1 Research Statement: James P. Crutchfield

Much of the scientific enterprise depends on successfully identifying structure in natural processes. Once identified through some discovery process, structures become the substrate for theory building, prediction, and, eventually, an understanding of internal mechanism. Engineering, too, depends on natural structures, but attempts to design them into systems that perform useful and controllable functions. Both, thus, rely on understanding *patterns*: how they form in nature and how they emerge in artificial systems. The single, overriding theme in my research is to understand, at a fundamental level, what patterns are. To what do we refer when we talk about *structure*, *regularity*, *modularity*, *organization*, and *symmetry*? A second, and equally important, theme is to use this understanding to investigate the process by which humans (and artificial systems) discover novel patterns.

1.1 Overview

These interests originally led me to consider low-dimensional dynamical systems since they present an inherent ambiguity. On the one hand, they are simple to define; on the other, they generate complexity. In analyzing them one initially focuses on the raw phenomena: the sheer diversity of behaviors and structures that emerge. Then one moves on to analyze the mechanisms responsible for the complexity. Finally, one confronts inference: Can these structures be detected in experiment? Are these complexities genuinely more difficult to learn? This trajectory—passing from phenomenology to analysis and then to inference—is a familiar one in the study of complex systems.

The bulk of my early research, then, concentrated on studying chaotic dynamical systems and testing them against a number of phenomena. Low-dimensional systems, however, fall far short when one turns to address the vast majority of natural systems. More interesting and challenging are high-dimensional dynamical systems, such as those in space (e.g., *lattice dynamical systems*, in particular) and those with nonregular connections between components—what I call *network dynamical systems*. These investigations, in low- and high-dimensional systems, led to many applications in a wide range of fields—some that I pursued directly, from solid state physics to astrophysics, and many that developed based on my and others early work.

By the late 1980s, the repeated successes of dynamical systems modeling led me to consider the common elements in the theory-building process by which they were attained. This led to the discovery of a general definition of pattern, which is articulated in my theory of *computational mechanics* and is closely related to that of *information processing architecture* which one finds in the theory of computation. This parallel led to the definition of *intrinsic computation*: (i) How much historical information does a process store? (ii) How (in what architecture) is that information stored? (iii) How is the stored information processed to produce future states? A number of applications of the general theory followed that analyzed intrinsic computation in chaotic and nonchaotic dynamical systems, stochastic systems (e.g., hidden Markov models), spatial systems (e.g., cellular automata and spin systems), and in quantum systems (automata and grammars).

In parallel with these phenomenological and analytical studies and based on having a well defined concept of pattern, I began to investigate the process of inference itself—the *dynamics of learning*. How can one design an autonomous agent whose dynamical behavior is learning? This is an active research topic. Although there is still much fundamental work remaining, there are a number of results that describe and constrain how an individual agent learns about and builds a model of its environment.

It is a short step from the dynamics of learning to study how learning agents interact in groups. Are groups more effective than noninteracting individuals? Many believe so. But how can one be precise about the net gain (or loss) in functionality that comes from groups solving nondecomposable tasks? Can we say that *collective cognition* emerges from agent-agent interactions? Are these interactions competitive or cooperative? These questions arise in many disciplines, from the formation of multicellular organisms to societies of insects and humans, and they remain largely open. My goal is to contribute some of the fundamental theory to answer these questions and to experimentally demonstrate the benefits and pitfalls of collective cognition in groups of autonomous robotic vehicles.

It is one thing to consider how an observer discovers patterns in a process, but in some ways this begs a much larger question, How would evolution produce pattern-discovering observers in the first place? This question led me to investigate evolutionary dynamics, both in the domain of biology—adapting statistical mechanics and nonlinear dynamics to analyze evolutionary innovations—and in computer science—analyzing optimal evolutionary search and average-case computational complexity.

This synopsis hopefully gives a sense in which the notion of pattern and the process of pattern discovery underlie my research. From this one can readily appreciate its range and anticipate the motivations that will determine its future directions. The following sections delve into more detail on dynamics (in low and high dimensions and in networks and learning), intrinsic computation, evolution, and collective cognition. The closing section comments on my abiding interest in experiment, in particular, in running a laboratory that allows in-house exploration of complex systems.

