differential, partial differential, and integral equations, along with minimization and variational problems. This section gives the motivation for studying the subject.

Section 2—Linear Algebra—and

Section 3—Analysis—good summaries of multivariate calculus and linear algebra, with references to the basic texts.

The remainder of the book is now developed within the background of Part I. A listing of the remaining section headings may suffice to indicate the scope of the book:

Part II—Nonconstructive Existence Theorems,
Section 4—Gradient Mappings and Minimization,
Section 5—Contractions and the Continuation Property,
Section 6—The Degree of a Mapping;

Part III—Iterative Methods,
Section 7—General Iterative Methods,
Section 8—Minimization Methods;

Part IV—Local Convergence,
Section 9—Rates of Convergence—General,
Section 10—One-Step Stationary Methods,
Section 11—Multistep Methods and Additional One-Step Methods;

Part V—Semilocal and Global Convergence,
Section 12—Contractions and Nonlinear Majorants,
Section 13—Convergence under Partial Ordering,
Section 14—Convergence of Minimization Methods.

An analysis of asymptotic convergence rates and appropriate indications of how to use the general theory in the presence of round-off errors are given. The book is not only well conceived mathematically, but it is beautifully written and "as self-contained as possible".

E. I.

19[2.45, 2.55, 12].—John K. Rice & John R. Rice, *Introduction to Computer Science*, Holt, Rinehart & Winston, Inc., New York, 1969, xv + 463 pp., 24 cm. Price $12.95.

Among those concerned with teaching computer science, there is a view, currently in the ascendant, that one ought to teach principles of algorithm construction rather than mere programming technique. The authors of this book hold this view, and have ably constructed a textbook, based upon it, for a first course in computer science. They have written a textbook as textbooks ought to be written: clear prose, ample and relevant exercises, clean organization, but with a great deal of inherent flexibility, and pleasing typography and layout, with two-color printing. It is a triumph of the pedagogue's craft.

Nevertheless, I disagree with the basic principle upon which this book is constructed: that one can, in a meaningful sense, study algorithms outside of the context of a programming language. The difficulty, as this reviewer sees it, is that only the necessity of writing a program for an algorithm can force one to define that algorithm rigidly. Any abstract notion of what an algorithm is implies an underlying notion of what a machine is, be it a Turing machine, an Algol machine, or an IBM 360/65. One can exhort the student to be more specific in reducing a vaguely specified computational process to an algorithm; but the only way to answer the question "Have I got an algorithm?" is to try to write a program in some specific programming language. Furthermore, many of the more vexing problems in algorithm construction are concerned with representational problems, and these are easily glossed over in the absence of the demand that a program be written.

The book consists of eight chapters plus three appendices: one on ALGOL, one on FORTRAN, and one on Programs, Compilers, and Systems. The eight chapters are:

1. The Formulation of Problems.
2. The Structure of Algorithms.
3. Languages for Algorithms.
4. Computer Organization.
5. The Construction of Familiar Algorithms.
6. The Representation of Information.
7. The Classification of Solution Methods.
8. The Nature of Errors and Uncertainty.

In keeping with their philosophy, the authors place all material concerning specific programming languages into the Fortran and Algol appendices, and the body of the book is almost entirely language independent. In Chapter 3, they introduce a "natural programming language" which introduces a number of notational conveniences, including normal mathematical symbolism. This natural programming language is not rigidly defined; rather, it is intended for communication with the reader, and according to the authors:

> The natural programming language used in this book is primarily intended for communication with the reader. The criterion for acceptability of a statement in this language is: Can a person understand what is meant? In particular, the reader should not attempt to learn this language in the way that he would learn algorithmic languages like Algol and Fortran.

Most algorithms given later in the book are given in natural programming language; often they are also given in Fortran and Algol.

There are places where, I feel, some distortion has crept into the material because of the authors' conception of it. For example, the first chapter is on problem formulation. The general procedure suggested by the authors is to form a model, get some answers from it, and if the answers are not reasonable, modify the model. This procedure has two very dubious underlying assumptions: that any problem can be modelled by simple equations, and that reasonable answers are necessarily correct answers. A specific example is given: how long should one keep a car before trading it in in order to minimize average annual cost. After many arbitrary assumptions, e.g., that the depreciation in a given year is 30% of the value at the beginning of the year, a conclusion is reached: that "Big Shot" should trade in his car after 2 years, that "Average Joe" can trade it in anytime after 3 years, and that "Penny-Pincher" should keep it for 12 years. Only linear equations are used, and constants (like the 30%) are chosen at will. The results are accepted because they are reasonable. And one of the exercises, requiring an analysis of this sort, is:

> You are a girl's advisor in a major college. A young lady comes to you for advice about the field in which she should major. Work out a schedule.

I submit that examples such as these do not lead to good judgment about when mathematical models should be used.

The book contains some excellent chapters on matters not usually treated adequately. For instance, there is a chapter on the representation of information, which shows the student how to encode information in a form in which a computer can deal with it. The problem is treated on several levels: the physical representation of information, the representation of alphabetic information numerically, the use of arrays and lists, the representation of graphical relationships, and the Polish prefix representation of algebraic expressions. There is also a chapter on errors: where they come from, how to classify them, and how to safeguard (as far as possible) against them. More generally, the book contains a great deal of material designed to give the student a "world view" of what is going on in the computer field.

In summary, this is an excellent textbook for those instructors who are in sympathy with the philosophy that a first computer science course should teach students about algorithms rather than about programming. I happen not to be in sympathy with that philosophy.

Paul W. Abrahams

Courant Institute of Mathematical Sciences
New York University
New York, New York 10012

20[4, 5, 13.15].—John R. Radbill & Gary A. McCue, *Quasilinearization and Nonlinear Problems in Fluid and Orbital Mechanics*, American Elsevier Publishing Co., Inc., New York, 1970, xxiii + 228 pp., 24 cm. Price $14.00.

In Chapter 1 of this book the authors summarize some elementary results on ordinary differential equations. In Chapter 2 they present the "quasilinearization method," which, as far as can be determined from this book, is, in fact, Newton's method. The discussion of the validity and applicability of this method is best described as minimal. Chapters 3 through 9 present a number of applications, drawn mainly from the theory of hydrodynamic stability and boundary layer theory, but including also the computation of electrostatic probe characteristics (Chapter 5) and optimum orbital transfer with "bang bang" control (Chapter 8). A computer program is given in Chapter 9. Some of these problems are difficult and important, but the presentation is too sketchy to be understood without extensive prior knowledge. One cannot make sense of, say, the Orr-Sommerfeld equation, with so few mathematical tools. The level of the mathematical discussion throughout the book is extremely low, and this is particularly true of the numerical aspects which