OMNITAB

A Computer Program For Statistical and Numerical Analysis

United States Department of Commerce
National Bureau of Standards
Handbook 101

Preparing This Publication

Extensive use was made of the digital computer in preparing this publication. In addition to the computer results which are obviously taken from the computer output, the text itself was prepared with the aid of the digital computer and a card- controlled typewriter. The first draft of the manuscript was punched on cards and processed on the IBM 870 Document Writer. After numerous editorial revisions, the cards were fed into an IBM 7094 computer which recomposed the lines and justified the text to the specified width. These justified cards were then fed through the Document Writer again to produce the camera- ready copy from which the pages in this Handbook were produced.

That the computer was instrumental in speeding up the production of this Handbook is now quite clearly established. Furthermore, the authors are satisfied that the computer- assisted copy preparation used in this Handbook greatly facilitated its production. They hope that such compromises as were forced upon them by the present limitations of the system and as are reflected in the copy do not seriously hamper the reader.

UNITED STATES DEPARTMENT OF COMMERCE • John T. Connor, *Secretary*
NATIONAL BUREAU OF STANDARDS • A. V. Astin, *Director*

OMNITAB

A Computer Program For Statistical and Numerical Analysis

Joseph Hilsenrath, Guy G. Ziegler,[1] Carla G. Messina, Philip J. Walsh, and Robert J. Herbold[2]

[1] Computer Department, General Electric Company, Phoenix, Arizona

[2] Present address: Computer Science Center, University of Maryland, College Park, Maryland

National Bureau of Standards Handbook 101
Issued March 4, 1966

Reissued January 1968, with corrections.

For sale by the Superintendent of Documents, U.S. Government Printing Office
Washington, D.C., 20402 - Price $3.00

Library of Congress Catalog Card Number : 65-61853

FOREWORD

As high-speed computers assume an increasingly important role in the scientific, technological, and economic life of modern society, it is becoming necessary to develop techniques that will provide the user with more convenient and efficient access to these machines.

Over the last few years there have been a number of continuing efforts to make it easier for scientists and engineers to use large digital computers. This Handbook presents the result of one of these efforts. It describes the OMNITAB general- purpose digital computer program, a tool that permits workers who are unfamiliar with programming to communicate with a large computer in a highly efficient manner by means of simple English sentences.

It is hoped that this publication will serve two purposes: first, to provide a manual for those who will be using the program, and secondly, to serve as a stimulus for further improvements in the man- machine interaction.

A. V. Astin, Director.

PREFACE

This Handbook describes the characteristics, operation, application and design of a general- purpose digital computer program developed and employed at the National Bureau of Standards for statistical and numerical analysis of experimental data, and for a wide variety of computations in applied mathematics, science, and engineering. Three important characteristics of the OMNITAB program have influenced the preparation of this work in the present format. The first of these is the generality of the program which permits its ready application to such diverse problems as frequency sharing of satellites, reference tables for thermocouples, the influence on range measurements of tropospheric refraction of radio waves, analysis and fitting of molecular spectra, and a variety of problems from biometrics to econometrics to sociology. The second feature is the interpretive character of the program which enables it to respond to simple English language instructions. This feature permits the use of the digital computer by nonprogrammers in a manner highly analogous to the way they use desk calculators. Finally, the conciseness of the instruction set for a wide class of problems makes the system an ideal one for use with remote computer stations.

This volume is essentially a user's manual. It discusses the general philosophy and motivation for the program design and specific instructions on the application of the program to a variety of calculations arising in research and development establishments. The wide spread use of the program at the National Bureau of Standards and the interest shown in it by other agencies have motivated the preparation of this volume in the NBS Handbook series.

A supplementary report is in preparation addressed to programmers. It contains a brief discussion of the important overall logical features of the program and a detailed symbolic listing of the main program and a number of special subroutines. That volume should provide the experienced programmer with sufficient information to add new subroutines or, if space is a consideration, to replace existing subroutines by ones more suited to the local machine, operating system, or problem applications. As the program now contains close to 100 subroutines and is far too large to fit into a 32K core, it has been designed as a complete system operating as a subsystem under the IBM system called IBSYS.

During the three years over which this program was developed and tested, the authors received considerable help from numerous colleagues at NBS and elsewhere. The following NBS programmers have contributed one or more subroutines to the system: Vernon Dantzler, J. D. Waggoner, Douglas McMillan, Steve Muchnick, Martin L. Reilly, Ruth N. Varner, and Bradley A. Peavy. Much valuable advice and encouragement was received from many colleagues inside of NBS and outside — who were motivated to use the program in its early stages. A listing of these would be either too long or too prone to serious omissions. Mention should be made however of the appreciation which the authors feel for the consistent encouragement and sterling advice proffered by Joseph M. Cameron and Alfred E. Beam, in the first instance in matters of the statistical treatment of data, and in the second in elucidating certain programming subtleties and for providing a critique of the OMNITAB concept and organization. They are, however, hereby absolved of any tarnish which may adhere to the finished product. The authors appreciate the valuable help received from a number of colleagues who volunteered to read the manuscript in its many versions. Among these are T. B. Douglas, M. Greenspan, G. C. Sherlin, R. Moore, and H. Matheson; last but not least Mrs. P. J. Fowler deserves special mention for her devoted and patient preparation of the manuscript.

It is a pleasure to acknowledge also the valuable assistance rendered to this program by a number of outside groups: In particular to Al Beam and John P. Menard of the Computer Science Center, University of Maryland for making staff and machine time available during various check out stages of the program; to B. D. Holbrook of the Bell Telephone Laboratories for supplying the subroutines of computing the elliptic, sin, cos, and exponential integrals; to Wm. H. Sickles of the University of Chicago for supplying a set of improved function subroutines prior to their release to the SHARE Library.

Special thanks are due to a number of University colleagues for their assistance in making OMNITAB multilingual. They are Professor Pierre Johannin, and Professor Audrie Johannin of College Scientifique Universitaire, Brest, France; Professor E. U. Franck of the Institut für Physikalische Chemie und Electrochemie, Technische Hochschule, Karlsruhe, Germany; and Professor Takahiko Shimanouchi of the Department of Chemistry, University of Tokyo, Japan, and Dr. Isao Suzuki of the University of Tokyo who provided a number of subroutines which formed the basis for the matrix operation portion of the program.

It would be unusual indeed if this Handbook were error-free. The authors sincerely hope that such errors as have escaped their attention will cause the reader no serious difficulty and would appreciate early

notification if any are found. As the program will shortly be available for distribution, they would of course welcome correspondence concerning the experience of others in applying the program.

The arrangement of a description of a system as extensive as this presents the authors with a number of difficult choices, each of which has drawbacks as well as advantages. We have tried to arrange the material in such a way as to provide the reader first with the underlying philosophy of and motivation for the preparation of the program, and then with a broad view of the entire repertoire of commands and operations. After this the material is divided into two major portions — in the first, sections 3 and 4 cover details on input, output, and mathematical operations in the ordinary operating mode. The latter chapters treat the mathematical and statistical analysis operations, the repeat mode, operations on matrices and arrays of numbers, and other more sophisticated operators.

The commands (exclusive of matrix and array operations) are treated together in chapter 3. Their operation is illustrated extensively but not completely in section 3.9 via examples which are worked out fully. The examples have been arranged, in general, in order of their complexity rather than in the logical order in which the instructions are given in chapter 3. In addition, a number of problems are solved again in succeeding chapters employing more sophisticated instruction. Problems so treated are cross— referenced.

Unlike normal programming techniques, the user need not master the entire vocabulary to perform many substantial calculations. This will become obvious on examination of the examples. Some of the more consistent users of OMNITAB who employ it to reduce or fit experimental data seldom use more than a dozen or so commands. In general, however, many users have been stimulated to tackle larger and more sophisticated calculations requiring a more complex instruction set.

Our advice to the reader is to read the introductory material through section 2.1, glance briefly through the entire instruction set in section 2.2, skip to section 4, and then to the problems in section 3.9. These should be sufficient to stimulate reference to the appropriate parts of section 3 for a detailed description of each of the commands. It would be profitable, before settling down to a detailed study of this Handbook, to read the first two or three pages of each of the later chapters to obtain a broader view of the system. From there on the reader must be governed by his own needs or inclinations.

Joseph Hilsenrath, Guy G. Ziegler, Carla G. Messina, Philip J. Walsh, and Robert J. Herbold.

Preface to the Second Printing

That the stock of the first printing of this Handbook should have been exhausted almost coincident with the appearance of the reviews of it, points up the keen interest in user-oriented systems for problem solving on modern computers. Equally gratifying is the recognition by the designers of more conventional computer languages of the need to free programmers from many of the tedious, annoying, and error-generating rules and restrictions of early versions of FORTRAN - restrictions which are largely absent in OMNITAB.

This edition differs from the first in the following ways: a number of typographical errors have been corrected; and the description of a number of commands has been amplified to clear up any ambiguity. A section has been added describing some branching instructions which are present in the program, but were kept subrosa because of the reluctance of the authors at that time to have OMNITAB viewed as still another programming language. The current trend toward instructing computers in a more natural language now makes that reservation seem academic.

An appendix, written by Dr. David Hogben, has been added. It amplifies the discussion of a large number of the commands and shows how the existing commands can be used to perform fairly sophisticated analysis of experiments which were not specifically provided for. Aside from its value for practicing statisticians, it shows off the power of certain of the manipulative commands in the hands of an ingenious problem solver.

TABLE OF CONTENTS

1. INTRODUCTION

In spite of the successful application of digital computers to important problems in virtually every subdivision of modern science, they are not yet being employed widely and effectively to assist in the routine day to day computations of a busy laboratory. They have not yet replaced the multicolumned work sheet, the use of which is wide spread in both experimental and theoretical research.

A brief catalog of these routine numerical calculations would include interpolation (linear and nonlinear), statistical analysis, smoothing, curve fitting, numerical differentiation and integration, and the generation of tables of elementary and special functions, and tabulation of various physical quantities. These mathematical operations obviously have general applicability. Their day to day use in any individual situation is, however, quite specialized — too specialized it would seem to justify the preparation of a multitude of computer programs tailor-made for each application.

Access to digital computers has been made easier for both the experienced programmer and the novice by the development recently of a variety of systems, compilers, and languages. These aids have encouraged many scientists to learn to program their own problems. In spite of the expanding corps of scientist programmers, there is now and will continue to be a need for painless access to digital computers by scientists who are not programmers.

In an earlier report° we described a general-purpose computer program which was designed to handle a variety of ad hoc computations which are ordinarily performed on desk calculators. That program, called OMNIFORM I, was essentially a function generator. Experience gained with that program indicated the feasibility of extending it to include numerical and statistical manipulation of tabular data. In the design of this program, which has been named OMNITAB, it became clear that programming techniques had developed

- - - - - - - - - - -

° Hilsenrath, J., and Galler, G. M., OMNIFORM I: A General Purpose Machine Program for the Calculation of Tables of Functions Given Explicitly in Terms of One Variable, National Bureau of Standards Technical Note 125 (PB 161626) May 1962. See also NBS Technical News Bulletin 43, p 8. (1960)

sufficiently in the last two years to permit the user to communicate with the machine via English sentences. Since any program for manipulation of tabular data must have certain arithmetic capability, the entire arithmetic and function generating capability of OMNIFORM I was included in OMNITAB.

The motivation for the development of the program and its strong user-orientation arose from the need of one of the authors to provide a group of nonprogrammers, engaged in a wide variety of thermodynamic calculations, with direct access to a large digital computer on a day to day basis. More direct and immediate access is needed than is afforded by employing one or more programmers as middlemen on the one hand, or by training the entire staff to program on the other hand. The first alternative seemed unattractive in view of the shortage of trained programmers and the second seemed even less attractive since it would divert the scientists from their primary duties in some degree if they just dabbled at programming and would divert them to a serious degree if they were to become truly proficient programmers.

Regardless of the existence of "simple" user-oriented programming languages, if the job is to be done correctly, the user must pay careful attention to the rules in minute detail. There is no doubt that these languages and others are important to the progress of computer science and that more powerful languages are still to be developed. Our point here is that the use of the computer should not be restricted, in the first instance, to those who have mastered programming (even the "simple" kind) or, in the second instance, by a system which requires the preparation of program after program for the miscellaneous ad hoc computing tasks which face every experimenter.

The early motivation toward a do-it-yourself system was such that we were prepared to pay a considerable price for this facility either in the length of the instruction set, or in inefficient use of the core, and even in increased machine time. Very early in the development of the program it became clear that there was really no need to sacrifice efficiency. The interpretive mode is really very fast. As the program grew, as more commands were added, and as it was applied to increasingly more complex problems, it became clear that in spite of the outwardly elementary character of the commands the instruction set was surprisingly short, and in many problems was even shorter than the corresponding set for the same problem written in FORTRAN.

Experience, extending over a 10 year period, with a variety of computer programs written in the conventional way — one might even say classical way — indicated to us the importance of freeing the user from annoying, time consuming, and often very costly consequences of fixed field input; of supplying the machine with a card count on each input of data or providing its equivalent, a flag word in some fixed location. Another annoying feature of conventional general-purpose programs is the cumbersome system of selecting options via integers in fixed field locations. OMNITAB frees the user from all of these unnecessary chores. The program accepts numbers and other input wherever they appear on the card; the program counts the cards read in; and the major subroutines which perform statistical analysis, curve fitting, finite differencing, etc., have virtually all options programmed in. In those instances where options are unavoidable, they are incorporated in the instructions in a conversational tone so that recourse to a dictionary of options or a complicated set of rules is unnecessary.

A good example of the amount and variety of results which it seems reasonable to provide on an automatic basis without option is provided by the instruction...

STATISTICAL ANALYSIS OF COL 6, WEIGHTS IN COL 2

This command sorts the data in increasing order, ranks them, differences them, computes both the weighted and unweighted mean, and the decile distribution. In addition, there are computed 34 measures of statistical characteristics of the data and the residuals, including various measures of central tendency, randomness, skewness or kurtosis, etc. These results are all printed out on two or more pages, without any further instruction (see figures 5-1 and 5-2). In addition, the deviations, ordered data, rank, and the 34 statistical characteristics are stored respectively in columns 46, 45, 44, and 43 of a 46 column work sheet so as to make them available for further calculations or to permit the selective storage of some of them for further analysis or comparison with like results from subsequent data sets (see figure 5-3).

Another command which provides a comprehensive automatic printout is:

POLYFIT COL 4,WEIGHTS IN COL 3,X IN COL 5,USE 7TH DEGREE

(See figures 5-6, 5-7, and 5-8.)

From the very beginning it has been our objective in designing the logic and structure of OMNITAB to provide the user with maximum flexibility, generality, and sophistication and yet keep to a minimum the requirement for prior detailed knowledge of computer hardware. During the past 3 years OMNITAB users have ranged from those who had never examined a punched card in detail to experienced programmers. As is to be expected these extremes formed a small minority. Even those without prior contact with punched cards learn very early the advantage of being able to punch their own cards. For this reason, we assume in this Handbook that the user is familiar with punched card equipment.

Even a casual contact with the professional computer literature and particularly with programming textbooks makes one realize the dependence of conventional programming on flow-charts. In view of the important role flow-charts play in other programming systems, the absence of any reference to flow-charts in this Handbook deserves some explanation. The explanation is quite simple. Flow-charts are no more necessary in carrying out OMNITAB calculations than they are in doing calculations on a desk calculator. At most, all that is required there is to keep track of which column of the work sheet contains which values. So it is with OMNITAB.

1.1. General-Purpose Programs

The motivation to facilitate communication by practicing scientists (including, of course, engineers, statisticians, economists, psychologists, etc.) with the computing machine in as natural and direct a manner as possible, is not peculiar to ourselves. The desire to address the machine via English words acceptable to a completely assembled general-purpose program has brought forth a number of systems. Considering their diverse origins, the following quotations show a remarkable degree of consistency.

> "COGO - a computer programming system for civil engineering problems ... has the following characteristics: The instructions or commands to the computer which the engineer uses to express the solution of a problem are at approximately the same technical language level as instructions which one engineer would use in describing his solution to another engineer." C. L. Miller°

> "... It is felt that this language is close to the language a statistician would use to express his problem to other statisticians so that the statistician wishing to perform certain multivariate analysis has a minimum of computing rules to learn before he can present his problem." B. E. Cooper°°

> "The program, hereafter referred to as DAM (Data processing and Multiple regression) facilitates the preparation of input data to be used in multiple regression analysis. It enables users without specific programming knowledge to write sequences of desired computations essentially in the form in which they would give instruction to a statistical assistant." R. R. Rhomberg and L. Boissonneault°°°

- - - - - - - - - - - - - -

° Miller, C. L. COGO - A Computer Programming System for Civil Engineering Problems (Civil Engineering Dept., Massachusetts Institute of Technology, Aug. 15, 1961).

°° Cooper, B. E. Designing the Data Presentation of Statistical Programs for the Experimentalist 34th Session of the International Statistical Institute, Ottawa, Aug. 1963 .

°°° Rhomberg, R. R. and Boissonneault, L. A General-Purpose Computer Program for Data Processing and Multiple Regression (DAM), International Monetary Fund, Washington, D. C., June 1962.

The above programs fall into two main classes, as do other numerous efforts along this line: Those that are interpretive and perform their work via English words, and those that depend largely upon numbers suitably coded to various entry-points and program options.

A notable representative of the first group – to which OMNITAB obviously belongs – is the very successful program, COGO°, devised by Professor C. L. Miller of the Civil Engineering Department of M.I.T., which solves problems involving such coordinate geometry calculations as arise in surveying and related work. We know of no others in this category and hasten to remark that – for all of its outward similarity – COBOL°° does not belong in this group, since it is a programming language from which a binary program is compiled, and not a fully compiled program or system of programs. While it might be argued that the OMNITAB vocabulary does indeed represent a programming language, it is our position that even though a portion of the command structure could possibly form the basis of a programming language, the majority of the specialized commands could not be so considered. In any event, such considerations are beyond the scope of this Handbook.

A most notable example of the second group, to which OMNIFORM I° belongs, is the extensive series of statistical programs prepared by Health Sciences Computing Facility of the School of Medicine at the University of California in Los Angeles called BMD°°°. Other statistical packages are referred to in chapter 5. They do not, however, exhaust the list. The few programs we have examined were too specialized to meet the need which the OMNITAB program was designed to satisfy.

If the existing general-purpose programs do not go far enough to satisfy the needs for more direct and efficient access to the machine, neither do the very useful and admittedly successful streamlined and simplified programming systems. This is, in our opinion, the case even for those systems whose stated objectives coincide closely with our own and the quotations above. A short bibliography of general-purpose computer programs is given at the end of this chapter (section 1.3).

° loc. cit.

°° COmmon Business Oriented Language.

°°° Dixon, W. J., Editor, BMD Biomedical Computer Programs Jan. 1964, UCLA Book Store, Los Angeles 24, California.

1.2. The Operating System

A few words are in order at this point concerning operating systems for large computers. In early computer designs and on many small computers even today, the operator manipulates the computer via a typewriter or a teletype key board. He enters data, modifies instructions or halts the computer when so desired. Conversely the computer often halts waiting for the operator to supply the next piece of information for a problem or to supply the next problem when the current one is finished. Even where the input is via paper tape or via punched cards, there is an attendant delay of time, and consequently a cost in dollars. Such delays can be tolerated on small machines and are often considered a desirable attribute of the smaller machines.

On the large machines, however, where the rental time runs to hundreds of dollars per hour, such delays cannot be tolerated. These machines must be run under a supervisory or monitor system which automatically goes from one problem to the next with a minimum of reliance on human operators. To achieve this it is necessary to prepare the input on magnetic tape sometime prior to its actual running on the machine. This is invariably done on a smaller auxiliary machine which performs the input and output functions for the larger machines.

Even under these systems, operators are still required to mount and unmount data tapes and the various system tapes. The amount of such manual manipulation varies with the system, with the number of tape units available, and with the nature and length of the problem being run.

Since OMNITAB is itself a system containing nearly 100 subroutines and over 250 commands, it must be integrated into one or more of the operating systems in use at the particular installation. At the National Bureau of Standards, for example, an early version has been running successfully on the Bell Monitor System and the current version has been incorporated as a subsystem under IBSYS as modified by the NBS Computation Laboratory.

The reader of this Manual need not be concerned at the outset, and perhaps not even later, with the characteristics of various operating systems. All that is required is to supply the few control cards demanded by the particular system. Figure 1-1 shows the arrangement of the control cards in relation to the OMNITAB instructions under IBSYS, while figure 1-2 shows the location of the control commands on the cards.

A run on the machine is initiated by a card carrying $JOB punched in columns 1 through 4 and such other information as may be required by the particular installation at which the program is run. The second card is a $EXECUTE card (punched in columns 1 through 8) and the system name OMNITAB punched in columns 16 through 22. This card pulls in the OMNITAB program from the IBSYS System tape and stores it in the core (the magnetic memory of the IBM 7094). A run may contain one or many independent problems each of which is preceded by a card carrying the word OMNITAB. No special instruction is required between problems other than the OMNITAB card which initializes the program parameters and clears out the working space for the new problem. The STOP card at the end of a series of problems ends the run by turning control back to the OMNITAB monitor.

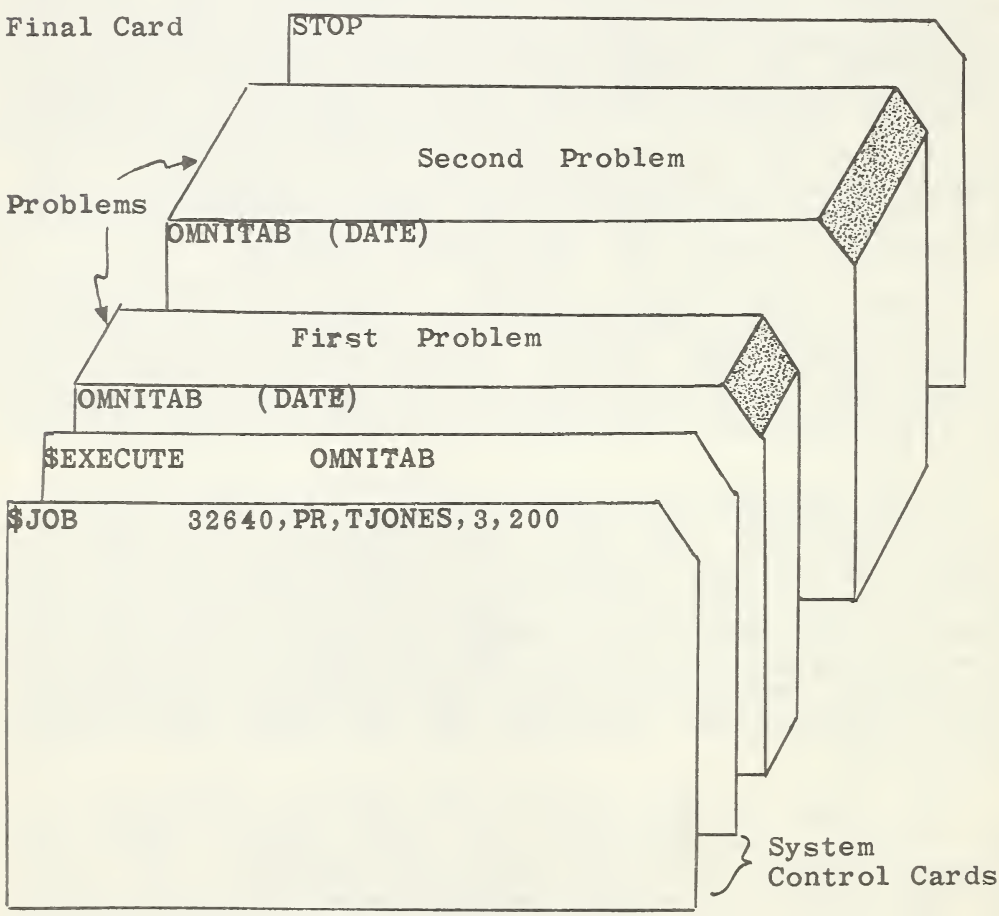


Figure 1-1. Arrangement of Control Cards for Omnitab runs.

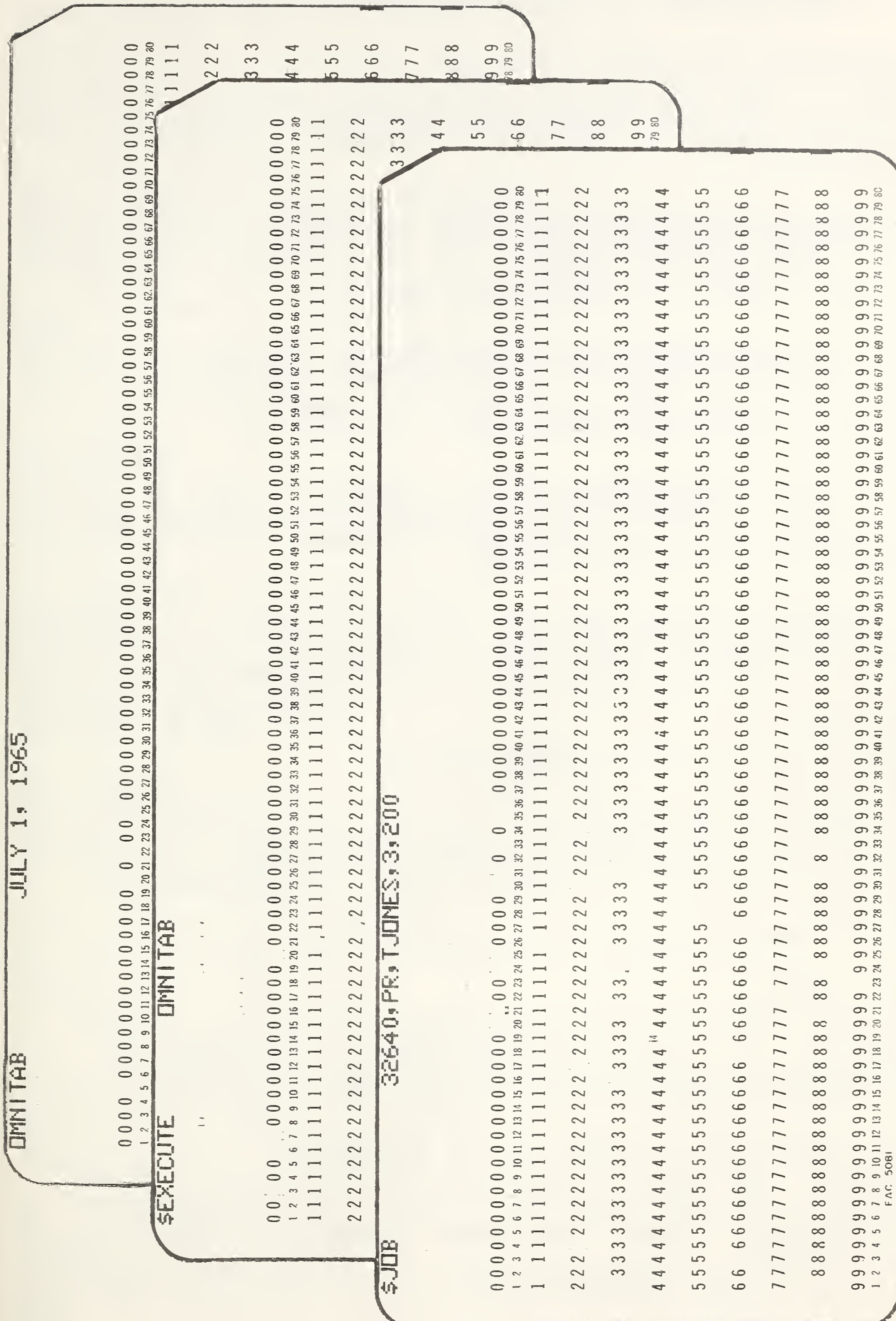


Figure 1-2. Location of Control Commands on the Cards.

1.3. Bibliography of General-Purpose Computer Programs

Anon.
A General Purpose Program for High-Speed Computers
National Bureau of Standards Technical News Bulletin 43, p. 8(1960).

M. Berman and R. Schoenfeld
Invariants in Experimental Data on Linear Kinetics and the Formulation of Models
J. Appl. Phys. 27, 1361-1370 (1956).

M. Berman, E. Shahn, and M. F. Weiss
The Routine Fitting of Kinetic Data to Models: A Mathematical Formalism for Digital Computers
Biophys. J. 2, 275-288 (1962).

M. Berman, M. F. Weiss, and E. Shahn
Some Formal Approaches to the Analysis of Kinetic Data in Terms of Linear Compartmental Systems
Biophys. J. 2, 289-316 (1962).

M. Berman
The Formulation and Testing of Models
Annals N. Y. Acad. Sci. 108, 182-194 (1963).

M. Berman
A Postulate to Aid in Model Building
J. Theoret. Biol. 4, 229-236 (1963).

M. Berman and M. F. Weiss
Manual for the NIH-OMR Computer Program 9B20
National Institute of Health, Office of Mathematical Research.
Report dated February 1963.