1.2 Dynamics

My work in the 1980s focused on the problem of inferring deterministic structure in complex physical processes. Despite attacks on this general problem by a number of disciplines, e.g., statistics and computer science, and the focus of contemporary physics on material phenomena, I would argue that this concern is fundamentally in the domain of contemporary physics. This derives partly from physics' mandate to build predictive mathematical models of the natural world and partly from the simple fact of nature's inherent nonlinearity and complexity.

My approach (and that of a small community then) was to investigate processes modeled by nonlinear dynamical systems and to quantifying unpredictable behavior in time and spacetime. The theoretical techniques used drew from dynamical systems theory, information theory, statistical mechanics, and computation theory. The goal throughout was (i) to understand how the geometry [5][1] of state-space structures gives rise to mechanisms that produce complex behavior and (ii) to quantify the observed complexity [7,9,11,27]. This resulted in projects on the effects of fluctuations on nonlinear systems, behavioral phenomenology, deducing models from data, and experimentation on spatially extended systems, for example. The following paragraphs discuss these in turn.

[1]Numbers in brackets refer to papers listed in the CV.

1.2.1 Low Dimensions

Chaotic systems are exponentially sensitive to fluctuations. The major physical consequence of this is that external noise must be taken into account at a basic level. It was established, for example, that noise acts as a disordering field for chaotic dynamics [4,10,12,13,15,16]. The theoretical description of this applied scaling and renormalization group methods to the period-doubling transition to chaos [10,12]. The identification of noise as a relevant variable and the dynamical entropy as a disorder parameter [4,11] completed Feigenbaum's phase transition picture of the routes to chaos.

Many major theoretical developments in dynamics have proceeded from conjectures based on numerical or experimental work with nonlinear systems. Due to the absence of general analytic methods, there has been (necessarily) a substantial effort in direct investigation of nonlinear phenomenology. In these, nonlinear systems are used as prototypes to (i) appreciate the diversity of behavior possible, (ii) study directly the geometric mechanisms underlying complex behavior, and (iii) validate statistical measures [3,7,11,14,15,22,27]. It was just this type of experimental mathematical investigation [19] of low-dimensional dynamical systems that led to a number of early applications in physical systems: e.g., solid-state turbulence [2,6] and chaotic attractors in fluid flows [17].

1.2.2 High Dimensions

To move more closely to statistical mechanical systems, I worked to extend dynamical systems techniques from low to high dimensions; particularly to spatially-extended systems [16,18,23,26]. This led to a new class of lattice dynamical systems [16,18]. One surprising result was the discovery of extremely long-lived transients that preclude the observation of the asymptotic attractors within practical experimental times. This suggested that the Lorenz-Ruelle-Takens hypothesis, that fluid turbulence is governed by a chaotic attractor, could be fundamentally in error. It also pointed to new types of complex *unsimulatable* behavior in physical systems [24].

I also investigated a related model class, network dynamical systems, in which the spatial translation symmetry is broken by arbitrary and dynamically changing coupling [25]. In the limit of high interconnectivity dimension [23] these deterministic systems are architecturally similar to spin glasses and so to neural networks. A basic question that came out of these studies was how to design high-dimensional systems to perform specified computations. (See, for example, Evolutionary Dynamics, below.) The answer lies in the geometry of attractors and basins in high dimensions [23-25].

To help further the study of networks, Duncan Watts (Sociology, Columbia) and I recently founded SFI's Network Dynamics Program, which has been generously funded by Intel's Research Division, and held a workshop on "Structure and Dynamics in Complex Interacting Networks". (See discuss.santafe.edu/dynamics.)

