SCIENTIFIC ANALYSIS ON THE POCKET CALCULATOR

JON M. SMITH

JMSA Systems Research and Analysis

Second Edition

SCIENTIFIC ANALYSIS ON THE POCKET CALCULATOR

To Laurie,
Mike, and Chris

PREFACE

This book is written for all those who own or operate a modern electronic pocket or desk calculator, and especially engineers, scientists, science students, mathematicians, statisticians, physicists, chemists, computer analysts, and science educators.

When the right numerical methods are used, the electronic pocket calculator becomes a very powerful computing instrument. "Micronumerical methods" that will help the reader to derive the most computing capability for every dollar he has spent on his pocket calculator are discussed here.

Most of the methods work on *any* pocket calculator. Special methods for certain types of machines are clearly indicated where necessary. Key stroke sequences for both algebraic and reverse-polish calculators are shown. Virtually all pocket calculator keyboards and capabilities were considered in preparing this book, to ensure that the numerical methods presented are the most universally applicable for general pocket calculator analysis.

Each part of this book provides a consistent and careful treatment of the methods and tabulated formulas that can be used with a pocket calculator. The aim is to supply the reader with a large number of numerical techniques, numerical approximations, tables, useful graphs, and flow charts for performing quick and accurate calculations with pocket calculators.

Emphasis is also given to numerical methods used in certain types of data processing, such as harmonic and statistical analysis. And they are presented in forms that are directly useful to engineers, scientists, and programmers.

The premise of this book is that the pocket calculator provides the scientific analyst with an important new dimension in analysis. Obviously the pocket calculator is useful both for numerically evaluating functions and for processing data. *In addition,* it enables the analyst to quickly gain detailed and quantified knowledge about any technical discipline (his own or another's) by *learning* its mathematical models and tools through use and experimentation on the pocket calculator. In short, the pocket cal-

culator becomes a *teaching machine* for the scientific analyst. A scientific analyst no longer need first develop a mathematical model for a complex process or system being studied and then turn it over to a computer programmer for its numerical evaluation. Instead, he can numerically evaluate complex functions (and thus analyze complex problems) in his home or office.

Finally, throughout the world scientific analysts working on pocket calculators are inventing their own numerical methods for evaluating problems in their specific disciplines. In this sense, the pocket calculator is a *research tool* which the analyst can use to develop his own numerical methods for his own purposes.

Throughout the book I give more attention to subjects of interest to the practitioner than to those of interest to the theorist. Though the treatment of this material is mathematical, I have not strived for conciseness or rigor beyond that required for pocket calculator analysis. Numerous examples of each technique and method are given, and their implementation is discussed in detail.

This book consists of five parts that are subdivided into 15 chapters, each dealing with topics in numerical analysis that are useful to the practical analyst. I have tried to avoid overgeneralization in the treatment of these topics, since numerical analysis is an art as much as it is a science. *Part I* of the book introduces the spectrum of pocket calculators (including their capabilities and their limitations) available to engineers and scientists. Particular attention is given to the unique computing features of interest to the scientific analyst. Part I also presents mathematical preliminaries and mathematical refresher material and develops certain elementary numerical methods particularly suited to analysis on the pocket calculator. Topics from arithemtic to algebra and analysis with complex variables are covered.

Part II presents numerical methods and formulas for numerically evaluating advanced mathematical functions. It also deals with the nested parenthetical form of the most frequently used functions in advanced engineering mathematics. It is the nesting of a sequence of arithmetic operations in parenthetical form that is the basis for performing advanced analysis on the pocket calculator. For example, 14 multiplies, 2 divides, 2 sums, and 108 data entries, totaling 126 key strokes and 5 data storage records, are needed for a three-digit floating-point evaluation of sin (x) $\approx x - x^3/3! + x^5/5!$. But only 54 key strokes and *no data storage records* (on a scratch pad) are needed to evaluate sin $(x) \approx x(1 - (x^2/6)(1 - x^2/20))$ to the same accuracy. Though we would evaluate sin (x) in this manner only on a four-function calculator, this example does illustrate the point

that many complex formulas usually requiring calculator memory to be numerically evaluated can be written in a "nested" form not requiring calculator memory and thus can be evaluated conveniently on even the simplest four-function pocket calculator.

The nested parenthetical form is considered to be a "fast" form for numerical evaluation. That is, functions written in nested parenthetical forms require fewer operations to numerically evaluate than do the same functions in their "simplest algebraic form." The nested forms are therefore evaluated more rapidly and with fewer chances for error than their unnested counterparts.

Part II also covers the topics of recurrence formulas for numerically evaluating advanced functions such as Bessel functions, Legendre polynomials, and many more not found on even the most sophisticated scientific pocket calculators. Recurrence formulas are unique in that they are *infinite memory forms* of otherwise *finite memory form* calculations. The formulas give the pocket calculator "virtually" an infinite memory for storing data. Many useful numerical methods for data processing can be rewritten in recursive form for pocket calculator analysis. Here, again, even the simplest four-function calculator is shown to be capable of doing sophisticated analysis without the need for extensive memory. Such concepts as nested parenthetical forms and recursion formulas, when combined with those of Chebyshev economization and rational polynomial approximation, provide flexibility and accuracy in the numerical evaluation of the most complex functions; even on the simplest four-function pocket calculator. In fact, the serious analyst can perform precise calculations unheard of until a few years ago—in the comfort and convenience of his home or while traveling on the job.

