite different answers, but this is not necessarily chaos. Chaotic systems exist in the real world, independently of computers. Nor is chaos necessarily non-deterministic: it may appear in both deterministic and non-deterministic systems.
- Second, the defining characteristic of chaotic behaviour is that very small changes in the initial state lead to very large changes in the trajectory. This is known as the "butterfly effect" because the weather is a chaotic system and, in principle, a butterfly flapping its wings in Medicine Hat could cause a snowstorm in St-Louis-Du-Ha-Ha.[3]

A limit cycle with a very long period and a chaotic trajectory are, for many practical purposes, indistinguishable.

[3] In a 1963 presentation, Lorenz said: "One meteorologist remarked that if the theory were correct, one flap of a seagull's wings would be enough to alter the course of the weather forever."

► Figure 5 shows the phase space of SP in chaotic motion (or possibly in a very long limit cycle: it is hard to tell which). It might appear that this image contradicts the rule that a trajectory may not cross itself. This is not the case, however, because the state space is now three dimensional (the variables are θ, ω, and the driving force) but Figure 5 shows only a two dimensional projection.

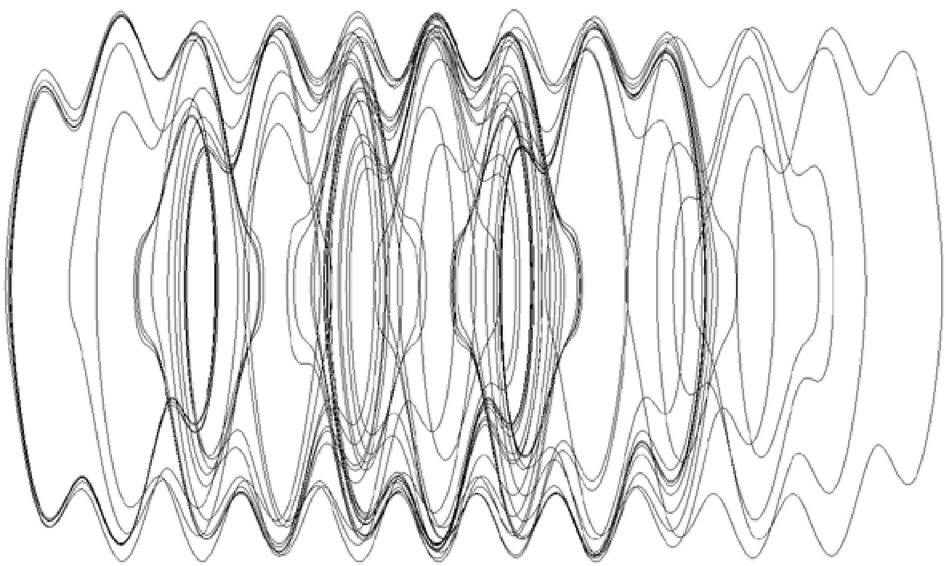

Figure 5: A single trajectory in the state space of a damped simple pendulum driven by a sinusoidal force.

In some systems, there is a single parameter that determines, at least in a general way, how the system will behave. Typically, small values of this parameter give stable behaviour and large values give chaotic behaviour. The most interesting behaviours often lie in the region where the behaviour is becoming less stable and more chaotic. Self-organizing systems, for example, often work best in this region. This phenomenon has led to the popular phrase ***the edge of chaos***.

A stationary state is also called an ***equilibrium state***. A system will remain in an equilibrium state unless it is perturbed (that is, moved to a nearby state). There are three main kinds of equilibrium, depending on the response to a perturbation:

Stable equilibrium: the system oscillates around the equilibrium point and, if it is dissipative, eventually returns to it.

Unstable equilibrium: the system moves away from the equilibrium point.

Metastable equilibrium: the system displays stable or unstable behaviour, depending on the form of the perturbation.

The conventional example of metastability is a mountain pass. Suppose the pass runs from south to north and you are at its highest point. A small movement east or west is stable, because you have to go uphill. The north-south direction is unstable because a small movement will send you sliding down the mountain.

► SP is in stable equilibrium when pointing straight down and in unstable equilibrium when pointing straight up. When it is close to stable equilibrium, the restoring force, $\sin\theta$, can be approximated as θ, and the system is linear. SP with damping and a driving force may display chaotic behaviour, as shown in Figure 5.

Attractors and Repellors A region of the state space from which there are no exit paths is called an ***attractor***. An attractor consisting of a single state whose only successor is itself is a ***point attractor*** (and an equilibrium state). An attractor consisting of a cycle of states called a ***limit cycle***. The set of states that eventually lead to an attractor is called the ***basin of attraction*** of the attractor.

The opposite of an attractor is a ***repellor***, which is a region of the state space that trajectories leave but do not enter. A point repellor is an unstable equilibrium state.

► SP has attractors at $\theta = 2k\pi$ (integer k), where the pendulum is pointing straight downwards, and repellors at $\theta = (2k+1)\pi$, where the pendulum is pointing straight upwards. As can be seen in Figure 2, every point in the state space that is not a repellor belongs to the basin of attraction of one attractor. The pendulum may make several complete revolutions before swinging back and forth.

► BN has three attractors: the individual states 0, 7, and the limit cycle $\{1, 2\}$. The basin of attraction of state 7 is $\{3, 4, 5, 6\}$.

Attractors clearly have a dimension lower than that of the phase space. For example, an attractor might be a 2D line in a 3D phase space. Some attractors have dimensions that are not integers. For example, a row of dots may contain too many dots to be considered zero-dimensional but not enough dots to form a line; such a row would have a dimension between 0 and 1. (See Appendix B on page 23 for a brief discussion of fractional dimensions.) A ***strange attractor*** is an attractor with a fractional dimension.

