AI Magazine Volume 21 Number 2 (2000) (© AAAI)

Review of *The Computational Beauty of Nature*

Elizabeth Bradley

Gary Flake's *The Computational Beauty of Nature* is about computation, fractals, chaos, complex systems, adaptation, and the connections among them. Its basic premise is that these "most interesting computational topics today" are deeply interrelated, and in some heretofore undescribed ways. The text is well crafted, and the scholarship is both broad and deep. The author is clearly a renaissance man as well as a wonderful teacher. He is equally good at succinct summaries and painting the big picture, and he makes particularly effective use of examples. Best of all is his infectious joy about his subject: The text is full of percolations of delight at the beauty of some concept or equation or at the sheer fun of hacking code. This book would be great for a course, as a reference, or just for plain old recreational reading.

A variety of underlying themes run through *The Computational Beauty of Nature*, beginning with the notion of *emergence* (p. 449):

> [Emergent] refers to a property of a collection of simple subunits that comes about through the interactions of the subunits and is not a property of any single subunit.... Usually the emergent behavior is unanticipated and cannot be directly deduced from the lower-level behaviors.

Classic examples are flocking behavior in birds or an anthill emerging out of the labors of a crowd of ants. Another distinguishing characteristic of emergence is that it involves no single omniscient entity or central controller; a group of construction workers building a house under the direction of an architect's drawings, for example, is not participating in emergent behavior.

The second theme is the presence of unexpected and interesting behavior in dynamic systems. Flake describes several examples of this, relating them to one another. Emergence—simple behavior in a complex system—is the first instance. Conversely, a simple system can also exhibit complicated behavior but only if it is nonlinear; this revelation was one of the driving forces behind the recent development of chaos theory. Flake also points out that boundaries—between order and chaos, for example, or between computability and incomputability—are extremely fertile areas and argues that evolution, adaptation, and learning derive their leverage by moving populations toward and away from these boundaries.

The Computational Beauty of Nature: Computer Exploration of Fractals, Chaos, Complex Systems, and Adaptation, Gary William Flake, The MIT Press, Cambridge, Massachusetts, 1998, 493 pp., *ISBN 0-262-06200-3.*

Flake's third theme is *universality*: that very different systems can behave in similar ways. In the throes of chaos, for example, predator-prey populations, stars, and human hearts all exhibit the same behavior—and all are equally unpredictable. Self-similarity arises when processes involve multiple spatial or temporal scales, which is quite common in nature, making fractals another universal concept. Both fractals and self-similarity are related to the notion of scaling behavior; there are lots of small earthquakes or avalanches and only a few large ones, for example. The size and frequency of events such as earthquakes are described by power-law distributions around some critical point, and the scaling exponent in each power law is related to the fractal dimension. The fourth and final theme in *The Computational Beauty of Nature* is a new way of thinking about the role of the computer in modern science. Flake views simulation as "... a form of experimentation in a universe of theories" (p. xiv). He believes that computers can, in principle, do anything that physics (or humans) can do, as long as the models are right, and enough cycles are available: "Nothing we know about nature indicates that it stands a level above computational processes" (p. 57). This belief leads to a strong emphasis on modeling coupled with a notion of model building as adaptation. The computer is not just viewed as a number-crunching simulation tool, however. It is also respected as a metaphor in the procedural epistemology sense: a medium that guides the expression and structure of thought. Moreover, because it blurs the line between experiment and theory, allowing scientists to make discoveries that have been relevant to multiple fields, the computer is also seen as reversing the current trend toward fragmentation of the sciences.

The book's target audience is broad, so it has the daunting task of describing this rich collection of themes and topics at a basic level while allowing readers who have more background to skip around effectively. Flake clearly devoted much time and thought to this presentation, and the results are very successful. He has divided the text into seven sections: (1) computation, (2) fractals, (3) chaos, (4) complex systems, and (5) adaptation plus (6) an introduction and (7) an epilogue. Each of the five topical sections contains several chapters—essentially a distillation of the "fun stuff" in the corresponding upper-division college class. The material is made accessible by a lively and effective arsenal of techniques: great figures; delightful quotes from Aristotle, Cole Porter, and everyone in between; useful anecdotes from net news debates on current topics; boxed "digressions" that delve into advanced topics, interesting tangents, and the odd biographical detail; a comprehensive glossary as a safety net; and so on. The examples and figures are particularly well selected and thought out. Discussing the feedback loop between room temperature and body heat, for example, or the fractal properties of the human body are effective ways to bring these concepts home to nonspecialists. The examples are also wonderfully innovative and playful, for example, making the connection between π and the Mandelbrot set, building And-Or-Not gates out of Life glider guns, using cake slicing to define game theory, or drawing a parallel between the pagan festivals coopted by Christian religions and memetic crossover in evolution. Each group of chapters is followed by a postscript, which serves both as a recapitulation for readers who might have missed something and as a capsule summary for experts who are skimming the section. For background, Flake assumes only a basic knowledge of computation; when the discussion calls for other prerequisites, he gives short descriptions (for example, calculus, Lisp, linear algebra, control theory, neo-Darwinism). Although these succinct descriptions are good as reviews, they might be a bit too difficult as first encounters; fortunately, each chapter also has a good, short bibliography. Readers with more background will have no trouble getting right to the interesting parts. The chapters and sections can stand in isolation, and Flake gives clear, in-line guidelines about what can be skipped and under what circumstances. With only a few exceptions—for example, the material about recursively enumerable sets, which the reader skims or skips at his/her later peril—these guidelines work well.

