book to read and study, and it offers many varied illustrations to excite the eye and challenge the mind.

Introduction to Numerical Computation in Pascal

by P. M. Dew and K. R. James
Published by Springer-Verlag, New York, Inc.
$16.00, 291 pages
Reviewed by Robert Ashworth

The first part of this book focuses on Pascal with an emphasis on the fundamental techniques needed to exploit the modular structure of the language. The authors have developed very readable code with particularly clear descriptions to show the results of quality mathematical software.

The central theme is a description of "mathlib" which is a collection of routines available from the Editor of the Computer Science series of the publisher. These run under UCSD Pascal on the Apple II microcomputer. A ten-page summary of all the routines is given in the appendix.

The authors take an algorithm or procedure for the solution of a given problem and then show, step-by-step, how to make it more comprehensive, accurate, or efficient. Stress is placed on: stability, i.e., the algorithm is not sensitive to rounding errors; accuracy, i.e., the algorithm performs according to the documentation; efficiency, i.e., the algorithm is a function of time and memory required; robustness, i.e., the algorithm solves correctly the problem it was intended for; adaptability, i.e., the ability to move a program from one computer to another with a minimum of modification; error control, i.e., the need to have results correct to a specified number of decimal places, and the respective mathematical method of analysis. Part I concludes with a detailed discussion of errors and how to manage them on your computer.

Part II explores the development of mathematical software. Different methods are given for the solution of nonlinear equations in one variable. The study of systems of linear equations is followed by different methods and an analysis of the strengths of each approach. Finally, the last two chapters deal with the fixed-point rules and the adaptive methods for numerical integration.

The text meets the authors' goal of covering core material in numerical analysis and methods for producing reliable mathematical software. Solutions are given for all exercises and four helpful appendices complement the well-designed text. It should be given a high priority for Pascal programmers.

Automation of Reasoning

Classical papers on computational logic, Vol. 1, 1957-1966; Vol. 2, 1967-1970
Edited by Jorg Siekmann and Graham Wilson
Published by Springer-Verlag: Berlin, Heidelberg, New York, 1983
$35.00 Vol. 1: 525 pages, Vol. 2: 637 pages.
Reviewed by Jay Halcomb

These are the first two volumes of a planned series devoted to automated theorem proving. The volumes collectively contain 61 articles discussing mechanical theorem proving, which has applications in automated program verification, program synthesis, and deductive data bases. The papers are of the highest quality, selected by an international committee of researchers in the field, and include translations of previously untranslated literature.

On the whole the papers are focused in scope, approaching computational logic in its strictest sense: the application of theories of first-order deduction and semantic methods to proof of mathematical or logical theorems. A few of the papers allude to possible wider applications, such as generalized problem solving and program verification, but the core concern is with theorem proving in logic and mathematics. In this sense the title and cover blurb are slightly misleading. However, for readers with professional or avocational interests in this field, the book is an excellent unified reference source.

The volumes contain three historical articles especially useful to the new student or casual onlooker:

Wos and Henkin: "Automated theorem proving, 1965-70"

Davis: "The prehistory and early history of automated deduction"

Maslov, Mints, and Orevkov: "Mechanical proof search and the theory of logical deduction in the USSR"

Background

The philosopher G. Leibnitz was perhaps the first to seriously envisage the possibility of algorithmizing and even mechanizing human reasoning. He proposed a *calculus ratiocinator* and *lingua characteristica* "to bring under mathematical laws human reasoning, which is the most excellent and useful thing we have . . . the mind will be freed from having to think directly of things themselves, and