Linearity An important property of a very wide class of continuous systems is that, whether they are linear or non-linear, they behave as linear systems in a small region of the phase space that encloses an equilibrium state. For this reason, much of the classical theory of complex systems focused on their behaviour near equilibrium states.

Researchers realized only recently that, although it has the convenient property of being easily analysed, behaviour near equilibrium is not very interesting. This realization led to interest in systems ***far from equilibrium***, with Ilya Prigogine leading the way [13, 14, 15].

► In the ideal case of a frictionless pendulum swinging in a vacuum, the system is *conservative*. If we restrict the motion to small swing angles, $\sin\theta \approx \theta$ and the equation of motion,

$$m\frac{d^2\theta}{dt} + mg\ell\theta = 0$$

is linear. Solving the equation gives

$$\begin{aligned} \theta &= \theta_0 \sin k\,t \\ \omega = \frac{d\theta}{dt} &= k\,\theta_0 \cos k\,t \end{aligned}$$

where $k = \sqrt{g\ell}$. It follows that

$$\begin{aligned} k^2\,\theta^2 + \omega^2 &= k^2\,\theta_0^2\,(\sin^2 k\,t + \cos^2 k\,t) \\ &= k^2\,\theta_0^2 \end{aligned}$$

Conditions for Chaos Continuous systems are often described by differential equations. In standard form, all of the equations are first order. An equation with derivatives of second or higher order is converted to several first order equations. For example, the equation of the frictionless simple pendulum is usually written

$$\frac{d^2\theta}{dt^2} + k^2 \sin\theta = 0$$

but in standard form is written

$$\begin{aligned} \dot{\omega} &= -k^2 \sin\theta \\ \dot{\theta} &= \omega \end{aligned}$$

The necessary conditions for chaotic behaviour are:

1. there must be at least three state variables in the description of the system, and
2. at least one of the equations must be non-linear.

That these are not *sufficient* conditions can be seen from a Newtonian 2-body system such as the earth and moon, which is non-linear with 12 degrees of freedom ($2 \times 3 \times |\{p, v\}|$) but not chaotic. A 3-body system, however, may display chaotic behaviour. Even 3-body Newtonian systems may not be chaotic if, as in the case of the sub/earth/moon system, there are large mass discrepancies.

There do not appear to be simple criteria for chaos in discrete systems.

► SP cannot exhibit chaotic behaviour, even with damping. With damping and a driving force, however, the conditions for chaos are met because there are three state variables (θ, ω, and F) and a non-linear term ($\sin\theta$):

$$\begin{aligned} \dot{\omega} &= F(t) - \mu\omega - k^2 \sin\theta \\ \dot{\theta} &= \omega \end{aligned}$$

2.4 Quantitative Analysis

One way to evaluate a dynamic system is to run it and see what happens. This is not a very satisfactory method, for obvious reasons. This section describes two standard techniques for analyzing system behaviour.

2.4.1 Lyapunov Exponents

Consider a dynamic system with a single state variable x and assume that state transitions are given by a function f. That is, if the system is in state x_i, the next state is $x_{i+1} = f(x_i)$. We use the notation f^n for n iterations, so that $x_{i+n} = f^n(x_i)$.

We consider two initial states separated by a small distance ε. As the system evolves, the separation between states is assumed to change exponentially:

$$f^n(x + \varepsilon) - f^n(x) = \varepsilon\, e^{n\lambda}$$

in which λ is the ***Lyapunov exponent***.[4] We can estimate $\lambda(x_0)$ for a single trajectory $x_0, x_1, x_2, \ldots$ using the formula (see Appendix A on page 22):

$$\lambda(x_0) = \lim_{n\to\infty} \frac{1}{n} \sum_{i=0}^{n-1} \log_e |f'(x_i)|$$

The Lyapunov exponent for the system is the average value of $\lambda(x)$ for many starting points x.

For a system with more than one state variable, $\mathbf{x} = (x_1, x_2, \ldots, x_k)$, there is a Lyapunov exponent associated with each variable:

$$f^n(\mathbf{x} + \varepsilon) - f^n(\mathbf{x}) = \varepsilon\, e^{n\Lambda}$$

where $\Lambda = \lambda_1 + \lambda_2 + \cdots + \lambda_k$.

For conservative systems, $\Lambda = 0$. For closed dissipative systems, $\Lambda < 0$. If the system is chaotic, then at least one of the λ_i must be positive. (Note that this implies, as we have already seen, that a system with one state variable cannot be chaotic.)

2.4.2 Derrida Plots

Lyapunov exponents are clearly not suitable for discrete systems with binary state spaces. A method for the Hamming metric was introduced by Bernard Derrida [5] and is now called a ***Derrida plot***.[5]

A Derrida plot is a 2D graph constructed by repeating the following steps many times:

1. Choose two random states represented by bit strings b_i and b_j.
2. Let the two points evolve to b_{i+1} and b_{j+1}.
3. Compute $x = \bar{H}(b_i, b_j)$ and $y = \bar{H}(b_{i+1}, b_{j+1})$.
4. Plot the point (x, y).

[4] After A.M. Lyapunov (1857–1918), a Russian mathematician best known for his work on stability.

[5] Do not confuse Bernard Derrida, a professor of physics at Ecole Normale Supérieure and Université Paris 6, with the late post-modernist Jacques Derrida. Confusion is made more likely by papers such as *On Fractal Thought: Derrida, Hegel, and Chaos Science* by Laurie McRobert, in which "Derrida" refers to Jacques!