The book's second level addresses its main premise: the interrelationships among its five chosen topics. The text is structured to bring out the natural connections among these areas: a form of book as emergent system, where this particular combination adds up to something even more interesting than the parts alone. Much of this development takes place in the postscripts, where Flake weaves a web of connections between the evolving topics. The second one, for example, uses time-space ideas to make the connection between the computation and fractals sections: A running program and a fractal can be infinitely complex, but the underlying processes (source code and recursion rule, respectively) are quite compact. The key to the complexity and richness, in both cases, is recursion.

The third postscript folds chaos into the mix using the notions of incomputability and strange attractors to connect to computation and fractals, respectively, and so on. Along with this increasingly dense network of connections, the series of postscripts also contains a faint but steadily building thread of speculativeness, leading to the new connections that are one of the contributions of this book. Like all new and exciting things, however, this development is not totally pat and comfortable. By the third postscript, there are some leaps of faith that might make a particularly picky mathematician wince a bit, and the later postscripts take these leaps as givens and edge even further onto the rim of the well established. It will be interesting to see how these connections are fleshed out over the next few years as the fields of chaos and complexity continue to grow. In the meantime, a bit more discussion seems warranted. Leaps of faith are useful things, but they should be identified as such, and the reader should be told how accepted these ideas are in their own communities.

The author is clearly a renaissance man as well as a wonderful teacher. He is equally good at succinct summaries and painting the big picture, and he makes particularly effective use of examples.

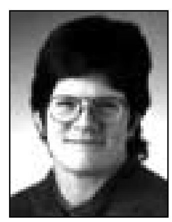
As in any book, there are a few disconnects, muddy spots, and bugs. Some discussion of what the dimension of a system means, as well as the difference between transients and attractors, would clear up chapters 10 and 12, for example, and all chaotic systems most certainly do not have fractal state-space structure (compare Anosov systems on the torus). The path taken into the theoretical computer science material (by way of Gödel) was a little awkward, and the neural net material is split in an odd way: The first chunk turns up in part 4, Complex Systems, and the rest, with some duplication, appears in part 5 under Adaptation. Some topics would benefit from a bit more follow-up, and others—like the lambda calculus and the contravariant basis—could easily be omitted. The notion of a fitness landscape turns up several times, in a somewhat scattered fashion before it all comes together, and the (very important!) notion of local versus global optima and how they affect movement across these landscapes only makes its first appearance on page 419. The reader is also left wondering about how one chooses a neural net architecture for a given problem. To be fair, no one book could possibly address every possible tangent or implication of this rich collection of concepts, and *The Computational Beauty of Nature* in general does a good job. Indeed, if it did not, the reader might not know enough to ask these questions.

The C source code for almost every example and figure in the book is available from a beautifully designed web site along with a variety of excerpts, reviews, addenda, suggestions for lecturers, related links, and other useful materials (including the full bibliography, in BibTeX format): mitpress.mit.edu/books/FLAOH/cbn-html/home.html. Readers who skip prefaces, incidentally, might be somewhat mystified when they first encounter a reference to one of the programs; a pointer in the text to the appropriate appendix would be useful. I had some difficulty installing the source code on my BSDI UNIX box, but the LINUX, SOLARIS, and NT versions are "known to work." Even better, the web site was recently updated to provide JAVA applets for many of the examples, which will make this book even more accessible to students and attractive as a textbook.

The Computational Beauty of Nature is not, however, simply a textbook; rather, it is a labor of love. This unusual book crackles with excitement and runs deep with thoughtfulness. The topics that it covers and the connections that it makes are fascinating, and the presentation is clear and lively. The writing style is precise and evocative: informal without being chatty and enthusiastic without being shallow. It is fun to read, and it will make the reader stop and think, whether he/she is a freshman, an academic, or a professional. I learned from all its parts, even in my own disciplines; I expect to reread it, use it as a reference, use some of the examples in my classes, and recommend it to my colleagues and students.

Acknowledgments

Thanks to B. Chandrasekaran and Andee Rubin for helpful comments.



Elizabeth Bradley received an S.B., an S.M., and a Ph.D. from the Massachusetts Institute of Technology in 1983, 1986, and 1992, respectively, and has been on the faculty of the Department of Computer Science at the University of Colorado since 1993. She is a member of the external faculty of the Santa Fe Institute and the recipient of a National Young Investigator Award, a Packard Fellowship, and the 1999 College of Engineering teaching award. Her research interests include nonlinear dynamics and chaos, scientific computation and AI, network theory and circuit design, and classical mechanics. Her e-mail address is lizb@cs.colorado.edu.