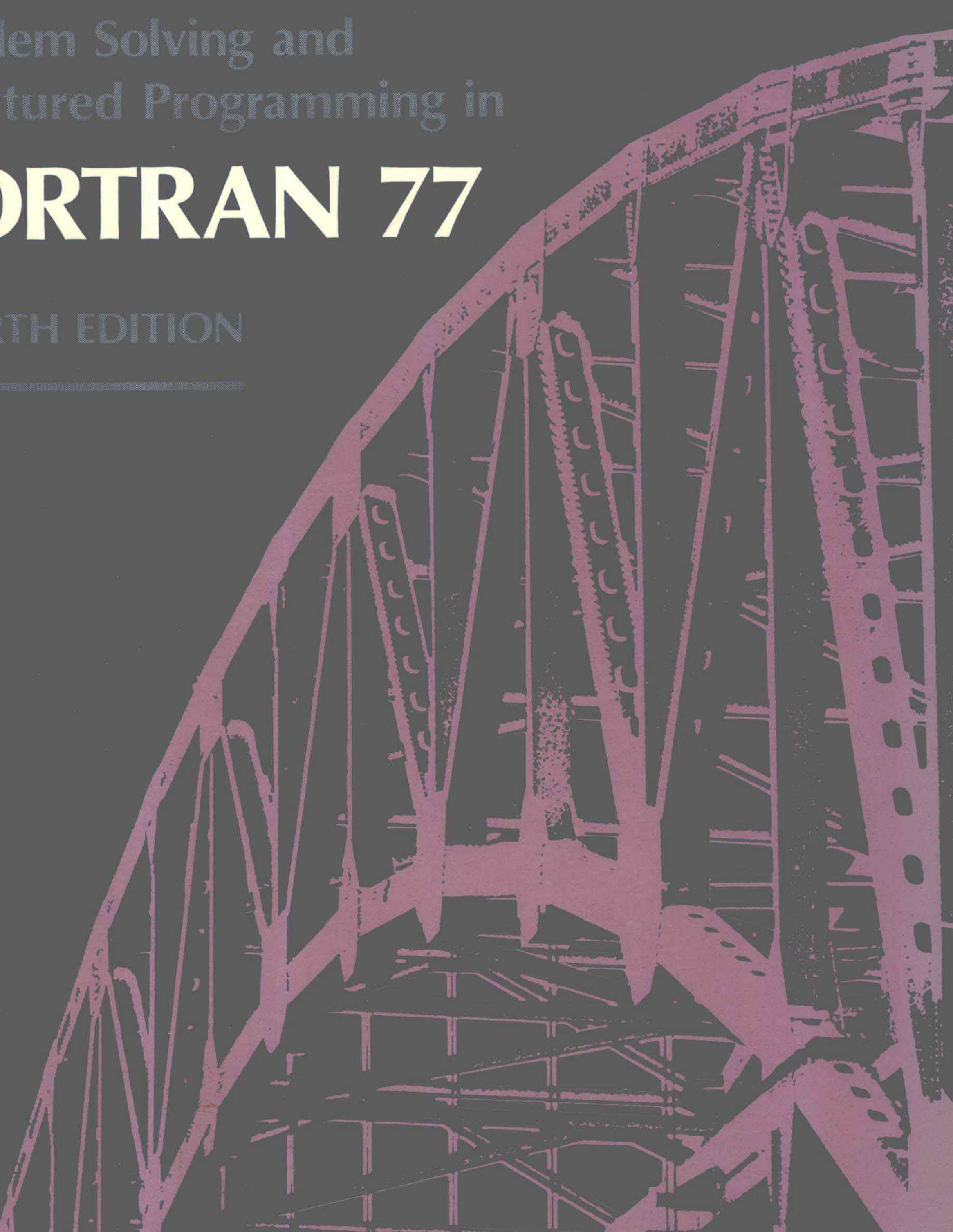
KOFFMAN / FRIEDMAN

Problem Solving and
Structured Programming in

FORTRAN 77

FOURTH EDITION



Problem Solving and Structured Programming in FORTRAN 77

FOURTH EDITION

Elliot B. Koffman
Frank L. Friedman

Temple University

江苏工业学院图书馆
藏书章

Addison-Wesley Publishing Company, Inc.
Reading, Massachusetts   Menlo Park, California   New York
Don Mills, Ontario   Wokingham, England   Amsterdam   Bonn
Sydney   Singapore   Tokyo   Madrid   San Juan

Keith Wollman, Sponsoring Editor
Bette J. Aaronson, Production Supervisor
Nancy Benjamin, Production Editor
Joyce Cameron Weston, Text and Cover Design
Roy Logan, Manufacturing Supervisor

Library of Congress Cataloging-in-Publication Data

Koffman, Elliot B.
Problem solving and structured programming in FORTRAN 77.