W. S. Brown
The ALPAK System for Nonnumerical Algebra on a Digital Computer - I: Polynomials in Several Variables and Truncated Power Series with Polynomial Coefficients.
Bell System Technical Journal 62, 2081 -2118, 1963.

W. S. Brown, J. P. Hyde, and B. A. Tague
The ALPAK System for Nonnumerical Algebra on a Digital Computer -II: Rational Functions of Several Variables and Truncated Power Series with Rational Function Coefficients
B. S. T. J. 63, 785-804, 1964.

W. S. Brown and S. P. Hyde
ALPAK Reference Manual mimeographed report.

J. M. Cameron and J. Hilsenrath
Use of General-Purpose Coding Systems for Statistical Calculations
Proceedings of the I.B.M. Scientific Computing Symposium on Statistics, October 21-23, 1963, pp. 281-299 IBM White Plains N.Y. (1965)

W. J. Dixon, Editor
BMD Biomedical Computer Programs
Health Sciences Computing Facility, School of Medicine, University of California, Los Angeles, Jan 1, 1964.

J. Felberbaum, A. Court, and K. Downes
TOOT
Brookhaven National Laboratories, Memorandum dated April 9, 1963.

S. Fenves, R. D. Logcher, S. P. Mauch, and K. F. Reinschmidt
STRESS: A User's Manual. A Problem-Oriented Computer Language for Structural Engineering
The M.I.T. Press, Cambridge, Mass.

J. Hilsenrath and G. M. Galler
OMNIFORM I: A General Purpose Machine Program for the Calculation of Tables of Functions Given Explicitly in Terms of One Variable
National Bureau of Standards Technical Note 125 (PB 161626) May 1962.

J. P. Hyde
The ALPAK System for Nonnumerical Algebra on a Digital Computer-III: Systems of Linear Equations and a Class of Side Relations (in press).

I. H. Pomper and H. Posten
An Investigation and Evaluation of the Generating Discriminant Function
IBM Report.

R. R. Rhomberg and L. Boissonneault
A General-Purpose Computer Program for Data Processing and Multiple Regression (DAM)
Department of Research and Statistics, International Monetary Fund, June 1962.

M. J. Romanelli
Thaumaturgy, A Simplified Method of Coding for ORDVAC
BRL Memorandum Report No. 1116, Dec. 1957
Ballistic Research Laboratory, Aberdeen Proving Ground, Md.

D. Roos and C. L. Miller
The Internal Structure of COGO-90
Research Report R64-5, Dept. of Civil Engineering, Mass. Inst. of Tech. Cambridge, Mass., February 1964.

D. Roos and C. L. Miller
COGO-90 Engineering User's Manual
Research Report R64-12, Dept. of Civil Engineering, Mass. Inst. of Tech., Cambridge, Mass., April 1964.

D. Roos, and C. L. Miller
COGO-90 Time-Sharing Version
Research Report R64-18, Dept of Civil Engineering, Mass. Inst. of Tech. Cambridge, Mass., May 1964.

I. E. Sutherland
A Man-Machine Graphical Computer System
Tech. Report No. 296, Lincoln Labs, Cambridge, Mass. January 30, 1963.

J. A. Yuhos and V. H. Lucke
Panacea User's Manual
Report No R64FPD4, Advanced Engine and Technology Dept., General Electric Co., Cincinnati, Ohio (Jan 1964).

J. A. Yuhos and V. H. Lucke
Panacea Programmer's Guide
Report No R64FPD5, Advanced Engine and Technology Dept., General Electric Co., Cincinnati, Ohio (Feb. 1964).

2. THE OMNITAB PROGRAM

2.1. General Characteristics

OMNITAB is a completely assembled interpretive program for the IBM 7090/7094 which permits direct use of the machine by scientists or engineers without knowledge of programming. Instructions, given in the form of English sentences, control the flow of calculations in a manner highly analogous to the logic which prevails in carrying out computations on a desk calculator. More precisely, OMNITAB simulates desk computing in that it replaces the desk calculator, mathematical tables, and the multicolumn work sheet. The name of the program has the following significance. It is one in a series of omnibus programs, hence the <u>omni</u>. It is able to handle a wide variety of tabular numerical operations, hence the <u>tab</u>. It has a versatile function generating capacity and can handle functions of two or more variables.

The instructions for the program are given via a series of sentences or certain abbreviations of them. Each sentence is punched on a card. The method in which OMNITAB is used is analogous to the method used in carrying out calculations on a desk calculator. The program simulates a work sheet containing 46 columns having 101 rows. Arguments are entered or generated in specified columns, then mathematical or manipulative operations are performed on desired columns. And finally, the results are stored in the designated column. The storing is done in two modes, either by addition or replacement. The mathematical operations include the arithmetic operations, the elementary and special functions, and various statistical and numerical analyses. The manipulative operations provide for inverting, promoting, demoting, exchanging, shortening, erasing, printing, and punching columns of numbers.

Although the instruction is given in sentence form, only the first word (or the first six characters, if the word has more than six) of the sentence and the numbers are crucial to the operation. The machine scans the whole card and picks up only the first word and the numbers and ignores the intervening words completely. For this reason, the intervening words are not really needed. Their use is, however, encouraged to enable the user to read his own instructions more easily.

To illustrate the use of the OMNITAB sentences, suppose it is desired to multiply a column of numbers listed in column one by those in column three and to store the result in column four. The instruction is:

MULTIPLY COL 1 BY COL 3, STORE IN COL 4,
or simply
MULT 1, 3, 4°

In this instruction, as in all the sentences involving a single operation, the storage is always by replacing the previous value in the storage column. To be able to add the result to the value already in column four, we resort to a corresponding sentence of the double operation type. In this category, after each mathematical operation, a multiplication is performed, after which the product is put away by adding to the result in the designated column. Thus to add the sum of the columns one and three to that in four, we might use the following instruction:

ADD 1 TO 3, MULT BY 1., ADD TO 4
or simply
ADD 1, 3, 1., 4
or even
ADD 1 3 1. 4

It should be noted that the ADD instruction applies only to columns 1 and 3, that the last figure, 4, denotes a storage column, and that the extra figure (in this case 1.) is always a multiplicative factor. The presence of this extra number in the sentence is enough for the machine to distinguish the sentence from ADD 1, 3, 4 (note the absence of decimal points).

A system such as this would be severely limited if it were restricted to operating only with numbers already stored. It is necessary to supply numbers such as coefficients or exponents or constants. OMNITAB achieves this through a very simple device. In an instruction statement, the absence of a decimal point in a number means that the number designates a column in the work sheet (certain few exceptions to this rule will be noted later). The presence of a decimal point causes

- - - - - - - - - -

° Ordinarily the entire word is used although only the first six characters are needed. In addition, four abbreviations have been incorporated: MULT, SUB, DIV, and EXP. Numbers must be separated by a comma or one or more blank spaces.

the number to be read as itself.° Thus ADD 1. TO 2 means add the value 1 to numbers stored in column 2 while ADD 1 TO 2 means add the numbers in column 1 to those in column 2. Similarly, RAISE 2 TO 3. means take the cube of the values in column 2, while RAISE 2 TO 3 means raise each of the numbers in column 2 to the exponent specified by the corresponding number in column 3.

The following example illustrates how the computer is instructed to carry out a calculation by the OMNITAB program.

Suppose we wish to compute a table of values of a function F for a series of values of T ranging from 1500, in steps of 100 to 6000, when the function F can be represented by the empirical relation:

$$F = a(1 - \ln T) + bT + cT^2 + dT^3 + eT^4 + k$$

where $a = 4.6083000$, $d = -1.686851 \times 10^{-11}$,

$b = -4.309474 \times 10^{-4}$, $e = 0.8721937 \times 10^{-15}$,
$c = 1.276702 \times 10^{-7}$, $k = -1.851497$,

The OMNITAB instructions for the computational part of this problem could be as follows:

```
GENERATE 1500. (100.)6000. IN COL 1

ADD    -1.851497 TO 0. AND STORE IN COL 2

RAISE 1 TO 4., MULT BY 0.8721937E-15, ADD TO COL 2

RAISE 1 TO 3., MULT BY -1.686851E-11, ADD TO COL 2

MULT 1 BY 1, MULT BY 1.276702E-7, ADD TO COL 2

MULT -4.309474E-4 BY 1, MULT BY 1., ADD TO COL 2

LOGE OF 1, STORE IN COL 3

SUBTRACT 3 FROM 1., MULT BY -4.6083, ADD TO COL 2
```

° Numbers written using E format may have the decimal point missing, provided that a numeral precedes the E. For example, 123E4 ($123. \times 10^4$) is a valid number (so is 123E+4 or 123E04), and even 123E 4, which is compatible with Fortran format even though it is an exception to the OMNITAB rule.

Since the arrangement of the OMNITAB program is such that only the first word of the sentence constitutes a command, the same operations would be performed if the other words were deleted. Thus, the program would work also if the instructions were as follows:

```
GENERATE 1500., 100., 6000., 1

ADD -1.851497, TO 0., 2

RAISE 1, 4., 0.8721937E-15, 2

RAISE 1, 3., -1.686851E-11, 2

MULT 1, 1, 1.276702E-7, 2

MULT -4.309474E-4, 1, 1., 2

LOGE 1, 3

SUB 3, 1., -4.6083, 2
```

In the above instructions, some of the results are simply stored while others are stored by adding into results previously contained in the column. The nature of the store is not optional. It is tied to the number of arguments (numbers) contained in the sentence. For example, in line 2 of the above instructions, the result is stored by replacement because there are three numbers in the sentence, while in line 8 the result is stored by addition because there are four numbers in the sentence.° This point should be more evident after an examination of the basic OMNITAB sentences contained in section 2.2. Note the alternate notation for the interval in the GENERATE instruction in the two examples.

It should be observed that nowhere in the above instructions was any mention made of the length of the column. That is indeed one of the tours de force of the program. The length of the column is determined from the GENERATE instruction. This length is stored in a special cell in the core (called NRMAX) and used by the subsequent subroutines to tell them how far down the column to operate.

° Actually the mode of storage is controlled by a determination by the program that the sentence contains either the minimum number of arguments required for the operation and storage or one additional argument.

At the start of a calculation the word OMNITAB clears out the work-sheet and sets various program parameters to zero or any other appropriate value. NRMAX in particular is set to zero and until it is reset to something else, no operations can be performed. The instructions GENERATE, READ, SET, and GQUAD set NRMAX to an appropriate value. As its name implies, it is always the maximum length (the length of the longest column). It can also be modified by the instruction

RESET LENGTH TO ,, ROWS
or simply
RESET 15

At the beginning of an OMNITAB calculation, all of the columns contain zeros. Subsequently the entire work sheet can be erased (replaced by zeros) by the instruction ERASE. Specific columns can be erased by instructions such as ERASE COL 2, or ERASE 3, 7, 9, etc.

The programming philosophy underlying OMNITAB is to simplify the use of the program by the user at the expense of the original programmer. We cite the following characteristics of OMNITAB to illustrate the extent to which this point has been pushed:

a. The input is variable. With the exception of the control cards, neither the words nor the numbers need to be in any fixed location on the cards. The only requirement is that numbers must be separated by a comma or one or more spaces or by any alphabetic character except E.

b. A card count is not required. The program counts the data cards following a READ statement.

c. The program distinguishes between data and computing instructions automatically.

d. Format statements are not ordinarily required. They may be used if necessary in inputting (where data are not separated by spaces) or, if desired, in providing a compact printout.

e. The instructions are given via sentences containing a leading word or acronym and a sequence of numbers with intervening words or spaces.

f. Except for the first word of the sentence, which must be a legitimate word in the program's vocabulary, the remaining words in the sentence are ignored. They may or may not be used as the user chooses. Only the first word must be used precisely as given in the vocabulary; the intervening words may be abbreviated, truncated, or left out entirely.

Before we discuss the vocabulary list, the reader is reminded that the program operates in two modes. The first of these is the ordinary mode, where the machine reads an instruction card, executes the instruction, and writes the instruction on an output tape for printing at the end of the problem. Once this instruction is executed it is no longer available. The other mode is the repeat mode, in which instructions can be stored in the core and used over and over again. A discussion of the repeat mode is deferred to a later section.

2-2. OMNITAB Vocabulary and Sentence Structure

```
LEGEND

++ = A COL NO. (NO DECIMAL POINT)     ,, = AN INTEGER (NO DEC POINT)*
** = A CONSTANT (12.34,1.2E-7,ETC.)   - = A BLANK SPACE
$$ = A COL NO. OR CONSTANT


INPUT INSTRUCTIONS (SEE SECTION 3.1 FOR MORE INFORMATION )

OMNITAB (YOUR NAME AND THE DATE HERE WILL IDENTIFY EACH PAGE)
GENERATE ** (**) ** (**) ** ... IN COL ++
READ COL ++,++,++,++ , ETC
SET IN COL ++, ARGUMENTS ON CARDS TO FOLLOW
FREAD  ,, CARDS INTO COLS ++,++,++ ETC
  (FREAD MUST BE PRECEDED BY A FORMAT CARD COMPATIBLE WITH THE DATA)

ORDINARY OUTPUT INSTRUCTIONS (SEE SECTION 3.2 FOR MORE INFORMATION )

PRINT COL ++,++,++,ETC   (8 COL LIMIT)
FIXED WITH ,, DECIMALS
FLOATING WITH ,, DECIMALS
PUNCH COL ++, ++, ++, ++ (4 COL LIMIT)
PLOT COLS ++, ++, ++, AGAINST ++  (5 COL LIMIT)
PLOT COLS ++, ++, ++, AGAINST ++ ABSC FROM ** TO ** ORD FROM ** TO **
NOSUMMARY    (SUPPRESSES SUMMARY)
SUMMARIZE    (RESTORES SUMMARY)
TITLE1-   (66 CHARACTERS MAXIMUM)
TITLE2-   (66 CHARACTERS MAXIMUM)
TITLE3-   (53 CHARACTERS MAXIMUM)
TITLE4-   (53 CHARACTERS MAXIMUM)
HEAD COL ++/   (12 CHARACTERS ARE ALLOWED AFTER / )
NOTE--      (AS APPROPRIATE)
FOOTNOTE---- (AS APPROPRIATE)
ABRIDGE ROW ,, OF COLS ++, ++, ++,   ETC.
STOP

SPECIAL OUTPUT INSTRUCTIONS (SEE SECTION 3.3 FOR MORE INFORMATION )


FORMAT   (AS DESIRED, IN ACCORD WITH FORTRAN RULES )
FPRINT COL ++,++,++,ETC (AS ALLOWED BY FORMAT)
FPUNCH COL ++,++,++,... (AS ALLOWED BY FORMAT)
FABRIDGE ROW ,, OF COLS ++,++,++,ETC  (AS ALLOWED BY FORMAT)
TPRINT COL ++,++,++,++... (AS ALLOWED BY FORMAT BUT DOES NOT PAGE)
SPLOT COLS ++, ++, ++,    ETC. AGAINST ++
SPLOT COLS ++, ++, ++, AGAINST ++ ABSC FROM ** TO ** ORD FROM ** TO **
NEWPAGE
SPACE ,, LINES
PAGE NUMBER AT LOCATION ,,
PAGE STARTING WITH ,,  AT LOCATION ,,
NOLIST          (SUPPRESSES LISTING OF INSTRUCTIONS)
COMMANDS  (LISTS CURRENT OMNITAB VOCABULARY AND SENTENCE STRUCTURE)
CGS  (SWITCHES THE FUND PHYS CONSTANTS TO THE CGS SYSTEM)
MKSA  (SWITCHES THE FUND PHYS CONSTANTS BACK TO THE MKSA SYSTEM)
MANUAL  (LISTS AN ABRIDGED VERSION OF SECTION 3 OF THIS HANDBOOK)
WATSNU  (LISTS PROGRAM CHANGES SINCE THE LAST EDITION OF THIS HBK)
```

- - - - - - - - - - -

* The symbol ,, is used also for the degree of a polynomial, the number of points in a process or terms in a formula, a row number, and the number of rows and columns of a matrix.

FUNCTION GENERATION (SINGLE OPERATION)*

```
ABSOLUTE     OF $$ AND STORE IN ++
ADD          $$ TO $$ AND STORE IN ++
SUBTRACT     $$ FROM $$ , STORE IN ++
MULTIPLY     $$  BY  $$ , STORE IN ++
DIVIDE       $$  BY  $$ , STORE IN ++
RAISE        $$  TO  $$ , STORE IN ++
EXP          OF $$ , STORE IN ++
NEGEXP       OF $$ , STORE IN ++
LOGE         OF $$ , STORE IN ++       (BASE E)
LOGTEN       OF $$ , STORE IN ++
ANTILOG      OF $$ , STORE IN ++       (BASE TEN)
SQRT         OF $$ , STORE IN ++       (SQUARE ROOT)
SIN          OF $$ , STORE IN ++
COS          OF $$ , STORE IN ++
TAN          OF $$ , STORE IN ++
COT          OF $$ , STORE IN ++
SIND         OF $$ , STORE IN ++
COSD         OF $$ , STORE IN ++
TAND         OF $$ , STORE IN ++
COTD         OF $$ , STORE IN ++
ARCSIN       OF $$ , STORE IN ++
ARCCOS       OF $$ , STORE IN ++
ARCTAN       OF $$ , STORE IN ++
ARCCOT       OF $$ , STORE IN ++
ASIND        OF $$ , STORE IN ++
ACOSD        OF $$ , STORE IN ++
ATAND        OF $$ , STORE IN ++
ACOTD        OF $$ , STORE IN ++
SINH         OF $$ , STORE IN ++
COSH         OF $$ , STORE IN ++
TANH         OF $$ , STORE IN ++
COTH         OF $$ , STORE IN ++
ASINH        OF $$ , STORE IN ++
ACOSH        OF $$ , STORE IN ++
ATANH        OF $$ , STORE IN ++
ACOTH        OF $$ , STORE IN ++
```

* SINGLE OPERATION SENTENCES ERASE THE COLUMN BEFORE STORING RESULTS. SUB,DIV,MULT ARE ALSO ALLOWED.

FUNCTION GENERATION (DOUBLE OPERATION)*

```
ABSOLUTE  $$, MULT BY $$ , ADD TO ++
ADD       $$  TO   $$, MULT BY $$ , ADD TO ++
SUB       $$ FROM  $$, MULT BY $$ , ADD TO ++
MULT      $$  BY   $$, MULT BY $$ , ADD TO ++
DIV       $$  BY   $$, MULT BY $$ , ADD TO ++
RAISE     $$  TO   $$, MULT BY $$ , ADD TO ++
EXP       OF $$, MULT BY $$ , ADD TO ++
NEGEXP    OF $$, MULT BY $$ , ADD TO ++
LOGE      OF $$, MULT BY $$ , ADD TO ++        (BASE E)
LOGTEN    OF $$, MULT BY $$ , ADD TO ++
ANTILOG   OF $$, MULT BY $$ , ADD TO ++        (BASE TEN)
SQRT      OF $$, MULT BY $$ , ADD TO ++        (SQUARE ROOT)
SIN       OF $$, MULT BY $$ , ADD TO ++
COS       OF $$, MULT BY $$ , ADD TO ++
TAN       OF $$, MULT BY $$ , ADD TO ++
COT       OF $$, MULT BY $$ , ADD TO ++
SIND      OF $$, MULT BY $$ , ADD TO ++
COSD      OF $$, MULT BY $$ , ADD TO ++
TAND      OF $$, MULT BY $$ , ADD TO ++
COTD      OF $$, MULT BY $$ , ADD TO ++
ARCSIN    OF $$, MULT BY $$ , ADD TO ++
ARCCOS    OF $$, MULT BY $$ , ADD TO ++
ARCTAN    OF $$, MULT BY $$ , ADD TO ++
ARCCOT    OF $$, MULT BY $$ , ADD TO ++
ASIND     OF $$, MULT BY $$ , ADD TO ++
ACOSD     OF $$, MULT BY $$ , ADD TO ++
ATAND     OF $$, MULT BY $$ , ADD TO ++
ACOTD     OF $$, MULT BY $$ , ADD TO ++
SINH      OF $$, MULT BY $$ , ADD TO ++
COSH      OF $$, MULT BY $$ , ADD TO ++
TANH      OF $$, MULT BY $$ , ADD TO ++
COTH      OF $$, MULT BY $$ , ADD TO ++
ASINH     OF $$, MULT BY $$ , ADD TO ++
ACOSH     OF $$, MULT BY $$ , ADD TO ++
ATANH     OF $$, MULT BY $$ , ADD TO ++
ACOTH     OF $$, MULT BY $$ , ADD TO ++
```

* DOUBLE OPERATION INSTRUCTIONS ADD RESULTS TO THOSE ALREADY IN THE COLUMN.

MANIPULATIVE INSTRUCTIONS (SEE SECTION 3.5 FOR MORE INFORMATION)

```
EXCHANGE COL ++ WITH  ++, ++ WITH ++, ++ WITH ++, ETC
ERASE COL ++, ++, ++, ++, ETC
DELETE ROWS ,, ,, ,, ETC     (PLACES ZEROS IN ROW)
TRANSPOSE ROW ,, INTO COL ++, ,,INTO ++ , ,, INTO ++, ETC
        ( 5 ROW LIMIT)
FLIP COL ++ INTO COL ++, ++ INTO ++, ++ INTO ++, ETC
PROMOTE ARRAY ,, ROWS
DEMOTE ARRAY ,, ROWS
PROMOTE ,, ROWS, COL ++ INTO ++, ++ INTO ++, ETC
DEMOTE ,, ROWS, COL ++ INTO ++, ++ INTO ++
RESET LENGTH TO ,, ROWS
DEFINE ROW ,,  COL ++ AS ROW ,, COL ++
DEFINE ROW ,, COL ++ AS COL ++
DEFINE ** AS ROW ,, COL ++
SHORTEN COL ++ FOR COL ++ = **, STORE IN COL ++ AND COL ++
CLOSE UP   ROWS HAVING ** IN COL ++,++,++ ....
SEPARATE FROM COL ++ EVERY ,, ROW STARTING WITH ROW ,, STORE IN ++
INSERT IN COL ++ FROM COL ++ AT EVERY ,, ROW,STARTING AS ,, ROW,STORE IN ++
```

ARITHMETIC OPERATIONS (SEE SECTION 3.6 FOR MORE INFORMATION)

```
CHANGE SIGNS OF COLS ++, ++, ++, ETC
ROWSUM COLS ++, ++, ++, STORE IN COL ++
ROWSUM ARRAY STORE IN COL ++
SUM COL ++, STORE IN COL ++
SUM COL ++, ROWS ,, TO ,, ,STORE IN COL ++
SUM COL ++, ROWS ,, ,, ,, ,, ,STORE IN COL ++
AVERAGE COL ++ STORE IN COL ++
RMS OF COL ++, STORE IN COL ++
MAXIMUM OF COL ++, STORE IN COL ++
MAXIMUM OF COL ++, STORE IN COL ++, CORRESP  VALUE OF ++ IN  ++
MINIMUM OF COL ++, STORE IN COL ++
MINIMUM OF COL ++, STORE IN COL ++, CORRESP  VALUE OF ++ IN ++
DIAGONALIZE $$ INTO COLS ++ TO ++
SORT COL ++, CARRY ALONG COL ++, ++, ++, ETC
ORDER COL ++, ++, ++, ++, ETC
PARSUM COL ++, STORE IN COL ++
PARPRODUCT OF COL ++ , STORE IN COL ++
LENGTH OF ARRAY STORE IN ++
COUNT LENGTH OF COL ++ STORE IN COL ++ (SEARCHES FOR 3 CONSEC ZEROS)
SINTEGRAL OF ++, X IN ++, STORE IN ++   (SUMMARY INTEGRAL)

CADD REAL IN ++,TO REAL IN ++,STORE REAL IN ++   (COMPLEX ARITHMETIC)
CSUB REAL IN ++,TO REAL IN ++,STORE REAL IN ++   (COMPLEX ARITHMETIC)
CMULT REAL IN ++,BY REAL IN ++,STORE REAL IN ++  (COMPLEX ARITHMETIC)
CDIV REAL IN ++,BY REAL IN ++,STORE REAL IN ++   (COMPLEX ARITHMETIC)
```

SPECIAL FUNCTIONS (SEE SECTION 3.7 FOR MORE INFORMATION)

```
LSUB  ,, OF ++   (LAGUERRE)
TSUB  ,, OF ++   (CHEBYSCHEV)  - THE POLYNOMIALS (FSUBN)
USUB  ,, OF ++   (CHEBYSCHEV)  - ARE STORED BACKWARD
PSUB  ,, OF ++   (LEGENDRE)    - STARTING WITH N=1
HSUB  ,, OF ++   (HERMITE))    - IN COLUMN 46
STRUVE SUB 0 OF COL ++, STORE IN COL ++
STRUVE SUB 1 OF COL ++, STORE IN COL ++
SININT OF $$, STORE IN COL ++
COSINT OF $$, STORE IN COL ++
EINT   OF $$, STORE IN COL ++
COMPLETE ELLIPTIC INTEGRAL K=$$,PHI=$$,STORE FIRST KIND IN ++,SECOND IN ++
INCOMPLETE ELLIP INTEG, K=$$, PHI=$$, STORE FIRST IN ++, SECOND IN ++
ERROR OF $$, STORE IN COL ++
CERF  OF $$, STORE IN COL ++ (COMPLEMENTARY ERROR)
GAMMA OF $$, STORE IN COL ++
```

NUMERICAL AND STATISTICAL ANALYSIS (SEE SECTION 5.1)

```
DIFFERENCE COL ++, ARG IN COL ++  (SDIFFERENCE SUPPRESSES PRINT OUT)
DIVDIF COL ++, ARG IN COL ++      (SDIVDIF  SUPPRESSES PRINT OUT)
STATISTICAL ANALYSIS OF COL ++    (SSTATIS SUPPRESSES PRINT OUT)
STATISTICAL ANALYSIS OF COL ++ WEIGHTS IN ++
FIT COL ++, WEIGHTS IN ++, VECTORS IN ++, ++, ++,ETC  (23 VECTORS MAX)
       (SFIT          SUPRESSES PRINT OUT)
POLYFIT COL ++, WEIGHTS IN ++, X IN ++, USE ,, DEGREE
       (SPOLYFIT      SUPRESSES PRINT OUT)
SOLVE FOR COL ++, COEFFICIENTS IN ++, ++, ++, ...
GQUAD WITH ** POINTS,A = **,B = ** ,STORE X IN ++,WTS IN ++
MAXMIN X IN ++, Y IN ++, STORE MAX IN ++, ++ AND MIN IN ++, ++
INTERPOLATE X IN ++, Y IN ++ FOR L= ,, FOR ,, ENTRIES IN ++
                              USE ,, POINTS, STORE IN ++
DERIVATIVE OF X IN ++,Y IN ++ FOR L=,, FOR ,, ENTRIES IN ++
                              USE ,, POINTS, STORE IN ++
LAGINT OF COL++, USE ,, POINTS,H=**, STORE IN ++   (UNIFORM INTERVALS)
HARMONIC ANALYSIS OF COL ++ FOR ,, ORDINATES(EVEN), STORE COEF IN ++
CENSOR COL ++ FOR $$ REPLACING BY $$  STORE IN COL ++
SELECT IN COL ++,VALUES APPROXIMATING  ++,TO ** TOLERANCE ,STORE IN ++
SEARCH IN COL ++ FOR VALUES IN ++,CARRY ALONG VALUES FORM ++ INTO ++ ETC.
```

BESSEL FUNCTIONS (SEE SECTION 3.8 FOR MORE INFORMATION)

```
BEJZERO  OF $$, STORE IN COL ++
BEJONE   OF $$, STORE IN COL ++
BEYZERO  OF $$, STORE IN COL ++
BEYONE   OF $$, STORE IN COL ++
BEIZERO  OF ++, STORE IN COL ++
BEIONE   OF ++, STORE IN COL ++
BEKZERO  OF ++, STORE IN COL ++
BEKONE   OF ++, STORE IN COL ++
BEIZERO OF ++, OPTION 1, STORE IN COL ++
BEIONE  OF ++, OPTION 1, STORE IN COL ++
BEKZERO OF ++, OPTION 1, STORE IN COL ++
BEKONE  OF ++, OPTION 1, STORE IN COL ++
BERBEI SUB 0 OF ++, STORE IN ++ AND ++  (SCALED)
BERBEI SUB 1 OF ++, STORE IN ++ AND ++  (SCALED)
KERKEI SUB 0 OF ++, STORE IN ++ AND ++  (SCALED)
KERKEI SUB 1 OF ++, STORE IN ++ AND ++  (SCALED)
BERBEI SUB 0 OF ++, OPTION 1, STORE IN ++ AND ++ (UNSCALED)
BERBEI SUB 1 OF ++, OPTION 1, STORE IN ++ AND ++ (UNSCALED)
KERKEI SUB 0 OF ++, OPTION 1, STORE IN ++ AND ++ (UNSCALED)
KERKEI SUB 1 OF ++, OPTION 1, STORE IN ++ AND ++ (UNSCALED)
DBERBEI SUB 0 OF ++, OPTION 1, STORE IN ++ AND ++  (UNSCALED)
DBERBEI SUB 1 OF ++, OPTION 1, STORE IN ++ AND ++  (UNSCALED)
DKERKEI SUB 0 OF ++, OPTION 1, STORE IN ++ AND ++  (UNSCALED)
DKERKEI SUB 1 OF ++, OPTION 1, STORE IN ++ AND ++  (UNSCALED)
```

