

*Computer Simulations with Mathematica: Explorations in Complex Physical and Biological Systems*, by Richard J. Gaylord and Paul R. Wellin. Springer-Verlag, 1995. \$54.95. 297 pages, CD-ROM.

Computer simulations of natural phenomena have been used for as long as there have been computers. The first Monte Carlo simulations done via a programmable computer (of neutron transport) were performed on ENIAC in the late 1940s. Sometimes, simulations are essential in order to extract predictions from theories that are analytically intractable, as in lattice QCD. In other cases, simulations have provided completely new insights that would not have been readily achieved otherwise. A good example of this is the numerical experiments on ergodicity by Fermi, Pasta and Ulam in the 1950s, which led to the discovery of solitons.

Simulation has become an integral part of the toolkit of computational scientists. In the past 10 years a number of sophisticated interactive mathematical/statistical packages have been developed; Mathematica, Maple and Matlab being the most famous. These packages all integrate high-level (interpreted) programming languages and interactive graphics with large collections of built-in functions, and have made the business of doing and teaching simulation easier than ever.

Using Mathematica to write simulations is the subject of the new book by Richard Gaylord and Paul Wellin, *Computation Simulations with Mathematica: Explorations in Complex Physical and Biological Systems*. The simulations described in this book are latticed based, such as cellular automata (CA) and attempt to model physical systems by using approximate descriptions of the local (“microscopic”) interactions of lattice elements with their neighbors. E.g., to simulate a gas, a direct microscopic approach such as molecular dynamics would involve numerical integration of the coupled time-dependent equations of motion of the particles. In contrast, the lattice gas puts a population of a small number of particle types on a lattice and marches forward in discrete time according to certain collision rules. Mathematica is well suited to handle these lattice-based calculations and the authors have made good use of color graphics and animation to present the results. The book is reasonably priced and comes with a CD-ROM containing hundreds of interesting and well-written (and, more importantly, well-explained) Mathematica programs.

There is much good to be said about Gaylord and Wellin’s book. The range of interesting topics covered is wide and serves to illustrate the gen-

erality of the ideas behind the modern theory of complex systems. The Mathematica codes are clearly explained. The book has a number of useful appendices (amounting to nearly 40% of the book), including an introduction to programming with Mathematica and an appendix by Todd Gayley on how to link C programs. However, I believe that the authors' underlying view of simulation and its relation to science is fundamentally misguided. The net result is that while the book might be an effective tool for teaching certain kinds of Mathematica programming, we do not end up learning much about simulation *per se* or the systems supposedly being studied. Let me quote from Gaylord and Wellin's preface:

The goal of scientific investigation is to understand how nature works. There are two ways to do this: by experimental observation and by theoretical modeling. Modeling has traditionally consisted of thinking up and writing down equations and then solving those equations, either by hand (*e.g.*, theoretical physics) or by computer (*e.g.*, computational physics). While the use of equations has been very useful in opening up the natural world to our understanding, as well as to our control, it has its limitations. The equational approach breaks down when the equations can't be solved (analytically or numerically) because of technical difficulties or when the system being studied cannot be represented in terms of equations.

A new approach to the modeling of natural phenomena does not use equations at all. In *algorithmic physics*, as we call this approach, equations are replaced by algorithms or computer programs. These programs, known as computer simulations, directly model the phenomenon under investigation.

But of course computer programs can be expressed in terms of equations, so writing computer programs does not do away with equations. More importantly, to the extent that the output of any computer program does indeed *simulate* (i.e., resemble) nature quantitatively, it can only be so because the rules used to make the simulation reflect the underlying physical processes, which are themselves expressible mathematically; for example, the conservation of mass and momentum, inter-particle force laws, competition

for resources, etc. In other words, I believe that “equational physics” is identical with physics. To give a simple example, in their discussion of cellular automata, the authors state the following rule for a CA which is supposed to model diffusion: the value of the temperature (or some other quantity) at any lattice point is to be replaced by the average of its value at its neighboring point. But this rule is nothing but a particular explicit finite difference approximation of the diffusion equation; other discretizations would give rise to other valid CA rules. This is not to say that the macroscopic picture is more valid than the microscopic one; after all, the diffusion equation is itself an (occasionally bad) approximation of nature. But what criterion does one use to judge the validity of a simulation? The authors provide no guiding insights into how their simulation rules are intuited. Those algorithms which are not simply discretizations of differential equations have usually been derived by building the simplest models that seem to capture the essential physics of a problem: for instance, Burridge and Knopoff’s slider block model of earthquakes, or the Ising model of ferromagnetism, or the lattice gas model. The lattice gas models the fluctuations of gas molecules on a triangular lattice in discrete time. The particles move about the lattice according to the small number of collision rules that conserve mass and momentum. The original model of Hardy, de Pazzis and Pomeau turned out to be too simple to give realistic macroscopic properties. This defect was remedied by Frisch, Hasslacher and Pomeau, who were able to develop a lattice gas that evolves according to the Navier-Stokes equation in the limit of large lattice size and low velocity. Had FHP not made this breakthrough, would anyone now care about the lattice gas model? After all, the HPP automata makes nice pictures too.

This interplay between abstraction and prediction is the way physics has always been done. The difference now is that we have better computational tools. These more powerful tools are not always a benefit, however, since there is a tendency to use the computer in lieu of thinking: thinking is hard and computer graphics provide instant gratification. Thus, it is especially important in a book intended for non-specialists that the relationship between writing computer programs and simulating nature be put into proper perspective. The simulations described in this book cover a vast spectrum of complex physical processes, from the Ising model to critical sandpiles, forest fires, waves in excitable media, spin glasses, lattice gases and much more. Yet none of these phenomena are ever explained in anything other than the

most heuristic terms. Absent physical explanation of the phenomena, students and others not already familiar will be left with nothing but intuition or the pictures generated by the computer programs to know whether these calculations really do simulate nature or whether they are little more than elaborate graphics.

*Computer Simulations* is at its best, which is very good indeed, when explaining how to program certain kinds of simulations in Mathematica. For people who already understand the physics behind the models used and want to work with small-scale simulations, I highly recommend this book. I would also recommend this book for teachers who are willing to supplement the Mathematica discussions with physical motivation and explanation.

Quibbles. The authors introduce too many terms that are never really used or explained: limit points, limit cycles, strange attractors, universal computers, and so on. Such terms lend the book a certain modern cachet, but the lack of context must be rather frustrating to people who do not already know what these things mean. In addition, the references provided are limited to a relatively small number of recent works. In a book such as this, I believe that a greater awareness of history is desirable. No doubt the authors' citations fit in with their view that *algorithmic physics* is a new field, but somehow I think that people like Fermi, who was doing Monte Carlo simulations by hand 60 years ago, would not be all that surprised at the current state of computer simulation. The Connection Machine was not the world's first parallel computer, as the authors claim. I'm not sure what was, but Burroughs introduced a 1-4 processor MIMD machine in 1962. The ILLIAC-IV and STAR-100 date were built not long after this.

John A. Scales, Department of Geophysics, Colorado School of Mines