1. FORTRAN 77 (Computer program language).
2. Structured programming. I. Friedman, Frank L.
II. Title.
QA76.73.F25K64 1990 005.26 89-18494
ISBN 0-201-51216-5

The programs and applications presented in this book have been included for their instructional value. They have been tested with care but are not guaranteed for any particular purpose. The publisher does not offer any warranties or representations, nor does it accept any liabilities with respect to the programs or applications.

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book, and Addison-Wesley was aware of a trademark claim, the designations have been printed in initial caps or all caps.

Copyright © 1990, 1987, 1981, 1977 by Addison-Wesley Publishing Company, Inc.

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of the publishers. Printed in the United States of America.

Reprinted with corrections December, 1990

4 5 6 7 8 9 10 DO 9594939291

To our families
Caryn, Richard, Deborah, and Robin Koffman
and
Martha, Shelley, and Dara Friedman

Preface

This is a textbook for a first course in problem solving and programming methods using the FORTRAN language. It assumes no prior knowledge of computers or programming. A course in high-school algebra is sufficient mathematics background for most of the material in this textbook, except for sections of the last chapter. In other chapters, examples that require more mathematical background are clearly marked.

The most significant change in this edition is a substantial increase in the number of engineering and science examples, including at least one new case study in each chapter. We have also retained most of the general examples. The net result is a textbook that can be used either in a FORTRAN course for engineering and science majors or in an introductory programming course for a general audience.

Problem Solving, Structured Programming, and Modularization

The first edition of this textbook was published in 1977, the same year that FORTRAN 77 became the standard version of the FORTRAN language. The first edition pioneered the teaching of problem solving and structured programming in a FORTRAN course, a tradition that has continued in successive editions.

Problem solving continues to be emphasized in this edition. Chapter 2 contains a new section on the art and science of problem solving, which includes a discussion of the engineering and scientific method.

This edition has an increased emphasis on modular programming, which is an important component of structured programming. Chapter 4 introduces the use of library functions; Chapter 5 is a new chapter on user-defined functions and subroutines. In this edition, we discuss both kinds of FORTRAN subprograms before coverage of arrays (Chapter 6). This allow us to use subprograms in many of the program examples that manipulate arrays.

Organization of the Book

The first seven chapters represent the core of the book and should be studied by all students. Students in a general course should skip many of the engineering examples; students in a predominately engineering or science section may skip some of the business-oriented case studies.

Chapter 1 provides an overview of computer hardware and software, including an expanded introduction to the history of computers. It contains an introduction to FORTRAN statements and data types, and the use of FORTRAN in different programming environments. It also introduces the essentials of using files, so that students can write programs that access data files prepared by their instructor or send their program results to an output file.

此为试读，需要完整PDF请访问：www.ertongbook.com

Chapter 2 describes problem solving, algorithm refinement, and decisions in programs. The chapter begins by discussing problem solving in engineering and science followed by a general discussion of problem solving and algorithm refinement. It also shows how to write conditions and implement decisions in FORTRAN using the IF statement. The chapter covers all aspects of the use of the IF statement, including the Block IF form for implementing decisions with multiple alternatives.

Chapter 3 describes two control structures for implementing loops: the DO loop and the WHILE loop. As in earlier editions, we include the WHILE loop because we believe students should know how to use this control structure, which is fundamental to structured programming. Even though the WHILE loop is not part of the FORTRAN 77 standard, it is provided as an optional feature in many versions of FORTRAN 77 and will be included in the new FORTRAN standard (FORTRAN 8X). For those who do not have access to this control structure, we show how to implement it in standard FORTRAN 77.

Chapter 4 provides additional details on FORTRAN data types, including the use of library functions in expressions. The chapter also introduces the FORMAT statement.

Chapter 5 is a new chapter on subprograms. The chapter covers both function and subroutine subprograms. It introduces preconditions and postconditions as part of a function's documentation. It also shows how to use structure charts to document the flow of control and data between the main program and its subprograms.

Chapter 6 introduces the array, focusing on one-dimensional arrays. Chapter 7 covers advanced features of arrays, including multidimensional arrays, searching an array, and sorting an array. The chapter also discusses additional subprogram concepts, including COMMON storage and the SAVE statement. The chapter ends with a discussion of random number generation and computer simulation.

The remaining chapters cover advanced material. These chapters do not have to be studied in any particular order. If time permits, instructors in classes with a general audience should concentrate on Chapters 8 and 9, which cover additional features of formatting, file processing, and string manipulation. Instructors in classes with science and engineering students should spend more time on Chapters 10 and 11, which cover graphing, computer-aided design, and numerical methods.