ITERATION AND REPEAT MODE INSTRUCTIONS (SEE SECTION 6.1)

```
BEGIN STORING INSTRUCTIONS
INDEX INSTRUCTIONS ,, ,, ,, ,, BY     ,,
INCREMENT INSTRUCTION ,, BY $$,$$,$$,ETC
COMPARE COL ++ AND ++ TO A TOLERANCE OF $$
ISOLATE X IN ++, FOR COL ++ EQUAL **, STORE IN ++
ISOLATE X IN ++, FOR COL ++ EQUAL **, USE ,, POINTS, STORE IN ++
ISETUP X IN ++,Y IN ++,NEW Y IN ++, STORE IN ++
ITERATE X IN ++,Y IN ++,NEW Y IN ++, STORE IN ++
FINISH STORING INSTRUCTIONS
REPEAT INSTRUCTIONS ,, THRU ,,  ,, TIMES
```

MATRIX OPERATIONS (SEE SECTION 7.1 FOR MORE INFORMATION)

(R IS NUMBER OF ROWS, C IS NUMBER OF COLUMNS)

```
MADD (A) IN ,, ++,R=,, C=,, TO (B) IN ,, ++,START STORING IN ,, ++
MSUB (A) IN ,, ++,R=,, C=,, FROM (B) IN ,, ++, START STORING IN ,, ++
SCALAR PRODUCT OF (A) IN ,, ++, R=,, C=,, BY **,START STORING IN ,,++
MMULT (A) IN ,, ++,R=,, C=,, BY (B) IN ,, ++, K=,, START STORING IN ,, ++
MRAISE (A) IN ,, ++, R=,, C=,, TO B=**,START STORING IN ,,++
DIAGVEC OF (A) IN ,, ++,R=,, STORE IN COL ++
MTRANS OF (A) IN ,, ++, R=,, C=,, START STORING ON ,, ++
VECMAT (A) IN ,, ++ R=,, C=,, START STORING IN ,, ++
RESTMAT FROM ,, ++ R=,, START STORING IN ,, ++
TRACE OF (A) IN ,, ++ R=,, STORE IN COL ++
COLNORN  (A) IN ,, ++ R=,, C=,, START STORING IN ,, ++
ROWNORM  (A) IN ,, ++ R=,, C=,, START STORING IN ,, ++
TRANSFORM (A) IN ,, ++,R=,, R=,, ROWS OF (U) IN ,, ++, START STORING IN ,, ++
BACKTRANS (A) IN ,, ++,R=,, C=,, COLS OF (U) IN ,, ++, START STORING IN ,, ++
SYMUTM (A) IN ,, ++ R=,,
SYMLTM (A) IN ,, ++ R=,,
EIGENVALUES OF (A) IN ,, ++ R=,, STORE ROOTS IN COL ++
EIGENVECTORS OF (A) IN ,, ++ R=,, STORE ROOTS IN ++, VECTORS IN ,, ++
LINEAR EQ, COEF IN ,, ++ R=,, RHSIDE IN COL ++, STORE SOLUTION IN ++
  (SOLVES A SET OF N LINEAR EQUATIONS IN N UNKNOWNS. N = 45)
INVERT (A) IN  ,, ++ R=,, STORE INVERSE STARTING IN ,, ++
  (THE MATRIX MAY BE AS LARGE AS 46 BY 46 AND NEED NOT BE SYMMETRIC)
```

ARRAY OPERATIONS AND MANIPULATIONS (SEE SECTION 8.1 FOR MORE INFORMATION)

```
ARAISE (A) IN ,, ++, R=,, C=,, TO B IN ++, START STORING IN ,, ++
ARAISE (A) IN ,, ++,R=,, C=,, TO (B) IN ,, ++, START STORING IN ,, ++
ALOGE  (A) IN ,, ++ R=,, C=,, START STORING IN ,, ++
ALOGTEN (A) IN ,, ++ R=,, C=,, START STORING IN ,, ++
ADIVIDE (A) IN ,, ++,R=,, C=,, BY $$, START STORING IN ,, ++
AMULT (A) IN ,, ++, R=,, C=,, BY (B) IN ,, ++, START STORING IN ,, ++
MINEL OF (A) IN ,, ++ R=,, C=,, STORE IN COL ++
MAXEL OF  (A) IN ,, ++ R=,, C=,, STORE IN COL ++
BLOCKTRANSFER OF (A) STARTING IN ROW ,, COL ++,R=,, C=,, TO ROW ,, COL ,,
APRINT STARTING IN ROW ,, COL ++, R=,, C=,,
COALESCE ON FIRST COL OF (A) IN ,, ++ R=,, C=,, STORE IN ,, ++
COALESCE ON ** IN THE FIRST COL OF (A) IN ,, ++ R=,, C=,, STORE IN ,, ++
ARRAYAVERAGE ON THE FIRST COL OF (A) IN ,, ++,R=,, C=,, STORE IN ,, ++
ARRAYAVERAGE ON ** IN FIRST COL OF (A) IN ,, ++,R=,, C=,, STORE IN ,, ++
DUPLICATE ,, TIMES THE ARRAY IN ,, ++ R=,, C=,, START STORING IN ,, ++
VECMULT (A) IN ,, ++ R=,, C=,, BY $$, START STORING IN ,, ++
```

SPECIAL OPERATORS (SEE SECTION 9.1 FOR MORE INFORMATION)

```
FORMULA  ,,OF COL ++ ,PARAMETERS FOR ,, TERMS IN COL ++, STORE IN ++
EXPAND $$ TO ,, POWER IN INTERVALS OF ,, START STORING IN  ++
EXPAND $$ TO ** POWER IN INTERVALS OF ** START STORING IN ++
PRODUCT  OF COLUMNS ++,++,++,++,ETC , STORE IN COLUMN ++  (DISCRETE COLS)
PRODUCT  OF COLUMNS ++ TO ++, STORE IN COLUMN ++ (CONSECUTIVE COLS)
CTOF  OF $$, STORE IN COL ++
FTOC  OF $$, STORE IN COL ++
ATOMIC MASS TABLE  STORE IN ++
MOLWT Z=,, AMOUNT=,,  Z=,,  AMOUNT=,,  ..... STORE SUM IN COL ++
EINSTEIN  TEMP IN $$, VIBRATIONAL FREQ(IN WAVE NO) IN $$
EINSTEIN  TEMP IN $$, FREQUENCY IN $$, GAS CONSTANT R=**
PFTRANS TEMP IN $$, MOLECULAR WEIGHT (M) IN $$
PFATOM TEMP IN $$, MOL WT IN $$, WAVE NO IN ++, G IN ++
PARTFUNCTION TEMP IN $$, WAVE NO IN ++, G IN ++, START STORING IN ++
```

2.3. A Few Simple Rules

1. Aside from two or more system cards which are required to get on the machine, the first card of any problem must carry the word OMNITAB.

2. The word OMNITAB must be punched in card fields 1–7 and should carry a date containing at least two numbers.

3. The first operation must introduce a set of arguments into the machine. This can be done via a GENERATE, READ, or SET instruction. After this point, instructions may follow in any order consistent with the problem at hand.

4. TITLE, HEAD, FORMAT, etc., are not considered operations. They may be introduced at any stage – between or even ahead of mathematical operations. In order that they be effective, they must precede the PRINT instruction to which they are to apply. These instructions can be updated or modified any number of times. The new titles or headings overwrite the previous ones.

5. Do not use a period when abbreviating words in the instructions; nor immediately after a TITLE2, etc.

6. In the ordinary operating mode certain instructions like ERASE, CHANGE SIGN, etc., can have as many as 25 column designations.

7. In the repeat mode, to be discussed later, a maximum of 10 numbers per command is allowed, otherwise the limit is 25 numbers.

8. Do not use an * anywhere except in calling for the fundamental physical constants.

9. Punch one card per instruction.

2.4. Bibliography of OMNITAB Applications

The following publications and reports refer to work involving calculations which were performed in whole or in part on OMNITAB.

G. F. Blackburn and F. R. Caldwell
Reference Tables for Thermocouples of Iridium-Rhodium Alloys Versus Iridium.
J. Res. NBS 68C (Eng. and Instr.) No 1., 41-59 (Jan-Mar 1964).

R. C. Butler and R. Moore
A Computer Program for the Archibold Method of Calculating Molecular Weight from Ultracentrifuge Data
Unpublished Report, American Red Cross, Washington, D. C.

E. Davis
Frequency Sharing by Satellites in Circular Equatorial Orbits.

J. J. Freeman
Real-Time Compensation for Tropospheric Radio Refractive Effects on Range Measurements NASA CR-109 - National Aeronautics and Space Administration, Wash. D. C. (October 1964).

Walter J. Lafferty, Arthur G. Maki, and Earle K. Plyler
High-Resolution Infrared Determination of the Structure of Carbon Suboxide
J. Chem. Phys. Vol 40, No. 1, 224-229, 1 Jan. 1964

Walter J. Lafferty and Robert J. Thibault
High-Resolution Infrared Spectra of $C_2^{12}H_2$, $C^{12}C^{13}H_2$, and $C_2 H_2$
Journal of Molecular Spectroscopy 14, 79-96 (1964).

R. Moore
A Computer Program for Calculating Sedimentation Coefficients from Ultracentrafuge Data
Unpublished Report, American Red Cross, Washington, D. C.

T. L. Porter, D. E. Mann, and N. Acquista
Emission Spectrum of CF
Unpublished work of the Molecular Spectroscopy Section, NBS.

Takehiko Shimanouchi and Tao Suzuki
Method of Adjusting Force Constants and its Application to to H_2O, H_2CO, CH_2, Cl_2, and their Deuterated Molecules.
J. Chem. Phys. Vol 41, (1964)

3. DISCUSSION OF THE OMNITAB COMMANDS

The basic instructions of OMNITAB are given on page 20 for a single operation, and on page 21 for a limited double operation. On page 20 only one mathematical operation is possible, after which the number must be stored in some column. The storage operation replaces what was in the column.

On page 21 we have added a multiplication operation (it is never anything else but multiplication). The third operation of the sentence, which appears to be an ADD instruction, is really a storage instruction except that the storage is accomplished without clearing out the previous result. This storage is therefore equivalent to an addition operation. We have used the words AND ADD TO to remind the user of this fact. The addition is, however, automatic as is also the multiplication. Only in the first word is the user at liberty to choose any of the mathematical or manipulative operations. After that, the program determines how many numbers there are in the sentence, at which time it knows whether the sentence is for a single operation, a double operation type, or for a special subroutine.

The cumulative multiplication in the double operation has been provided to facilitate the evaluation of polynomials or series, so that, when computed, a term can be added to the previous terms without resorting to another instruction for each term. An instruction is a line written on a coding sheet or punched on a card. If the automatic addition feature is not wanted, the result must be put into a blank column.

From the foregoing it will be seen that for each mathematical operation there are generally two types of sentences: one, where the operation is performed and its result is stored by erasing what was previously in the column, and the other where the result is multiplied by some value, after which it is stored by adding to the values previously stored in the designated column. The two types of sentences apply only to the arithmetic operations and the elementary functions listed on page 20 and 21.

All of this is achieved simply by having the program distinguish how many parameters are contained on the card and whether they carry a decimal point or not. Thus the crucial information on any OMNITAB instruction is a legitimate vocabulary word, and a series of numbers appropriate to the operation involved.

A current list of commands is part of the machine program. The use of the instruction COMMANDS will cause them to be printed out.

3.1. Input Instructions

1. OMNITAB
   a. must be the first card following the system control cards.
   b. clears out the working storage and initializes the program parameters, hence is required between problems.
   c. this card should carry a date containing at least two numbers.

2. °READ COL ++,++,++,++, ...
   a. cards that follow are treated as data and read into columns°° specified by the READ instruction.
   b. card columns 1 through 72 are read.
   c. READ is terminated when any word in the OMNITAB vocabulary is encountered.°
   d. if later data fields are missing on any data card, the data from the preceding card are used.
   e. a maximum of 101 data cards is stored and additional data cards are ignored, except for triggering a diagnostic.
   f. if a decimal point or E is not contained in the number the decimal is placed at the extreme right of the number. Any number larger than 131071 must contain a decimal point or an E.

3. °GENERATE ARGUMENTS **(**)**(**)** IN COL ++
   a. generates a column of numbers in indicated increments.
   b. a maximum of 101 numbers is generated in the column.

- - - - - - - - - -

° The program automatically counts the number of cards after the READ or the length of the column GENERATED or SET, and sets NRMAX equal to this length if NRMAX was previously smaller. Except for matrix and array operations, NRMAX controls how far down the column subsequent operations are carried out.
°° Unless modified by the word "card", "column" refers to a column in the work sheet (101x46).

c. the parentheses enclose the increments in accord with common notation.

4. °SET IN COL ++,THE ARG ON THE FOLLOWING CARDS
a. reads data cards and stores columnwise.
b. set is terminated when an executable statement is encountered.
c. if a decimal point or E is not contained in the number the decimal is placed at the extreme right of the number. (See 2f above.)
d. a maximum of 101 numbers will be read; each card may contain as many as 25 numbers.

5. FREAD ,, CARDS INTO COLS ++, ++, ++, ...
a. required only for data punched without spaces separating the numbers or when it is desired to store non-numeric information (words or labels) in columns.
b. this instruction must be preceded by a FORMAT statement and must contain the count of the number of cards to follow.
c. its use is to be avoided except for fancy format control.
d. the data following this instruction are not printed out at the end of the calculation as are the data following a READ or SET instruction.

° The program automatically counts the number of cards after the READ or the length of the column GENERATED or SET, and sets NRMAX equal to this length if NRMAX was previously smaller. Except for matrix and array operations, NRMAX controls how far down the column subsequent operations are carried out.

3.2. Ordinary Output Instructions

1. PRINT COL ++, ++, ++, ... (8 col limit).
   a. NRMAX values in the specified columns are printed fixed or floating point depending on 2 or 3 below.
   b. If the PRINT instruction is not preceded by the FIXED or FLOATING instructions, the output is printed in floating format with one digit to the left and six digits to the right of the decimal point.
   c. titles and column headings are also printed.
   d. when no values are entered in a column or a portion of it, zeros will be printed.

2. FIXED WITH ,, DECIMALS
   a. provides fixed field format with the specified decimal places (for example 1.7983, 275.01, etc.).
   b. maximum decimal field is 7.

3. FLOATING WITH ,, DECIMALS
   a. provides floating field format with the specified number of decimal places (e.g. - 1.7983E01, 2.7501E02).
   b. maximum decimal field is 7.

4. PUNCH COL ++, ++, ++, ++, (4 col limit)
   a. provides NRMAX cards of output in accord with the built-in FORMAT.
   b. punching is limited to 60 characters including spaces.

5. PLOT COLS ++, ++, ++, AGAINST COL ++
   a. plots of one to five functions are provided.
   b. scale ranges are chosen by the machine from the coordinates.
   c. The assignment of plots symbols, depends upon the number of columns plotted. Thus:

	1st	2nd	3rd	4th	5th
1 COLUMN	.				
2 COLUMNS	*	.			
3 COLUMNS	+	*	.		
4 COLUMNS	,	+	*	.	
5 COLUMNS	0	,	+	*	.

   d. Where symbols would coincide, priority is given in the order . * + , 0 .

PLOT COLS ++, ++, AGAINST ++, ABSC FROM ** TO ** ORD FROM ** TO **
   a. same as above except that ranges for the abscissa and ordinate are provided by the user.

6. NOSUMMARY
   a. suppresses the automatic summary of the printed columns.
   b. can be counteracted by the instruction SUMMARIZE.

7. SUMMARIZE
   a. provides a one page summarization of the results with the execution of a PRINT statement. (See fig. 4-8)
   b. required only to counteract the NOSUMMARY instruction.

8. TITLE1 (66 characters maximum).
   TITLE2 (66 characters maximum).
   a. information in card col 7-72 is printed, left adjusted, as the first or second line or both on all pages obtained via a PRINT or FPRINT instruction.
   b. the words TITLE1 etc. must start in col 1 of the card.
   c. card column 7 must not contain either E, ., +, -, or *.
   d. titles can be updated at any time with a new title card.
   e. they may appear anywhere in the program prior to the print instruction to which they apply.
   f. the title starts on the second type bar position and extends to the 67th type bar on the printer.

9. TITLE3 (53 characters maximum)
   TITLE4 (53 characters maximum)
   a. provide a continuation across the page of TITLE1 and TITLE2, respectively.
   b. comments b, c, d and e of item 8 apply here also.
   c. titles 3 and 4 start on type bar position 68 and extend through the 120th.

10. HEAD COL ++/ (12 characters allowed after /)
   a. provides a heading for the designated column in place of the standard heading.
   b. heading will be printed only if the PRINT instruction is used.
   c. can be updated by another HEAD instruction.

11. NOTE-- (as appropriate)°
   a. information punched into card columns 7-72 of this card is printed as a single line at the time it is read.
   b. any number of NOTE cards may be used to provide text or documentation information either preceding, following, or interspersed between computation results obtained via one of the PRINT instructions.

° - denotes a blank.

12. FOOTNOTE---- °
   a. provides 1 line of information as a footnote for all subsequent PRINT instructions unless updated or deleted.
   b. the word FOOTNOTE followed by blanks deletes the FOOTNOTE instruction.

13. ABRIDGE ROW ,, COL ++, ++, ++, ++, (8 col limit)
   a. prints the designated row.
   b. may be repeated any number of times.
   c. prints according to the built-in FORMAT statement.

14. STOP
   a. required at the end of a series of OMNITAB runs.
   b. writes the latest instruction set on the output tape and returns control to IBSYS.

3.3. Special Output Instructions

1. FORMAT (°° as appropriate °°)
   a. provides tailor-made format according to FORTRAN rules°°.
   b. no limit to the number of columns except that 120 characters cannot be exceeded on any printed line or 72 characters on any card.
   c. use FPRINT only to obtain specified format.
   d. for further instructions, see a FORTRAN Manual.

2. FPRINT COL ++, ++, ++, ++, ... (as allowed by format)
   a. same as PRINT except that no headings are printed.
   b. can be used only when a FORMAT instruction preceded it.

3. FPUNCH COLS ++, ++, ++, ... (as allowed by format)
   a. must be used to obtain results in accord with a specific FORMAT.
   b. punching is limited to 72 characters including spaces on each card.

4. FABRIDGE ROW ,, COL ++,++,++ etc.
   a. prints the designated row in accordance with instructions on the current FORMAT card.
   b. may be repeated any number of times.

- - - - - - - - - - -

° - denotes a blank.

°° The novice need not be concerned with formating in order to use OMNITAB.

5. TPRINT COL ++,++,++ .... (as allowed by format)
   a. performs the same function as FPRINT but does not call for a new page.

6. NEWPAGE
   a. ensures printing on a new page.
   b. used largely with ABRIDGE or TPRINT instruction.

7. SPACE ,,
   a. provides ,, blank lines where desired in the output.
   b. used largely with NOTE, ABRIDGE and TPRINT instructions.

8. PAGE NUMBER AT LOCATION ,,
   a. provides for the printing of a page number at the top of the page in the designated type-bar location as well as extreme left-hand side. The OMNITAB card usually printed is omitted.
   b. pages are numbered starting with the current page number.

9. PAGE STARTING WITH ,, AT LOCATION ,,
   a. performs the same operation as the above instruction except that paging starts with the designated number.

10. NOLIST
   a. prevents listing of the input commands and data after the job is completed.
   b. should be placed immediately before the STOP card or the next OMNITAB card.

11. MKSA
   a. use of this instruction supplies fundamental constants in the MKSA system when called for. (See chapter 12)
   b. if the summary has not previously been suppressed, the list of fundamental constants is printed out. (See section 12.2)

12. CGS
   a. use of this instruction supplies fundamental constants in the CGS system when called for.
   b. if the summary has not previously been suppressed, the list of fundamental constants is printed out. (See section 12.2).

13. COMMANDS
   a. causes the OMNITAB vocabulary and sentence structure to be listed as contained in the current version of the program.
   b. this material is kept current with the program additions.

14. MANUAL

a. provides for a listing of an abridged version of section 3 of this Handbook.

b. this material is kept current with revisions or additions to the program.

15. WATSNU

a. lists all changes and additions since this edition of this Handbook.

16. SPLOT ++,++,++, ETC AGAINST ++ or
SPLOT ++,++,++, ETC AGAINST ++, ABSC FROM ** TO **, ORDINATE FROM ** TO **

a. provides a plot with higher resolution than does PLOT.

b. only one symbol is used but as many as 20 curves may be drawn on the same plot.

c. see section 4.5 for further discussion of this instruction.

3.4. Function Generation

1. LOGE, LOGTEN (Base e and 10 respectively)

a. if $x \leq 0$, the program prints out an appropriate error message and takes the log of the absolute value of x.

2. DIVIDE

a. on division by zero, a zero is returned, and the calculation proceeds.

3. RAISE

a. provides for exponentiations of the form a^b where either or both a and b can refer to column numbers or particular constants.

b. when a is negative and b is nonintegral, the program truncates b to the next lowest integer, prints out a diagnostic statement, and proceeds with the calculations. The results are thus obviously not correct for such cases.

4. EXP

a. e^x will be out of range for $x \geq 87.3$ and the program will stop with a suitable error statement.

b. if $x \leq -87.3$, underflow will occur, a zero will be substituted, and the calculation will continue without any notification.

5. SIN, COS, etc.
   a. SIN, COS, SIND, COSD compute the trigonometric functions for the angle in radians or degrees respectively.
   b. for arguments above 900 radians the accuracy diminishes by 2 or more places, resulting in a diagnostic but no interruption of the calculations.
   c. if x is larger than $2\pi \cdot 2^{19}$ degrees, the program stops with an appropriate error statement.

6. ARCSIN, ARCCOS, ... ASIND, ACOSD ....
   a. the inverse functions provide the principal value (0 to $+\pi/2$ for arc sin and arc cos and $-\pi/2$ to $+\pi/2$ for arc tan and arc cot).
   b. the D denotes an angle in degrees.
   c. x may be as large as 2^{128} for the arctan.

7. SINH, COSH, etc. denote hyperbolic functions while ASINH, ACOSH, etc., denote their inverses.

3.5. Manipulative Instructions

1. EXCHANGE COL ++, WITH ++, ++ WITH ++, etc
   a. interchanges NRMAX values of the designated columns pair by pair.

2. ERASE COLS ++, ++, ++, etc
   a. replaces NRMAX values of the designated columns by zeros.
   b. erases the entire work sheet when no columns are specified, and sets NRMAX to zero.
   c. unlike the word OMNITAB, it does not initialize the program parameters or destroy the stored instructions.

3. DELETE ROWS ,, ,, ,, ,, etc
   a. replaces the designated rows by zeros across the whole array.

4. TRANSPOSE ROW ,, INTO COL ++, ,, INTO ++, ,, INTO ++, etc
   a. stores the values in the designated rows into the designated columns.
   b. the transposed vector starts in the first row of the column.
   c. up to five rows can be transposed in one instruction.
   d. the new column is only NRMAX long.

5. FLIP COL ++ INTO COL ++, ++ INTO ++, ++ INTO ++
   a. turns a column of NRMAX elements end for end, up to NRMAX.
   b. without parameters this flips the entire array up to NRMAX.

6. PROMOTE ARRAY ,, ROWS
   a. discards the designated number of lines from the top of each of the 46 columns and moves the array up.

   b. leaves zeros in the vacated elements.
   c. leaves NRMAX unchanged.

7. DEMOTE ARRAY ,, ROWS
   a. depresses the entire array the designated number of rows, and increases NRMAX.
   b. provides duplication of the first ,, rows.

8. PROMOTE BY ,, ROWS, COL ++ INTO COL ++,
   COL ++ INTO ++, etc
   a. promotes columns selectively and stores as desired.
   b. leaves intact the original column, the remainder of the storage column, and NRMAX.

9. DEMOTE BY ,, ROWS, COL ++ INTO COL ++,
   COL ++ INTO ++, etc
   a. demotes NRMAX values from each of the designated columns by ,, rows and stores as indicated leaving intact the original column and the top ,, rows of the storage column.
   b. NRMAX is increased by ,, but only after all of the columns have been demoted.

10. RESET NRMAX TO ,, ROWS
   a. as NRMAX controls how far down the column subsequent instructions operate,this command provides a simple way of altering this important program parameter.

   RESET NRMAX TO THE VALUE IN ROW ,, OF COL ++

   b. enables NRMAX to be set to a computed value n.
   c. if $0. \leq n \leq 101.$, NRMAX is set to the integer equal to or less than n.
   d. if n is greater than 101. or negative, the program stops with the appropriate error statement.

11. DEFINE ROW ,, COL ++, AS ROW ,, COL ++
a. moves one element of the stored array into any location in the array.
b. the original element is not destroyed.

12. DEFINE ROW ,, COL ++ AS COL ++
a. vectorizes the designated element into the designated column so as to make available NRMAX identical values.
b. the original element is not destroyed.

13. DEFINE ** AS ROW ,, COL ++
a. replaces the designated element by the designated number.

14. SHORTEN COL ++ FOR COL ++ =**, STORE IN COL ++ AND ++
a. shortens the designated column at and including a particular value in that column or another column and stores both shortened columns respectively as indicated.
b. if the value is not found exactly, shortens at the next highest number if the function increases or the next lower number if the values decrease (the function is assumed monotonic).
c. resets NRMAX to agree with the shortened column.
d. if the value is not found, a diagnostic is printed and the columns are transferred to the storage location in accordance with NRMAX.

15. CLOSE UP ROWS HAVING ** IN COLS ++, ++, ++.....
a. deletes in each column those rows containing the indicated value and closes up the resultant spaces.
b. puts zeros in the vacated rows at the bottom of each of the columns up to NRMAX.

16. SEPARATE FROM COL ++ EVERY ,, ROW START WITH ROW ,, STORE IN COL ++
a. extracts every nth value from the designated column starting at the mth row, and stores these results consecutively in the indicated column starting in row one.
b. NRMAX is left unchanged as is the original column.

17. INSERT IN COL ++ FROM COL ++ AT EVERY ,, ROW STARTING AS ,, ROW, STORE IN COL ++
a. performs the reverse of the SEPARATE instruction.
b. inserts current NRMAX or fewer values from the top of the designated column into the specified locations in the first named column starting at the designated row, and stores the composite column as indicated.
c. NRMAX is increased by the number of inserted rows up to a maximum of 101. (See appendix I)
d. if the list to be inserted is shorter than NRMAX and is followed by zeros, these values can be deleted by using the CLOSE UP instruction.

3.6. Arithmetic Operations

1. CHANGE SIGN OF COLS ++, ++, ++, ...
   a. changes the signs of NRMAX values in the designated columns.

2. ROWSUM COLS ++, ++, ++, ... AND STORE IN COL ++
   a. performs the designated additions for each row of the array, and stores the sum in the corresponding row of the indicated column.

   ROWSUM ARRAY, STORE IN COL ++
   a. provides a column of NRMAX values each row of which is the sum of the corresponding row of the 46 column array.

3. SUM COL ++, STORE IN COL ++
   a. sums the first mentioned column and stores this sum NRMAX times in the indicated column.

4. SUM COL ++, ROWS ,, TO ,, STORE IN COL ++
   a. sums the indicated column for the consecutive inclusive rows indicated, and stores the result NRMAX times in the indicated column.

5. SUM COL ++, ROWS ,, ,, ,, ... STORE IN COL ++
   a. sums the designated column for the discrete rows indicated and vectorizes it as directed.°
   b. a minimum of 3 discrete rows must be specified, except that a blank row can be specified to achieve the addition of only two rows.

6. AVERAGE COL ++, STORE IN COL ++
   a. the arithmetic mean is computed and vectorized as indicated.
   b. assumes NRMAX to be the length of the column unless three consecutive zeros are encountered before NRMAX is reached, in which case the actual length above the three zeros is used.

7. RMS OF COL ++, STORE IN COL ++
   a. computes the root mean square of NRMAX values in the designated column and vectorizes it in the indicated column.

- - - - - - - - - - -

° vectorize here means to store the results NRMAX times.

8. MAXIMUM OF COL ++, STORE IN COL ++
   a. vectorizes into the designated column the maximum value of the first named column.
   b. the function need not be monotonic.

9. MAXIMUM OF COL ++ STORE IN COL ++, STORE CORRESP VALUE OF ++ IN ++
   a. vectorizes the maximum value in the designated column and the corresponding value (the value in the same row) in another column and stores these results as directed.
   b. the function need not be monotonic.

10. MINIMUM OF COL ++, STORE IN COL ++
   a. vectorizes into the designated column the minimum value in the first named column.
   b. the function need not be monotonic.

