regular tetrahedron whose base is the triangle formed by the midpoints of the edges of the face. Repeating this process creates a succession of complicated polyhedra, but astonishingly they converge (except for a set of measure zero) to a perfect cube.

The other half of the book concerns topics from number theory. Prime numbers and the Riemann zeta function are treated with increasing sophistication, and the Prime Number Theorem and related approximations are clearly illustrated. Wagon gives a thorough treatment of some basic algorithms of number theory, such as computing the greatest common divisor and solving equations in integers. One of the most satisfying parts is Chapter 9, in which the reader is introduced to a complex version of the Euclidean algorithm and is systematically led through a fairly elaborate package that efficiently computes all ways to represent a given integer as the sum of two squares. Along the way, the reader is exposed to both interesting mathematical theory as well as a variety of *Mathematica* techniques such as packages and contexts. The book concludes with some spectacular calculations using the Riemann formula to approximate the prime distribution function.

There are extensive references for further reading and detailed indexes for terminology and *Mathematica* objects. More elaborate code is relegated to an appendix, and all code in the book is available from the author (`wagon@macalstr.edu`) on Macintosh disk or via ftp.

Wagon's book is an excellent way to learn the `In`'s and `Out`'s of *Mathematica*. It's also a lot of fun to read.

References

[1936] D. H. Lehmer. 1936. "On the Converse of Fermat's Theorem," *American Math. Monthly* 43, 347–354.

Mr. Wizard's Workshop

Brian Hayes

Exploring Mathematics with Mathematica: Dialogs Concerning Computers and Mathematics by Theodore W. Gray and Jerry Glynn. Addison-Wesley Publishing Co. 535 pp. + CD-ROM disk. $46.25 hardcover; $34.50 paperbound.

Mathematica for the Sciences by Richard E. Crandall. Addison-Wesley Publishing Co. 299 pp. $35.50 hardcover.

I grew up watching Mr. Wizard on TV. Every week some earnest, crewcut youth would come in the back door and confess that he'd been wondering lately about the nature of air pressure or the mechanism of a nuclear reactor. Mr. Wizard would say, "Gee, Bobby, that's a good question," and off they would go into the basement workshop to build a barometer or to model a chain reaction with ping-pong balls and mousetraps. I loved every minute of it. The new book by Theodore W. Gray and Jerry Glynn brings back that sense of a world with no end of wonders waiting to be discovered. For the most part Gray plays Mr. Wizard—even though he represents the younger generation—and Glynn is the eager assistant who never runs out of questions. The basement workshop is *Mathematica*.

The word "exploring" has a legitimate place in their title, for the authors often don't know when they set out where they will wind up. In an early chapter they decide to see what happens when the `Sin` function is iterated; that is, they look at the series of functions `Sin[x]`, `Sin[Sin[x]]`, `Sin[Sin[Sin[x]]]`, etc. The iteration is performed with *Mathematica*'s `Nest` operator, as in `Nest[Sin,x,3]`, which produces the last of the three expressions given above. Thinking about this process, one expects the function values to be monotonically decreasing; for example, `Sin[Pi/2]` is 1, `Sin[1]` is about 0.84, `Sin[0.84]` is 0.74, and so on. It turns out this intuition is correct, but nevertheless a surprising structure emerges when the nesting depth reaches a few hundred or a few thousand. In a graph of the iterated `Sin` function, the amplitude of the sine waves becomes progressively smaller with each iteration, as expected, but the waves also change shape and become square. Even Mr. Wizard admits he had not anticipated this development, and to explain it he has to call on outside help (from Dana Scott and Jerry Keiper).

The reader gets to explore, too. One obvious question that is left open for the curious reader is what happens when the `Cos` function is iterated instead of `Sin`. In one sense the result is less dramatic: Those strange square waves do not appear. Instead, the iterated `Cos` function quickly reaches a fixed point for all argument values. Still, the way in which the function approaches that fixed point offers at least a small reward for the seeker after mathematical trophies: An animation of successive graphs looks for all the world like the damped vibration of a plucked string.