1.3 Dynamics of Learning

The behavioral diversity of nonlinear processes suggests that dynamical systems in the presence of fluctuations may be a sufficient class with which to model a very wide range of natural processes. With that ansatz the question of detecting structure within complex data is simplified to deducing the underlying *dynamic*—that is, the state-to-state mapping or the equations of motion. Although distinguishing between a deterministic chaotic dynamic and stochastic perturbations may seem an arbitrary one, a distinction can be made systematically and parametrized by a given level of predictive capability [1,22,27]. The theory puts into a single framework a number of previously disparate chaotic data analysis methods and introduces a quantitative measure of the structural complexity of nonlinear equations of motion [22]. This led to a number of applications, including recent work on automated modeling and adaptive learning of complex dynamics.

I extended this work to computation-theoretic models of nonlinear dynamical systems with a method that reconstructs the minimal equivalent *causal architecture* from data [27,29,49,50,69,76,79]. This provides a concrete connection between nonlinear processes and the well developed theory of computation. This in turn yielded a method for measuring *intrinsic computation*—a system's inherent information-processing capabilities. With our minimality and optimality theorems, which characterize the causal representation (ϵ-machine), there is now a well defined approach to understanding how physical substrates support computation. The theory, *computational mechanics*, is constructive and so experimental applications have been carried out. Computational mechanics synthesizes aspects of statistical mechanics, dynamical systems, and computation theory, into a unified approach to modeling nonlinear systems. There are several notable consequences. First, the theory suggests a simple and geometric interpretation of the partition function: it summarizes the path algebra of phase space. (That algebra, by the way, is what is meant by *pattern*.) Second, it relates fluctuation theory to the intrinsic computational capability of a physical process. Finally, it points out that physical systems capable of computation must be in a critical state; that is, they must have a subspace in which the Kolmogorov-Sinai entropy vanishes.

Another off-shoot of this work, which is still on-going, is a theory of individual agent learning [65,82,83,89,90].

1.4 Intrinsic computation

Not surprisingly, having a well defined notion of intrinsic computation led to direct studies of how various classes of process support information processing. In addition to the deterministic dynamical systems already mentioned, intrinsic computation has been analyzed in several stochastic systems:

1. Hidden Markov models, also called probabilistic nondeterministic finite automata [46,49].
2. Quantum systems, in particular, finite and pushdown quantum automata and regular and context-free quantum grammars [63].

Intrinsic computation has also been extensively investigated in spatially extended systems:

1. Cellular automata [38-41,55,59,66,70,79].
2. Spin systems in one and two spatial dimensions [60,67,94].
3. Low-dimensional materials; specifically in polytypes [91-93].

The net result is a new way to think about how various substrates support information processing. It appears that there may be new (very much smaller) spatial and temporal scales at which to implement computation and, moreover, the kinds of information processing possible may be much more diverse than our current concept of computation allows. Thus, one of the exciting directions suggested by this work would be to adapt them to nanotechnology engineering.

1.5 Evolutionary Dynamics

In parallel with the work (1990s) on intrinsic computation and the dynamics of learning, I became interested in the question of how biological nature would evolve organisms that learn and adapt. As a prerequisite to addressing this, I focused on how evolution produces structural complexity.

The accepted *gradualist* view is that incremental adaptation is the dominant mechanism. Another, somewhat contrasting, view is that change occurs via *epochal evolution* [84], in which long periods (*epochs*) of stasis in a population are punctuated by rapid *innovations* to higher-fitness organisms that reflect the emergence of new structures or novel functions. Importantly, there is a compelling biological motivation for epochal dynamics. It has been observed in a number of natural evolutionary systems: sudden morphological changes in the paleontological record (so-called *punctuated equilibria*), in bacterial colonies, and in the evolution of tRNA secondary structures, for example. It was also discovered to be an important mechanism in evolving cellular automata that perform spatial computations [43,45,48,52-54,56,57,59,65,70,72,77,86].

Through my work [61,62,68,71,73,74,78,84,85] and that of others at SFI, we now know that large degeneracies in the genotype-to-fitness mapping are the main source of the epochal nature of the evolutionary dynamics. For every genotype, there is a range of genotypic change that mutation can induce, without affecting phenotype or fitness. In this way, the space of genotypes is broken into strongly and weakly connected sets with respect to fitness. Sets of equal-fitness genotypes are strongly connected clusters—called *neutral networks*; while genotype sets of differing fitness are only weakly connected to each other. Stated most simply, then, epochs occur because members of an evolving population must search most of the space of neutral variants before higher-fitness genotypes are discovered.