11. MINIMUM OF COL ++, STORE IN COL ++, STORE CORRESP VALUE OF ++ IN ++
   a. vectorizes the minimum value of the designated column and the corresponding value (the value in the same row) in another column and stores these results as directed.
   b. the function need not be monotonic.

12. DIAGONALIZE $$ INTO COL ++ THRU ++
   a. stores the constant or designated column as the principal diagonal of a matrix defined by the designated consecutive columns.
   b. the off-diagonal terms are unaffected, as is the value of NRMAX.

13. SORT COL ++ CARRY ALONG ++, ++, ++, ...
   a. sorts the numbers in the designated column in increasing order and carries along the element in the corresponding row of the designated columns.
   b. results remain in the original columns.

14. ORDER COL ++, ++, ++, ...
   a. arranges the values in each column independently in increasing order.

15. PARSUM COL ++, STORE IN COL ++
   a. provides NRMAX partial sums $s_n = \sum_{i=1}^{n} X_i$, $1 \leq n \leq$ NRMAX and stores these in the designated column.

16. PARPRODUCT COL ++, STORE IN COL ++
   a. provides NRMAX partial products $\prod_i X_i$ and stores these in the designated column.

b. if the numbers in the column are consecutive integers (starting with 1 or 2), the nth row will contain n factorial.
c. if the product exceeds 10^{38}, zeros are returned for the rest of the column.

17. LENGTH OF ARRAY TO BE STORED IN COL ++
a. enter the current value of NRMAX into the designated column.

18. COUNT LENGTH OF COL ++, STORE IN COL ++
a. counts up to NRMAX or up to, but not including, the first of 3 consecutive zeros.
b. the count is vectorized as indicated.

19. SINTEGRAL OF COL ++, X IN COL ++, STORE IN ++
a. computes the integral

$$\int_{x_1}^{x_n} f(x)\, dx = \sum_{i=1}^{n} h\, a_i y_i$$

for NRMAX values of the function y_i computed at uniform values of the variable.
b. the integration is equivalent to a four point Lagrangian integration.
c. if the arguments are in column one, the result should agree with the integral given in the OMNITAB Summary.
d. the interval, h, is determined from the designated argument column.
e. the coefficients a_i are unity except for the ends of the column where they are 0.348611111, 1.24583333, 0.879166667, and 1.06638889 respectively for a_1, a_2, a_3, a_4 and for a_n, a_{n-1}, a_{n-2}, and a_{n-3}.

f. the number of values must obviously be greater than 8.

20. Complex Arithmetic

CADD, REAL IN ++, TO REAL IN ++, STORE REAL IN ++
CSUB, REAL IN ++ FROM REAL IN ++, STORE REAL IN ++
CMULT, REAL IN ++ BY REAL IN ++, STORE REAL IN ++
CDIV, REAL IN ++ BY REAL IN ++, STORE REAL IN ++

a. provides for complex arithmetic of numbers whose real parts are stored in the indicated columns.
b. the imaginary part of each of the three complex numbers is assumed to be in the next higher column.
c. CMULT and CDIV cannot be stored over the original values.

3.7. Special Functions

1. TSUB ,, OF COL ++
2. USUB ,, OF COL ++
3. PSUB ,, OF COL ++
4. HSUB ,, OF COL ++
5. LSUB ,, OF COL ++

a. These commands represent respectively the following polynomials, with n= 1, 2, 3, ... 40:

the Chebyshev polynomials,°

$$T_n(x) = \cos(n \cos^{-1} x) \qquad \text{for } -1 \leq x \leq +1.$$

$$U_n(x) = [T_{n+1}(x)] / (n+1)$$

the Legendre polynomials,°°

$$P_n(x) = \frac{1}{2^n n!} \frac{d^n}{dx^n} [x^2 - 1]^n$$

the Hermite polynomials,°°°

$$H_n(x) = (-1)^n e^{x^2} \frac{d^n}{dx^n} (e^{-x^2})$$

the Laguerre polynomials,°°°°

$$L_n(x) = \frac{e^x}{n!\sqrt{x}} \int_0^{\infty} e^{-t} t^{(n+1/2)} J_1(2\sqrt{tx})\, dt$$

° Anonymous. Tables of Chebyshev Polynomials Sn(x) and Cn(x), National Bureau of Standards, Applied Mathematics Series 9 U. S. Government Printing Office, Washington 25, D. C. (1952).

°° Kopal, Z., Numerical Analysis, p. 368, John Wiley & Sons, New York (1955).

°°° Kopal, Z., Numerical Analysis, p. 371, John Wiley & Sons, New York (1955).

°°°° Abramowitz, M., and Stegun, I. A., Handbook of Mathematical Functions, National Bureau of Standards Applied Mathematics Series 55, Superintendent of Documents, U. S. Government Printing Office, Washington, D. C. 1964.

b. These polynomials are computed from the initial values and the recursion relations given in table A. The program computes consecutively the polynomials from 1, 2, 3, 4....n and stores the values of $F_1(x)$ in column 46, $F_2(x)$ in column 45, etc., until .... $F_n(x)$ which is stored in column (47−n). (See table B)

TABLE A. Initial Values and Recursion Formulas for Certain Special Functions

Symbol	n = 0	n = 1	n = 2	n = 3
$T_n(x)$	1	x	$2x^2-1$	$4x^3-3x$
$U_n(x)$	1	2x	$4x^2-1$	$8x^3-4x$
$P_n(x)$	1	x	$\frac{3}{2}x^2-\frac{1}{2}$	$\frac{5}{2}x^3-\frac{3}{2}x$
$H_n(x)$	1	2x	$4x^2-2$	$8x^3-12x$
$L_n(x)$	1	−x+1	$\frac{x^2-4x+2}{2}$	$\frac{-x^3+9x^2-18x+6}{6}$

Recursion Formula

$$T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x)$$

$$U_{n+1}(x) = 2xU_n(x) - U_{n-1}(x)$$

$$P_{n+1}(x) = xP_n(x) + \frac{n}{n+1}[xP_n(x) - P_{n-1}(x)]$$

$$H_{n+1}(x) = 2xH_n(x) - 2nH_{n-1}(x)$$

$$L_{n+1}(x) = [(1+2n-x)L_n(x) - nL_{n-1}(x)]/(n+1)$$

6. STRUVE SUB 0 OF COL ++, STORE IN COL ++
STRUVE SUB 1 OF COL ++, STORE IN COL ++
   a. computes the Struve functions $S_0(x)$ and $S_1(x)$.

defined as follows°:

$$S_0(x) = \frac{2}{\pi}\int_0^{\pi/2} \sin(x \sin\theta)\, d\theta$$

$$S_1(x) = \frac{2}{\pi} - \frac{2}{\pi}\int_0^{\pi/2} \cos(x \sin\theta)\, \sin\theta d\theta$$

7. SININT OF $$, STORE IN COL ++
   a. computes the sine integral.

$$S(x) = \int_0^{x} \frac{\sin t}{t}\, dt$$

8. COSINT OF $$, STORE IN COL ++
   a. computes the cosine integral.

$$C(x) = -\int_x^{\infty} \frac{\cos t}{t}\, dt$$

9. EINT OF $$, STORE IN COL ++
   a. computes the exponential integral.

$$E(x) = \int_x^{\infty} \frac{e^{-u}}{u}\, du$$

10. COMPLETE ELLIPTIC INTEGRAL K = $$, STORE FIRST KIND IN ++, SECOND IN ++
   a. computes the complete elliptic integral of the first and second kind as follows:

the first kind:

$$K(k) = \int_0^{\pi/2} \frac{d\phi}{\sqrt{1-k^2\sin^2\phi}}$$

- - - - - - - - - - -

° Dwight, H. B., Tables of Integrals and Other Mathematical Data, pp. 199, 203, The Macmillan Co., New York (1961).

and the second kind

$$E(k) = \int_0^{\pi/2} \sqrt{1-k^2\sin^2\phi}\; d\phi$$

for the specified values of $k = \sin\theta$

11. INCOMPLETE ELLIP INTEG K = $$, PHI = $$, FIRST KIND IN ++, SECOND IN ++
   a. computes the incomplete elliptic integral $F(k,\phi)$ of the first kind and $E(k,\phi)$ of the second kind where

$$F(k,\phi) = \int_0^{\phi} \frac{d\phi}{\sqrt{1-k^2\sin^2\phi}}$$

and

$$E(k,\phi) = \int_0^{\phi} \sqrt{1-k^2\sin^2\phi} \cdot d\phi$$

where $k = \sin\theta$.

12. ERROR OF $$, STORE IN COL ++
   a. generates the error function.

$$\mathrm{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt$$

13. CERF OF $$, STORE IN COL ++
   a. gives $[1 - \mathrm{erf}(x)]$.

14. GAMMA OF $$, STORE IN COL ++
   a. generates the gamma function.

$$\Gamma(n) = \int_0^\infty x^{n-1}e^{-x}\, dx \quad (\text{for } n > 0)$$

For n a positive integer, $\Gamma(n) = (n-1)!$

3.8. Bessel Functions°

1. BEJZERO OF $$, STORE IN ++
   BEJZERO is the Bessel function of the first kind of zero order $J_0(x)$

2. BEJONE OF $$, STORE IN ++
   BEJONE is the Bessel function of the first kind of first order $J_1(x)$

3. BEYZERO OF $$, STORE IN ++
   BEYZERO is the Bessel function of the second kind of zero order $Y_0(x)$

4. BEYONE OF $$, STORE IN ++
   BEYONE is the Bessel function of the second kind of first order $Y_1(x)$

5. BEIZERO OF ++, STORE IN ++
   BEIZERO OF ++, OPTION 1, STORE IN ++
   BEIZERO is the modified Bessel function of the first kind of zero order $I_0(x)\exp(-x)$

6. BEIONE OF ++ STORE IN ++
   BEIONE OF ++, OPTION 1 STORE IN ++
   BEIONE is the modified Bessel function of the first kind of first order $I_1(x)\exp(-x)$

7. BEKZERO OF ++, STORE IN ++
   BEKZERO OF ++, OPTION 1, STORE IN ++
   BEKZERO is the modified Bessel function of the second kind of zero order $K_0(x)\exp(x)$

8. BEKONE OF ++ STORE IN ++
   BEKONE OF ++, OPTION 1, STORE IN ++
   BEKONE is the modified Bessel function of the second kind of first order $K_1(x)\exp(x)$

9. BERBEI SUB 0 OF ++, STORE IN ++ AND ++ (SCALED)
   BERBEI SUB 1 OF ++, STORE IN ++ AND ++ (SCALED)
   BERBEI SUB 0 OF ++, OPTION 1, STORE IN ++ AND ++ (UNSCALED)
   BERBEI SUB 1 OF ++, OPTION 1, STORE IN ++ AND ++ (UNSCALED)

   BERBEI ,, gives the real part, $e^{-x/\sqrt{2}}\,\mathrm{ber}_n x$, and the imaginary part $e^{-x/\sqrt{2}}\mathrm{bei}_n x$ of the Bessel-Kelvin function of argument $xi\sqrt{i}$ of the first kind:

° Option 1 removes the exponential scale factors.

$$J_n(xi\sqrt{i}) = i^n I_n(x\sqrt{i}) = ber_n x + i\, bei_n x$$

for n = ,, = 0 or 1.

KERKEI SUB 0 OF ++, STORE IN ++ AND ++ (SCALED)
KERKEI SUB 1 OF ++, STORE IN ++ AND ++ (SCALED)
KERKEI SUB 0 OF ++, OPTION 1, STORE IN ++ AND ++ (UNSCALED)
KERKEI SUB 1 OF ++, OPTION 1, STORE IN ++ AND ++ (UNSCALED)

KERKEI ,, gives the real part, $e^{+x/\sqrt{2}}\, ker_n x$, and the imaginary part, $e^{+x/\sqrt{2}}\, kei_n x$ of the Bessel-Kelvin function of argument $xi\sqrt{i}$ of the second kind

$$i^{-n} K_n(x\sqrt{i}) = ker_n x + i kei_n x$$

for n = ,, = 0 or 1.

10. DBERBEI SUB 0 OF ++, OPTION 1, STORE IN ++ AND ++
DBERBEI SUB 1 OF ++, OPTION 1, STORE IN ++ AND ++
DKERKEI SUB 0 OF ++, OPTION 1, STORE IN ++ AND ++
DKERKEI SUB 1 OF ++, OPTION 1, STORE IN ++ AND ++ (UNSCALED)

DBERBEI ,, and DKERKEI ,, give

$$e^{-x/\sqrt{2}}\, ber'_n x, \quad e^{-x/\sqrt{2}}\, bei'_n x$$

and

$$e^{-x/\sqrt{2}}\, ker'_n x, \quad e^{-x/\sqrt{2}}\, kei'_n x$$

where $ber'_n x = \frac{d}{dx}(ber_n x)$ etc.

The inclusion of the option number in the instructions for the Bessel-Kelvin functions removes the factor $e^{-x/\sqrt{2}}$ thereby bringing the results in accord with the functions ber, bei, ker, and kei as tabulated by Dwight° and Lowell°°.

- - - - - - - - - - -

° Dwight, H. B., Tables of Integrals and Other Mathematical Data, pp. 199, 203, The Macmillan Co., New York (1961).

°° Lowell, H. H., Tables of the Bessel-Kelvin Functions ber, bei, ker, kei and their Derivatives for the Argument Range 0(0.01)107.50, National Aeronautics and Space Administration Technical Report R-32 (NASA TR R-32), Superintendent of Documents, U.S. Government Printing Ofice, Washington 25, D. C.

3.9. Sample Problems in the Ordinary Operation Mode

Problem 3 - 1

Compute the Einstein functions

$$-G = -\ln(1 - e^{-x})$$

$$H = x\,e^{-x} / (1 - e^{-x})$$

$$C = x^2 e^{-x} / (1 - e^{-x})^2$$

$$S = -G + H$$

for x = 1.01(.01)1.5

The OMNITAB instructions are as follows:

```
OMNITAB  PROBLEM  3 - 1
TITLE1 EINSTEIN FUNCTIONS
GENERATE 1.01(.01)1.5 IN COL 1
NEGEXP OF COL 1,STORE IN COL 2
MULTIPLY COL 2 BY -1. STORE IN 3
ADD 1. TO COL  3    STORE IN 3
LOGE OF COL 3,MULT BY -1.,ADD INTO 4
RAISE COL 3 TO -1.,MULT BY COL 2,ADD 5
MULTIPLY COL 5 BY COL 1,STORE IN 5
ADD COL 4 TO COL  5  STORE IN COL 6
DIVIDE COL5 BY COL2,MULT BY 5,ADD 7
HEAD COL 1/         X
HEAD COL 4/         G
HEAD COL 5/         H
HEAD COL 6/         S
HEAD COL 7/       CSUBP
FIXED POINT  5 DECIMALS
PRINT  1,4,5,6,7
        STOP
```

See opposite page for results and chapter 9 for a more succinct solution to this problem.

EINSTEIN FUNCTIONS

X	G	H	S	CSUBP
1.01000	0.45290	0.57860	1.03150	0.91916
1.02000	0.44722	0.57523	1.02245	0.91763
1.03000	0.44162	0.57188	1.01350	0.91609
1.04000	0.43611	0.56855	1.00466	0.91454
1.05000	0.43069	0.56523	0.99592	0.91298
1.06000	0.42535	0.56193	0.98727	0.91140
1.07000	0.42008	0.55864	0.97872	0.90981
1.08000	0.41490	0.55536	0.97026	0.90822
1.09000	0.40980	0.55210	0.96190	0.90661
1.10000	0.40477	0.54886	0.95363	0.90499
1.11000	0.39982	0.54563	0.94545	0.90335
1.12000	0.39494	0.54241	0.93735	0.90171
1.13000	0.39013	0.53921	0.92934	0.90006
1.14000	0.38540	0.53602	0.92142	0.89839
1.15000	0.38073	0.53285	0.91358	0.89671
1.16000	0.37613	0.52970	0.90583	0.89503
1.17000	0.37160	0.52655	0.89815	0.89333
1.18000	0.36713	0.52343	0.89055	0.89162
1.19000	0.36272	0.52031	0.88304	0.88990
1.20000	0.35838	0.51722	0.87560	0.88817
1.21000	0.35410	0.51413	0.86823	0.88643
1.22000	0.34988	0.51106	0.86095	0.88468
1.23000	0.34572	0.50801	0.85373	0.88292
1.24000	0.34162	0.50497	0.84659	0.88115
1.25000	0.33758	0.50194	0.83952	0.87937
1.26000	0.33359	0.49893	0.83252	0.87758
1.27000	0.32966	0.49593	0.82559	0.87577
1.28000	0.32578	0.49294	0.81873	0.87396
1.29000	0.32196	0.48998	0.81193	0.87214
1.30000	0.31818	0.48702	0.80520	0.87031
1.31000	0.31446	0.48408	0.79854	0.86847
1.32000	0.31079	0.48115	0.79195	0.86663
1.33000	0.30717	0.47824	0.78541	0.86477
1.34000	0.30360	0.47534	0.77894	0.86290
1.35000	0.30008	0.47245	0.77253	0.86102
1.36000	0.29660	0.46958	0.76618	0.85914
1.37000	0.29317	0.46672	0.75990	0.85724
1.38000	0.28979	0.46388	0.75367	0.85534
1.39000	0.28645	0.46105	0.74750	0.85343
1.40000	0.28315	0.45824	0.74139	0.85151
1.41000	0.27990	0.45543	0.73534	0.84958
1.42000	0.27669	0.45264	0.72934	0.84764
1.43000	0.27353	0.44987	0.72340	0.84570
1.44000	0.27040	0.44711	0.71751	0.84374
1.45000	0.26732	0.44436	0.71168	0.84178
1.46000	0.26427	0.44163	0.70590	0.83981
1.47000	0.26127	0.43891	0.70017	0.83783
1.48000	0.25830	0.43620	0.69450	0.83584
1.49000	0.25537	0.43351	0.68888	0.83385
1.50000	0.25248	0.43083	0.68331	0.83185

Problem 3 - 2

Compute the ideal gas thermal functions for the electron from the equations

$$-(F^\circ - H^\circ_0)/RT = 2.5 \ln T + 1.5 \ln M_e + 3.665108$$

$$(H^\circ - H^\circ_0)/RT = 2.5$$

$$C^\circ_p/R = 2.5$$

$$S^\circ/R = -(F^\circ - H^\circ_0)/RT + (H^\circ - H^\circ_0)/RT$$

where M_e = .00054876

Tabulate results for T = 273.15, 298.15, 1000.(100.)10,000.°K.

The OMNITAB instructions are as follows:

```
OMNITAB    PROBLEM 3 - 2
TITLE1                              THERMAL FUNCTIONS          E-
GENERATE  273.15(25.)298.15(701.85)1000.(100.)10000.  IN COL 1
LOGE   COL 1  AND STORE IN COL 2
MULTIPLY  2.5  BY  COL 2 AND STORE IN COL 2
LOGE   .00054876  AND STORE IN COL 3
MULTIPLY    1.5 BY  COL 3 AND STORE IN COL 3
ADD  3.665108  TO COL 3 AND STORE IN COL 3
ADD COL 2 TO COL 3 AND STORE IN COL 3
ADD      2.5000  TO  COL 4  AND STORE IN  COL 4
ADD      2.5000  TO  COL 6  AND STORE IN  COL 6
ADD      COL3    TO  COL 4  AND STORE IN  COL 5
MULTIPLY COL 4 BY COL 1 AND STORE IN COL 7
HEAD   COL 1/  T K
HEAD   COL 3/-(F-HO)/RT
HEAD   COL 4/ (H-HO)/RT
HEAD   COL 5/    S/R
HEAD   COL 6/     CP/R
HEAD   COL 7/ (H-HO)/R
FORMAT   (1F10.2,4F12.4,1F12.1)
PRINT  1,3,4,5,6,7
FPRINT 1,3,4,5,6,7
FPUNCH 1,3,4,5,6,7
        STOP
```

Problem 3 - 3

Compute the vapor pressure of liquid hydrogen fluoride from the equations of Simons and of Hildebrand given below. Tabulate results at 10° intervals from $0^\circ C$ to $190^\circ C$ and in atmospheres as well as millimeters of mercury.

$$\log P\ (\text{mm Hg}) = 7.37 - 1315./T$$

$$\log P\ (\text{mm Hg}) = 7.3739 - 1316.79/T$$

The OMNITAB instructions are as follows:

```
OMNITAB  PROBLEM 3 - 3
TITLE1  VAPOR PRESSURE OF HF
GENERATE ARG 273.15(10.)463.15 IN COL 1
DIVIDE 1315. BY COL 1 STORE IN COL 2
SUBTRACT COL 2 FROM 7.37 AND STORE IN COL 3
RAISE 10.TO    COL 3  STORE IN COL 4
DIVIDE COL 4 BY 760. AND STORE IN COL 5
DIVIDE 1316.79 BY COL 1 STORE IN COL 6
SUBTRACT COL 6 FROM 7.3739 AND STORE IN COL 7
RAISE 10. TO COL 7 STORE IN COL 8
DIVIDE COL 8 BY 760. AND STORE IN COL 9
SUBTRACT 273.15 FROM COL 1 AND STORE IN COL 10
HEAD COL 1/DEG K
HEAD COL 4/SIMON (MMHG)
HEAD COL 5/ SIMON (ATM)
HEAD COL 8/HILDB.(MMHG)
HEAD COL 9/ HILDB.(ATM)
HEAD COL 10/  DEGREES C
PRINT 1,10,8,9
PRINT 1,10,4,5
PRINT 1,10,4,8,5,9
       STOP
```

Problem 3 - 4

Compute the current, I, from the relation

$$I = E[R^2 + (2\pi fL - \frac{1}{2\pi fC})^2]^{-1/2}$$

for E = 1 volt to E = 100 volts in steps of 1 volt, where

f = 1000 cps L = 0.03 henry

R = 150 ohms C = $4 \cdot 10^{-6}$ farad

The OMNITAB instructions are as follows:

```
OMNITAB  PRCBLEM 3 - 4
GENERATE 1.(1.)100.,1
MULTIPLY 3.14159 BY 2. MULT BY 1000. ADD TO 2
MULTIPLY 4.E-6 BY COL 2 STORE 3
DIVICE -1. BY 3 STORE 4
MULT 2 BY .03 BY 1. ACD TO 4
MULT 4 BY 4 STCRE 4
MULT 150. BY 150. MULT BY 1. ADD TO 4
SQRT OF 4,4
DIVICE 1 BY 4 STORE 5
HEAD CCL 1/         E
HEAD COL 5/         I
PRINT 1,5
       STOP
```

Problem 3 - 5

Compute a table of reciprocal temperatures as follows:

$$T = \frac{1000}{T + 0.1\ k}$$

for T = 100(1)200 and k = 0, 1, 2, ... 9 and print results in two pages, five columns to a page.

The OMNITAB instructions are as follows:

```
OMNITAB   PROBLEM  3 - 5
TITLE1 RECIPROCAL TABLE, 1000./T - T = 100.0 - 200.0 BY 0.1
GENERATE 100.(1.)200.,1
DIVIDE 1000. 1 2
ADD     0.1  1  20
DIVIDE 1000. 20 3
ADD   0.1 20 20
DIVIDE 1000. 20 4
ADD   0.1 20 20
DIVIDE 1000. 20 5
ADD   0.1 20 20
DIVIDE 1000. 20 6
ADD   0.1 20 20
DIVIDE 1000. 20 7
ADD   0.1 20 20
DIVIDE 1000. 20 8
ADD   0.1 20 20
DIVIDE 1000. 20 9
ADD   0.1 20 20
DIVIDE 1000. 20 10
ADD   0.1 20 20
DIVIDE 1000. 20 11
FORMAT (1H F9.0,10F10.5)
FPRINT 1,2,3,4,5,6
FPRINT  1,7,8,9,10,11
       STOP
```

See chapter 6 for a shorter instruction set.

Problem 3 − 6

Compute a table of vertical heights of an object which subtends an angle θ at a distance d. Use

$$h = d \tan \theta$$

for d = 0.(.01).5 meters for θ = 10.(10.)80.°

The OMNITAB instructions are as follows:

```
OMNITAB      PROBLEM  3 - 6
TITLE1     TABLE OF VERTICAL HEIGHTS
FIXED 6
NOSUMMARY
GENERATE  0. .01 .5 1
TAND 10. 1 2
TAND 20. 1 3
TAND 30. 1 4
TAND 40. 1 5
TAND 50. 1 6
TAND 60. 1 7
TAND 70. 1 8
TAND 80. 1 9
HEAD 1/       M
HEAD 2/  THETA = 10
HEAD 3/  THETA = 20
HEAD 4/  THETA = 30
HEAD 5/  THETA = 40
HEAD 6/  THETA = 50
HEAD 7/  THETA = 60
HEAD 8/  THETA = 70
HEAD 9/  THETA = 80
PRINT 1 2 3 4 5
PRINT 1 6 7 8 9
      STOP
```

See opposite page for results and chapter 6 for a more succinct instruction set.

PAGE 1 OMNITAB PROBLEM 3 - 6

TABLE OF VERTICAL HEIGHTS

M	THETA = 10	THETA = 20	THETA = 30	THETA = 40
0.	0.	0.	0.	0.
0.010000	0.001763	0.003640	0.005773	0.008391
0.020000	0.003527	0.007279	0.011547	0.016782
0.030000	0.005290	0.010919	0.017321	0.025173
0.040000	0.007053	0.014559	0.023094	0.033564
0.050000	0.008816	0.018199	0.028868	0.041955
0.060000	0.010580	0.021838	0.034641	0.050346
[illegible]70000	0.012343	0.025478	0.040415	0.058737
[illegible]	0.014106	0.029118	0.046188	0.067128
[illegible]	[illegible]15869	[illegible]	0.051962	0.075519

PAGE 3 OMNITAB PROBLEM 3 - 6

TABLE OF VERTICAL HEIGHTS

M	THETA = 50	THETA = 60	THETA = 70	THETA = 80
0.	0.	0.	0.	0.
0.010000	0.011918	0.017321	0.027475	0.056713
0.020000	0.023835	0.034641	0.054950	0.113426
0.030000	0.035753	0.051962	0.082424	0.170138
0.040000	0.047670	0.069282	0.109899	0.226851
0.050000	0.059588	0.086603	0.137374	0.283564
0.060000	0.071505	0.103923	0.164849	0.340277
0.070000	0.083423	0.121244	0.192323	0.396990
0.080000	0.095340	0.138564	0.219798	0.453703
0.090000	0.107258	0.155885	0.247273	0.510415
0.100000	0.119175	0.173205	0.274748	0.567128
0.110000	0.131093	0.190526	0.302222	0.623841
0.120000	0.143010	0.207846	0.329697	0.680554
0.130000	0.154928	0.225167	0.357172	0.737267

Problem 3 - 7

Add certain third virial corrections to a table of ideal- gas thermodynamic functions of nitrogen.