The third obvious permutation—replacing `Sin` or `Cos` with `Tan`—leads to graphs that can best be described as a numerical mess. Regions where the function value is unbounded seem to introduce noise, which eventually takes over the entire graph.

Any discussion of iterated functions leads inevitably to the subject of iterated quadratic mappings, period-doubling, deterministic chaos, and all the boisterous activity that has surrounded these ideas over the past decade. Gray and Glynn do not neglect the area. A chapter gives the expected pictures of strange attractors and bifurcation diagrams for the logistic function `lambda x (1 - x)`, accompanied by mock-expasperated apologies for the overfamiliarity of all this material. The chapter goes on to describe a technique for identifying points of bifurcation along the path to chaos, a technique that few readers will find overfamiliar. (It was totally new to me.) The key idea is to use *Mathematica*'s built-in `Plot` procedure as a means of auto-

---

Brian Hayes is editor of American Scientist; *he has written on aspects of computing for various other publications, including* Scientific American, The Sciences *and* Pixel.

matically narrowing the search for a bifurcation point; `Plot` is set up to draw a smooth curve whenever possible, and it will recursively subdivide an interval in an attempt to eliminate discontinuities; this is just what is needed in an efficient search for the critical transitions of an iterated mapping.

In this case too, being able to manipulate and recalculate the graphs is tremendously helpful in understanding their structure. As Jerry Glynn says of a bifurcation diagram: "I agree that these pictures have become commonplace but, now that we have them under our control, I have many more questions and interests than I did before. When the pictures sat in books... I reacted in a more passive way than I do now that they are in electronic form. I'm sure I'm not the only person who will react this way—I'm sure readers of this book will want to zoom in on their own favorite areas."

The CD-ROM that makes such zooming convenient is more than a software supplement to *Exploring Mathematics with Mathematica*. It includes the entire contents of the book, and not merely as static text and pictures, but as a sequence of *Mathematica* Notebooks. In fact, the CD-ROM includes two full copies of the book, one copy meant for machines with a color monitor and the other for black-and-white computers. Reading the book as a "live" document—where you can double-click to animate a series of graphs, or where you can alter an expression, changing a `Sin` to a `Cos`, and immediately see the result—is an unnerving experience. I find it exciting, but I also find that it challenges some taboo, doubtless acquired in early childhood, that forbids defacing the received text. The fact that the CD-ROM itself cannot be altered is helpful in this respect; and no doubt future generations will have lost my quaint inhibitions. In the end I read about half the book from the disk and half from the printed version; I might have done without the printed book altogether if it weren't so awkward taking a computer to bed.

Even if you do not have a CD-ROM drive available, the disk is not useless. All of the sounds from the *Mathematica* Notebooks are provided as separate audio tracks that can be reproduced through an ordinary audio CD player. Sounds? Yes, *Exploring Mathematics with Mathematica* is haunted by an abundance of beeps and gurgles, buzzes, whoops and warbles. (Playing them from within the Notebooks, rather than from the separate audio tracks, requires Version 2.0 of *Mathematica*.) It seems we have already reached that point in the history of the world where the mere visualization of mathematical or scientific ideas has lost is novelty; now we must hear our data as well.

We all know what a sine wave sounds like, but until this small silver disk entered my life I had never heard the chiming of a Bessel function or the ethereal whistling of an Airy function. For some more elaborate music, Gray and Glynn and their "visiting professor" Dan Grayson construct a very plausible aleatoric composition by converting to sounds the sequences of integers produced by the Collatz mapping (in which `x -> x/2` for even `x` and `x -> 3x + 1` for odd `x`). The music has a steady beat and a mysterious melodic structure, in which delicate peeps and tweets are occasionally punctuated by the rude blatt of a bassoon. As Glynn says: "It's very attractive, in an integer sort of a way." For a finale to the chapters on sound, Gray and Glynn borrow a recording of Beethoven's Ninth Symphony and digitize a few seconds of the choral fantasy, producing a list of some 28,000 numbers. Then, in a process akin to the Gödel numbering of theorems, they calculate a single number that embodies all the information captured in the digitized passage. The number begins 0.50585... and in very small type goes on for more than three pages. The authors thereupon remark:

> Jerry: Do you really expect me to believe that this number, just a bit more than 1/2, is a piece of the glorious Ninth Symphony? I have a hard time adjusting to the notion that something so rich and multifaceted could be reduced to a fraction.
>
> Theo: Well now, I wouldn't be so quick to conclude that it's a fraction (rational). It might be irrational. Many great artists are...

After failing to come to agreement on the question of Beethoven's rationality, they finally supply this vast number as an argument to a function named `playFloatingPointNumber`. What comes out of the speaker is Beethoven again, seemingly unharmed by his passage through the world of mathematical abstraction. But there are limits. The square root of the Ninth Symphony, it turns out, is mere noise.

Scientific Computing

Richard E. Crandall's *Mathematica for the Sciences* lacks the multimedia showmanship of Gray and Glynn, but it is full of engaging material for those who read the old-fashioned way. Here we get *Mathematica* at work, whereas much of Gray and Glynn can be read as *Mathematica* at play.

The contrast between the two approaches is particularly clear in those areas where the books take up similar subject matter. Both of them, for example, discuss primality and the factoring of integers. Gray and Glynn rely on the built-in facilities `Prime`, `PrimeQ` and `FactorInteger`, and they draw a few graphs. Crandall, in contrast, implements a series of recent, industrial-strength algorithms for those who are looking to factor some *really* big numbers. The code is given in full; the explanations are spare.

Crandall's book has a much broader scope, encompassing all of the sciences (and a bit of engineering as well) rather than just mathematics. The topics include classical and quantum mechanics, special and general relativity, solitons, the dynamics of chemical reactions, population biology, the propagation of nerve impulses, and electronic circuit analysis. In addition to all this, there is after all a fair amount of pure mathematics, including chapters on identities and expansions and on real and complex analysis. Finally, there is a good deal of practical and methodological advice. For example, Crandall gives an extended treatment of signal processing in *Mathematica*, with applications to speech recognition and the enhancement and analysis of bitmapped images; I would never have guessed that either of these things could be done without special-purpose software. (As far as image processing is concerned, the key fact I had missed is that `DensityPlot` can be used to display a grayscale bitmap.)

One of Crandall's projects that particularly attracted my attention is his account of signal propagation in nerve fibers, as described by the Hodgkin-Huxley equation. Some sciences are founded on equations of great simplicity and elegance, but not neurobiology; the Hodgkin-Huxley equation is a dreadful kludge, with more than a dozen empirically determined constants, coefficients and functions, and with a structure that precludes straightforward analytic solution. Merely looking at the equation does not offer a whole lot of intuition about what kind of behavior it predicts. This very opacity, however, makes the equation a good candidate for simulation and analysis with *Mathematica.* Crandall's implementation of the Hodgkin-Huxley model is based on the idea of searching for an impulse-transmission speed that yields an action potential with a stable and plausible shape. The method works, and it is fascinating to watch a long series of graphs gradually evolve toward the familiar double-peaked textbook waveform. Experimentation would be more fun, however, if the process did not require an overnight run to converge.

Crandall's final chapter takes up three "great problems of history": Fermat's Last Theorem, the Riemann zeta function, and theories of gravitation. Even with the help of *Mathematica*, Crandall does not find a proof (or disproof!) of Fermat's famous assertion about Diophantine equations, but he certainly does some heavy-duty work on the problem. For example, using a method developed in *Mathematica* but subsequently rewritten as a C program, he has demonstrated that $x^p + y^p = z^p$ has no integer solutions for all prime exponents p less than 1,000,000. The treatments of the zeta function and gravitation focus more on expository issues than on pushing back the frontiers of new knowledge. A particularly neat trick is plotting the orbit of Mercury, as predicted by general relativity, in a solar system where the mass of the sun has been increased by a factor of a million; the precession of the perihelion points is immediately visible.