All of the resulting population-dynamical behavior derives from the interplay of this neutral-network architecture, the infinite-population dynamics, and the stochasticity and discretization arising from finite-population sampling. Through our *statistical dynamics* analysis of these interactions one comes to appreciate a number of fundamental trade-offs and basic mechanisms that drive and inhibit epochal evolution. Our analysis formalized a number of oft-used, but ill-defined concepts, such as the emergence of increasing complexity, how phenotypic structural constraints guide evolution, and the persistence of frozen accidents, as these phenomena occur in epochal evolution.

This and other work in the mid-1990s led to an SFI workshop on evolutionary dynamics, the establishment of SFI's Evolutionary Dynamics Program (funded by the Keck Foundation), and the book "Evolutionary Dynamics: Exploring the Interplay of Selection, Accident, Neutrality, and Function", edited by myself and Peter Schuster. Details on the book and workshop can be accessed from www.santafe.edu/∼chaos.

1.6 Collective cognition

Studies of the evolution of competing organisms, on the one hand, and the dynamics of learning, on the other, are naturally complemented by questions concerning the interaction of learning agents. Many forms of individual learning are enhanced by communication and collaboration with other intelligent agents. I refer to this as *collective cognition*, by analogy with the well known concept of *collective action*. People (and other intelligent agents) often think better in groups and sometimes think in ways that are simply impossible for isolated individuals. Perhaps the most spectacular and important instance of collective cognition is modern science. An array of formal organizations and informal social institutions also can be considered means of collective cognition. For instance, Hayek famously argued that competitive markets effectively calculate an adaptive allocation of resources that *could not* be calculated by any individual market-participant.

Collective cognition involves an interaction among three elements—the individual abilities of the agents, their shared knowledge, and their communication structure. Cognitive collectives therefore resemble many other complex systems which are collectives of goal-directed processes. Typically, the individual processes know little of the detailed dynamics and the state of the overall system and, therefore, must use adaptive techniques to achieve their goals. There are many naturally occurring examples, including human economies, human organizations, ecosystems, and even spin glasses. In addition, it has recently become clear that many of the engineered systems of the future must be of this type, with massively distributed computational elements. There is optimism in the multiagent system (MAS) field that widely applicable solutions to large, distributed problems are close at hand. Some experts now believe that, in the information and telecommunications networks of today, one sees nascent examples of artificial cognitive collectives.

While it is sometimes possible to hand-craft very small collectives, what is needed is a full-fledged science of collective design. One needs to know how to make a large collective work towards a specific goal (e.g., minimize packet throughput in a router, win a soccer game) in a decentralized, adaptive way. Only with such methods can MAS—as the engineering discipline it claims to be—fulfill its promise and meet the very real technical needs of the near future.

I am now adapting computational mechanics, network dynamical systems, and the dynamics of learning to address these issues. For example, recent work shows that simple reinforcement-learning agents in a collective become a kind of distributed game-theoretical dynamical system, with the full range of dynamical complexities [88].

This research program is funded by DARPA's Agent-Based Computing Program. The project details can be found at www.santafe.edu/∼dynlearn. Last year I organized a workshop—"Collective Cognition: Mathematical Foundations of Distributed Intelligence"— to stimulate more analytical work in this area.

1.7 Experiment

I have always pursued experiment and theory concurrently. While this has certain costs, these are often outweighed by benefits. The first benefit is that, at one and the same time, experimentation focuses one on concepts that are grounded in reality and suggests new theoretical directions. The second benefit is that it gives one immediate access to experimental discoveries proposed by theory. Additionally, within nonlinear dynamics, the analyses suggested by theory and numerical simulation often require new types of experimental automation and on-line data analysis. Notably, the goals of qualitative dynamics often demand a tremendous amount quantitative analysis.