The OMNITAB instructions are as follows:

```
OMNITAB  PROBLEM  3 - 7
TITLE1 THERMO FUNCTIONS FOR NITROGEN INCLUDING 3RD VIRIAL CORRECTION
TITLE2 T = 200C
READ 31,1,2,3,4,5,6                              UNCORRECTED
4.   0.      1.0015    2.8972    3.8987   28.3089   7.33284E+00     2.0R
4.   0.20    1.0023    2.8972    3.8995   27.8475   1.16318E+01     2.0R
4.   0.40    1.0037    2.8971    3.9009   27.3855   1.84604E+01     2.0R
4.   0.60    1.0059    2.8971    3.9029   26.9227   2.93212E+01     2.0R
4.   0.80    1.0093    2.8969    3.9062   26.4586   4.66301E+01     2.0R
4.   1.00    1.0148    2.8967    3.9115   25.9924   7.43037E+01     2.0R
4.   1.20    1.0235    2.8963    3.9198   25.5229   1.18768E+02     2.0R
4.   1.40    1.0372    2.8957    3.9329   25.0481   1.90758E+02     2.0R
4.   1.60    1.0589    2.8948    3.9537   24.5649   3.08670E+02     2.0R
4.   1.80    1.0934    2.8934    3.9868   24.0684   5.05130E+02     2.0R
4.   2.00    1.1480    2.8911    4.0391   23.5510   8.40571E+02     2.0R
4.   2.20    1.2346    2.8874    4.1220   23.0003   1.43267E+03     2.0R
4.   2.40    1.3718    2.8816    4.2534   22.3968   2.52298E+03     2.0R
4.   2.60    1.5892    2.8725    4.4617   21.7096   4.63251E+03     2.0R
4.   2.80    1.9339    2.8579    4.7918   20.8900   8.93420E+03     2.0R
4.   3.00    2.4801    2.8349    5.3150   19.8602   1.81591E+04     2.0R
READ 1,12,13,14,15,30,16                         CORRECTIONS
 0.     0.00000    0.00000    0.00000   -0.00000   2.0    1.6356E-05
 0.2    0.00001    0.00000    0.00001   -0.00000   2.0    6.5115E-05
 0.4    0.00001    0.00000    0.00002   -0.00001   2.0    2.5923E-04
 0.6    0.00004    0.00000    0.00004   -0.00001   2.0    1.0320E-03
 0.8    0.00009    0.00001    0.00010   -0.00003   2.0    4.1085E-03
 1.0    0.00022    0.00003    0.00025   -0.00008   2.0    1.6356E-02
 1.2    0.00056    0.00007    0.00063   -0.00021   2.0    6.5115E-02
 1.4    0.00141    0.00018    0.00159   -0.00053   2.0    2.5923E-01
 1.6    0.00354    0.00045    0.00399   -0.00132   2.0    1.0320E+00
 1.8    0.00889    0.00112    0.01001   -0.00333   2.0    4.1085E+00
 2.0    0.02233    0.00282    0.02515   -0.00836   2.0    1.6356E+01
 2.2    0.05609    0.00709    0.06318   -0.02099   2.0    6.5115E+01
 2.4    0.14089    0.01780    0.15869   -0.05273   2.0    2.5923E+02
 2.6    0.35390    0.04472    0.39862   -0.13245   2.0    1.0320E+03
 2.8    0.88895    0.11234    1.00129   -0.33270   2.0    4.1085E+03
 3.0    2.23295    0.28218    2.51513   -0.83571   2.0    1.6356E+04
ADD 2,12,12
ADD 3,13,13
ADD 4,14,14
ADD 5,15,15
ADD 6,16,16
PRINT 1,12,13,14,15,16,30
NOSUMMARY
HEAD 1/         LOG
HEAD 12/         Z
HEAD 13/         E/RT
HEAD 14/         H/RT
HEAD 15/         S/R
HEAD COL 16/   P(ATM)
```

```
HEAD CCL 30/
FIXEC  WITH 4 CECIMALS
PRINT 1,12,13,14,15,16,30
FORMAT(10X,1F4.2,4F9.4,1PE13.5,1F8.1)
TITLE2 T = 2000 LOG     Z     E/RT      H/RT        S/R       P(ATM)
FPRINT 1,12,13,14,15,16,30
       STOP
```

Results

PAGE 4 OMNITAB PROBLEM 3 - 7

THERMO FUNCTICNS FCR NITROGEN INCLUDING 3RD VIRIAL CORRECTION

T = 2000 LCG	Z	E/RT	H/RT	S/R	P(ATM)
0.	1.0015	2.8972	3.8987	28.3089	7.33286E 00
0.20	1.0023	2.8972	3.8995	27.8475	1.16319E 01
0.40	1.0037	2.8971	3.9009	27.3855	1.84607E 01
0.60	1.0059	2.8971	3.9029	26.9227	2.93222E 01
0.80	1.0094	2.8969	3.9063	26.4586	4.66342E 01
1.00	1.0150	2.8967	3.9117	25.9923	7.43201E 01
1.20	1.0241	2.8964	3.9204	25.5227	1.18833E 02
1.40	1.0386	2.8959	3.9345	25.0476	1.91017E 02
1.60	1.0624	2.8952	3.9577	24.5636	3.09702E 02
1.80	1.1023	2.8945	3.9968	24.0651	5.09238E 02
2.00	1.1703	2.8939	4.0642	23.5426	8.56927E 02
2.20	1.2907	2.8945	4.1852	22.9793	1.49778E 03
2.40	1.5127	2.8994	4.4121	22.3441	2.78221E 03
2.60	1.9431	2.9172	4.8603	21.5771	5.66451E 03
2.80	2.8228	2.9702	5.7931	20.5573	1.30427E 04
3.00	4.7130	3.1171	7.8301	19.0245	3.45151E 04

Problem 3 - 8

Calculate an abridgement of a table of Squares, Cubes and Roots and a table of Factors for Computing Probable Errors from the Handbook of Chemistry and Physics, 38th Edition, pages 182 and 206, respectively.

The OMNITAB instructions are as follows:

```
OMNITAB  PROBLEM  3 - 8
TITLE1                      POWERS AND ROOTS
GENERATE ARG 1.(1.)20.  IN 1
RAISE  1, 2. ,STORE IN 2
RAISE  1, 0.5, 3
MULTIPLY  1, BY 10.STORE IN 4
MULTIPLY  1, BY100. STORE IN 5
RAISE 4 TO 0.5, STORE IN 6
RAISE 1, TO  3.0, STORE IN 7
DIVIDE 1. BY 3.     30
RAISE 1 TO COL 30,STORE IN 8
RAISE 4 TO COL 30,STORE IN 9
RAISE 5 TO COL 30,STORE IN 10
TITLE2      X      (N)2      (N)1/2    (10N)1/2       (N)3
TITLE4 (10N)1/3  (100N)1/3
FORMAT(2F9.0,2X,2F11.6,F11.0,4F11.6)
FPRINT 1 2 3 6 7 8 9 10
RAISE 1, TO -0.5 STORE IN 11
SUBTRACT 1. FROM 1 STORE 12
RAISE 12 TO -.5, STORE 13
MULTIPLY 11 BY 13 ,STORE 14
MULTIPLY .6745 BY 13 , STORE 15
MULTIPLY .6745 BY 14, STORE 16
MULTIPLY .8453 BY 13 , STORE 17
DIVIDE 17 BY 1, STORE 18
MULTIPLY .8453 BY 14, STORE 19
TITLE1          FACTORS  FOR PROBABLE ERROR CALCULATIONS
TITLE2      N      1/()      1/()()   .6745/()    .6745/()()
TITLE4 .8453/()()
FORMAT(F9.0,8F11.6)
FPRINT 1 11 14 15 16 18 19
        STOP
```

PAGE 1 OMNITAB PROBLEM 3 - 8

POWERS AND ROOTS

X	(N)2	(N)1/2	(10N)1/2	(N)3	(N)1/3
1.	1.	1.000000	3.162278	1.	1.000000
2.	4.	1.414214	4.472136	8.	1.259921
3.	9.	1.732051	5.477225	27.	1.442250
4.	16.	2.000000	6.324555	64.	1.587401
5.	25.	2.236068	7.071068	125.	1.709976
6.	36.	2.449490	7.745966	216.	1.817121
7.	49.	2.645751	8.366600	343.	1.912931
8.	64.	2.828427	8.944271	512.	2.000000
9.	81.	3.000000	9.486832	729.	2.080084
10.	100.	3.162278	10.000000	1000.	2.154435
11.	121.	3.316625	10.488088	1331.	2.223980
12.	144.	3.464102	10.954451	1728.	2.289428
13.	169.	3.605551	11.401754	2197.	2.351335
14.	196.	3.741657	11.832159	2744.	2.410142
15.	225.	3.872983	12.247448	3375.	2.466212
16.	256.	4.000000	12.649110	4096.	2.519842
17.	289.	4.123105	13.038404	4913.	2.571281
18.	324.	4.242640	13.416407	5832.	2.620741
19.	361.	4.358899	13.784048	6859.	2.668402
20.	400.	4.472136	14.142135	8000.	2.714417

PAGE 2 OMNITAB PROBLEM 3 - 8

FACTORS FOR PROBABLE ERROR CALCULATIONS

N	1/()	1/()()	.6745/()	.6745/()()	.8453/()
1.	1.000000	0.	0.	0.	0.
2.	0.707107	0.707107	0.674500	0.476944	0.422650
3.	0.577350	0.408248	0.476944	0.275363	0.199239
4.	0.500000	0.288675	0.389423	0.194711	0.122009
5.	0.447214	0.223607	0.337250	0.150823	0.084530
6.	0.408248	0.182574	0.301646	0.123146	0.063005
7.	0.377964	0.154303	0.275363	0.104078	0.049299
8.	0.353553	0.133631	0.254937	0.090134	0.039937
9.	0.333333	0.117851	0.238472	0.079491	0.033207
10.	0.316228	0.105409	0.224833	0.071099	0.028177
11.	0.301511	0.095346	0.213296	0.064311	0.024301
12.	0.288675	0.087039	0.203369	0.058708	0.021239
13.	0.277350	0.080064	0.194711	0.054003	0.018771
14.	0.267261	0.074125	0.187073	0.049997	0.016746
15.	0.258199	0.069007	0.180268	0.046545	0.015061
16.	0.250000	0.064550	0.174155	0.043539	0.013641
17.	0.242536	0.060634	0.168625	0.040898	0.012431
18.	0.235702	0.057166	0.163590	0.038559	0.011390
19.	0.229416	0.054074	0.158981	0.036473	0.010486
20.	0.223607	0.051299	0.154741	0.034601	0.009696

Problem 3 - 9

Quantities of interest in designing a light gas gun of constant diameter are the nondimensional barrel length, L_B; the chamber length, L_c; the acceleration which the projectile experiences, $\dot{U}$; and the efficiency of the gun with respect to a theoretical one which yields constant acceleration. The expressions for these quantities are:

$$L_B = \frac{2}{(\gamma+1)} \left[1.0 - \frac{1.0 - \left(\frac{\gamma+1}{2}\right) U}{\left(1.0 - \frac{\gamma+1}{2} U\right)^{\frac{\gamma+1}{\gamma-1}}} \right]$$

$$L_c = \frac{2}{(\gamma+1)} \left[\frac{1.0}{\left(1.0 - \frac{\gamma+1}{2} U\right)^{\frac{\gamma+1.}{2.(\gamma-1)}}} - 1.0 \right]$$

$$\dot{U} = \left[1.0 - \frac{\gamma-1}{2} U \right]^{\frac{2.\gamma}{\gamma-1.}}$$

$$\text{Efficiency} = \frac{U^2}{L_B}$$

where U is the projectile velocity and γ is the specific heat ratio of the propelling gas.

These quantities have been calculated for several values of U for a driver gas with a specific heat ratio of 1.4.

The OMNITAB instructions are as follows:

```
OMNITAB      PRCBLEM 3 - 9        7/5 DRIVER
NOSUMMARY
TITLE1 U=V/A,L=PAX/MC
TITLE2 UDOT=(VCOT)M/PA
GENERATE 0.0(.05)4.95,1
MULT 1,.2,2
SUB 2,1.,2
RAISE 2,6.,2
MULT 1,1.2,3
SUB 3,1.,3
DIV 3,2,2
SUB 2,1.,2
MULT 2,.8333,2
MULT 1,.2,3
SUB 3,1.,3
RAISE 3,3.,3
DIV 1.,3,3
SUB 1.,3,3
MULT 3,.8333,3
RAISE 1,2.,4
DIVICE 4,2,4
MULT 1,.2,5
SUB 5,1.,5
RAISE 5,7.,5
HEAD 1/  U=V/A
HEAD 2/ LBARREL
HEAD 3/ LCHAMBER
HEAD 4/ EFFIC.
HEAD 5/ UDOT
PRINT 1,2,3,4,5
       STOP
```

PAGE 1 OMNITAB PROBLEM 3 - 9 7/5 DRIVER

U=V/A,L=PAX/MC

UDCT=(VDCT)M/PA

U=V/A	LBARREL	LCHAMBER	EFFIC.	UDOT
0.	8.691996E-08	-3.725141E-08	0.	1.000000E 00
5.000000E-02	1.310014E-03	2.550747E-02	1.908376E 00	9.320653E-01
1.000000E-01	5.495918E-03	5.206660E-02	1.819532E 00	8.681256E-01
1.500000E-01	1.297717E-02	7.973234E-02	1.733814E 00	8.079829E-01
2.000000E-01	2.422728E-02	1.085628E-01	1.651031E 00	7.514475E-01
2.500000E-01	3.978021E-02	1.386201E-01	1.571133E 00	6.983373E-01
3.000000E-01	6.023815E-02	1.699700E-01	1.494070E 00	6.484776E-01
3.500000E-01	8.628095E-02	2.026828E-01	1.419780E 00	6.017009E-01
4.000000E-01	1.186757E-01	2.368334E-01	1.348212E 00	5.578466E-01
4.500000E-01	1.582889E-01	2.725016E-01	1.279307E 00	5.167609E-01
5.000000E-01	2.060995E-01	3.097727E-01	1.213006E 00	4.782968E-01
5.500000E-01	2.632141E-01	3.487379E-01	1.149255E 00	4.423133E-01
6.000000E-01	3.308840E-01	3.894943E-01	1.087995E 00	4.086755E-01
6.500000E-01	4.105256E-01	4.321461E-01	1.029168E 00	3.772547E-01
7.000000E-01	5.037428E-01	4.768049E-01	9.727186E-01	3.479278E-01
7.500000E-01	6.123531E-01	5.235899E-01	9.185877E-01	3.205771E-01
8.000000E-01	7.384175E-01	5.726295E-01	8.667183E-01	2.950903E-01
8.500000E-01	8.842756E-01	6.240609E-01	8.170529E-01	2.713605E-01
9.000000E-01	1.052585E 00	6.780318E-01	7.695343E-01	2.492855E-01
9.500000E-01	1.246366E 00	7.347009E-01	7.241053E-01	2.287679E-01
10.000000E-01	1.469057E 00	7.942390E-01	6.807086E-01	2.097152E-01
1.050000E 00	1.724576E 00	8.568298E-01	6.392876E-01	1.920391E-01
1.100000E 00	2.017387E 00	9.226718E-01	5.997857E-01	1.756557E-01
1.150000E 00	2.352589E 00	9.919786E-01	5.621466E-01	1.604853E-01
1.200000E 00	2.736009E 00	1.064981E 00	5.263140E-01	1.464520E-01
1.250000E 00	3.174312E 00	1.141929E 00	4.922326E-01	1.334839E-01

Problem 3 - 10

Generate and plot the values, for x between 0 and 6 in steps of 0.2, of the polynomial whose roots are at x = 0, 1, 2, 3, 4, 5, 6. Carry out the computations of the function in the form:

$$y = x(x-1)(x-2)(x-3)(x-4)(x-5)(x-6)$$

The OMNITAB instructions are as follows:

```
OMNITAB            PROBLEM 3 - 10
TITLE1    ELEMENTARY SYMMETRIC FUNCTION
GENERATE  0. .2 6. IN COL 1
SUB  1. FROM 1, MULT BY 1, ADD TO 9
SUB  2. FROM 1, MULT BY 9, ADD TO 8
SUB 3. FROM 1,MULT BY 8 ADD TO 7
SUB 4. 1 7 6
SUB 5. 1 6 5
SUB 6. 1 5 4
NOSUMMARY
FIXED 4
PRINT 1,4
DEMOTE 1 LINE COL 1 INTO 1 COL 4 INTO 4
DEFINE 10. AS ROW 1 COL 1
DEFINE 0. AS ROW 1 COL 4
PLOT COL 4 AGAINST 1
       STOP
```

In order to compress the plot of this function to fit the printed page we have introduced the DEMOTE and the two DEFINE instructions to provide a point with the coordinated (10,0) so that the plot on page 65 would have a uniform abscissa from 0. to 10. Note should be taken that the zeros of the function do indeed fall at the integral values 0, 1 .... 6 and that this fact is confirmed in the printed results on the next page. The reader is encouraged at this point to see what happens when the results are printed to more than 4 decimals as for example via the built-in format. See Problem 6-5 for an automatic way of isolating roots in a function of this type.

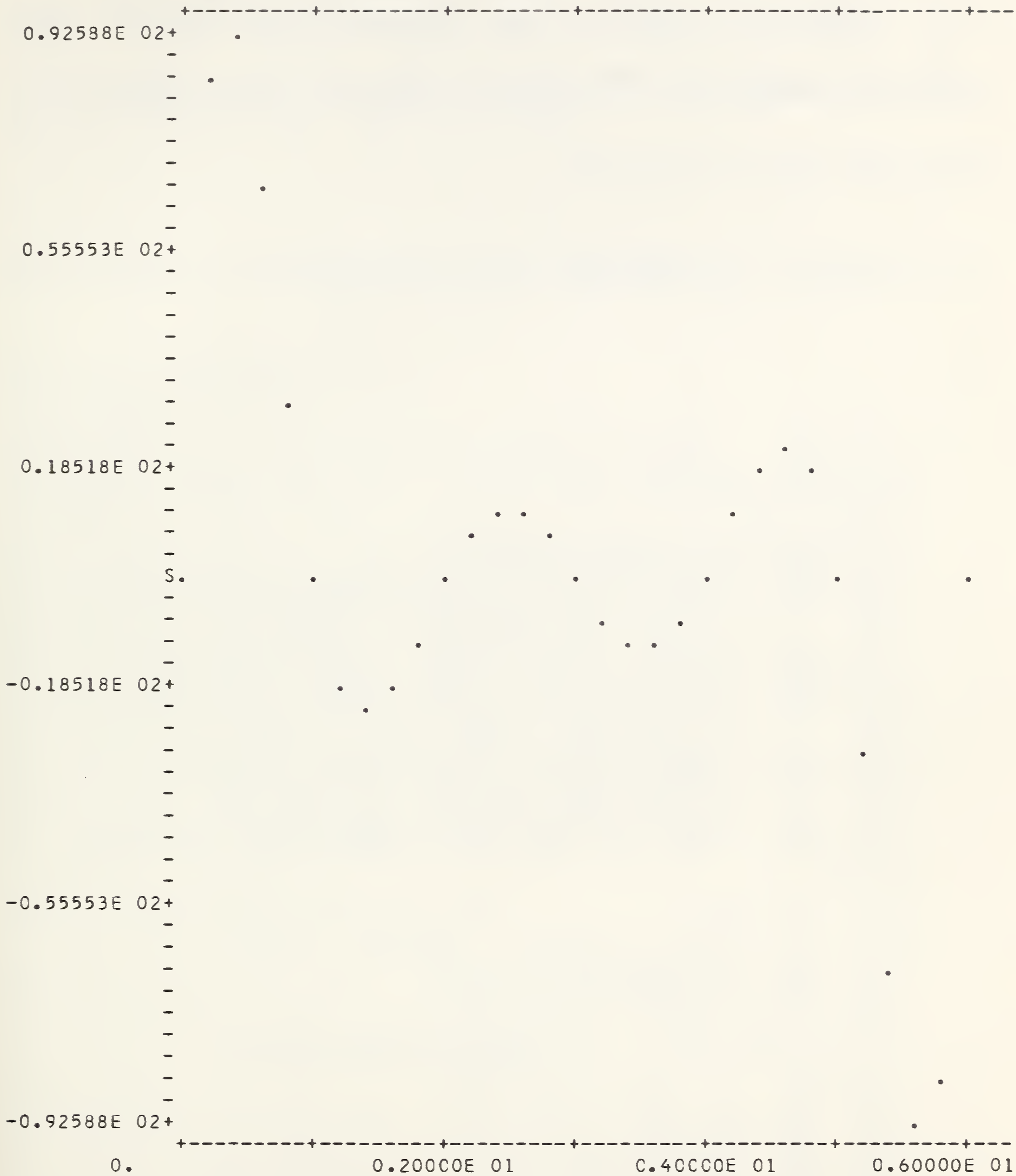
PAGE 1 OMNITAB PROBLEM 3 - 10

ELEMENTARY SYMMETRIC FUNCTION

COLUMN 1	COLUMN 4
0.	0.
0.2000	85.3107
0.4000	92.5876
0.6000	65.1442
0.8000	29.5207
1.0000	0.0000
1.2000	-17.6505
1.4000	-23.1469
1.6000	-19.3020
1.8000	-10.2187
2.0000	-0.0000
2.2000	8.0898
2.4000	12.0766
2.6000	11.4057
2.8000	6.8125
3.0000	0.0000
3.2000	-6.8125
3.4000	-11.4057
3.6000	-12.0766
3.8000	-8.0898
4.0000	-0.0000
4.2000	10.2187
4.4000	19.3020
4.6000	23.1469
4.8000	17.6505
5.0000	0.0000
5.2000	-29.5207
5.4000	-65.1442
5.6000	-92.5876
5.8000	-85.3107
6.0000	0.

PAGE 2 OMNITAB PROBLEM 3 - 10
ELEMENTARY SYMMETRIC FUNCTION

ABSCISSA - COLUMN 1
ORDINATES - COLUMN 4 (.),



Problem 3 - 11

Compute a table of the following functions of the quantum numbers J, J^2, J(J+1), J^2(J+1), J+1/2, $(J+1/2)^2$, $(J+1/2)^2$ for J = 0(1)100

The OMNITAB instructions are as follows:

```
OMNITAB  PROBLEM 3 - 11
NOSUMMARY
TITLE1  J=0 TO J= 100
GENERATE 0.(1.)100. STORE IN COL 1
MULTIPLY   COL 1 BY COL 1 AND STORE IN COL 2
ADD        1. TO COL1AND STORE IN COL 3
MULTIPLY   COL 1 BY COL 3 AND STORE IN COL 4
MULTIPLY   COL 3 BY COL 3 AND STORE IN COL 5
MULTIPLY   COL 2 BY COL 5 AND STORE IN COL 6
ADD        1. TO COL 2 AND STORE IN COL 7
MULTIPLY   COL 2 BY COL 7 AND STORE IN COL 8
ADD        .5 TO COL 1 AND STORE IN COL 9
MULTIPLY   COL 9 BY COL9 AND STORE IN COL 10
MULTIPLY   COL   9  BY COL 10 AND STORE IN COL 11
FIXED  0
HEAD COL 1/J
HEAD COL 2/JXJ
HEAD COL 4/J(J+1)
HEAD COL 8/JXJ(JXJ+1)
PRINT 1,2,4,8
PRINT 1,2,4,8          (FOR A SECOND COPY)
FIXED 3
HEAD COL 1/J
HEAD COL 9/J+.5
HEAD COL 10/(J+.5)2
HEAD COL 11/(J+.5)3
PRINT 1,9,10,11
       STOP
```

Problem 3 - 12

Tabulate: x^t and $(x-1.893)^t$ for $t = -2., -1., -2/3, -1/2, -2/5, -1/3, -2/7, -1/4, -2/9, -1/5$, and $-2/11$. Use $x = 0(0.1).5(0.5)10.(1.)50.(5.)100.$ for x^t, and $x = 2.0(1.)100.$ for $(x-1.893)^t$.

The OMNITAB instructions are as follows:

```
OMNITAB  PROBLEM 3 - 12
NOSUMMARY
GENERATE 0.(0.1).5(0.5)10.(1.)50.(5.)100.     IN COL 1
GENERATE          2.(0.5)10.(1.)50.(5.)100.    IN COL 2
SUBTRACT 1.893 FROM COL 2 AND STORE IN COL 3
TITLE1       X  TO THE  T POWER
RAISE COL 1 TO 2.        POWER AND STORE IN COL  4
RAISE COL 1 TO -1.       POWER AND STORE IN COL  5
RAISE COL 1 TO -0.666667 POWER AND STORE IN COL  6
RAISE COL 1 TO -0.5      POWER AND STORE IN COL  7
RAISE COL 1 TO -0.4      POWER AND STORE IN COL  8
RAISE COL 1 TO -0.33333  POWER AND STORE IN COL  9
RAISE COL 1 TO -0.28571  POWER AND STORE IN COL  10
RAISE COL 1 TO -0.25     POWER AND STORE IN COL  11
RAISE COL 1 TO -0.22222  POWER AND STORE IN COL  12
RAISE COL 1 TO -0.2      POWER AND STORE IN COL  13
RAISE COL 1 TO -0.18182  POWER AND STORE IN COL  14
FIXED 6
PRINT 1,4,5,6,7,8
PRINT COLS 1,9,10,11,12,13,14
TITLE1 (X - 1.893)  TO THE  T POWER
RAISE COL 3 TO 2.        POWER AND STORE IN COL  4
RAISE COL 3 TO -1.       POWER AND STORE IN COL  5
RAISE COL 3 TO -0.666667 POWER AND STORE IN COL  6
RAISE COL 3 TO -0.5      POWER AND STORE IN COL  7
RAISE COL 3 TO -0.4      POWER AND STORE IN COL  8
RAISE COL 3 TO -0.33333  POWER AND STORE IN COL  9
RAISE COL 3 TO -0.28571  POWER AND STORE IN COL  10
RAISE COL 3 TO -0.25     POWER AND STORE IN COL  11
RAISE COL 3 TO -0.22222  POWER AND STORE IN COL  12
RAISE COL 3 TO -0.2      POWER AND STORE IN COL  13
RAISE COL 3 TO -0.18182  POWER AND STORE IN COL  14
PRINT COLS 2,4,5,6,7,8
PRINT COLS 2,9,10,11,12,13,14
STOP
```

Problem 3 - 13

Compute

$$E(r) = 4.5 - (0.6667r^5 + 3r^4 + 6r^3 + 9r^2 + 9r + 4.5)e^{-2r}$$

for r = 1.(.5)10.

The OMNITAB instructions are as follows:

```
OMNITAB   PROBLEM 3 - 13
NOSUMMARY
GENERATE  1.(0.5)10.   IN COL  1
ADD      4.5 TO COL 2 AND STORE IN COL 2
MULTIPLY COL 1 BY 9. ,MULT BY 1., ADD TO COL 2
RAISE COL 1 TO 2. POWER MULT BY 9. ADD TO COL 2
RAISE COL 1 TO 3. POWER MULT BY 6. AND ADD TO COL 2
RAISE COL 1 TO 4. POWER MULT BY 3. AND ADD TO COL 2
RAISE COL 1 TO 5. POWER MULT BY .66667 AND ADD TO COL 2
MULT COL 1 BY 2. STORE IN COL 3
NEGEXP OF COL 3 MULT BY COL 2 AND ADD TO COL 4
SUBTRACT COL 4 FROM 4.5 AND STORE IN COL 5
HEAD COL 1/    R
HEAD COL 5/   E(R)
PRINT COLS 1,5
       STOP
```

Problem 3 - 14

Compute the vibrational-rotational lines of a parallel band for ethane from the formula:

$$v = 2753.493 + 1.31596m - 0.010564m^2 - 2.92(10^{-6})m^3$$

for m 0(1)40 for the R branch and for negative values for the P branch. Compare these values with the experimentally determined values supplied for $\pm m$ = 4(1)17.

The OMNITAB instructions are on the following page.

```
OMNITAB  PRCBLEM 3 - 14
NOSUMMARY
TITLE1 CALCULATIONS FOR K=3
GENERATE 0.(1.)40. IN COL 1
MULT CCL 1 BY 1.31596  STORE IN 2
MULT     1   -1.31596  STORE IN 5
ACC    2753.493  TO 2,         2
ADD   2753.493      5,         5
MULT  CCL 1 BY COL 1,MULT BY -0.01054 AND ADD TO COL 2
MULT  CCL 1 BY COL 1,MULT BY -0.01054 AND ADD TO COL 5
MULT  CCL 1 BY COL 1,MULT BY  COL 1 AND ADD TO COL 30
MULT CCL 30 BY -2.92E-6  MULT BY 1. AND ADD TO COL 2
MULT CCL 30 BY  2.92E-6  MULT BY 1. AND ADD TO COL 5
HEAD CCL 1/    M
HEAD CCL 2/   R
HEAD CCL 5/   P
FIXEC  3 DECIMALS
PRINT CCLS 1,2,5
READ CCLS 6 6 7      (see footnote°)
 0.,0.,0,
 1.,0.,0,
 2.,0.,0.
 3.,0.,0.
 4.,58.584,48.C74
 5.,59.795,46.645
 6.,61.006,45.216
 7.,62.184,43.774
 9.,64.478,40.795
 8.,63.329,42.298
10.,65.584,39.279
11.,66.671,37.742
12.,67.741,36.189
13.,68.811,34.620
14.,69.832,33.013
15.,70.840,31.400
16.,71.828,29.749
17.,72.797,28.C91
ACC     2700. TC COL 6 AND STORE IN COL 6
ACC     2700. TC COL 7 AND STORE IN COL 7
PRINT CCLS 1,6,7
SUB   CCL 6 FRCM COL 2 AND STORE IN COL 8
SUB   CCL 7 FRCM COL 5 AND STORE IN COL 9
HEAD CCL 8/  CELTA R
HEAD CCL 9/  DELTA P
RCWSUM COLS 2,5   STORE 6
PRINT CCLS 1,2,5,8,9,6
       STOP
```

- - - - - - - - - - -

° The repetition of the 6 in this instruction causes the second number on the cards to write over the first number. This is a convenient way of ignoring data not required in the calculations. If the second column were to be ignored the instruction would be READ COL 6,7,7 .

Problem 3 - 15

Compute

$$y = \left[\frac{x}{480} - \frac{313x^2}{3360} + \frac{923x^3}{2240} - \frac{18029x^4}{33600} + \frac{14643x^5}{44800}\right]e^{-x}$$

for x = .01(.01)1.

