traced Conway's discovery of three new simple groups [2] from Leech's work on sphere packings [4]. Leech's work was related to Golay's (23, 12) 3-error correcting code [3]. Thompson uses a hands-on approach, and assumes that the reader has had advanced calculus and a first course in algebra. The book is also unusually clear, because one of Thompson's main goals is to give the evolution of the mathematics.

It would be a lively choice for an upper level topics course. There are several interesting historical observations. Here are two facts that the reviewer did not know. It was Cocke [1], not Hamming or Golay, who found the infinite family of 1-error correcting codes over a general finite field $GF(q)$ (the so-called Hamming 1-codes). Bell Labs was able to patent (in 1951) Hamming's original (7, 4) 1-error correcting code. This led to a delay in its publication which caused a priority dispute.

DENNIS STANTON

School of Mathematics
University of Minnesota
Minneapolis, Minnesota 55455

1. J. COCKE, "Lossless symbol coding with nonprimes," *IRE(IEEE) Trans. Inform. Theory*, v. 5, 1959, pp. 33–34.

2. J. CONWAY, "A perfect group of order 8,315,553,613,086,720,000 and the sporadic simple groups," *Proc. Nat. Acad. Sci. U.S.A.*, v. 61, 1968, pp. 398–400.

3. M. GOLAY, "Notes on digital coding," *Proc. IRE(IEEE)*, v. 37, 1949, p. 657.

4. J. LEECH, "Some sphere packings in higher space," *Canad. J. Math.*, v. 16, 1964, pp. 657–682.

27[65-01, 65-04].—WEBB MILLER, *The Engineering of Numerical Software*, Prentice-Hall Series in Computational Mathematics, Prentice-Hall, Englewood Cliffs, N. J., 1984, viii + 167 pp., $23\frac{1}{2}$ cm. Price \$27.95.

This book is primarily a textbook suitable for the senior undergraduate or first-year graduate level. It ought to appeal to a greater audience, however: anyone likely to develop or use computer programs for serious scientific computation. Thus, it should interest engineers, mathematicians, and scientists, as well as computer scientists.

The author presents material related to the production and testing of numerical software that has never before been gathered together. In the Preface he states, "My goal is to present principles for writing numerical software. The ideal textbook about the production of numerical software remains to be written, but I hope that I have verified its worth and feasibility and hastened its arrival." I believe that the author has succeeded admirably in verifying worth and feasibility. While still not the ideal textbook, this book is a valuable first effort to organize and codify principles and concepts that have hitherto only been found scattered through the literature. The text is supplemented with exercises and programming assignments, some quite challenging, designed to enhance the reader's understanding of fundamental issues.

The book contains six chapters. Chapter 1 introduces terminology and illustrates concepts that will be used throughout the book. The distinction between similar terms, particularly those related to programming "mistakes" of various kinds, is sometimes subtle. Fortunately, examples in later chapters make the distinctions, and the reasons for them, understandable.

Chapter 2 is an introduction to those details of floating-point arithmetic necessary for understanding the design and testing of numerical software. Without introducing unnecessary complications, the author outlines the important differences between floating-point arithmetic systems and the mathematical real number system. He introduces and briefly discusses parameters that characterize the former, and surveys various schemes for making these parameters available. The chapter concludes with a lucid discussion of uncertainty diagrams and their use in approximate error analysis.

Chapters 3 through 6 proceed in an orderly way from discussions of software whose behavior can be completely analyzed, and is therefore well understood, to discussions of software whose behavior can only be discovered empirically. Chapter 3 discusses the design and testing of software for the sine function, drawing heavily from material in [1]. The author takes time to explore the nuances and numerics of critical algorithmic details, using the material to illustrate broad principles of software design and implementation.

In Chapter 4, software for the solution of linear equations forms the necessary backdrop for an escalated discussion of testing procedures. Early sections of the chapter concentrate on general implementation issues for linear algebra software, using material from [2], and describe three well-known algorithms that provide fodder for the subsequent discussion of testing methodology. The main topic is the reliability of testing as a means for determining whether or not a program meets design specifications. The chapter concludes with the important point that an algorithm may be a practical success even though theoretically it is unreliable. Only extremely sophisticated tests will detect the unreliability in such cases.