The authors of both of these books bring with them a great deal of expertise in *Mathematica*, along with a certain amount of insider knowledge. Theodore W. Gray is the principal architect of the Macintosh front end for *Mathematica*, and he designed the Notebook system common to several versions. Jerry Glynn is a noted mathematics teacher—he was Gray's teacher for a time—whose involvement with computers in the teaching of mathematics goes back to the PLATO project at the University of Illinois almost 20 years ago. Richard E. Crandall teaches at Reed College, where he initiated a program introducing computing into the liberal arts curriculum, and he is also director of scientific computation at NeXT, Inc.

Recreating Number Theory

A. L. Szilard

Computational Recreations in Mathematica by Ilan Vardi. Addison-Wesley Publishing Co. 304 pp. $29.95.

For those who think that serious mathematics is only for hard-nosed practitioners and that computers are for humorless nerds, Ilan Vardi's *Computational Recreations in Mathematica* should come as a pleasant surprise. Using the potentially infinite precision of *Mathematica*, the book reveals the exquisite secrets of computing number-theoretic functions, which will delight a wide mathematical readership, from amateurs to specialists.

The author uses the term computational recreations in the same sense that an endurance swimmer might call a breaststroke passage across the Nile recreational swimming after completing a triple crossing of the English Channel in a full-stroke butterfly.

Vardi assumes that the reader is interested in computational experiments in combinatorial number theory, is at least familiar with LISP and C as well as competent in *Mathematica*, has a copy of *Concrete Mathematics* by Graham, Knuth and Patashnik, and has access to *Mathematica* Version 2.0 on a high-performance computing platform. This last is needed because the solutions of some moderately complex computational problems presented in this book might take weeks to run on even a RISC-based workstation.

The ten loosely autonomous chapters of the book are enhanced with problems and exercises whose solutions can be found in the text or in an appendix. A second appendix collects over 70 *Mathematica* functions used throughout the text. The material is well researched, with a bibliography of 136 references.

Each chapter is graced with a work of graphic art created by Scott Kim, who appropriately used the graphics tools in *Mathematica*. However, this is the only window the book opens to *Mathematica*'s graphics universe.

Though several entangled golden threads are woven into it, *Computational Recreations* does not adhere to one central theme. The carefully crafted pieces in this kaleidoscope do not always come together to form a cohesive image. The book does, however, explore the tradeoff between elegance and efficiency, and celebrates the occasions when one yields the other. The author advocates a functional style of programming and subliminally promotes his conviction that serious researchers in number theory and combinatorics can no longer ignore the advanced mathematical tools available in computer science.

Written in the style of an advanced textbook, *Computational Recreations* begins by demonstrating its dedication to elegant programming. Readers are asked to program five easy pieces: a function for run-length encoding; one for generating the list of all sublists of a given repetition-free list; functions for evaluating x^n and $x^n(\mathrm{mod}\, n)$ efficiently; and a function to generate the Conway sequence, 3, 13, 1113, 3113, 132113, 1113122113 ...

Vardi explains his succinct solutions (see *The Mathematica Journal*, Winter 1991) starting with the innermost functions and then tracing the results through successive function compositions. Although it is clear that the programs solve the problems, it is not always clear how to adapt them to problems that are just a bit different from the ones discussed. For example, to generate the subsets in Gray code order—an order in which consecutive subsets differ in exactly one element—an elegant functional program should return `{{}, {a}, {a,b}, {b}, {b,c}, {a,b,c}, {a,c}, {c}}` for `GraySubsets[{a,b,c}]`. This should not be too difficult, but the author claims otherwise. He proposes two solutions, a

A. L. Szilard is professor of computer science at the University of Western Ontario.