The OMNITAB instructions are as follows:

```
OMNITAB  PRCBLEM 3 - 15
NOSUMMARY
GENERATE  .01(.01)1. IN COL 1
MULT COL 1 BY COL 1 AND STORE IN COL 2
MULT CCL 1 BY COL 2 AND STORE IN COL 3
MULT COL 1 BY CCL 3 AND STORE IN COL 4
MULT COL 1 BY CCL 4 AND STORE IN COL 5
DIVICE COL 1 BY 480. AND STORE IN COL 6
DIVICE -313. BY 3360. MULT BY COL 2 AND ADD TO COL 6
DIVICE 923. BY 2240. MULT BY COL 3 AND ADD TO COL 6
DIVICE -18029. BY 33600. MULT BY COL 4 AND ADD TO COL 6
DIVIDE 14643. BY 44800. MULT BY COL 5 AND ADD TO COL 6
NEGEXP CF CCL 1 MULT BY COL 6 AND ADD TO COL 7
HEAD CCL 1/   X
HEAD CCL 7/    Y
PRINT CCLS 1,7
       STOP
```

Problem 3 - 16

Compute a table of corrected optical pyrometer temperatures from the Planck radiation law.

$$T = \frac{10^8(C_2/\lambda)}{\ln[\epsilon(\exp(10^8\ C_2/\lambda T) - 1) + 1]}$$

where C_2 = 1.438, λ = 6500, ϵ = .02(.02).20 and T = 5000.(50.)10,000°K

The OMNITAB instructions are as follows:

```
OMNITAB   PROBLEM 3 - 16
GENERATE 5000.(50.)10000. IN COL 1
DIVIDE  1.438  BY 6500., MULT BY 1.E8  AND STORE IN COL 30
DIVIDE  30  BY  1  , AND STORE IN COL 29
EXP         29 ,STORE IN COL 2
ADD   -1. TO COL 2        , 2
MULT  .02 BY COL 2 ,STORE IN COL 3
ADD   1.  TO COL 3         ,3
LOGE  COL  3, STORE IN COL 3
DIVIDE COL 30 BY 3 ,STORE IN  4
MULT  .04 BY COL 2 ,STORE IN COL 3
ADD   1.  TO COL 3         ,3
LOGE  COL  3, STORE IN COL 3
DIVIDE COL 30 BY 3 ,STORE IN  5
MULT  .06 BY COL 2 ,STORE IN COL 3
ADD   1.  TO COL 3         ,3
LOGE  COL  3, STORE IN COL 3
DIVIDE COL 30 BY 3 ,STORE IN  6
MULT  .08 BY COL 2 ,STORE IN COL 3
ADD   1.  TO COL 3         ,3
LOGE  COL  3, STORE IN COL 3
DIVIDE COL 30 BY 3 ,STORE IN  7
MULT  .10 BY COL 2 ,STORE IN COL 3
ADD   1.  TO COL 3         ,3
LOGE  COL  3, STORE IN COL 3
DIVIDE COL 30 BY 3 ,STORE IN  8
MULT  .12 BY COL 2 ,STORE IN COL 3
ADD   1.  TO COL 3         ,3
LOGE  COL  3, STORE IN COL 3
DIVIDE COL 30 BY 3 ,STORE IN  9
MULT  .14 BY COL 2 ,STORE IN COL 3
ADD   1.  TO COL 3         ,3
LOGE  COL  3, STORE IN COL 3
DIVIDE COL 30 BY 3 ,STORE IN 10
MULT  .16 BY COL 2 ,STORE IN COL 3
ADD   1.  TO COL 3         ,3
LOGE  COL  3, STORE IN COL 3
DIVIDE COL 30 BY 3 ,STORE IN 11
MULT  .18 BY COL 2 ,STORE IN COL 3
ADD   1.  TO COL 3         ,3
LOGE  COL  3, STORE IN COL 3
DIVIDE COL 30 BY 3 ,STORE IN 12
MULT  .20 BY COL 2 ,STORE IN COL 3
ADD   1.  TO COL 3         ,3
LOGE  COL  3, STORE IN COL 3
DIVIDE COL 30 BY 3, STORE IN 13
FORMAT(1X,1F6.0,10F8.0,1F6.0)
FPRINT COLS 1,4,5,6,7,8,9,10,11,12,13,1
      STOP
```

Problem 3 - 17

Compute values of log P from the relationship

$$\log P = .218541A - .109271(B+C) - 4.70287 \cdot 10^4/T + .218541k_i/T$$

for values of A, B, C given in tabular form for seven temperatures. Let k_i assume each of the values:

$$k_1 = 9.2 \times 10^4, \quad k_2 = 1.245 \times 10^5, \quad k_3 = 1.845 \times 10^5$$

The OMNITAB instructions are as follows:

```
OMNITAB    PROBLEM 3 - 17
TITLE1                    LOG P BEF G
NOSUMMARY
READ  1,2,3,4
 500.     45.817     1.8527     5.497
 750.     48.776     3.1949     9.134
1000.     50.953     4.3686    13.5752
1250.     52.691     5.4013    17.7537
1750.     55.390     7.378     24.4841
2000.     56.486     8.349     27.2683
2250.     57.463     9.209     29.7681
MULTIPLY COL 2 BY 0.218541 STORE IN COL 5
MULT 0.109271 BY COL 3 STORE IN COL 6
MULTIPLY COL 4 BY 0.109271 STORE IN COL 7
DIVICE 4.70287E4  BY COL 1 STORE IN COL 8
DIVICE 0.218541 BY COL 1 MULT BY 9.2E4  ADD TO COL 9
DIVICE 0.218541 BY COL 1 MULT BY 1.245E5   ADD TO COL 10
DIVICE 0.218541 BY COL 1 MULT BY 1.845E5 ADD TO COL 11
SUB   CCL 6 FRCM COL 5 AND STORE IN COL 12
SUB  CCL 7 FROM COL 12 AND STORE IN COL 12
SUB  CCL 8 FROM COL 12 AND STORE IN COL 12
ACC COL 9  TO COL 12 STORE IN COL 13
ADC COL 10 TO COL 12 STORE IN COL 14
ACC COL 11 TO COL 12 STORE IN COL 15
HEAD CCL 13/D 92000
HEAD CCL 14/D 124500
HEAD CCL 15/D 184500
HEAD CCL 1/T DEG. K
PRINT CCL 1 5 6 7 8
PRINT CCL 1 9 10 11
PRINT CCL 1 13 14 15
       STOP
```

Problem 3 - 18

Compute the transition probability kernel for

$Z = .002(.002).2$ and $Z = -.002(-.002)-.2$
from the following formulas:

$$F(Z) = (1 + Z)\left[\exp(-Z(2 + Z))\ 0.8862269 \cdot [-\text{erf}(-0.1 + 4.95Z) \pm \text{erf}(10 + 4.95Z)] + 0.8862269[\text{erf}(0.1 + 5.05Z) \mp \text{erf}(10+5.05Z)]\right]$$

$$F(Z) = (1 + Z)\left[\exp(-Z(2 + Z))\ 0.8862269 \cdot [\text{erf}(10 + 4.95Z) \pm \text{erf}(-0.1 + 4.95Z)] + 0.8862269(\text{erf}(10. + 5.05Z) \pm \text{erf}(0.1+5.05Z)]\right]$$

The OMNITAB instructions are as follows:

```
OMNITAB  PROBLEM  3 - 18
TITLE1 TRANSITION KERNEL X=1
NOSUMMARY
GENERATE 0.002(0.002)0.200 STORE IN COL 1
MULTIPLY COL 1 BY -1. STORE IN COL 2
ADD 1. TO COL 1 MULTIPLY BY 0.8862269 STORE IN COL 3
ADD 1. TO COL 2  MULTIPLY BY 0.8862269 STORE IN COL 4
ADD 2. TO COL 1  MULTIPLY BY COL 1 STORE IN COL 5
ADD 2. TO COL 2  MULTIPLY BY COL 2 STORE IN COL 6
NEGEXP OF COL 5   STORE IN COL 7
NEGEXP OF COL 6   STORE IN COL 8
MULTIPLY COL 1 BY 4.95  STORE IN COL 9
MULTIPLY COL 2 BY 4.95  STORE IN COL10
MULTIPLY COL 1 BY 5.05  STORE IN COL 11
MULTIPLY COL 2 BY 5.05  STORE IN COL 12
ADD 10. TO COL 9  STORE IN COL 13
ADD 10. TO COL 10  STORE IN COL 14
SUBTRACT 0.100 FROM COL 9  STORE IN COL 15
SUBTRACT 0.100 FROM COL 10  STORE IN COL 16
ADD 10. TO COL 11  STORE IN COL 17
ADD 10. TO COL 12  STORE IN COL 18
ADD 0.100 TO COL 11  STORE IN COL 19
ADD 0.100 TO COL 12  STORE IN COL 20
```

```
PRINT CCLS 13,14,15,16
PRINT CCLS 17,18,19,20
PRINT CCLS 13,14,15,16
PRINT CCLS 17,18,19,20
ERROR CF COL 13  STORE IN 13
ERROR CF COL 14  STORE IN 14
ERROR CF COL 15  STORE IN 15
ERROR CF CCL 16  STORE IN 16
ERROR CF COL 17  STORE IN 17
ERROR CF COL 18  STORE IN 18
ERROR CF COL 19  STORE IN 19
ERROR CF COL 20  STORE IN 20
PRINT CCLS 13,14,15,16
PRINT CCLS 17,18,19,20
PRINT CCLS 13,14,15,16
PRINT CCLS 17,18,19,20
SUBTRACT COL 15 FROM COL 13  MULTIPLY BY COL 7  ADD TO COL 21
ACC COL 16 TO COL 14  MULTIPLY BY COL 8  ADD TO COL 22
SUBTRACT COL 19 FROM COL 17  STORE IN COL 23
ACC COL 20 TC CCL 18  STORE IN COL 24
ACC COL 21 TC CCL 23  MULTIPLY BY COL 3  STORE IN COL 25
ACC COL 22 TO CCL 24  MULTIPLY BY COL 4  STORE IN COL 26
SUBTRACT COL 23 FROM COL 21  MULTIPLY BY COL 3  STORE IN COL 27
SUBTRACT COL 22 FROM COL 24  MULTIPLY BY COL 4  STORE IN COL 28
HEAD CCL 25/F(Z),E=0.1
HEAD CCL 26/F(Z),E=0.1
HEAD CCL 27/F(Z),E=10.
HEAD CCL 28/F(Z),E=10.
HEAD CCL 1/Z
HEAD CCL2/Z
PRINT CCL,S 1,COL 25,COL 27 ,COL 2,COL,26 COL28
       STCP
```

4. SPECIAL FEATURES

As important as mathematical features are, they are no more important to the overall efficiency of a program than are a variety of auxiliary characteristics such as: flexible input and output, to make the access to the machine less painful; versatile plotting capabilities — to present the results in more dramatic and meaningful form; manipulative capabilities — to obviate laborious hand sorting or editing of data prior to entering the machine; built-in diagnostics and error checking — to assist in trouble shooting a calculation which has gotten out of control or, alternatively, to supply assurance of the accuracy of the results; and finally, an economy of operation such as to encourage the user to adopt an experimental approach — thereby making it possible for the program itself to teach the user many of its less obvious characteristics.

In the sections which follow we discuss the special features mentioned above and illustrate their application or utility as the case may be by citing concrete examples taken from some of the productive work to which OMNITAB has been applied.

Certain manipulative operations of OMNITAB are worthy of special mention. These are transposing of a row into a column, the promoting or demoting of the numbers in a column into another column with appropriate line shifts, the saving of columns for later use, stringing together of columns, the inversion of columns, the shortening of columns, the sorting of one or more columns to agree with an ordered arrangement of another column, and the abridging of tables for certain selected values or certain groups of values. Many of these operations can be achieved with a single instruction. Others may require two or more instructions. There are enough manipulative operations in OMNITAB to perform any of the card handling operations which are normally done by the sorter, lister, duplicating punch, and collator. Some of these features are employed in the examples in later chapters.

4.1. Input of Tabular Data

OMNITAB has been programmed to accept tabular arrays of data in as many forms as possible. This flexibility enables it to accept outputs from other programs. The program will accept either fixed field or variable field input. Numbers will be read between card columns 1–72. They may originate in any column of the card and must be separated by a space or some symbol other than E or a decimal point. The format of the input may vary from card to card or even from number to number. Furthermore, no card count need be supplied as the program does its own counting of data cards.°

The numbers on each card are read and assigned in sequence to the designated columns in the machine. Each number on the card may be assigned to any of the 46 columns in the work-sheet simply by indication the storage columns in one-to-one correspondence with the columns of the table to be read in. The program clears out the designated column before storing the table. If a column in the input data is not wanted, it cannot be ignored. It may be read into some column and then written over. Thus, if columns 1, 4, and 5 of a particular 5 column input are to be read into columns 1, 2, and 3 respectively, the instruction READ 1,2,2,2,3 or READ 1,40,40,2,3 would accomplish this.

Numbers such as dates, identification, etc., which do not usually carry decimal points, are also read by the program. For example, the data 3/4/62 would be read as 3 numbers. The program automatically floats the numbers (assuming a decimal point at the end of the number) which do not carry decimal points. Examples of mixed input acceptable by the OMNITAB program are given in the figures which follow.

The above discussion applies to data being supplied on cards following a READ or SET instruction. Data following an FREAD instruction must be punched on the cards in strict compliance with the FORMAT statement which preceded it. If the format remains fixed for numerous sets of data, it need not be read in again prior to each FREAD. It can, however, be updated at will by reading in a new FORMAT instruction when the data require it or when the format changes for the output.

° The FREAD is an exception; a card count must be supplied (see page 19).

```
OMNITAB          7 / 5 / 64
READ  COL 17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32
D 1,0,0,0,0,0,0,0,0,0,0,0,0,0,0
D 0,1
D 0,0,1
D 0,0,0,1
D 0,0,0,0,1
D 0,0,0,0,0,1
D 0,0,0,0,0,0,1
D 0,0,0,0,0,0,0,1
D 0,0,0,0,0,0,0,0,1
D 0,0,0,0,0,0,0,0,0,1
D 0,0,0,0,0,0,0,0,0,0,1
D 0,0,0,0,0,0,0,0,0,0,0,1
D 0,0,0,0,0,0,0,0,0,0,0,0,1
D 0,0,0,0,0,0,0,0,0,0,0,0,0,1
D 0,0,0,0,0,0,0,0,0,0,0,0,0,0,1
D 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1
READ   1, 2, 3, 4, 5, 6, 7
0  36265. 0 0 0 0 0
5158  0  0  0  0  0  0
188  0  0  9979  12795  2679  0
20  38  77  0  30  128  145
69  2225  424  10425  0  128  0
49  173  4165  2398  3252  0  145
128  913  1080  1209  696  319  0
3794  1109  12225  1835  2087  3890  8475
2915  905  1388  709  1361  638  1065
869  1682  77  709  907  383  1937
524  324  0  1501  121  64  97
326  23  0  3878  242  0  0
10  0  0  250  136  0  0
7845  2942  10297  2252  1996  4719  14286
3537  2527  926  1877  1165  510  1840
5444 2821  501  1772  1119  191  630
READ   8, 9, 10,11, 12,13,14, 15, 16
0 4646  3169  8939  0  421  0  1355  0
0 1382 103 2466 0 120 0 6995 5
0 0 1062 0 0 0 909 0 5
264 29  85  99  335  0  0  5  0
275  882  874  83  0  601  2434  253  104
550 2705 86 17 0 3065 0 0 30
```

Figure 4-1. Examples of variable input accepted by OMNITAB. The numbers are automatically floated by placing a decimal point at the end of the number. When a number exceeds 131071 ($2^{17}-1$), it must always carry a decimal or be in the E notation. Note that when numbers are missing at the end of a card, values are stored from the previous card.

```
OMNITAB  8/28/64
READ  1,10
      0         0
      0         0
     02   -42885.89   CF P02 0,0
      0         0
     04   -42885.89   CF P02 0,0
     05    42886.74   CF P02 0,0
     06    42888.11   CF P02 0,0
     07    42890.09   CF P02 0,0
     08    42892.53   CF P02 0,0
     09   -42895.62   CF P02 0,0
     10    4289.172   CF P02 0,0
     11   -42903.34   CF P02 0,0
     12    42908.09   CF P02 0,0
     13    42913.40   CF P02 0,0
     14    42919.29   CF P02 0,0
     15    42925.74   CF P02 0,0
     16   -42932.77   CF P02 0,0
     17   -42940.41   CF P02 0,0
     18   -42948.60   CF P02 0,0
READ COLUMNS 1,2,3,4
       .0000001 10000                                 -1  178.624
        0000011   11000                              2.      8.0006235E1
       0000012.   120.                               3   -7.21E-4
       000013      13000                            -4 3.14159E21
     0000014E-0     14000                            5            -.00624
-.000000015          15000                          -6.    527.111
 .0000016E+0          1600000.                       7   0000000.
  -.000017              17000                        8.
   .00018                18000                       9
  -.0019E000                190                      0  9999999.
     54321.4E-5                  7654321E-36         .0 .22222222
```

Figure 4—2. Further examples of the variable input accepted by OMNITAB. Note in the above example the program would ignore the data beginning with CF. If the READ statement had contained 5 numbers, only the alphabetic characters would be ignored.

4.2. Output Provisions

If the user cares only about the numbers and not the format, he can obtain results simply by a statement PRINT COLS 1,2,7,9, etc. The columns can be printed in any order or even repeated on a page. Any number of print statements can be given. Each is subject only to certain limitations given below. Figure 4–3 shows a page resulting from a PRINT statement which employs the built–in format. The instruction FIXED 6 will provide results to six decimals (see figure 4–4).

Four options are available for controlling the format of the printout. Three of these are selected by a statement which precedes the PRINT instruction. These instructions, which start with the words FIXED, FLOATING, or FORMAT, are discussed in section 3.2 and 3.3. If none of the above words precede the PRINT instruction, the program will print the results in floating format with one number to the left of the decimal and 6 to the right, followed by a suitable exponent, for example 3.782797E–02. In this printing mode a single page will accommodate a maximum of 8 columns. Since the format also controls the punching of results, this built–in mode will accommodate only four columns of punched results (only 72 columns are available for punching on one card).

The ability of OMNITAB to accept a FORMAT statement gives the OMNITAB user the capability of formating his results in as sophisticated a manner as any programmer using FORTRAN. We illustrate in this section and in succeeding chapters a variety of formats but suggest that the novice put off the use of formats until he has gained some confidence in the use of the program. Alternatively we suggest that until one has mastered the format statement,he ask for an ordinary PRINT (either in fixed field or floating) prior to using the FPRINT so as to insure some output even if the formated output gets mixed up.

When the built–in format is used, the column headings are centered over the columns. If no headings are supplied, the program automatically prints the appropriate column numbers as headings. This is illustrated in one or more of the sample problems. With a little experience, it is often possible to get headings centered properly even when the printing mode is controlled by the instructions FIXED or FLOATING.

Before going on to the other output instructions, it should be emphasized that each PRINT or FPRINT (also

PUNCH and FPUNCH) instruction gives a column whose length is from 1 to 101 lines (cards) long. Since 50 lines of results are provided on a page, a column of 100 rows will extend over two pages. Each page will carry the word OMNITAB and such other information as was supplied on the OMNITAB card. The program expects a date on this card. If it does not find it it prints out a statement to that effect (see figure 4-9). We cannot over-emphasize the importance of a date on this card since it will be carried on every page of the printout. Below this line provision is made for two lines for titles. TITLE1 and TITLE3 run contiguously across 119 type bars on the first of these and TITLE2, and TITLE4 follow on the next line. The next line accommodates column headings which are entered as desired. Failure to supply column headings will cause the column numbers themselves to be used as headings.

When a FORMAT instruction is used, it is not possible to provide headings over the columns via the HEAD instruction. The instructions FPRINT and FPUNCH must be used instead of PRINT and PUNCH in order that the special FORMAT instruction prevail. Since TITLE2 and TITLE4 extend over the last 119 type bars it is possible to use these to distribute headings over the columns when a format statement controls the printout.

Space is available at the end of each PRINT for a one line footnote which information is read from a card carrying the word FOOTNOTE followed by 4 spaces. The single entry of a footnote causes the same footnote to be printed after every PRINT. To remove it, it is necessary to read in a card which is blank except for the word FOOTNOTE. The footnote can be updated at will.

If a printout is shorter than 50 lines, an appropriate number of lines of text can be placed below it via the NOTE instruction. The use of this provision is illustrated in some of the illustrative problems and especially in chapter 11.

The TPRINT instruction obeys either the built-in format or a specified format but does not start a new page. Its principal use is in mixing text with computer results. Applications of this command are to be found in chapter 11.

PAGE 1 OMNITAB PROBLEM 3 - 2

THERMAL FUNCTIONS E-

T K	-(F-HO)/RT	(H-HO)/RT	S/R	CP/R
2.731500E 02	6.428387E 00	2.500000E 00	8.928387E 00	2.500000E 00
2.981500E 02	6.647326E 00	2.500000E 00	9.147326E 00	2.500000E 00
1.000000E 03	9.672722E 00	2.500000E 00	1.217272E 01	2.500000E 00
1.100000E 03	9.910997E 00	2.500000E 00	1.241100E 01	2.500000E 00
1.200000E 03	1.012853E 01	2.500000E 00	1.262853E 01	2.500000E 00
1.300000E 03	1.032863E 01	2.500000E 00	1.282863E 01	2.500000E 00
1.400000E 03	1.051390E 01	2.500000E 00	1.301390E 01	2.500000E 00
1.500000E 03	1.068638E 01	2.500000E 00	1.318638E 01	2.500000E 00
1.600000E 03	1.084773E 01	2.500000E 00	1.334773E 01	2.500000E 00
1.700000E 03	1.099929E 01	2.500000E 00	1.349929E 01	2.500000E 00
1.800000E 03	1.114219E 01	2.500000E 00	1.364219E 01	2.500000E 00
1.900000E 03	1.127736E 01	2.500000E 00	1.377736E 01	2.500000E 00
2.000000E 03	1.140559E 01	2.500000E 00	1.390559E 01	2.500000E 00
2.100000E 03	1.152757E 01	2.500000E 00	1.402757E 01	2.500000E 00
2.200000E 03	1.164386E 01	2.500000E 00	1.414386E 01	2.500000E 00
2.300000E 03	1.175499E 01	2.500000E 00	1.425499E 01	2.500000E 00
2.400000E 03	1.186139E 01	2.500000E 00	1.436139E 01	2.500000E 00
2.500000E 03	1.196345E 01	2.500000E 00	1.446345E 01	2.500000E 00
2.600000E 03	1.206150E 01	2.500000E 00	1.456150E 01	2.500000E 00
2.700000E 03	1.215585E 01	2.500000E 00	1.465585E 01	2.500000E 00

Figure 4-3. Typical output in floating point with column headings. A maximun of eight columns is available in this format. Note that the headings are quite well centered in this format.

PAGE 4 OMNITAB PROBLEM 3 - 12
X TO THE T POWER

X	T + -1/3	T + -2/7	T + -1/4
36.000000	0.302857	0.359210	0.408248
37.000000	0.300104	0.356409	0.405461
38.000000	0.297448	0.353704	0.402767
39.000000	0.294883	0.351089	0.400160
40.000000	0.292405	0.348558	0.397635
41.000000	0.290009	0.346108	0.395188
42.000000	0.287688	0.343733	0.392815
43.000000	0.285441	0.341430	0.390511
44.000000	0.283262	0.339195	0.388273
45.000000	0.281148	0.337024	0.386097
46.000000	0.279096	0.334914	0.383982
47.000000	0.277102	0.332863	0.381923
48.000000	0.275164	0.330866	0.379918
49.000000	0.273279	0.328923	0.377964
50.000000	0.271445	0.327030	0.376060
55.000000	0.262957	0.318245	0.367206
60.000000	0.255440	0.310431	0.359304
65.000000	0.248715	0.303412	0.352186
70.000000	0.242646	0.297055	0.345721
75.000000	0.237130	0.291257	0.339809
80.000000	0.232083	0.285936	0.334370
85.000000	0.227440	0.281025	0.329341
90.000000	0.223148	0.276473	0.324668
95.000000	0.219162	0.272235	0.320309
100.000000	0.215447	0.268275	0.316228

Figure 4-4. Typical output in fixed point resulting from the instruction FIXED WITH 6 DECIMALS. A total of eight columns can be accommodated here. The headings could have been centered better by moving them to the extreme right of the 12 allotted spaces.

PAGE 1 OMNITAB PROBLEM 3 - 5
RECIPRCCAL TABLE, 1000./T - T = 100.0 - 200.0 BY 0.1

100.	10.00000	9.99001	9.98004	9.97009	9.96016
101.	9.90099	9.89120	9.88142	9.87167	9.86193
102.	9.80392	9.79432	9.78474	9.77517	9.76563
103.	9.70874	9.69932	9.68992	9.68054	9.67118
104.	9.61538	9.60615	9.59693	9.58773	9.57854
105.	9.52381	9.51475	9.50570	9.49668	9.48767
106.	9.43396	9.42507	9.41620	9.40734	9.39850
107.	9.34579	9.33707	9.32836	9.31966	9.31099
108.	9.25926	9.25069	9.24214	9.23361	9.22509
109.	9.17431	9.16590	9.15751	9.14913	9.14077
110.	9.09091	9.08265	9.07441	9.06618	9.05797
111.	9.00901	9.00090	8.99281	8.98473	8.97666
112.	8.92857	8.92061	8.91266	8.90472	8.89680
113.	8.84956	8.84173	8.83392	8.82613	8.81834
114.	8.77193	8.76424	8.75657	8.74891	8.74126
115.	8.69565	8.68810	8.68056	8.67303	8.66551
116.	8.62069	8.61326	8.60585	8.59845	8.59107
117.	8.54701	8.53971	8.53242	8.52515	8.51789
118.	8.47458	8.46740	8.46024	8.45309	8.44595
119.	8.40336	8.39631	8.38926	8.38223	8.37521
120.	8.33333	8.32639	8.31947	8.31255	8.30565
121.	8.26446	8.25764	8.25083	8.24402	8.23723
122.	8.19672	8.19001	8.18331	8.17661	8.16993
123.	8.13008	8.12348	8.11688	8.11030	8.10373
124.	8.06452	8.05802	8.05153	8.04505	8.03859
125.	8.00000	7.99361	7.98722	7.98085	7.97448
126.	7.93651	7.93021	7.92393	7.91766	7.91139
127.	7.87402	7.86782	7.86164	7.85546	7.84929
128.	7.81250	7.80640	7.80031	7.79423	7.78816
129.	7.75194	7.74593	7.73994	7.73395	7.72798
130.	7.69231	7.68640	7.68049	7.67460	7.66871
131.	7.63359	7.62777	7.62195	7.61615	7.61035
132.	7.57576	7.57002	7.56430	7.55858	7.55287
133.	7.51880	7.51315	7.50751	7.50188	7.49625
134.	7.46269	7.45712	7.45156	7.44602	7.44048
135.	7.40741	7.40192	7.39645	7.39098	7.38552
136.	7.35294	7.34754	7.34214	7.33676	7.33138

Figure 4-5. Output resulting from the statement: FORMAT (1H F9.0,10F10.5). Note the absence of headings.

PAGE 5 OMNITAB 1 - 1

CF A-X EMISSION SPECTRUM 1-1 BAND HEAD AT 2308.674

0.	0.		0.	-43393.66		0.
		0.			-0.40	
1.	-43389.73		0.	-43393.26		0.
		-3.46			0.	
2.	43386.27		0.76	-43393.26		0.
		-2.70			0.40	
3.	43383.57		0.30	-43393.66		0.82
		-2.40			1.22	
[illegible]	[illegible]		0.	43394.88		0.63
					1.85	
[illegible]	[illegible]		[illegible]	[illegible]3396.73		0.61
					2.46	
						[illegible]
19.	[illegible]					
		8.06				
20.	43434.27		0.48	43501.40		[illegible]
		8.56			11.99	
21.	43442.83		0.89	43513.39		0.63
		9.45			12.62	
22.	-43452.28		0.42	43526.01		0.60
		9.87			13.22	
23.	-43462.15		0.62	43539.23		0.61
		10.49			13.83	
24.	43472.64		0.69	43553.06		0.58
		11.18			14.41	
25.	43483.82		0.56	43567.47		0.70
		11.74			15.11	
26.	43495.56		0.63	43582.58		0.58
		12.37			15.69	

Figure 4—6 Lefthand portion of a page resulting from the instructions:

```
FORMAT (1H F3.0, 4(F11.2,F13.2),2F10.2/1H F21.2, 3F24.2)
FPRINT 1,2,26,3,27,4,28,5,29,10,11,18,19,20,21
```

```
PAGE    1        OMNITAB        FORMAT TEST


     1.   0.100E 01   1.000E 00       10.        1. 1C0.000E-02      1000.       10.
     2.   0.200E 01   2.000E 00       20.        2. 2C0.000E-02      2000.       20.
     3.   0.300E 01   3.000E 00       30.        3. 300.000E-02      3000.       30.
     4.   0.400E 01   4.000E 00       40.        4. 400.000E-02      4000.       40.
     5.   0.500E 01   5.000E 00       50.        5. 5C0.000E-02      5000.       50.
     6.   0.600E 01   6.000E 00       60.        6. 6C0.000E-02      6000.       60.
     7.   0.700E 01   7.000E 00       70.        7. 7C0.000E-02      7000.       70.
     8.   0.800E 01   8.C00E 00       80.        8. 8C0.000E-02      8000.       80.
     9.   0.900E 01   9.000E 00       90.        9. 9C0.000E-02      9000.       90.
    10.   0.100E 02   1.000E 01      100.       10. 1C0.000E-01     10000.      100.

    20.   0.200E 02   2.000E 01      200.       20. 2C0.000E-01     20000.      200.
    30.   0.300E 02   3.000E 01      300.       3C. 3C0.000E-01     30000.      300.
    40.   0.400E 02   4.000E 01      400.       4C. 4C0.000E-01     40000.      400.
    50.   0.500E 02   5.000E 01      500.       50. 5C0.000E-01     50000.      500.
    60.   0.600E 02   6.000E 01      600.       6C. 6C0.000E-01     60000.      600.
    70.   0.700E 02   7.000E 01      700.       7C. 7C0.000E-01     70000.      700.
    80.   0.800E 02   8.000E 01      800.       80. 8C0.000E-01     80000.      800.
    90.   0.900E 02   9.000E 01      900.       90. 900.000E-01     90000.      900.
   100.   0.100E 03   1.000E 02     1000.      100. 1C0.000E 00    100000.     1000.
   200.   0.200E 03   2.000E 02     2000.      200. 2C0.000E 00    200000.     2000.


    OMNITAB       FORMAT TEST
    NOSUMM
    GENERAL. 1. 10. 10. 100. 100. 1000. 1
    FORMAT(1F9.0,1E12.3,1P1E12.3,1F9.0,0P1F9.0,3P1E12.3,1F11.0,1P1F9.0)
    FPRINT1 1 1 1 1 1 1 1
          STOP
```

Figure 4-7. Results from an experiment designed to test various format provisions in OMNITAB.

