flow charts adequate for programming are supplied with the discussion of particular numerical methods. With each topic, there is, in general, a good discussion of error. The exercises supplied are aptly chosen and adequate to check the student's understanding of the material. Answers for computational exercises are supplied. Briefly, the topics covered by this text are: an introduction to machine computation, with a discussion of problem formulation and flow charting; iterative methods for solving functional equations, with emphasis on Newton's method for finding real and complex roots of an equation; an introduction to matrices and linear equations (both iterative and elimination methods are presented); the characteristic value problem for matrices, with a discussion of the method of iteration for finding characteristic roots and vectors; interpolation, using the Aitken-Neville, and Lagrange methods; the Weierstrass theorem, and Bernstein polynomials as an approximation device; numerical differentiation and integration; a thorough discussion of Simpson's rule and the trapezoidal rule; solution of ordinary differential equations by Euler's and Heun's methods; and an introduction to difference equations.

The book is remarkably free of typographical errors. However, on page 47 the third component of the vector should be -9, not -20.

The author indicates that he is defining *analytic* on page 20; however, the correct definition is given as a footnote on page 30. The definition of *inner product* on page 44 could be made a bit more explicit. Since the Bernstein polynomials are discussed as an approximating device, their shortcomings in this role should be indicated.

All in all, the reviewer believes this to be a good text for use at the sophomore or junior level. It has many things to recommend it, including lucid presentation and good selection of topics. This text is aimed at an understanding of numerical analysis rather than a proficiency in problem solving. In developing a curriculum for the student who wishes to become a numerical analyst, it seems that two alternatives present themselves. They are either the inauguration at the sophomore or junior level of a course using a text similar to the one being reviewed or the inclusion of topics presented in this text in existing courses such as calculus, matrix theory, and differential equations.

James E. Scroggs

The University of Texas
Austin, Texas

32 [X].—K. A. Redish, *An Introduction to Computational Methods*, John Wiley & Sons, Inc., New York, New York, 1962, xii + 211 p., 25 cm. Price $5.75.

The intended readers of this book are, in the author's words, the "occasional" computer and students of science and engineering. The tools assumed available are tables and, preferably, a desk calculator. The knowledge assumed is that of a sophomore at college, although more advanced terms are used occasionally. Matrix notation is used only briefly, and familiarity with it is not necessary.

Someone approaching this subject for the first time could teach himself from this book, for it is admirably clear in style and nearly every method is illustrated by several worked out examples designed to cover various cases which can occur. The procedures chosen for discussion are those which are the simplest mathematically, not necessarily the most economical computationally. Emphasis is laid on always trying to estimate accuracy and on using a helpful layout. Actually, only a small part of the book would be irrelevant to someone with access to a high-speed automatic computer.

Problems of varying difficulty, taken largely from the London University examinations, follow each chapter, and answers are given where appropriate.

Chapter headings, with the methods chosen for explanation, are: Linear Algebraic Equations (elimination with row interchanges, attainable accuracy when the data are approximate, iteration on the residuals, Gauss-Seidel and relaxation methods); Non-linear Algebraic Equations (graphs, regula falsi, Newton and Graeffe methods); Finite Differences (the basic difference operators, propagation of table errors); Interpolation, Differentiation, and Integration (usual basic formulas, preparation of tables, inverse interpolation); Ordinary Differential Equations (graphs; series; Picard, predictor-corrector and Fox-Goodwin methods; applications to linear, non-linear, first- and second-order, initial-value and boundary-value problems); Functions of Two Variables (basic formulas, partial differential operators, relaxation); Miscellanea (brief notes on: approximating functions, difference equations, Gauss integration formulas, Lagrangian interpolation, eigenvalues and vectors, Runge-Kutta methods, and summation of series).

B. N. Parlett

New York University
New York 3, New York

33 [X, Z].—Bruce W. Arden, *An Introduction to Digital Computing*, Addison-Wesley Publishing Co., Inc., Reading, Massachusetts, 1963, ix + 389 p., 23 cm. Price $8.75.

As the author states in his preface, this is a text for an undergraduate introductory course in digital computing techniques. Integral calculus is the only prerequisite.

The first third of the book introduces the student to computer programming, the basic structure of digital computers and number systems. In his treatment of computer programming, the author uses the MAD language as an example and carefully guides the student through the various types of statements which go into a computer program. Use of flow charts is illustrated and machine language is discussed briefly. Arithmetic operations, scaling, and rounding are also presented in introductory fashion.

A little more than a third of the book is devoted to numerical methods. The topics include finite differences, interpolation, numerical integration, the solution of linear algebraic equations, least-squares approximation, and the solution of ordinary differential equations. In most instances the methods are presented with accompanying programs and flow charts, but with little or no analysis.

The final sixty pages deal with non-numerical problems such as sorting, compilers, and formal differentiation.

E. K. Blum

Computer Laboratory
Wesleyan University
Middletown, Connecticut