At the end of this process, there will be many points corresponding to each value of x. Replace these points by their average. The result is a curve that characterizes the system; it is called the ***Derrida curve***.

The diagonal $x = y$ corresponds to "parallel" evolution in which corresponding states remain the same distance apart. The area below the diagonal is called the *ordered region*, because trajectories converge in state space. The area above the diagonal is called the *chaotic* region because trajectories diverge.

The ***Derrida coefficient*** D_c is $\log s$, where s is the slope of the Derrida curve at the origin. If $s = 1$ then $D_c = 0$, giving "parallel" evolution. $D_c < 0$ indicates that the system is ordered and $D_c > 0$ indications that it is chaotic.

Figure 6 shows two typical Derrida plots.[6] Note that the upper curve, representing a chaotic system, starts off above the diagonal but eventually falls below it. This is inevitable because the right side of the figure corresponds to normalized Hamming distances close to 1. Clearly, if we start with two states differing in most of their bit positions, they are very unlikely to get further apart, even in a chaotic system.

3 Examples

3.1 The Logistic Map

A state is a real value x. Here is the formula used to obtain the successor, x', of a state x:

$$x' = \lambda x(1 - x)$$

The behaviour of the system depends on the value of the constant λ.

Figure 7 summarizes the behaviour of the logistic equation. A vertical section through the diagram corresponds to a value of λ. A dot corresponds to a state (value of x). For small values of λ (not shown here), the system converges to a single state. As λ increases, this state turns into a limit cycle consisting of two states, as at the left of the diagram. As λ continues to increase, the limit cycle continues to double in size. After about eight states, the diagram becomes fuzzy, indicating the onset of chaotic behaviour. As λ increases still further, the behaviour is mainly chaotic but there are brief intervals in which the behaviour reverts to a limit cycle.

3.2 The Lorenz Attractor

The Lorenz Attractor [12] is important in the history of dynamic systems because it was the first modern study of a strange attractor. Poincaré and others had realized that dynamic systems have curious properties but, without computers, they were unable to appreciate the full richness of the

[6] Taken from `http://www.cogs.susx.ac.uk/users/andywu/derrida_plots.html`

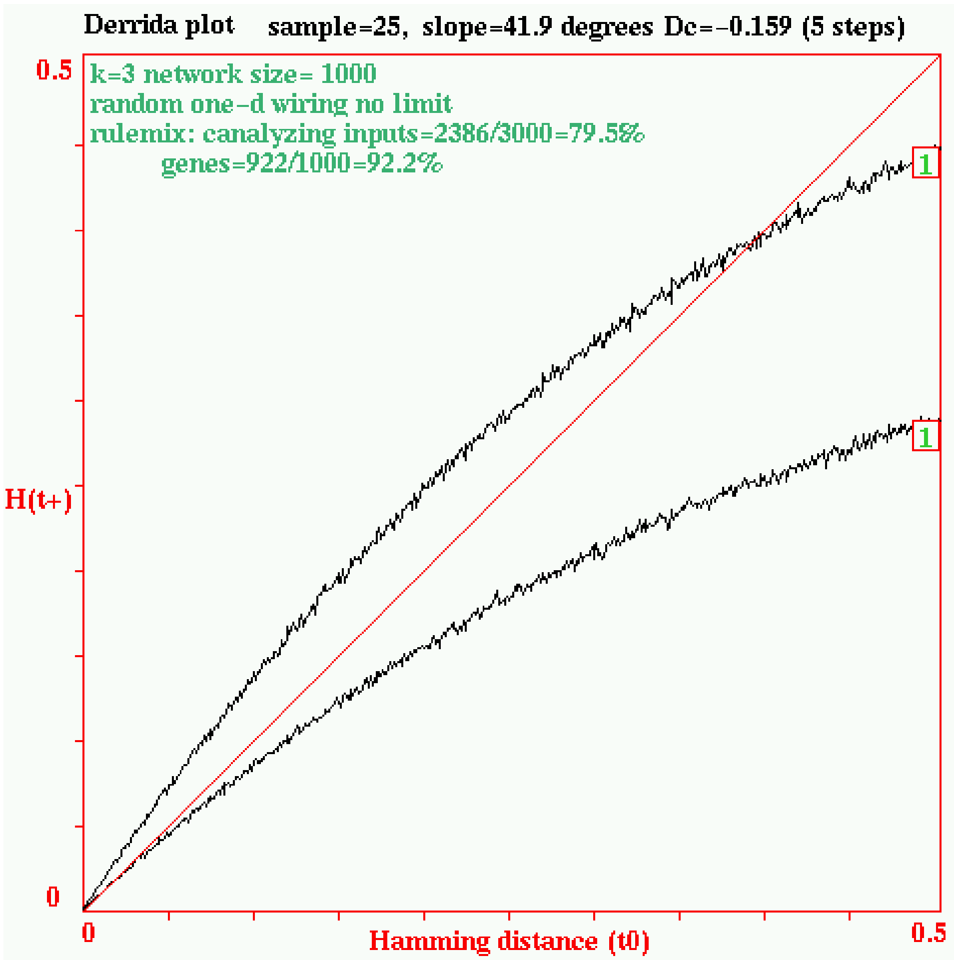

Figure 6: A Derrida Plot

behaviour. In the early 1960s, Ed Lorenz, working in meteorology and investigating sensitivity to initial conditions, discovered the equations

$$\begin{aligned} \dot{x} &= a\,(y - x) \\ \dot{y} &= x\,(b - z) - y \\ \dot{z} &= x\,y - c\,z \end{aligned}$$

in which a, b, and c are constants. The system described by these equations is now called the ***Lorenz Attractor***. Paul Bourke has created beautiful images of the Lorenz Attractor; his web page[7] also shows a simple physical model of it.