Chapter 5, dealing with software for the solution of a nonlinear equation, concentrates on the principles of measuring the performance of, as opposed to merely testing, numerical software. Loosely speaking, the distinction is that testing determines compliance with a specification, while performance measurements are descriptive. It is shown, however, that the results of carefully selected performance measurements can suggest improvements to an algorithm. The discussion revolves around methods for root determination based on bisection and linear interpolation. Curiously, nowhere in the discussion of bisection is a continuity condition imposed. Thus, the bisection scheme described can converge on a binary machine to consecutive floating-point arguments that bracket a singularity of the function. It is not clear whether this omission is deliberate or an oversight.

Chapter 6 discusses performance measurement in greater detail in the context of software for automatic quadrature. The algorithm discussed is a simple automatic scheme using Simpson's rule. This chapter is the least satisfying of those in the book, because it contains little that is definitive. In that sense, it faithfully reflects its subject. Good algorithms and good software for automatic quadrature exist, but all are demonstrably fallible. Software that sparkles on one integrand will fail with a slightly perturbed integrand. Because acceptable performance specifications do not exist, it is not possible to conduct meaningful tests of such software. Thus, we are reduced to amassing data from experiments in an attempt to determine performance characteristics. The proper design of such experiments, and proper data reduction techniques are research areas. The author presents a clear picture of how little we know about these matters.

In summary, this book is a useful, understandable introduction to the engineering and testing of numerical software that faithfully and fairly reflects the present status of the field. While it necessarily includes some numerical analysis, it is not a numerical analysis text. I recommend it to anyone either interested in working on numerical software or simply curious about what is going on.

W. J. CODY

Mathematics and Computer Science Division
Argonne National Laboratory
Argonne, Illinois 60439

1. W. J. CODY & W. WAITE, *Software Manual for the Elementary Functions*, Prentice-Hall, Englewood Cliffs, N. J., 1980.

2. J. J. DONGARRA, J. R. BUNCH, C. B. MOLER & G. W. STEWART, *LINPACK Users' Guide*, SIAM, Philadelphia, Pa., 1979.

28[60K05, 60K10, 65D07, 65D20, 65D30].—LAURENCE A. BAXTER, ERNEST M. SCHEUER, WALLACE R. BLISCHKE & DENIS J. MCCONALOGUE, *Renewal Tables: Tables of Functions Arising in Renewal Theory*, Technical Report, Graduate School of Business Administration, University of Southern California; 22 pages of typewritten text + 312 pages of tables xeroxed and reduced from computer printout sheets, deposited in the UMT file.

Let $\{N(t),\ t \geqslant 0\}$ denote an ordinary renewal process with inter-renewal distribution function F. In many applications of renewal theory, knowledge of the renewal function

$$H(t) = E[N(t)] = \sum_{n=1}^{\infty} F^{(n)}(t), \tag{1}$$

the variance function

$$V(t) = \operatorname{var}[N(t)] = 2H^{(2)}(t) + H(t) - [H(t)]^2, \tag{2}$$

and $\int_0^t H(u)\,du$, where $P^{(n)}$ denotes the n-fold recursive Stieltjes convolution of P, are required. With the exception of the Poisson process, exact expressions do not usually exist and numerical evaluation is quite difficult [2] so, other than the partial tabulations of Soland [6] and White [7], numerical values are not readily available.

The Cléroux-McConalogue cubic spline algorithm [3], [4] partially resolves the numerical problems; this algorithm generates very accurate piecewise polynomial approximations to convolutions of the form $F^{(n)}(t)$ where $F \in C^2[0, \infty)$ is a distribution function whose density is bounded. McConalogue [5] (see also [1]) generalized this algorithm, permitting its application to a subclass of those distribution functions F for which F' exhibits a singularity at the origin.

The generalized algorithm was used to compute $H(t)$, $V(t)$ and $\int_0^t H(u)\,du$ for $t = 0(.05)20$ for the five probability distributions most commonly encountered in applications of renewal theory: the Weibull, gamma, lognormal, inverse Gaussian, and truncated normal distributions. Each of these was tabulated to 4 decimal places