As an example, I built a data analysis and experiment control system for spacetime signal analysis of pattern generating systems. The particular nonlinear spatially extended systems to which this was applied included surface instabilities in ferrofluid, nonlinear image dynamics in video feedback [18,26], and pattern formation in chemical oscillators. One example of the symbiosis of theory and experiment comes from the investigation of video feedback. That experimental program led to the development of lattice dynamical systems [18,26], the discovery of super-transients [24], and an emphasis on information processing within the state structure of low-dimensional dynamical systems [28].

Another example is the laboratory that I established at SFI for distributed robotics to investigate practical and robust forms of collective cognition. The lab includes facilities for developing autonomous, wirelessly communicating robotic vehicles, an experimental arena in which they interact and are monitored, and a 128 CPU Linux cluster to run the distributed (and compute-intensive) learning algorithms and data analyses.

2 Teaching Statement: James P. Crutchfield

I enjoy teaching and take much personal satisfaction in it. In fact, after leaving UCB, I continued to be a research advisor to a number of Ph.D. students. Since SFI does not grant degrees, the students obtain Ph.D.s from their home universities, but are in residence at SFI during their dissertation research years. I also regularly involve graduate, undergraduate, and high school students in my research here.

I believe research training, in contrast with course work, is one of the best ways to give students access to the creative processes of scientific and intellectual life. Despite the pressure to specialize I also believe that students must be grounded both in theory and in experiment. In particular for the latter, it is important that students develop the physical-motor skills to build and manipulate experimental apparatus. The ultimate goal in this is to have students integrate the physical sense of the natural world that comes through hands-on experimentation and an appreciation of the mediation that instrumentation introduces with the theoretical enterprise of modeling and prediction.

The same teaching philosophy carries over to computational research and, specifically, programming. I tend to keep advanced students away from software packages, having them write their own basic routines and libraries, up until the point at which they understand in a concrete way modeling, the software design process, and their pitfalls.

In addition to research mentoring, there are a number of courses required to prepare students in complex systems, nonlinear dynamics, and modern computational thinking. These address, at the undergraduate level, a persistent lack in students' scientific training in nonlinear mathematics and, at the graduate, advanced concepts and methods in complex systems. The first half concerns developing a familiarity with the theoretical and experimental developments in the physics of nonlinear systems during the last half century; the second, a facility with fundamental computational tools for scientific research and technology development. I believe these aspects of training in complex systems would be substantially enhanced by courses such as those outlined below. My experience at UCB and SFI leads me to believe that these courses, or some variant, would be quite attractive to students.

2.1 Course: Statistical Dynamics and Intrinsic Computation

An advanced graduate-level course intended to bring students to the research frontier in nonlinear dynamics and complex systems. Emphasis is placed on statistical mechanical and computation-theoretic descriptions of complex physical, chemical, and biological systems. The intent is to deepen students' appreciation of classical dynamics and thermodynamics via geometric and statistical analysis of state-space structures and to contrast these with notions of organization imported from computation theory.

Topics:

1. Dynamical Systems Theory: abstract theory and basic differential topology; dissipative and Hamiltonian systems; forms of randomness; bifurcation theory; bifurcation sequence renormalization group and scaling theory.

2. Computational Mechanics (see fourth course below):

    (a) Statistical Mechanics of Complexity: symbolic dynamics, information theory, measurement theory, dimension and entropy, fluctuation theory.

(b) Computation within Physical Systems: formal language theory, quantitative measures of complexity,

(c) Causal Inference and Learning Theory: machine learning and modern statistical methods of structural inference, computational learning theory for physical problems, relation to statistical mechanics.

Projects: Experimentation with nonlinear systems through simulation and data analysis; code development; application to a phenomenon chosen by each student.

Audience: Advanced physics graduate students; theory students in mathematics, biology, chemistry, computer science, economics, and engineering.

Prerequisite: Junior level mathematical physics or equivalent.

Text: A large number are currently available; supplemented with my own manuscripts.

Duration: Two semesters.