3.3 Roots of Unity

Newton's rule says that x approximates a zero of a well-behaved function f, then $x' = x - f(x)/f'(x)$ is a better approximation. The rule has an obvious geometric explanation for real x but turns out

[7] http://astronomy.swin.edu.au/~pbourke/fractals/lorenz/

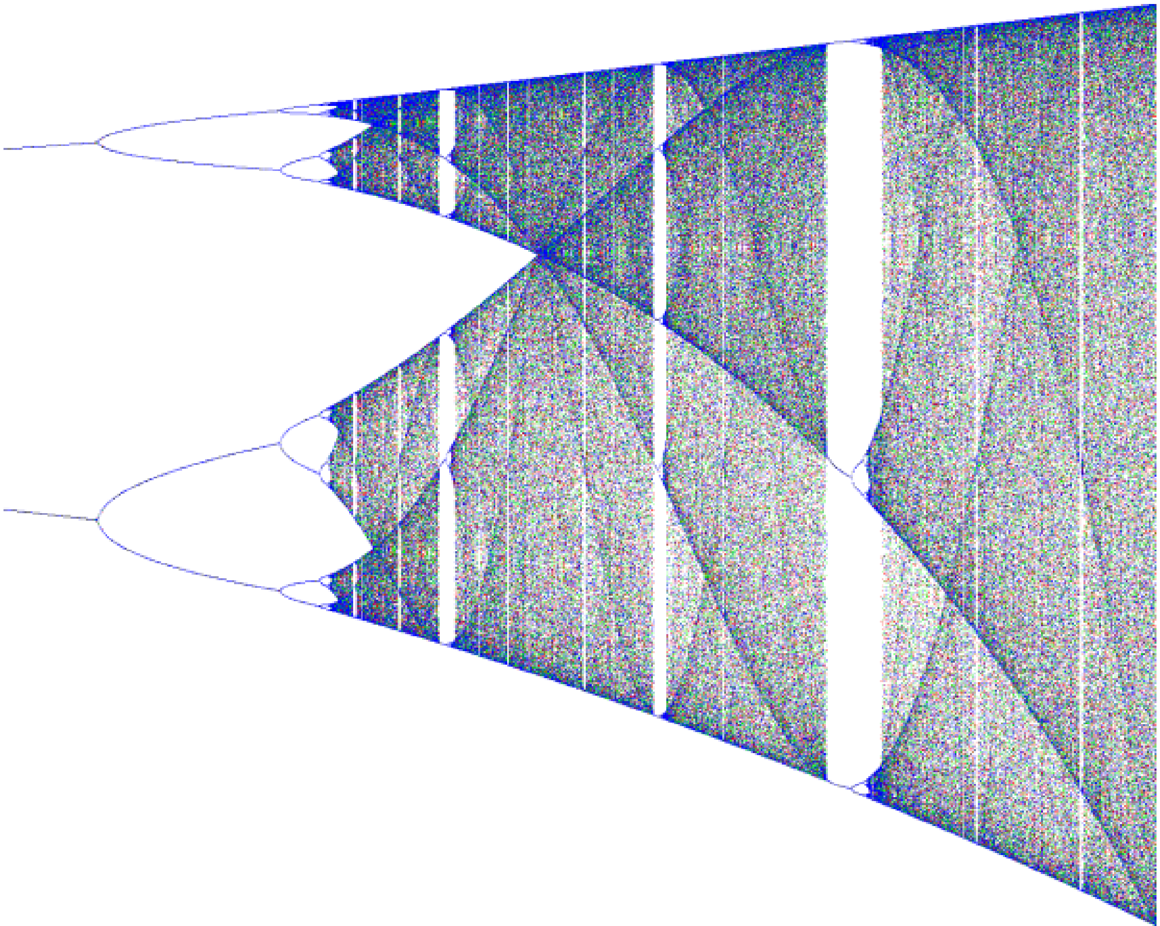

Figure 7: The Logistic Map

to work for complex values as well. We can use it to find the complex values of $\sqrt[n]{1}$ by iteratively solving the equation $z^n - 1 = 0$. The attractors of this system are the values of $\sqrt[n]{1}$ and the union of the basins of attraction is the entire complex plane.

The case $n = 2$ provides no surprises: the basins of attraction are $\Re z < 0$ and $\Re z > 0$ where $\Re z$ is the real part of z. For $\Re z = 0$ the iterated value is undefined.

The cases $n \geq 3$ are much more interesting: the basins of attraction are fractal and the boundaries between them are elusive. Consider the particular case $n = 3$ (we are solving $z^3 - 1 = 0$ and the roots are $z = 1$, $(-1 - \sqrt{3})/2$, and $(-1 + \sqrt{3})/2$) and suppose that we colour the basins red, green, and blue respectively. We find that between any red and green point, there is a blue point, and similarly for the other combinations.

3.4 Mandelbrot Set

The Mandelbrot set $M \subset \mathbb{C}$ is not strictly dynamic system but has some similarities with the systems described here. It is the set of points $c \in \mathbb{C}$ for which the iteration

$$z = z^2 + c$$

does ***not*** diverge to infinity. Thus M is a kind of strange attractor.[8]

3.5 Gait

We walk and run; four-legged animals have different ways of moving, such as cantering, trotting, and galloping. These are called ***gaits***. Studies show that motion is a dynamic system with particular gaits as attractors [10]. There are even quite accurate laws of gaits: a study of many animals in the Serengeti showed that a particular ratio was 1 for walking, 6 for trotting, and 9 for cantering. (Joints were modelled as pendulums with springs; the ratio used was the ratio of spring torque to torque induced by gravity.)