Part III examines the methods and formulas for performing advanced "types" of analysis on the pocket calculator. Included are such topics as numerical evaluation of definite integrals and methods for numerical differentiation of data sets, solving differential equations, simulating linear processes, performing statistical analysis, and performing harmonic analysis.

Part IV deals exclusively with analysis on the advanced programmable pocket calculator. The chapters illustrate conclusively the leap in computing capability produced by the pocket calculator. They are based on personal experience in solving a very large number of problems on the programmable pocket calculators developed by Hewlett-Packard, Texas Instruments, Sinclair, and National Semiconductor (NOVUS brand). The discussion is general, however, recognizing that even more programmable pocket-style calculators are being developed.

Part V covers the concepts and calculations of financial analysis. As the economics of systems and design become more important, the engineer and scientist are being asked to evaluate the economics and financial implications of their work. Part V is intended to acquaint the technical professional with the field of finance through discussions of the mathematics of finance as currently preprogrammed into the more versatile scientific pocket calculators.

This book grew out of eleven years of study on numerical methods for analysis on the digital computer. These methods were revised over a period of three years to make them applicable to desk calculator analysis and eventually to pocket calculator analysis. A number of the methods have been available to the analyst in scattered literature, such as user's guides and manuals for desk-top and pocket calculators, journal articles, and some textbooks. A large part of the material was developed by me or was provided by my associates in industry. I am particularly indebted to my associates at Software Research Corporation and McDonnell Douglas Corporation. They generously shared with me many of their "tricks of the trade" and suggested interesting problems for this book. I express my sincere appreciation to one of the great numerical analysts of our time, Dr. Richard Hamming, of Bell Laboratories, for his review and improvements to the manuscript.

My thanks to the people at Hewlett-Packard who reviewed and critiqued the manuscript, and in particular to the HP-65 chief engineer, Mr. Chung Tung.

I want to thank Joseph and Sara Goldstein, who taught me the Goldstein algorithm -- "one at a time."

To my wife, Laurie, my special appreciation for putting up with the 4 a.m. writing schedule.

My appreciation also to Mrs. Florence Piaget who typed the manuscript and helped me prepare it for publication.

Finally, my thanks to the readers of the first edition who recommended changes, corrections, and additions which have improved this book.

Jon M. Smith

Washington, D. C.
May 1977

CONTENTS

PART IV   THE PROGRAMMABLE POCKET CALCULATOR

PART V   FINANCIAL ANALYSIS FOR ENGINEERS AND SCIENTISTS

APPENDIXES

INDEX

INTRODUCTION TO POCKET CALCULATOR ANALYSIS

THE POCKET CALCULATOR

1-1 INTRODUCTION

This chapter discusses the mathematical differences among the various pocket calculators and certain mathematical concepts, useful for analysis on the pocket calculator, that appear throughout this book.

We are not so much concerned with the hardware implementation of mathematical operands and operations as with the different ways in which they can be assembled in a computing machine—the hardware architecture. Only the most obvious mathematical aspects of calculator design are examined, such as the language used, the size and type of memory, the instruction set, type of input/output, and whether the calculator is programmable. There are some 432 types of calculators that could be hardware implemented. An entire book could be written on this subject alone. Here we limit our discussion to the more important mathematical differences that result from the various hardware implementations in order to:

1. Understand pocket calculators and the organization of mathematics within them.

2. Determine, in a cursory way, the combinations of hardware implementation that result in a significant jump in calculating capability.

The purpose is to narrow the types of calculator to be considered in this book to three.

Three hypothetical calculators that are typical of the available and anticipated pocket computing machines are discussed. Care has been taken throughout not to limit the methods of analysis to any particular hardware implementation. In fact, if there is bias throughout the writing it is in the direction of anticipated developments in the pocket calculator field, though its overall effect on the material is negligible.

3

The following mathematical aspects are covered in this chapter:

1. Arithmetic calculations.
2. Function evaluation with and without memory.
3. Computational accuracy.

The first is a thorough introduction to what appears to be a mundane subject, arithmetic on the pocket calculator. In fact, it is found to be quite the opposite because the different languages used by different calculators lead to different capabilities for handling complex problems.

Particular attention is given to nested parenthetical forms of complex scientific functions that permit function evaluation on memoryless and limited-memory calculators. Nested parenthetical forms are used as a means of providing *implicit memory* to the memoryless calculators. They are also "fast" in the sense that their evaluation involves fewer key strokes than does the standard algebraic form.