FORTRAN Versions and Computers

The text uses FORTRAN 77 as its language of instruction rather than FORTRAN 8X. The reason for this choice is the current widespread availability of FORTRAN 77 compilers. At the time of publication of this book, FORTRAN 8X was still awaiting final approval so there were no FORTRAN 8X compilers. As mentioned earlier, we have included one important new control structure from FORTRAN

8X, the WHILE loop. Appendix D summarizes many of the important features of FORTRAN 8X.

The text is not oriented toward any specific computer. Chapter 1 discusses booting a personal computer and logging onto a timeshared computer. Most of the examples are written as interactive programs, which is the most prevalent mode of programming today. All of the new programs have been tested on an IBM-compatible personal computer using Microsoft FORTRAN 77. Appendix C describes how to use this popular compiler for microcomputers.

Pedagogical Features

We employ many pedagogical features to enhance the usefulness of this book as a teaching tool. Some of these features are discussed below.

End of Section Exercises: Most sections end with a number of self-check exercises. These include exercises that require analysis of program segments as well as short programming exercises. Answers to these exercises appear in the back of the book.

End of Chapter Exercises and Projects: Each chapter ends with a set of quick-check exercises with answers. There are also chapter review exercises whose solutions appear in the back of the book. Finally, there is a set of programming projects whose solutions appear in the instructor's manual.

Examples and Case Studies: The book contains a large number and variety of programming examples. Whenever possible, examples contain complete programs or subprograms rather than incomplete program fragments. There are also substantial case studies that help a student integrate and apply concepts studied over several chapters.

Syntax Display Boxes: The syntax displays describe the syntax and semantics of each new FORTRAN feature and also provide examples.

Program Style Displays: The program style displays discuss issues of good programming style.

Error Discussions and Chapter Reviews: Each chapter ends with a discussion of common programming errors. A chapter review includes a table of new FORTRAN statements.

Appendices and Supplements

A reference table of FORTRAN statements appears on the inside covers of the book. There are appendices on FORTRAN Library Functions and Character Sets. Also, there is an appendix on MS-DOS and the Microsoft FORTRAN 77 compiler and one covering FORTRAN 8X.

Supplements include an instructor's manual, transparency masters, and a program disk with all the programs that appear in the book. Use the reference numbers below to order these supplements from your Addison-Wesley sales representative.

Instructor's Manual: 1-201-51217-3
Transparency Masters: 1-201-52147-4
Program Disk: 1-201-51218-1

We are very grateful to Thomas Cunningham at Indiana University of Pennsylvania for his work on producing the instructor's manual.

Acknowledgments Many people participated in the development of this book. Professors Bart Childs and Glen Williams, and Mr. Mitch Fincher of Texas A & M University carefully reviewed the previous edition and provided many suggestions for improvement. They also prepared a host of engineering examples, exercises, case studies, and programming projects for inclusion in the book. Their help was invaluable in increasing the number of engineering and science applications presented.

The principal reviewers were most essential in suggesting improvements and finding errors. The principal reviewers include: Professor James Allert, University of Minnesota; Walter Brainerd, UNICOMP; Professor William H. Dodge, Rensselaer Polytechnic University; Professor George J. Dudas, Pennsylvania State University, Behrend College; Professor Sundaresan Jayaraman, Georgia Institute of Technology; Professor Richard A. Lejk, University of North Carolina at Charlotte; Professor Ernie Rilki, William Rainer Harper College; Abolfazl Sirjani, IBM; Professor Conrad L. White, Northern Illinois University.

We would also like to acknowledge the contribution of Professor Richard Epstein of George Washington University. Professor Epstein provided the original draft of much of the material in Chapters 10 and 11.

The personnel at Addison-Wesley responsible for the production of this book worked diligently to meet a very demanding schedule. The sponsoring editor, Keith Wollman, was closely involved in all phases of the manuscript preparation and provided much help and guidance. His assistant, Debbie Lafferty, did an excellent job of coordinating the reviewing process. Bette Aaronson supervised the design and production of the book, while Nancy Benjamin coordinated the conversion of the manuscript to a finished book.

E.B.K.
F.L.F.

Application Areas

Contents

3. Repetition and Loops 113

4. Simple Data Types 178

5. Subprograms 234

此为试读，需要完整PDF请访问：www.ertongbook.com

8. Formats and Files 397

9. String Manipulation 457

10. Plotting Functions and Computer-Aided Design 497

11. Introduction to Numerical Methods 527

1. Introduction to Computers and Programming

IN THIS chapter, we introduce computers and computer programming. We begin with a brief history of computers and a description of the major components of a computer, including memory, central processor, input devices, and output devices. We also discuss how information is represented in a computer and how it is manipulated.