3.6 Evolution

Evolution can be considered as a dynamic system with each possible individual considered to be a point in state space (sometimes called a ***fitness landscape***). A state transition is a genetic mutation. Local optima in the fitness landscape act as attractors.

3.7 Cellular Automata

Cellular automata (CA) can be constructed in any number of dimensions. A 1D CA consists of a single row of cells; a 2D CA consists of square cells distributed on a grid; a 3D CA consists of cubes distributed on a 3D lattice; and so on.

A 1D CA is a function $f : B \to B$, where B is the set of infinite bit strings. The key feature of CAs is that a particular bit of $f(b)$ depends only on the corresponding bit of b and its neighbours. The neighbourhood is usually symmetric and therefore has an odd number of bits. The smallest neighbourhood has 1 bit but gives only trivial results. However, even the smallest practical neighbourhood, 3 bits, provides $2^{2^3} = 256$ different functions with ordered, fractal, and chaotic behaviour [19, pages 24–7].

Cellular automata have a distinguished history.

- John von Neumann, inspired by a suggestion from Stanislaw Ulam, worked on 2D automata in the late 1940s [17].
- Konrad Zuse devised "computing spaces", which were, in fact, CAs. He was the first to suggest the idea[9] (later claimed by Wolfram — see below) that the universe is a CA.
- Tommaso Toffoli built the CAM–6[10] at MIT in the early 1980s and demonstrated applications in various fields [16].

[8] There are, of course, many other strange attractors. Their images can be very striking and have become a sort of popular art form. See, for example, `http://xaos.theory.org/gallery.html`.

[9] See *Zuse's Thesis: the Universe is a Computer* at `http://www.idsia.ch/~juergen/digitalphysics.html`.

[10] The original CAM–6 was a circuit board designed for an IBM PC-AT. CAM–6 simulators for modern PCs have been written by a number of people.

- John Horton Conway invented the *Game of Life*, a 2D CA, in 1970 and it became — and remains — very popular [6].
- For many years, Stephen Wolfram was fascinated by 1D CAs, founding Mathematica to fund his hobby, and writing an encyclopedic tome full of extravagant claims [19].

4 Boolean Networks

We discuss one particular kind of dynamic system, the Boolean network (BN), in some detail. BNs are emerging as the simplest realistic model for biological systems such as the brain and the genome. We encountered BNs previously in the simple example of §2. In this section, we look at some more definitions and models of behaviour.

BNs are more general than CAs. Whereas a node of a CA directly affects only a small, fixed neighbourhood, the edges of the graph may join arbitrary nodes. Biological observations suggest that BNs may be more appropriate as models for biological systems than CAs. For example, dendrites ("nerves") connecting neurons may be several feet long and genes (using the gene regulatory system discussed in §4.4) may influence other genes that are far away on the genome.

Following Kauffman [9], we define a ***NK network*** as a directed graph $G = (V, E)$ with $|V| = N$ and average in-degree K (consequently $|E| \approx NK$). Each node has a state which is either 0 or 1. The state of the network is a Boolean string of N bits, one for each node. Consequently, the network has 2^N states.

At each node, there is a Boolean function. Edges propagate binary signals $s \in \{0, 1\}$. Each state of a NK network can be represented as a string

A NK network is ***fully-connected*** if its underlying undirected graph is connected; otherwise it is ***modular***. (Some authors use modular in a more general sense: the underlying graph has clusters of connected components joined by only a few edges.)

4.1 Boolean Functions

The table in Figure 8 describes a Boolean function with three inputs. The states are listed in descending order (from 1 1 1 to 0 0 0) following Wolfram's convention [19]. We can describe this function by writing the output column as a binary string 10001100. The string has $8 = 2^3$ bits. In general, we need 2^k bits to describe a Boolean function with k inputs. It follows that there are 2^{2^k} such functions. An NK network has N such functions and there are consequently about[11] $\left(2^{2^K}\right)^N$ such networks.

The function output 10001100 can be represented as a single decimal (140) or hexadecimal (8C) number. Such numbers are often used to encode Boolean functions. An NK network can be described by N records, one for each node, each record giving the integer code for the Boolean function at the node and a list of the edges emanating from it.

[11] Not exactly, because K is an average.

Inputs			Output
1	1	1	1
1	1	0	0
1	0	1	0
1	0	0	0
0	1	1	1
0	1	0	1
0	0	1	0
0	0	0	0

Figure 8: A 3-input Boolean function

Kauffman discovered that the overall behaviour of the network is determined by K. Assume that N is large, that the nodes are connected randomly, and that the Boolean functions are chosen randomly. For $K \leq 2$, the system usually behaves in an orderly way. For $N \geq 4$ and random Boolean functions, the system usually behaves chaotically. Chaotic systems can be "tamed" by restricting the Boolean functions used.

The string representing a random Boolean function has, on average, about the same number of 0s and 1s. We can construct networks with ***biased Boolean functions*** — functions whose strings contain an unbalanced distribution of 0s and 1s. Even better, we can require that the functions be canalizing. A ***canalizing function*** has the property that one value of a particular input bit determines the output. The function in Figure 8 is canalizing because, if the second input bit is zero, then the output is zero for all values of the other input bits. In standard, 2-input logic, AND and OR gates are canalizing, but XOR gates are not.