4.3. Summary Calculations

Recognizing that, for the types of problems which OMNITAB is designed to handle, the major time delay is the time between the preparation of the long-hand instructions and their actual entry into the machine, we have incorporated a number of automatic operations. One example is the summary (see figure 4-8) which is computed and printed for each of the columns called for in any of the PRINT statements. The summary contains ten numbers which are printed in the following order: the number of values in the column, the sum of the values, the sum of the absolute values, the sum of squares of values, the average value, the average of the absolute values, the minumum value, the maximum value, the RMS value, and the definite integral of the column. The last of these is meaningful only if the values are tabulated at uniform intervals of the argument, which is taken to be in column 1, and if the interval of tabulation is close enough for four-point Lagrangian integration.

The summary feature is automatic unless it is suppressed by the instruction NOSUMMARY, in which case it can be reactivated again by the instruction SUMMARIZE.

```
PAGE   2      OMNITAB  PROBLEM 3 - 3
OMNITAB SUMMARY-------

   COLUMN  1      COLUMN 10      COLUMN  8      COLUMN  9

NUMBER OF VALUES

        20             20             20             20

SUM OF VALUES

  7.363000E 03   1.900000E 03   2.051364E 05   2.699164E 02

SUM OF ABSOLUTE VALUES

  7.363000E 03   1.900000E 03   2.051364E 05   2.699164E 02

SUM OF SQUARES OF VALUES

  2.777188E 06   2.470000E 05   4.167323E 09   7.214894E 03

AVERAGE OF VALUES

  3.681500E 02   9.500000E 01   1.025682E 04   1.349582E 01

AVERAGE OF ABSOLUTE VALUES

  3.681500E 02   9.500000E 01   1.025682E 04   1.349582E 01

MINIMUM VALUE

  2.731500E 02   0.             3.573898E 02   4.702497E-01

MAXIMUM VALUE

  4.631500E 02   1.900000E 02   3.394549E 04   4.466512E 01

RMS VALUE

  3.726384E 02   1.111306E 02   1.443489E 04   1.899328E 01

INTEGRAL ASSUMING INDEPENDENT VARIABLE IS IN COL  1 AND UNIFORMLY SPACED

  6.994850E 04   1.805000E 04   1.875973E 06   2.468385E 03
```

Figure 4—8. This output is printed for each column in the PRINT instruction. It may be suppressed by the instruction NOSUMMARY, and reinstated again with the word SUMMARIZE.

4.4. Diagnostic Features

A general-purpose computer program places requirements on the programmer to provide for handling overflow, and other disruptive situations, which may arise in the course of a calculation. The general philosophy in OMNITAB has been to ensure that underflow and certain other conditions do not disrupt the calculations wherever feasible. When such conditions occur, appropriate action is taken and a diagnostic statement is supplied. The appropriate action may be to set a very small number to zero and to proceed, or to truncate to an integer the exponent of a negative number. Some of these render subsequent operations meaningless. In that case, the program goes on to the next problem. In other instances, the computations are allowed to proceed after a diagnostic statement is printed.

The concept of a general-purpose program rests in some measure on the assumption that the user, though not a programmer, is familiar with the behavior of the mathematical functions he is using or trying to compute. Thus, we would expect him to know that Gamma functions for large positive numbers are extremely large, and that they are infinite for negative integers, that the inverse trigonometric functions have limiting values or infinite branches, etc. Nevertheless, diagnostic features are incorporated which will call some of these singularities to his attention if he has overlooked them. Such diagnostic statements are, however, no substitute for sound mathematical analysis which is necessary to avoid the more serious pitfalls of numerical computations.[°] The characteristics and, perhaps, limitations of the more important subroutines in OMNITAB have been given in the discussion of the functions in earlier sections. Typical diagnostic statements are illustrated in figures which follow and more fully in a later section.

- - - - - - - - - - - -

° Stegun, I. E., Abramowitz, M. Pitfalls in Computations, J. Soc. Indust. Appl. Math. 4, 207219, 1956.

Abramowitz, M., On the Practical Evaluation of Integrals, J. Soc. Indust. Appl. Math. 2, 2035, 1954.

```
*********************************************
*                                           *
*                                           *
*                                           *
*                                           *
*                                           *
*                                           *
*  YOU FORGOT THE DATE ON THE OMNITAB CARD  *
*                 ----                      *
*                                           *
*                                           *
*                                           *
*                                           *
*                                           *
*********************************************
```

Figure 4-9 An automatic reminder that the date was omitted from the OMNITAB card. This is printed on the first page of the output.

```
PAGE   1       OMNITAB              OCT 28. 1964
LIST OF OMNITAB COMMANDS, CATA ANC DIAGNOSTICS


     OMNITAB              OCT 28. 1964
     NOSUMMARY
     GENERATE  1.  1.  40.  1
     BEJZERC 1 2
     DEMOTE  2 RCWS 2 3 3 4 4 5 5 6 6 7 7 8 8  9 9 10
     MULT 1 .01732 11
     ADC  .2  3  3
     ADC  .4  4  4
     ADC  .6  5  5
     ADC     .8  6  6
     ADC    1. 77
          ***ILLEGAL INSTRUCTION   OCOADD     ***ILLEGAL INSTRUCTION***
     ADC  1.2 8 8
     ADC  1.4  9  9


PAGE   1       OMNITAB  PROBLEM  5 - 2
LIST OF OMNITAB COMMANDS, CATA ANC DIAGNOSTICS


     OMNITAB  PRCBLEM  5 - 2
     TITLE1 EINSTEIN FUNCTIONS
     GENERATE 1.01(.01)1.5 IN COL 1
     NEGEXP CF CCL 1, STORE IN COL 2
     ADC 1. TO CCL 3 STORE IN 3
     LOGE OF COL 3 ,MULT BY 1., ACD INTO 4
          ***ILLEGAL INSTRUCTION   OCLOGE     ***ILLEGAL INSTRUCTION***
     RAISE CCL 3 TO -1. , MULT BY COL 2 ADD 5
     MULTIPLY COL 5 BY COL 1, STORE IN 5
```

Figure 4-10. A typical printout of the OMNITAB instruction with diagnostic statements. In the first instance, the storage location was omitted, in the second, a zero was punched instead of an "O" in the word LOGE.

```
REPEAT          1,15,10
           ***UNDERFLOW**POLYFI 00012     ***UNDERFLOW**POLYFI 00012
           ***UNDERFLOW**POLYFI 00012     ***UNDERFLOW**POLYFI 00012
           ***UNDERFLOW**POLYFI 00012     ***UNDERFLOW**POLYFI 00012
GENERATE        -1.0(.04)+1.0 IN COL 1
MULTIPLY        1,1,2
MULTIPLY        1,2,3
MULTIPLY        1,3,4
MULTIPLY        1.4.5
           ***UNDERFLOW**MULTIP 00015     ***UNDERFLOW**MULTIP 00015
PRINT           1.2.3.4.5
           ***UNDERFLOW**OPRINT 00015     ***UNDERFLOW**OPRINT 00015
           ***UNDERFLOW**OPRINT 00015     ***UNDERFLOW**OPRINT 00015
PLOT            1,2,3,4,5,1



PAGE    1      OMNITAB   1 1 1
LIST OF OMNITAB COMMANDS, DATA AND DIAGNOSTICS


     OMNITAB   1 1 1
     GENERATE 1. 1. 10.      1
     RAISE 1 3. 30. 4
     RAISE 1 2. 60. 5
     HEAD 1/      X
     HEAD 2/ X**5
              ***ILLEGAL CONSTANT                   ***ILLEGAL CONSTANT***
     HEAD 3/ 5*X**4
              ***ILLEGAL CONSTANT  X                ***ILLEGAL CONSTANT***
     HEAD 4/ 30*X**3
              ***ILLEGAL CONSTANT  X                ***ILLEGAL CONSTANT***
```

Figure 4–11. More error diagnostics. The underflow indicated that one or more numbers are less than 1.E–38. This is an example of a diagnostic which does not halt the calculations. Because the asterisk plays a special role in designating fundamental constants (see chapter 10), it must not be used elsewhere. In this case the calculations are halted.

4.5. Graphical Presentation of Data

In this section we illustrate a variety of uses to which the plot routines in OMNITAB have been put in presenting data and computed results in a dramatic and useful way. The main routine provides a plot extending over 100 type bars (approximately 10 inches) for the abscissa and 50 ordinates spaced six to the inch. The program ordinarily will plot 5 functions against a single variable. The instructions for plotting, given in section 3.2, indicate the priority of symbols used.

The plot routine accepts scale limits from the PLOT instructions. If these are missing it computes scale limits from the data to be plotted. Ordinarily if two or more functions are to be plotted together they must all be tabulated for a single set of arguments (x values). If the tables are so short that they can be placed under one another within the 101 rows available, it is possible to plot a number of functions tabulated at noncoincident values of x.

Returning once more to many functions tabulated for a single set of arguments (x) and assuming again that the tables are short enough to be placed under one another on the 101 row work-sheet, it is possible to plot more than 5 functions at a time. Under these circumstances there is no distinction made via a plot symbol between the functions in a single column. If, as is often the case, the functions are quite distinct the use of identical plot symbols poses no problem. The following figures will illustrate the variety of plotting available through imaginative use of the program.

If the scale limits are known in advance of the run, and are supplied, the plot scales will be in terms of round numbers. To insure against an inadvertent slip or a wrong estimate of the range of the numbers, it is wise to make two plots; one without scale limits and one with. In such a case the first plot will permit the correction of a bad estimate. In the examples which follow it will be obvious also how the values in one function may set the range so as to lose information on others. In such a case the only alternative is to plot them separately so that each has its maximum scale.

It will be obvious from the plots given that the resolution of 1/6 inch vertically and 1/10 inch horizontally is not good enough for some purposes. Another plot routine is available in OMNITAB which, when coupled with an off-line 407 lister having a special 40 lines to the inch gear, gives much higher resolution. The instruction SPLOT, used in the same way as PLOT, results in a plot whose ordinate is divided into 40 to the inch (10" total). It can handle 10 functions all tabulated for a single set of arguments. Periods are used for the points on each of the functions. The SPLOT instruction, oddly enough, does not produce a plot directly as does the PLOT instruction. Instead, it punches a deck of cards (approximately 400) containing periods punched in one or more of sixty columns corresponding to 60 type bars. It is these cards which, when run through a lister (407) with an appropriate forms tractor, produce the plots shown in figures 4-19 and 4-20. These plots are especially good since the functions were computed at uniform intervals of the argument so that the position along the x axis is exact. The only unevenness is in the vertical position.

The same cards when run through a normal lister, which prints six lines per inch, produce the effect of expanding the vertical scale of the plot somewhat over 7 to 1. The result of such a listing is shown in part in figure 4-18. Although it is obviously no substitute for the earlier plot it is still quite useful in many instances where graphical interpolation is either useful or necessary. It is because of this last application that we have included this discussion of the instruction SPLOT.

```
PAGE  43      OMNITAB          4/6/63
                          ELECT ENERGY CALC RUN 30 33 E (
ABSCISSA - COLUMN 15
ORDINATES - COLUMN 17 (.),
```

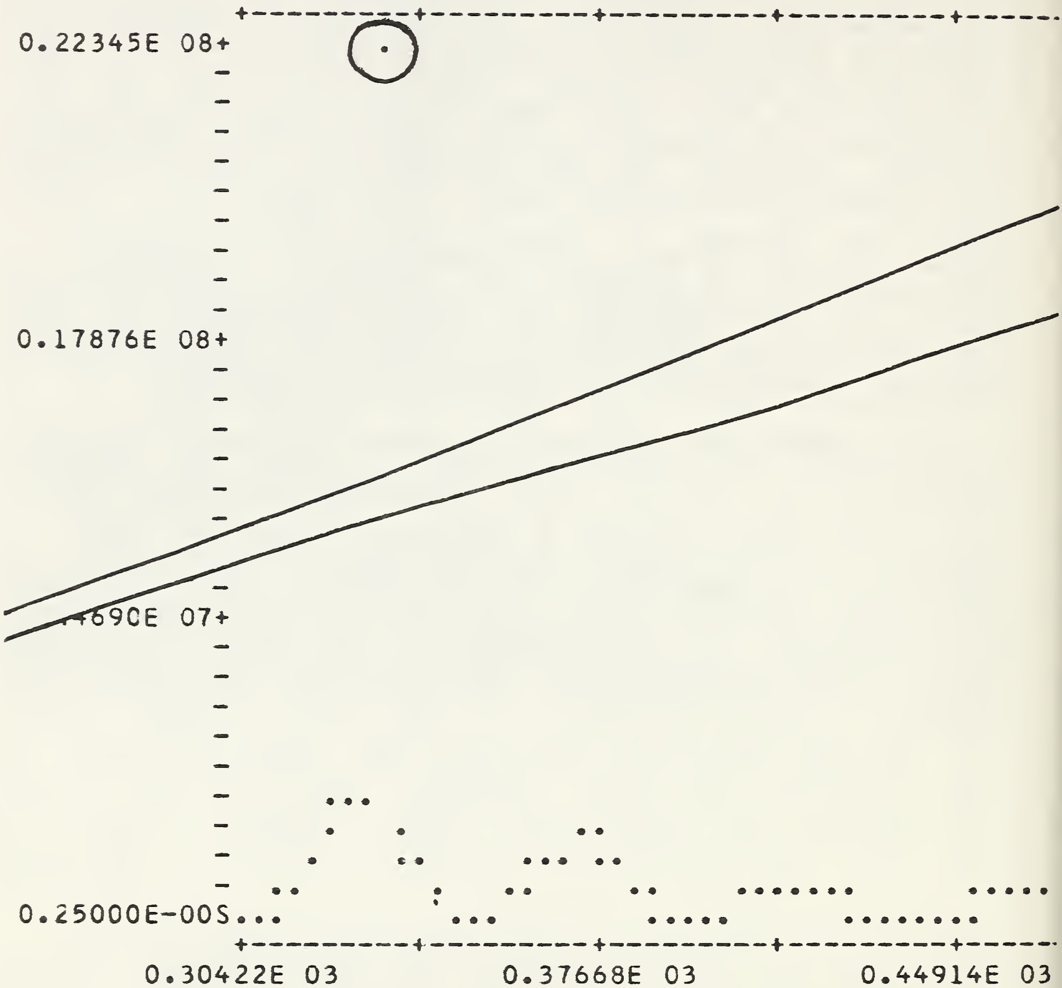


Figure 4-12. An illustration of the use of the PLOT instruction to find gross errors.

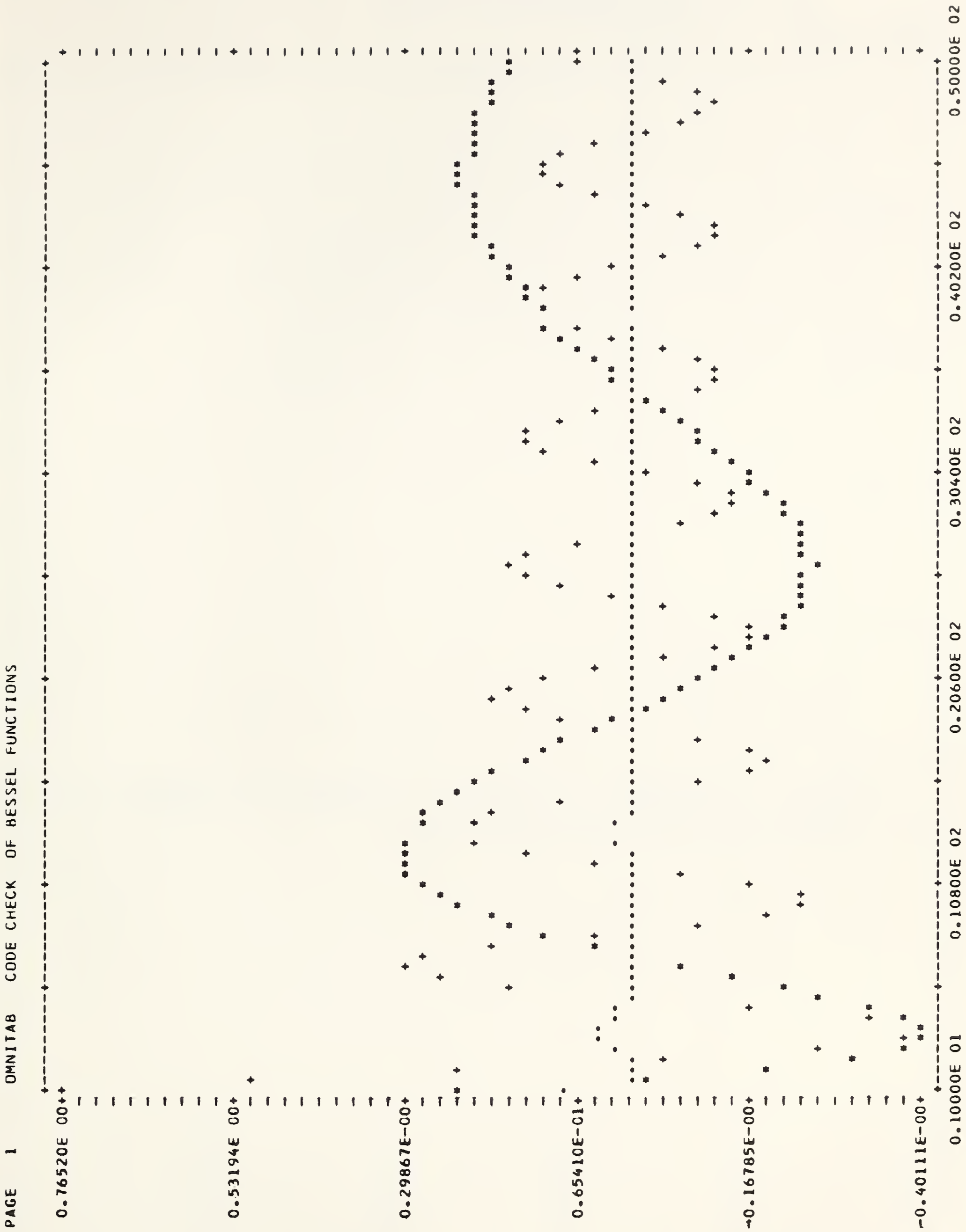
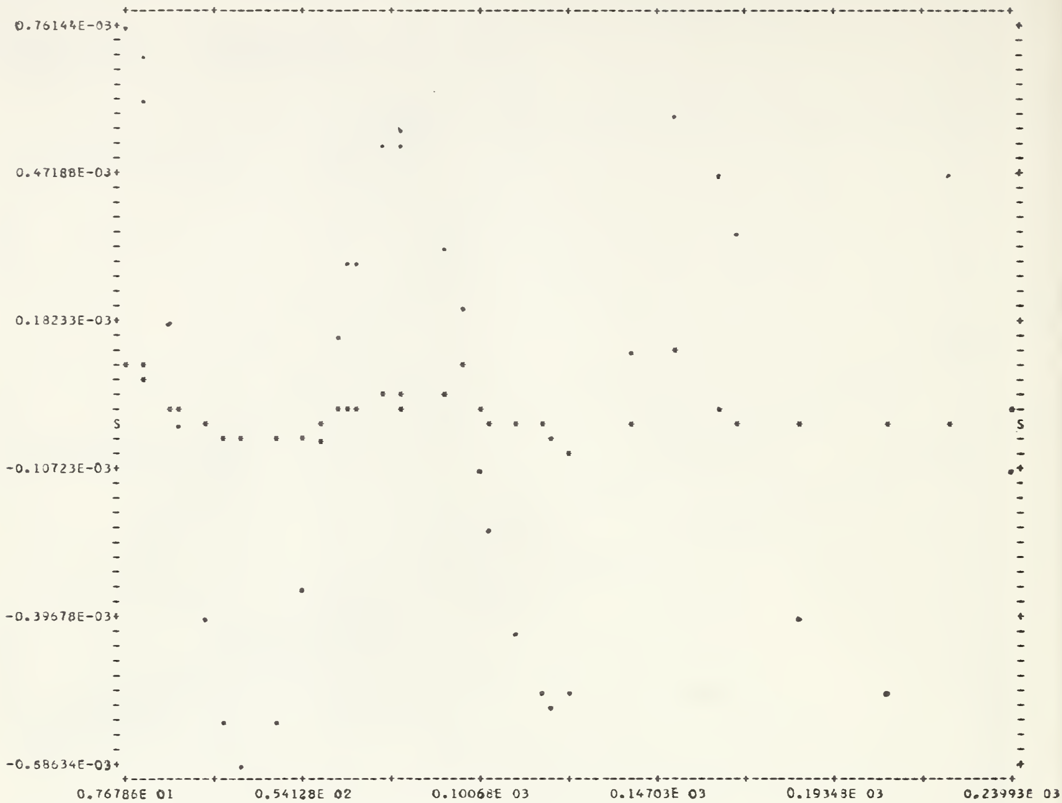


Figure 4–13 Plot of certain Bessel functions (scale limits not supplied).

PAGE 3 OMNITAB BF3 50 DEGREE ISOTHERM DENSITY VS Z 6/11/64
INCLUDING THE CORRECTION OF THE GAS HEAD
ABSCISSA - COLUMN 1
ORDINATES - COLUMN 31 (.), COLUMN 26 (*),



NOTE 6 TH DEG DENSITY FIT IN 26, 6TH DEG PRESSURE FIT IN 31 VS Z

Figure 4-14. Comparison of deviations (scale limits not supplied). The note at the bottom was supplied via a NOTE instruction.

Figure 4—15. Plot of the deviations of a set of data from the least—squares fit (scale limits not supplied).

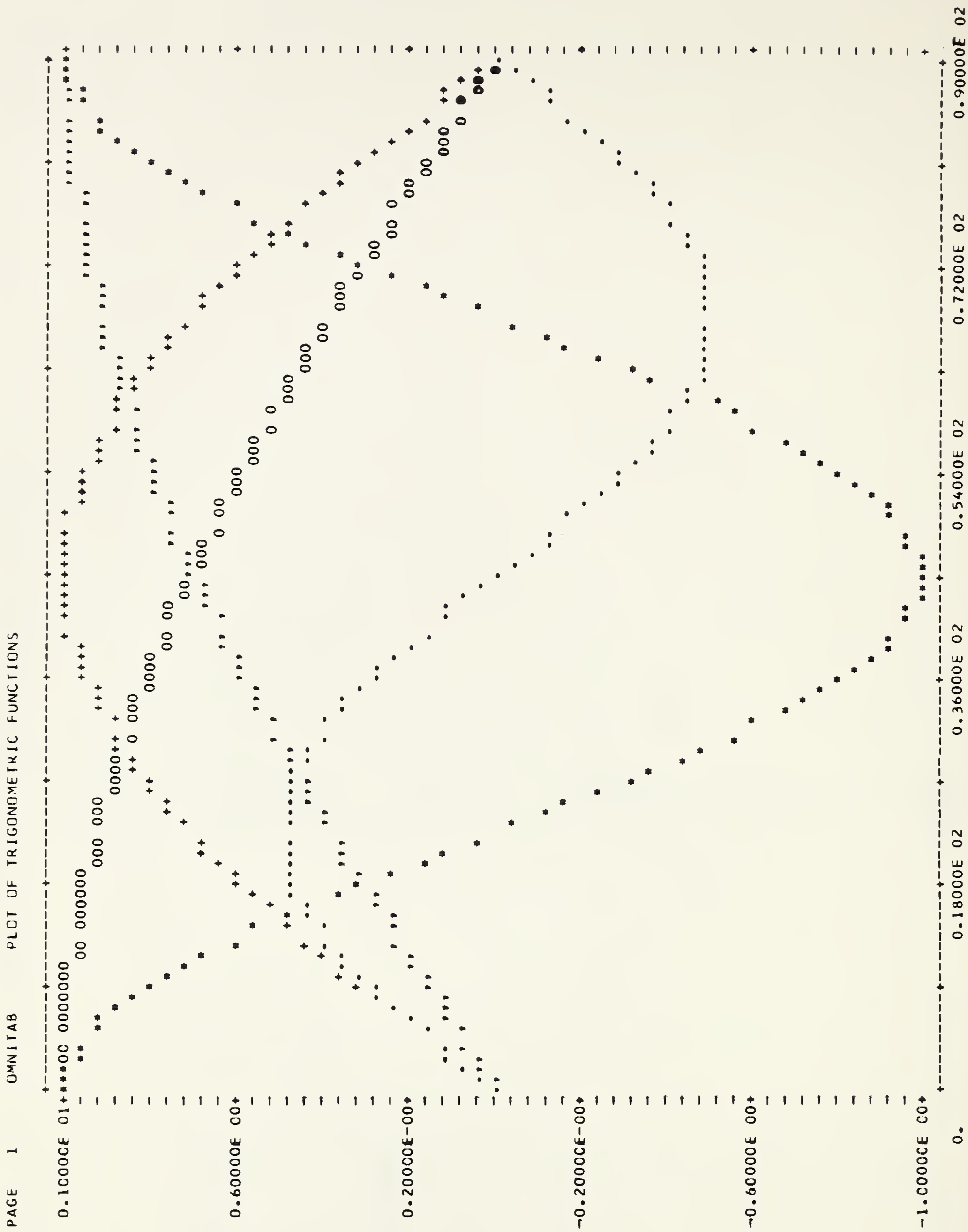


Figure 4-16. Plot of certain trigonometric functions (scale limits supplied).

Figure 4-17. Typical plot of a function tabulated so closely as to dramatize the effect of the coarse resolution (one sixth of an inch).

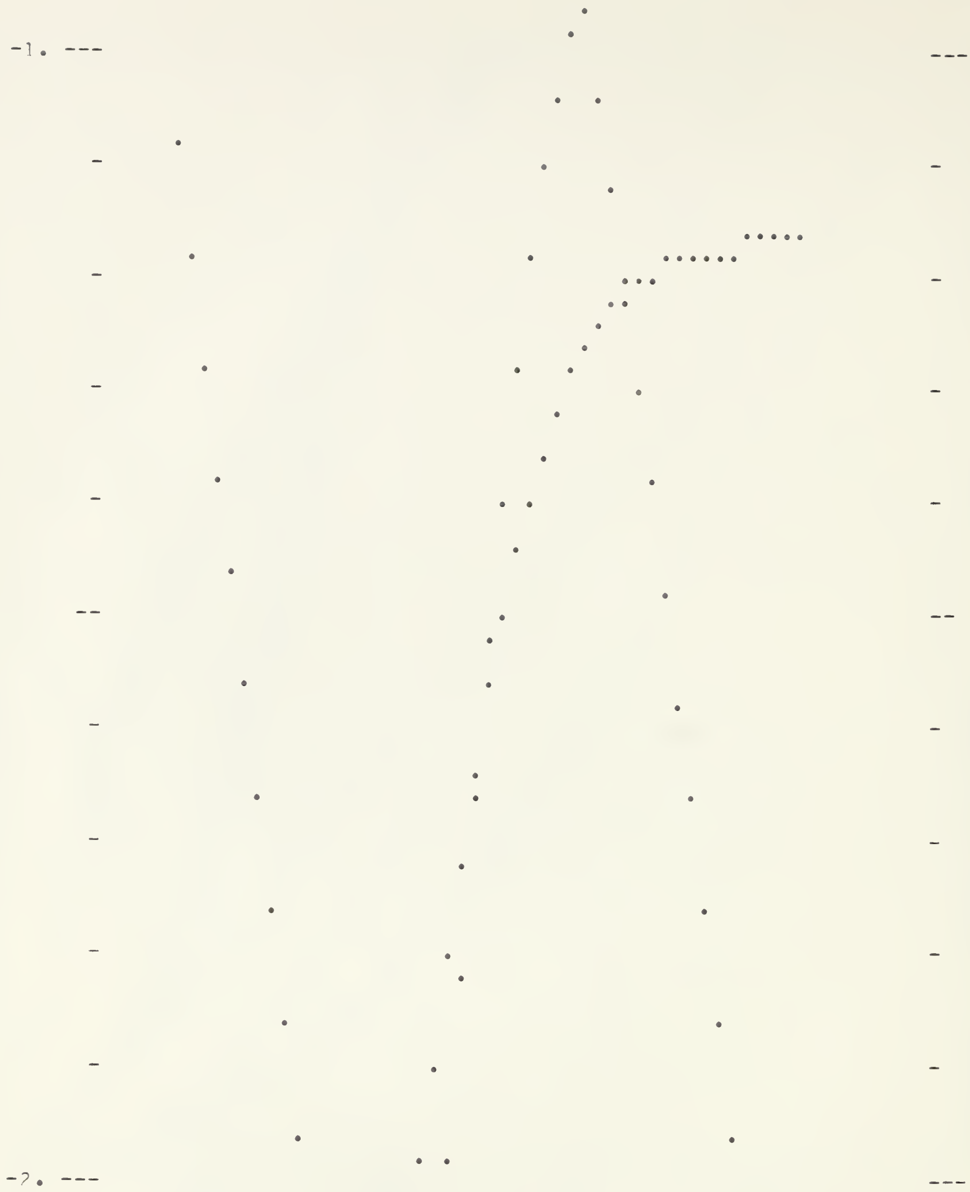


Figure 4-18. A portion of a plot obtained by listing the cards, resulting from the SPLOT instruction, on an ordinary lister six lines to the inch. The vertical scale is logarithmic.