Following this introduction, we begin a discussion of the main topics of this book: problem solving, programming, and FORTRAN. We first discuss problem solving with a computer. Then languages for computer programming are described. Finally, some simple computer operations are presented along with some short FORTRAN programs that demonstrate these operations. The FORTRAN programs contain statements for reading and displaying information and for performing simple computations.

Also described are the steps involved in creating a FORTRAN program and the roles performed by special programs that are part of a computer system. These programs include the operating system, compiler, editor, and loader.

1.1 Electronic Computers Then and Now

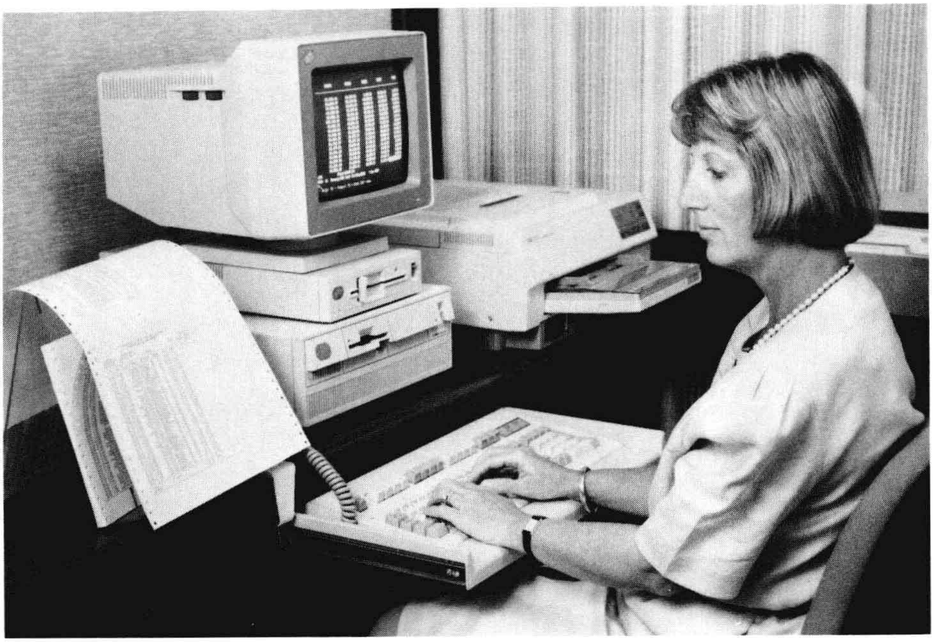
It is difficult to live in today's society without having some contact with computers. Computers are used to provide instructional material in schools, print transcripts, send out bills, reserve airline and concert tickets, play games, and even help authors write books.

However, it wasn't always this way. Just a short time ago, computers were fairly mysterious devices that only a small percentage of our population knew much about. Computer "know-how" turned around when advances in solid-state electronics led to large reductions in the size and cost of electronic computers. Today, a personal computer (see Fig. 1.1), which costs less than $3000 and sits on a desk, has as much computational power as one that 10 years ago would cost more than $100,000 and would fill a 9′ × 12′ room. This price reduction is even more remarkable when we consider the effects of inflation over the last decade.

If we take the literal definition for a *computer* as a device for counting or computing, then the abacus might be considered the first computer. However, the first electronic digital computer was designed in the late 1930s by Dr. John Atanasoff at Iowa State University. Atanasoff designed his computer to perform mathematical computations for graduate students.

The first large-scale, general-purpose electronic digital computer, called the ENIAC, was built in 1946 at the University of Pennsylvania with funding supplied by the U.S. Army. The ENIAC was used for computing ballistics

Figure 1.1. *IBM Personal Computer*



tables, for weather prediction, and for atomic energy calculations. The ENIAC weighed 30 tons and occupied a 30′ × 50′ space (see Fig. 1.2).

Although we are often led to believe otherwise, computers cannot think! They are basically devices for performing computations at incredible speeds (more than one million operations per second) and with great accuracy. However, in order to accomplish anything useful, a computer must be *programmed*, or given a sequence of explicit instructions (the *program*) to carry out.

To program the ENIAC it was necessary to connect hundreds of wires and arrange thousands of switches in a certain way. In 1946, Dr. John von Neumann of Princeton University proposed the concept of a *stored program computer* in which the instructions of a program would be stored in computer memory rather than be set by wires and switches. Since the contents of computer memory can be easily changed, it would not be nearly as difficult to reprogram this computer to perform different tasks as it was to reprogram the ENIAC. Von Neumann's design is the basis of the digital computer as we know it today.

Brief History of Computers

Table 1.1 lists some of the important milestones along the path from the abacus to modern-day electronic computers. We often use the term *first gener-*

此为试读,需要完整PDF请访问: www.ertongbook.com