A function can be canalizing on more than one input. For example, AND and OR gates are canalizing on both inputs. Canalizing functions are important for two reasons: networks that use only canalizing functions tend to be stable, and functions in biological systems seem to be canalizing.

Canalizing becomes more significant as the number of inputs increases. This is because the proportion of canalizing functions drops rapidly with increasing in-degree. Randomly connected networks with high in-degree behave chaotically unless the functions are picked carefully — not randomly.

4.2 Visualizing Boolean Networks

It was at first thought to be difficult to visualize the behaviour of BNs. However, Andrew Wuesche's doctoral research provided algorithms that explore the topology of the state space efficiently [20]. The Discrete Dynamics Laboratory (DDLab[12]) is a program based on these algorithms and Figure 9 is an example of its output.

[12] `http://www.ddlab.com/`

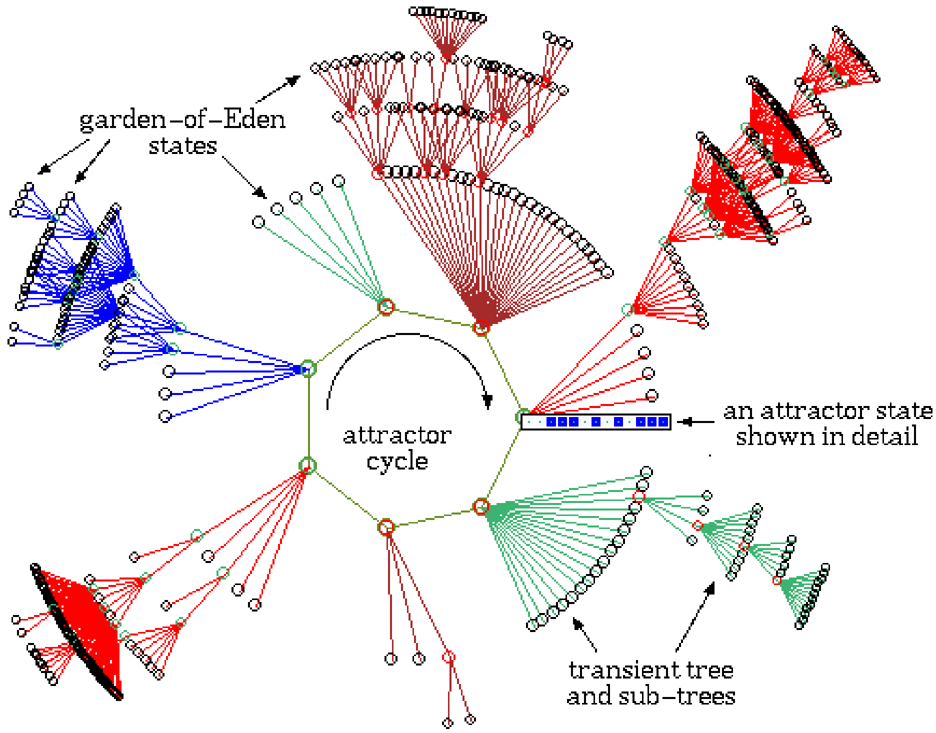

Figure 9: A Boolean network with a single attractor displayed by DDLab, exhibiting Garden of Eden states, transients, and the attractor.

4.3 Scale-free Networks

People have long been interested in random graphs. A popular question is "How many people have to attend a party for there to be a 50% chance that two have the same birthday?" The answer is not 182.5, as you might think, but about 23. The mathematical properties of random graphs were established in a series of eight classic papers (summarized in [8]) by Paul Erdős and Alfréd Rényi. The random graphs were constructed by an algorithm of the form "pick two nodes at random and, if they are not already connected, add an edge to the graph". Kauffman constructed his random BNs in a similar way.

In Erdős/Rényi random graphs, all nodes look much the same: degrees will vary but, in a large graph, they will all be fairly close to the average degree. Recently, Barabási and others [3] pointed out that real networks do not look like these random graphs at all. In a wide variety of situations, ranging from neuronal connections through road systems to the internet, there are a few nodes with high degree (often called "hubs") and a large number of nodes with small degree. Analysis of these networks showed that the degree of nodes obeyed a *power law*. Specifically, if N_k is the number of nodes with degree k then

$$N_k \propto k^{-\gamma}$$

where γ is a constant ($\propto$ means "proportional to"). Figure 10 gives some examples.

A classical random graph has a "scale" defined by the average degree of a node: looking at a node,

Phenomenon	Number of nodes	Average degree	γ
Actor collaboration	212,250	28.78	2.3 ± 0.1
World wide web	325,729	5.46	2.1 ± 0.1
Power grid	4,941	2.67	4
Scientific paper citation			3

Figure 10: Scaling exponents from [4]

we can say whether it is “typical” (close to the average degree) or not. The degree distribution is similar to the classical bell curve (Gaussian distribution) and a typical node is one that is near the peak of the bell. If the distribution of degrees follows a power law, however, the distribution has no peak and there are no “typical” nodes. Consequently, graphs with a power-law distribution are called ***scale-free*** [3, page 70]. Scale-free graphs have significantly different properties from classical random graphs.

In particular, it is harder to construct scale-free networks than classical random networks, but a number of algorithms have been proposed [1]. The assumptions made in classical random graph construction are:

1. There is an existing node set to which edges are added randomly.
2. All nodes have the same probability of being connected.

Both assumptions are changed for scale-free graph construction:

1. Construction begins with an empty set of nodes; adding nodes is part of the construction process.
2. The probability of a node being connected increases with the number of edges that it receives.