No chapter on mathematical preliminaries in a book on numerical analysis would be complete without a discussion of computational accuracy. Here we examine:

1. The accuracy limitations of the typical pocket calculator.
2. Ways in which to accurately evaluate functions in general, and on the pocket calculator in particular.

1-2 MATHEMATICAL DIFFERENCES IN POCKET CALCULATORS

Today's pocket calculators differ mathematically in many ways. Only the six more commonly encountered mathematical distinctions are covered here. In a sense, these are the major distinctions because they are the fundamental issues addressed in the *conceptual design* of every pocket calculator. The important mathematical distinctions that are associated with the subtleties of *detailed design* are not discussed because the hardware implementations vary widely. Perhaps the best known difference is that between the use of fixed-point and floating-point numbers.

The *fixed-point numbers* are those whose decimal point is fixed by the electronic circuitry. A difficulty occurs when two large numbers are multiplied together and the most significant digit exceeds the size of the numeric display. Then the number is truncated in the most significant digits. Most fixed-point arithmetic calculators have a symbol that is illuminated to indicate the *overflow* conditions.

Floating-point numbers have a decimal point that moves so as to retain

the most significant digits in any calculation. When a number is computed that is larger than the calculator's field of numbers and the decimal point location is unknown, most calculators display the most significant digits and illuminate a symbol indicating that the decimal point location is unknown.

In these two number systems, it should be noted, the number fields are dramatically different. In the floating-point number system the numbers are "bunched" around zero. In the fixed-point number system the numbers are uniformly distributed over the range of the number field. To see this, consider the process of incrementing each of these types of numbers on a pocket calculator.

The smallest possible increment between any two numbers is the least significant digit in the numeric display. For an eight-digit display with a decimal point fixed in the third place, the smallest increment that can be added to *any digit* is 0.001. Now consider the addition of an increment to a floating-point number. Since the decimal point "floats" in the floating-point number system, the decimal point precedes the far-left digit. For an eight-digit display, the smallest number that can be added to *zero* in a floating-point number system is 0.00000001. Now consider incrementing a floating-point number when the decimal point is after the far-right digit. In this case, the smallest number that can be added to 99999998 is 1. The difference in the size of the "smallest number" when incrementing a full and empty register in floating-point numbers is a factor of 10^8.

Now consider the full range of the positive numbers in both number systems. The fixed-point numbers range from 0.001 to 99999.999. The difference between numbers, no matter where a number is over the range of the calculator, is 0.001. Thus the numbers over the range of fixed-point numbers are uniformly distributed.

Again consider the range of the positive floating-point numbers, from 99999999 to 0.00000001. Clearly, the range is greater in the floating-point number system than in the fixed-point number system, but note also that when the numbers are very small the distance between them is 0.00000001. When the register is full, the difference between the numbers is 1. Obviously, over the range of floating-point numbers, the distribution is not uniform. In fact, there are as many numbers grouped between 0 and 1 as there are between 1 and the full register size 99999999.

It follows, then, that in fixed-point arithmetic the absolute difference remains fixed over the entire range of the number system, while in floating-point numbers the absolute difference varies significantly. It is worth emphasizing that in floating-point arithmetic the percentage difference remains fixed, while in the fixed-point arithmetic system the constant difference remains fixed over the range of numbers. As used here,

percentage difference is the ratio of the difference between two consecutive numbers divided by the larger of the two. For most engineering analysis, percentage difference and percentage error are usually the measure of accuracy of most interest.

The floating-point number system is usually extended by powers of 10, permitting the positive floating-point numbers to range from 10^{-99} to 99999999×10^{99}. In fact, calculators are usually configured to display this extended number field in scientific notation. Interestingly, this even further bunches the floating-point numbers in the neighborhood of zero. Because of this grouping property of the floating-point numbers, the absolute errors are smaller for calculations with numbers between 0 and 1 than for numbers betwen 1 and the full range of the calculator.

From a hardware architecture viewpoint, fixed-point numbers are usually displayed with greater accuracy than floating-point numbers; and floating-point numbers are usually displayed with a greater dynamic range than fixed-point numbers. This can be seen by considering a register with eight *display elements* where we configure both fixed- and floating-point numbers. In fixed-point arithmetic, eight mantissa digits can be displayed. If the decimal point is allowed to be set by the decimal point key $\boxed{\cdot}$, and a display element is used to show the decimal point, then only 7 digits remain to display the mantissa. If scientific notation is used to increase the dynamic range of the display, $m+1$ display elements are required to display m digits in the exponent. The extra display element is used to show the sign of the exponent.

Power of 10	Display	Required Display Elements
$10^{\pm x}$	$(\pm)(x)$	2
$10^{\pm xx}$	$(\pm)(x)(x)$	3
.		
.		
.		
$10^{\pm xxx}$	$(\pm)\underbrace{(x)(x)\cdots(x)}_{m \text{ digits}}$	$m+1$

If 99 orders of magnitude are to be shown in the display register (the usual case with scientific pocket calculators) three display elements are required to display the exponent and its sign, leaving only five digits for displaying the mantissa. In this sense, then, the effect of increasing the