2.2 Course: Computational Modeling Methods for Complex Systems

Students will be introduced to the interactive use of computers and contemporary computational concepts for scientific research. They will learn how to translate complex problems into appropriate computational tasks in ways that emphasize physical content. Students will use scientific workstations and associated software research tools in a computational laboratory to explore physical phenomena and theories.

The course is designed to complement standard Computational Physics courses which tend to focus on numerical techniques. The course synthesizes a range of techniques and concepts necessary for the effective research use of computers, emphasizing the complementary roles of building analytical, predictive theories and numerical simulation. Alternative offerings from applied mathematics and computer science for the same breadth would require several semesters of major-specific courses.

Topics:

1. Algorithms: automata theory, universality, computational complexity.
2. Languages: formal languages, assembly, HLL, LISP, and object oriented.
3. Machines: digital versus analog, microcontrollers to supercomputers.
4. Operating Systems: multitasking, multiuser, multiprocessing, real-time.
5. Networking: distributed processing, Internet collaboration, and web computing.
6. Scientific Communication: Object, bitmap, web, and page layout editors, information retrieval techniques, and electronic publishing.
7. Research Software: design and management, code development environments.
8. Symbolic Manipulation: basics, appropriateness, applications.
9. Advanced Computer Graphics: data visualization and rendering.

10. Data Reduction and Database Techniques.

11. Numerical Methods and Simulation.

Projects: Experiment control, signal analysis, complex system simulations, data reduction and display, solution of moderate to large systems.

Audience: Upper division physics majors; physics graduate student auditors; majors in engineering, computer science, mathematics.

Prerequisite: Mathematical Methods for Physicists, Introduction to Programming.

Text: several, e.g., Mathematica, Numerical Recipes, Interactive Computer Graphics.

Duration: One semester.

2.3 Course: Nonlinear Dynamics

This is a follow-on course to sophomore-level mechanics designed to introduce undergraduates to the diversity of complex behavior in simple nonlinear physical and biological systems. Both dissipative and Hamiltonian dynamics will be covered, as well as standard geometric and statistical analyses of complex behavior. The central example will be the chaotic driven pendulum. Both analytic solutions and interactive simulations will be used as teaching tools. The subjects covered would complement an undergraduate laboratory course on nonlinear dynamics.

Topics:

1. Basic Dynamical Systems: Hamiltonian and dissipative systems, attractor-basin portrait, the chaotic hierarchy, ODEs and maps, elementary differential topology.

2. Bifurcations: space of dynamical systems, bifurcation types, scaling theory of cascades.

3. Information Theory of Measurements and Unpredictability.

4. Phenomenology of Complex Behavior.

5. Conventional and Chaotic Data Analysis: statistics and Fourier analysis; attractor reconstruction, embedding, and dimension and entropy computation.

6. Statistical Mechanics of Chaos.

7. Computational Aspects of Physical Processes.

Projects: Numerical simulations and data analysis of nonlinear system of student's choice.

Audience: Upper division undergraduates; other science and engineering majors.

Prerequisite: Analytic mechanics or equivalent.

Text: A number of texts are currently available; supplemented with my own dynamical system simulation system.

Duration: One semester.

2.4 Course: Computational Mechanics

An upper-division undergraduate course on intrinsic computation in classical and quantum systems—low-dimensional, spatially extended, and networks.

Topics:

1. Algorithms: formal languages, automata theory, universality, computational complexity.
2. Symbolic Manipulation: basics, appropriateness, applications.
3. Continuous, Stochastic, and Quantum Computation.
4. Numerical Methods and Simulation of Complex Systems.

Advanced topics:

1. Intrinsic computation in low-dimensional dynamical systems.
2. Hidden Markov models.
3. Cellular automata computational mechanics.

Projects: Numerical simulations and data analysis of nonlinear system of student's choice. Formal proof-theoretic methods for establishing patterns and intrinsic computation.

Audience: Graduate students and upper division undergraduates in the sciences and engineering.

Prerequisite: Nonlinear Dynamics or Computational Modeling Methods for Complex Systems.

Text: Own manuscripts, supplemented with own software tools.

Duration: One semester.