Another aspect in which scale-free networks differ from random networks is their robustness. Both kinds of graph are fairly insensitive to the removal of a few edges. The situation is quite different with respect to nodes. Deleting one of the many lightly connected nodes of a scale-free network has a negligible effect, but deleting a hub is quite likely to break the network into disconnected fragments. There are many examples of this phenomenon: closing an airport does not make much difference unless the airport is a hub, such as Toronto, Chicago, or Atlanta; a blown circuit-breaker can turn off power distribution to half a continent; integrity of the internet depends on the functioning of about six major domain-name servers.

The fact that real-life networks are scale-free is not surprising. Consider roads as an example. A city usually has more roads leading to it than a town, a town more than a village, and a village more than a farmhouse. If you built a new house, you would not build a road to some other house chosen at random, you would build a road to the nearest hub — a village or a town. Moreover, it is likely that you would build your house *near* an existing hub, so hubs act as attractors as the network grows.

4.4 Applications

The properties of Boolean networks have been studied intensively for many years and are now well-know. Recently research has been directed at determining if they provide useful models of real phenomena, particularly in biology. For example, Wuensche mentions genetic regulatory networks in cell differentiation, protein networks in cell signalling, cell networks in tissues and organs, and the actual organs themselves [21].

Gene Regulation The genome does not consist simply of thousands of independent genes. Genes are *coupled* in the sense that, when a gene is enabled (that is, the transcription machinery synthesizes the protein it codes for), other genes may be enabled or disabled. This behaviour suggests that a BN might be an appropriate model.

Harris *et al.* examined the hypothesis that BNs can be constructed to behave like known gene regulatory networks [7]. They analyzed published data for more than 150 systems, classified by the average number of "direct molecular inputs", K. For most systems, $3 \le K \le 5$. For a few systems, $7 \le K \le 9$. As we have seen above, this implies that, if we use BNs to model gene regulation, the Boolean functions cannot be random.

The authors tried to match the observed behaviour of gene regulatory networks with two kinds of Boolean functions: biased functions (in which 0 are 1 outputs are not equally probable) and canalizing functions. Using Derrida coefficients, they found that canalizing functions could explain the observations but that biased functions could not. Specifically, a gene required an average of 2.6 canalizing inputs to maintain stability.

The paper makes a number of predictions of phenomena that might be observed in gene networks if their hypothesis is correct. If it is correct, BNs provide a plausible explanation (or, perhaps, model) for cell differentiation. The genome has the potential for describing an very large number of different cell types, depending on which genes are enabled. However, if the regulatory system ensures that there are a relatively small number of attractors, then these attractors might correspond to cell types. Cancer might be the result of a perturbation of the system leading to an undesired attractor.

Memory and Learning BNs have also been proposed as a model of memory. The attractors represent the information that is "remembered" and the basins of attraction enable memories to be retrieved.

If this model is to be useful, it must also explain learning — how memories come to be acquired. There may be a link between learning and the growth of scale-free networks. A problem with this idea is that we do not appear to learn by growing neurons but rather by modifying links and, especially, weights. However, the topology of the links must allow learning and so, perhaps, the growth phase of the brain is adapted to producing an effective learning engine.

5 Evaluation

As mentioned in the introduction, dynamic system theory applies to a very wide variety of systems. This is both a strength and a weakness: a strength because it provides a qualitative explanation for practically anything that moves, and a weakness because, by itself, it can provide a detailed, quantitative description almost nothing.

The well-known physicist Steven Weinberg has a similar opinion of theories of complexity. The following, rather long, quotation is from his review [18] of Wolfram's book [19].

> Scientists studying complexity are particularly exuberant these days. Some of them discover surprising similarities in the properties of very different complex phenomena, including stock market fluctuations, collapsing sand piles, and earthquakes. Often these phenomena are studied by simulating them with cellular automata, such as Conway's Game of Life. This work is typically done in university physics departments and in the interdisciplinary Santa Fe Institute. Other scientists who call themselves complexity theorists work in university departments of computer science and mathematics and study the way that the number of steps in a computer calculation of the behavior of various systems increases with the size of the systems, often using automata like the Turing machine as specific examples of computers. Some of the systems they study, such as the World Wide Web, are quite complex. But all this work has not come together in a general theory of complexity. No one knows how to judge which complex systems share the properties of other systems, or how in general to characterize what kinds of complexity make it extremely difficult to calculate the behavior of some large systems and not others. The scientists who work on these two different types of problem don't even seem to communicate very well with each other. Particle physicists like to say that the theory of complexity is the most exciting new thing in science in a generation, except that it has the one disadvantage of not existing.
>
> It is here I think that Wolfram's book may make a useful contribution . . . . Wolfram goes on to make a far-reaching conjecture, that almost all automata of any sort that produce complex structures can be emulated by any one of them, so they are all equivalent in Wolfram's sense, and they are all universal. . . . The trouble with Wolfram's conjecture is not only that it has not been proved – a deeper trouble is that it has not even been stated in a form that could be proved. What does Wolfram mean by complex? If his conjecture is not to be a tautology, then we must have some definition of complex behavior independent of the notion of universality. The pattern produced by the rule 110 cellular automaton certainly looks complex, but what criterion for complexity can we use that would tell us that it is complex enough for Wolfram's conjecture to apply?

We can find striking correlations between the predictions of dynamic systems theory and the structure and behaviour of living systems. Are striking correlations sufficient, or should we be looking for more precise models?

A Calculating Lyapunov Exponents

The following derivation of the Lyapunov exponent is given by Baker and Golub [2, pages 85–6]. We can solve

$$f^n(x + \varepsilon) - f^n(x) \quad = \quad \varepsilon \, e^{n\lambda}$$

for λ, giving

$$\begin{aligned}\lambda &= \frac{1}{n}\log_e\left[\frac{f^n(x+\varepsilon)-f^n(x)}{\varepsilon}\right] \\ &\approx \frac{1}{n}\log_e\left|\frac{df^n(x)}{dx}\right|.\end{aligned}$$

The chain rule for differentiation tells us that

$$\begin{aligned}\frac{d}{dx}f(f(x_0)) &= f'(x_0)\,f'(f(x_0)) \\ &= f'(x_0)\,f'(x_1)\end{aligned}$$

Applying the chain rule n times and taking the limit as $n \to \infty$ gives:

$$\lambda = \lim_{n\to\infty}\frac{1}{n}\sum_{i=0}^{n-1}\log_e|f'(x_i)|$$

This formula provides a general method for calculating Lyapunov exponents. For particular systems, there may be easier ways of computing or estimating them.

B Fractional Dimensions

There are many ways of defining fractional dimensions; here is a simple one.

Suppose we cover a line of length L with N_ε short segments of length ε. Then

$$N_\varepsilon = L/\varepsilon.$$

Similarly, if we cover a square of side L with N_ε small squares of side ε, we have

$$N_\varepsilon = (L/\varepsilon)^2.$$

The dimension d of a general self-similar object has the property that, if an object of size L can be covered with N_ε small objects of size ε then

$$N_\varepsilon = (L/\varepsilon)^d.$$

Taking logarithms gives

$$d = \frac{\log N_\varepsilon}{\log L + \log(1/\varepsilon)}$$

For small values of ε, we can neglect the term $\log L$, and define the ***dimension*** of the object as

$$d = \lim_{\varepsilon\to 0}\frac{\log N_\varepsilon}{\log(1/\varepsilon)}$$

For example, the *Cantor set* is constructed from the unit line segment by removing the middle third and applying the same operation recursively on the remaining segments. We have

$$\begin{aligned} N &= 1, \quad 2, \quad 4, \quad \ldots, \quad 2^n \\ \varepsilon &= 1, \quad 1/3, \quad 1/9, \quad \ldots, \quad (1/3)^n \end{aligned}$$

and

$$\begin{aligned} d &= \lim_{n \to \infty} \frac{\log 2^n}{\log 3^n} \\ &= 2/3 \end{aligned}$$

References

[1] Réka Albert and Albert-László Barabási. Topology of evolving networks: local events and universality. *Physics Review Letters*, 85:5234–5237, 2000.

[2] G. L. Baker and J. P. Golub. *Chaotic Dynamics: an Introduction*. Cambridge University Press, 1990.

[3] Albert-László Barabási. *Linked: the New Science of Networks*. Perseus Publishing, 2002.

[4] Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *Science*, 286:509–512, October 1999.

[5] B. Derrida and D. Stauffer. Phase transitions in two dimensional kauffman cellular automata. *Europhysics Letters*, 2:739–745, 1986.

[6] Martin Gardner. Mathematical games: the fantastic combinations of John Conway's new solitaire game "life". *Scientific American*, 223(10):120–123, October 1970.

[7] Stephen E. Harris, Bruce K. Sawhill, Andrew Wuensche, and Stuart Kauffman. A model of transcriptional regulatory networks based on biases in the observed regulation rules. *Complexity*, 7(4):23–40, 2002.

[8] Michal Karonski and Adrzej Rucinski. The origins of the theory of random graphs. In R. L. Graham and J. Nesetril, editors, *The Mathematics of Paul Erdős*. Springer, 1997.

[9] Stuart Kauffman. *At Home in the Universe*. Oxford University Press, 1995.

[10] J. A. Scott Kelso. *Dynamic Patterns: the self-organization of brain and behavior*. MIT Press, 1995.

[11] Stanislaw Lem. *The Cyberiad: Tales for the Cybernetic Age*. Avon, 1974. Translated from Polish by Michael Kandel.

[12] E. N. Lorenz. Deterministic nonperiodic flow. *J. Atmos. Sci.*, 20:130–141, 1963.

[13] Grégoire Nicolis and Ilya Prigogine. *Exploring Complexity: an Introduction*. W. H. Freeman, 1989.

[14] Ilya Prigogine. *From Being to Becoming: Time and Complexity in the Physical Sciences.* W. H. Freeman, 1980.

[15] Ilya Prigogine and Isabelle Stengers. *Order out of Chaos: Man's New Dialogue with Nature.* Bantam, 1984.

[16] T. Toffoli and N. Margolus. *Cellular Automata Machines.* MIT Press, 1987.

[17] John von Neumann. *Theory of Self-Reproducing Automata.* University of Illinois Press, 1966. Edited and completed by Arthur W. Burks.

[18] Steven Weinberg. Is the universe a computer? *New York Review of Books*, 49(16), October 2002.

[19] Stephen Wolfram. *A New Kind of Science.* Wolfram Media, 2002.

[20] Andrew Wuensche. *Attractor Basins of Discrete Networks.* PhD thesis, University of Sussex, 1997. Cognitive Science Research Paper 461. Abstract available from `http://www.cogs.susx.ac.uk/users/andywu/thesis.html`.

[21] Andrew Wuensche. Basins of attraction in network dynamics: a conceptual framework for biomolecular networks. Sante Fe Institute working paper 02–02–004, 2002. Available from `http://www.santafe.edu/research/publications/wpabstract/200202004`.