Figure 4-19. A plot obtained by listing the cards, resulting from SPLOT, on a lister having a special forms tractor (40 to the inch).

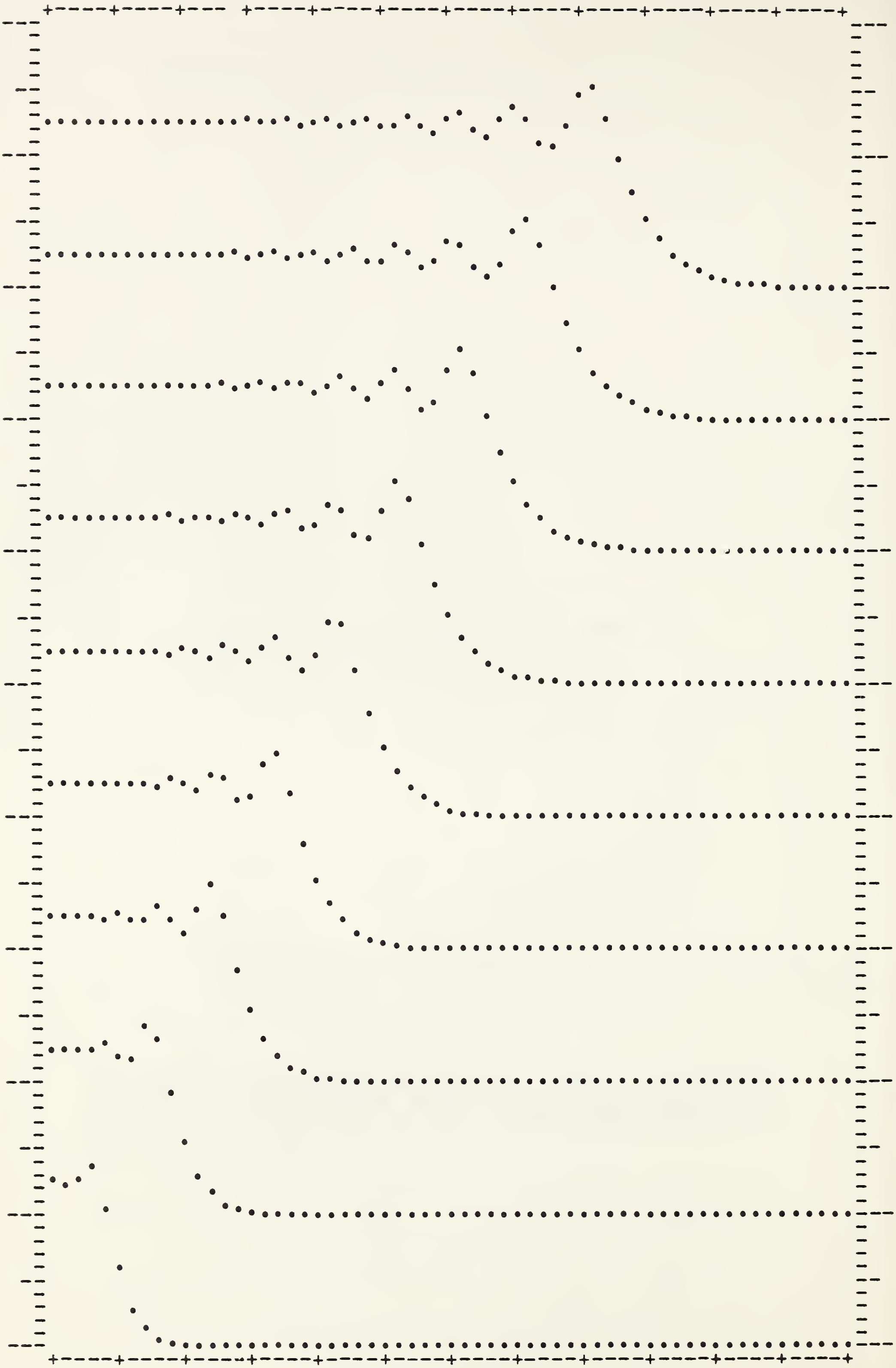


Figure 4-20. Another SPLOT product.

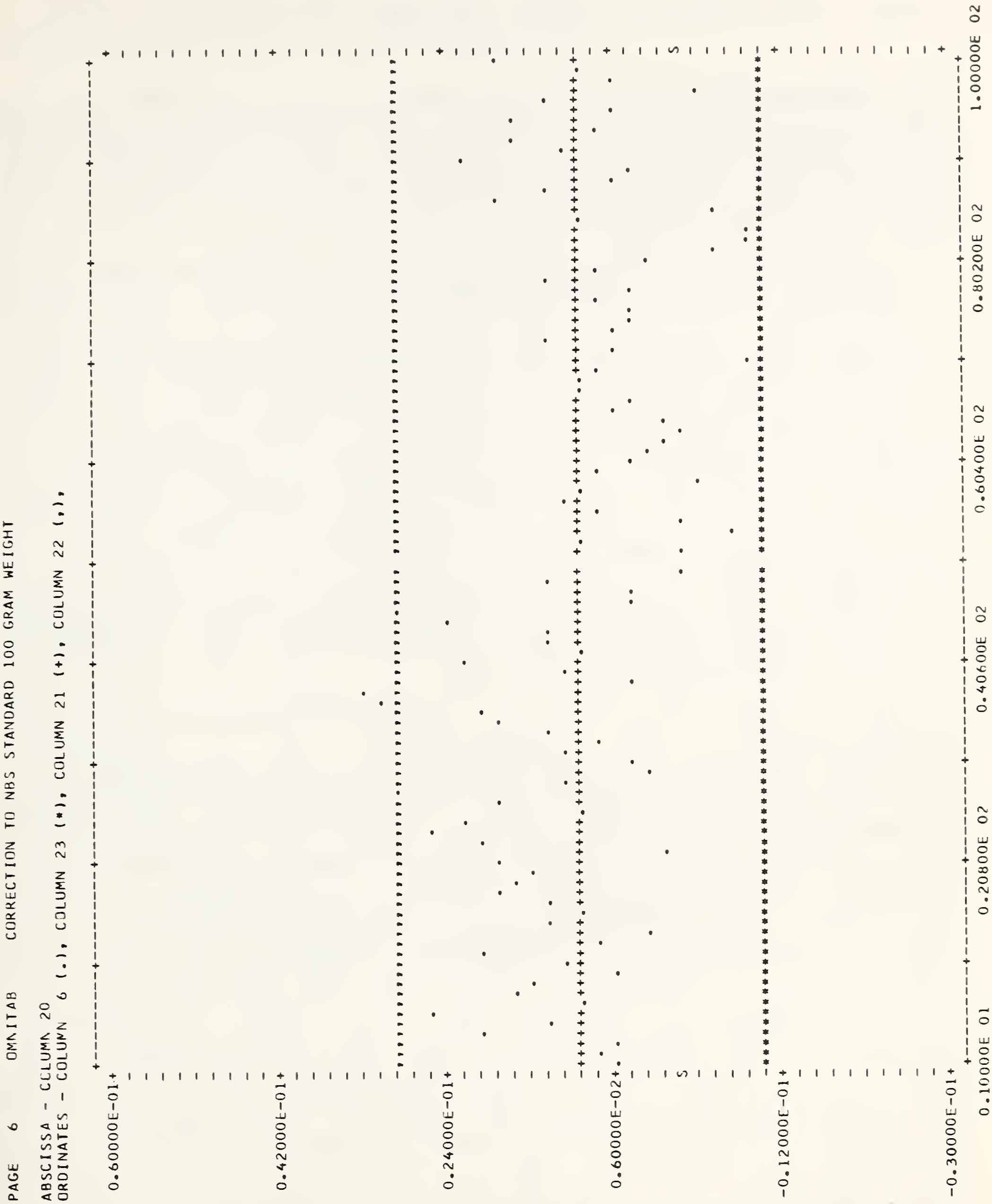


Figure 4-21. Statistical Control Chart Produced by the PLOT instruction.

4.6. Self Teaching

Experience with the program has shown that it is many things to many people, in addition to being a convenient computing tool. For some it is a teacher, for others it may be a task master, for most it is an efficient hand maiden.

The mathematical and statistical repertoire make it a useful tool in university and even secondary instruction. It is our view that a system of this kind — based as it is largely on ordinarily obvious English words — can be used effectively in assisting the teacher of mathematics science, psychology, economics, etc. in giving the student considerable first-hand experience in analyzing in depth, problems involving computations too lengthy to perform by hand. Such computer assisted instruction will be more feasible in the future when the development of inexpensive remote stations and time sharing systems will make computer time available and economical for groups who could not otherwise afford a computer. This application of OMNITAB has been discussed by one of the authors elsewhere.°

Quite apart from its application to formal pedagogy, OMNITAB can, if used imaginatively, be an effective teacher, and it is our purpose here to encourage the reader to use the program as a teacher. Such use might be divided into two main categories. We treat first the resolution of the many questions — some trivial, some serious — which may arise either in the reading of this text or in the course of using the program. While it is likely that many of the important questions will already have been anticipated and answered somewhere in the text, answers to all may not be found. We suggest strongly that the reader let the program answer his questions, and hasten to say that this will probably be more efficient, less time consuming and, in some instances, more exciting than digging the answers out of the book or asking the "experts." Furthermore, if the questions are placed at the end of a productive problem, such education can be achieved virtually free of charge because the few extra commands will require only a few microseconds for execution.

° Hilsenrath, J,. Computer Assisted Instruction in Mathematics, a talk presented at the May 2, 1964 meeting of the Maryland Section, Mathematical Association of America.

What is suggested may be better appreciated after considering the following examples:

a. Nothing is mentioned in this text concerning the ability of the program to use negative arguments in the Laguerre polynomials. In fact, since these polynomials are not orthogonal for negative arguments, the topic is not discussed in textbooks. Whether or not OMNITAB can be handled $L_n(-x)$ can be answered by generating a few negative arguments and using the LSUB instruction on them. The result gives values which show the answer to be yes.

b. An excellent example of how much can be learned from an OMNITAB experiment is afforded by figure 4-7 which gives the printout of a set of numbers in various formats. This problem was especially well designed since it shows up some of the less well known and often troublesome features characteristic of the format provision under FORTRAN.

c. Finally if one would be inclined to check on the verity of a particular statement in this Handbook concerning ranges or limits to see if they are the ultimate, one need only exceed the limit in an experiment and see. We can guarantee that in some instances the program will respond with a surprise in spite of the printed word.

The second way in which the program can be made to serve as a teacher is to use its repertoire of numerical mathematical methods to assist in problem analysis (either before or after the fact). Let us be more specific. Suppose a decision has been made to use a 5-point interpolation on a set of data on some basis. At the very least, after the table has been computed one additional instruction (DIFFER) used on the answer will provide (at almost no cost) the first six finite differences. This will document the wisdom (or lack of it) in choosing the 5-point approximation. If the choice was wise, the evidence is there. If it turns out to have been unfortunate it would be much more helpful if differences had been taken of the original table as well.

Another example of the assistance which the program can render in the matter of problem analysis is afforded by considering the evaluation of the following 10 integrals:

$$\int_0^\infty x^5 \exp(-\alpha^3 x^3) \exp(-1/x)\, dx$$

for α = 2., 1., 0.6(0.1)0.1, 0.05, 0.01

Important considerations in the analysis of problems of this type are the mesh size for integration and how far the integration should be carried to yield a practical, not mathematical, infinity. Figure 4-22 shows how the second question is answered for the 10 integrals in a single pass through the machine. In this pass we simply compute the integrand for each of the given values of α for 101 values of r between 0. and 100. This helps to determine the magnitude of the integrands, the value and location of the maximum integrand and the value of r, if any, for which the integrand becomes negligible. It is easy to see from figure 4-22 that the range over which the integration must be carried out depends strongly of the value of α ; that for α = 2., the integral is practically zero, and for α = 0.01, the integrands are so large as to put in question the physical significance of the integral for this case.

Having settled upon a set of values for the upper integration limit one needs next to consider a suitable mesh. In the OMNITAB program there is, in this case, no need to look for sophisticated integration techniques. It is by far simpler to generate 100 integrands uniformly spaced within the integration interval. In the problem at hand 101 values were chosen arbitrarily. The integrands were multiplied by the interval and summed.

It should be obvious from figure 4-22 that for the first four integrals accurate results would have been obtained by computing integrands uniformly for 101 values for 0. to 10. In these instances it was really not necessary to come off the machine. But this is hindsight. A computer technique must be programmed in such a way as to anticipate the need for refining the mesh or to provide enough information with the answers to permit the user to judge the adequacy of the mesh.

The evaluation of a single integral can often be achieved in one pass through the OMNITAB program. If use is made of the <u>repeat</u> mode, the instructions for computing the integrands can be stored and used two or more times on a successively finer mesh of points. Results can be printed out for various meshes. Alternatively use can be made of the COMPARE instruction which, together with the REPEAT instruction will allow the program to refine the mesh until a designated tolerance is satisfied. Such use of the program should be postponed, however, until the reader has had more experience with the <u>ordinary</u> and <u>repeat</u> modes of operations.

INTEGRAL OF EXP-ACUBEXCUBE XTO5 EXP-1/X 0.(1.)199. INTERVA

COLUMN 1	COLUMN 21	COLUMN 22	COLUMN 23	COLUMN 24	COLUMN 25
	α=2.	α=1.	α=0.6	α=0.5	α=0.4
0.	0.	0.	0.	0.	0.
1.000000E 00	1.234098E-04	1.353353E-01	2.964134E-01	3.246525E-01	3.450727E-01
2.000000E 00	3.112842E-27	6.510988E-03	3.447799E 00	7.140165E 00	1.163172E 01
3.000000E 00	0.	3.272583E-10	5.105478E-01	5.957960E 00	3.093005E 01
4.000000E 00	0.	1.279030E-25	7.907513E-04	2.675288E-01	1.326955E 01
5.000000E 00	0.	0.	4.808841E-09	4.189291E-04	8.582934E-01
6.000000E 00	0.	0.	3.596953E-17	1.237152E-08	6.526607E-03
7.000000E 00	0.	0.	9.715050E-29	3.492004E-15	4.263991E-06
8.000000E 00	0.	0.	0.	4.637859E-24	1.699037E-10
9.000000E 00	0.	0.	0.	0.	2.887477E-16
1.000000E 01	0.	0.	0.	0.	1.451203E-23
1.100000E 01	0.	0.	0.	0.	0.
1.[illegible]					

INTEGRAL OF EXP-ACUBEXCUBE XTO5 EXP-1/X 0.(1.)199. INTERVA

COLUMN 1	COLUMN 26	COLUMN 27	COLUMN 28	COLUMN 29	COLUMN 30
	α=0.3	α=0.2	α=0.1	α=.05	α=.01
0.	0.	0.	0.	0.	0.
1.000000E 00	3.580796E-01	3.649481E-01	3.675117E-01	3.678335E-01	3.678791E-01
2.000000E 00	1.563850E 01	1.820572E 01	1.925433E 01	1.938958E 01	1.940883E 01
3.000000E 00	8.399256E 01	1.402923E 02	1.694788E 02	1.735305E 02	1.741124E 02
4.000000E 00	1.416660E 02	4.779336E 02	7.480515E 02	7.911375E 02	7.974410E 02
5.000000E 00	8.754820E 01	9.412320E 02	2.257898E 03	2.518867E 03	2.558214E 03
6.000000E 00	1.930051E 01	1.169265E 03	5.303545E 03	6.406899E 03	6.580820E 03
7.000000E 00	1.384993E 00	9.370097E 02	1.033916E 04	1.395815E 04	1.456462E 04
8.000000E 00	2.867324E-02	4.811638E 02	1.733023E 04	2.712491E 04	2.890286E 04
9.000000E 00	1.495342E-04	1.549361E 02	2.548924E 04	4.823725E 04	5.280086E 04
1.000000E 01	1.700669E-07	3.035395E 01	3.328711E 04	7.985162E 04	9.039330E 04
1.100000E 01	3.632822E-11	3.492326E 00	3.885404E 04	1.245163E 05	1.468602E 05
1.200000E 01	1.251054E-15	2.270014E-01	4.066813E 04	1.844622E 05	2.285412E 05
1.300000E 01	5.948369E-21	8.001123E-03	3.820891E 04	2.612410E 05	3.430484E 05
1.400000E 01	3.338997E-27	1.465505E-04	3.220439E 04	3.553498E 05	4.993757E 05
1.500000E 01	0.	1.335219E-06	2.430857E 04	4.658921E 05	7.080070E 05
1.600000E 01	0.	5.787568E-09	1.639028E 04	5.903339E 05	9.810195[illegible]
1.700000E 01	0.	1.140748E-11	9.840323E 03	7.244139E 05	1.33[illegible]E 07
1.800000E 01	0.	9.767783E-15	5.241191E 03	8.622524E 05	[illegible]90029E 07
1.900000E 01	0.	3.469802E-18	2.466520E 03	9.966832[illegible]	5.612687E 07
				[illegible]E 05	6.416122E 07
2.000000E 01	0.	4.881946E-22	1.021127E 03	[illegible]327E 04	7.305645E 07
2.100000E 01	0.	2.596646E-26	3.701820E [illegible]	[illegible].296001E 04	8.287535E 07
2.200000E 01	0.	4.982377E-31	1.16[illegible]		
2.300000E 01	0.	0.	[illegible]771E-20	3.350327E 04	9.368019E 07
2.400000E 01	0.	0.	1.322275E-22	2.050340E 04	1.055346E 08
2.500000E 01	0.	0.	8.509480E-25	1.213125E 04	1.185032E 08
2.600000E 01	0.	[illegible]	4.244214E-27	6.935238E 03	1.326517E 08
2.700000E 01	0.	[illegible]	1.631011E-29	3.828474E 03	1.480465E 08
2.800000E 01	0.	0.	0.	2.039511E 03	1.647542E 08
2.900000E 01	[illegible]	0.	0.	1.047821E 03	1.828418E 08
[illegible]	[illegible]	0.	0.	5.188355E 02	2.023761E 08
3.00[illegible]	0.	0.	0.	2.474412E 02	2.234235E 08
[illegible]0000E 01	0.	0.	0.	1.135871E 02	2.460497E 08

?

Figure 4-22. Values of the integrands as a function of α resulting from a preliminary problem analysis run.

```
OMNITAB              7 / 20 / 64
TITLE1 CALCULATION OF SEDIMENTATION COEFFICIENT - SCHACHMAN, 4-55
TITLE2TABLE ONE-TABULATION OF EXPERIMENTAL DATA-PROGRAM A-2,0.7 CERULOPL
READ COLS 1, 2, 3, 4, 5
880  1  131.166  139.127  39.88
880  2  131.166  142.234  52.88
880  3  131.166  143.237  56.88
880  4  131.166  144.269  60.88
880  5  131.166  145.204  64.88
HEAD COL 1/1 PLATE NO.
HEAD COL 2/2 FRAME NO.
HEAD COL 3/3 MENISCUS
HEAD COL 4/4 X PEAK, MM
HEAD COL 5/5 TIME, MIN
FOOTNOTE    X303 RUN 1/24/64 LOG 5-13 HAND CALC 75-101 (S=7.3)
PRINT COLS 1, 2, 3, 4, 5
NOTE  1-THE METHOD IS TAKEN FROM - METHODS IN ENZYMOLOGY,-4-55
TITLE1 FIGURE ONE - COL 4 (.) (MEASUREMENT OF PEAK) VS. COL 5 (TIME)
PLOT COL 4 AGAINST COL 5
NOTE  2-THE POINTS SHOULD FORM A ST. LINE  DEVIANT PTS. MAY BE ERRORS
MULTIPLY COL 5 BY 60.0, STORE IN COL 6
HEAD COL 6/6 TIME, SEC.
ADD 5.9780E04 TO COL 7, STORE IN COL 7 (THIS INTRODUCES THE RPM SPEED)
HEAD COL 7/7 SPEED, RPM
MULTIPLY COL 7 BY 0.10472, STORE IN COL 8
HEAD COL 8/8 OMEGA,/SEC.
MULTIPLY COL 8 BY COL 8, STORE IN COL 9
HEAD COL 9/ OMEGA, SQ
MULTIPLY COL 4 BY 0.046860, STORE IN COL 10
HEAD 10/10 X PEAK, CM. (PLATE MEAS/CAMERA LENS MAGNIFICATION CONSTANT)
TITLE1 TABLE TWO - INTERMEDIATE CALCULATIONS
TITLE2PLATE MEASUREMENTS BY TRAUTMAN-S METHOD ARE CONVERTED TO REAL MEAS
FOOTNOTE    DATA AND CONSTANTS ARE ON COLORED CARDS, FOR EASE OF CHANGE
NOSUMMARY
PRINT COLS 6, 7, 8, 9, 10
NOTE  3-USUALLY ONLY A FEW OF THESE MANY COLUMNS ARE NEEDED OR PRINTED
NOTE  4-THE PROGRAM CAN BE CONDENSED USING MULTIPLE-STEP INSTRUCTIONS
NOTE  5-ALL ENTRIES IN COLUMNS ARE IN FLOATING POINT NOTATION
NOTE  6-TIMER WAS STARTED WHEN SPEED REACHED 2/3 RDS OF PRESENT SPEED
LOGE COL 10, STORE IN COL 11
HEAD COL 11/11 LN X    (THIS IS THE NATURAL LOG OF RADIUS OF PEAK)
TITLE1 FIGURE TWO - COL 11 (.) (LN X) VS. COL 6 (TIME)
PLOT COL 11 AGAINST COL 6
NOTE  7-THIS PLOT SHOULD BE A ST. LINE, WITH SLOPE PROP. TO SED. COEF.
DIVDIF COL 11, ARG IN COL 6
EXCHANGE COL 41 WITH COL 12
```

Figure 4-23. A sample of input wherein full advantage was taken of the opportunity to annotate both the instructions and the final results.

4.7. Multilingual Options

At various stages during the development of the program an opportunity presented itself to expand the vocabulary list by the inclusion of synonyms for commands in a number of foreign languages. An early version of the program is available in English, French, German, and Japanese. As the vocabulary list grew beyond 100 commands, the space taken up by the multiplicity of synonyms became too precious, and the foreign vocabulary was removed.

Figure 4-24 shows a portion of the multilingual vocabulary. If such a provision is desired, the main vocabulary list can easily be augmented and reassembled to achieve this. This is, however, a chore which will require the services of an experienced systems programmer as it will be necessary to place the revised program on the system tape.

* * * * * * * * * * * *

COMPLETE	COMLETE	VOLLSTAENDIG	ZENSEKIBUN
DEFINE		DEFINIEREN	UTSUSU
DERIVATIVE		ABLEITUNG	BIKEISU
DELETE			TORU
DEMOTE			SAGERU
DIAGONALIZE			TAIKAKUKA
DIFFERENCE	DIFFERENCE	DIFFERENZ	KAISA
EXCHANGE	EXCHANGER	AUSTAUSCHEN	TORIKAE
FIXED	POINTF	FIXPUNKTE	KOTEISHOSU
FLIP			GYAKUTEN
FLOATING	POINTM		IDOSHOSU
GENERATE			OKOSU
HEAD	TETE	KOPF	HYODAI
INCREMENT			HENKA
INCOMPLETE		UNVOLLSTAENDIG	FUKANZEN

Figure 4-24. A sample portion of a listing of the current status of the multilingual vocabulary.

TABLE B. Location of Results Stored Automatically by Various OMNITAB Commands

COMMANDS		COL 41	COL 42	COL 43	COL 44	COL 45	COL 46
DIFFER DIVDIF		Δ_i	Δ_i^2	Δ_i^3	Δ_i^4	Δ_i^5	Δ_i^6
STATISTICAL ANALYSIS°°°				Analysis in 34 rows See Fig 5-1	Rank	Ordered X	Deviations $X_i - \bar{X}$
FIT POLYFIT SOLVE					s.d. of A s.d. of B s.d. of C etc.	s.d. A B C etc.	Deviations $Y_{exp}-Y_{calc}$
LINEAR INVERT							$\|C\|$°°
TSUB USUB PSUB HSUB LSUB	$F_n(X_i)$		$F_5(X_i)$	$F_4(X_i)$	$F_3(X_i)$	$F_2(X_i)$	$F_1(X_i)$
ISOLATE						Roots	Overflow
SELECT		2nd closest if any	3rd closest if any	4th closest if any	5th closest if any	6th closest if any	7th closest if any
ISETUP					Averaged Y_i	Averaged X_i	Desired Y_i
MINEL						Row of min	Col of min
MAXEL						Row of max	Col of max

° Note that the standard deviations of the coefficients start in row 2.

°° The value of the determinant of the coefficients is in row 1.

°°° STATISTICAL ANALYSIS cannot be performed on columns 43 through 46.

5. STATISTICAL AND NUMERICAL ANALYSIS

The interest in general-purpose programs referred to in chapter 1 is nowhere so strong as in the field of statistics, where numerous comprehensive programming packages and systems have been devised, and where a number of committees are active in surveying and evaluating statistical programs. Mention has already been made of BMD (U. C. L. A.). Other programs in this class include Princeton University's P-STAT, IBM's STORM (statistically oriented, matrix programs), the University of California Berkeley Computer Center's STATPAK, and, no doubt, many others.

From the point of view of general statistics and analysis of variance, OMNITAB may seem relatively limited. It contains only two major subroutines in statistics. One is an orthonormalization program for performing least-squares fitting (not just of polynomials, but of any function set). The other is a comprehensive statistical analysis subroutine which computes 34 statistical quantities representing various measures of central tendency, randomness, skewness, kurtosis, etc. These operations, augmented by the mathematical, manipulative, and plotting capabilities, enable OMNITAB to solve many of the curve fitting, statistical, and regression problems arising in the analysis of laboratory data in the physical sciences. The versatility of the program is enhanced by the variety of function generation, and numerical mathematical analysis which can be performed interchangeably with the statistical analysis.

As an example, suppose it is desired to pass a curve through the points of maxima and another through the minima of a set of data. This can be achieved by use of the instructions: READ, MAXMIN, and POLYFIT. The addition of a PLOT instruction provides also a graphical solution. (See section 5-4.) Or suppose a comparison is sought between results from fitting a particular set of data to a 5th degree polynomial and to Tchebyschev polynomials. This can be accomplished simply by the instructions: READ, POLYFIT, TSUB, and FIT. The use of a few more commands would enable one to compare the residuals and the quality of the fit, graphically and numerically. Finally, consider the analysis of a body of data from which it is desired to remove a linear trend. This can be accomplished with the instructions: FIT and STATIS. The process of fitting a straight line to the data leaves the deviations in column 46 of the work sheet. These are then analyzed via the statistical analysis command.

5.1. Numerical and Statistical Analysis Commands

1. DIFFERENCE COL ++, ARGUMENT IN COL ++
   a. six differences are computed and stored automatically. (See table B)
   b. meaningful only for values given for uniformly spaced arguments.
   c. output is automatic with a fixed format. (See fig. 5-14)
   d. the average of the absolute differences is also printed.
   e. SDIFFER suppresses the printout which is otherwise automatic.

2. DIVDIF COL ++, ARGUMENT IN COL ++
   a. gives divided differences.
   b. applicable to a table for non-uniformly spaced arguments.
   c. gives wrong values when two adjacent arguments are equal.
   d. output is automatic and results are stored in the work sheet. (See table B)
   e. SDIVDIF suppresses the printout which is otherwise automatic.

3. STATISTICAL ANALYSIS OF COL ++, WEIGHTS IN ++
   a. computes the mean of a column of numbers, the deviations of each of the numbers from their mean, the rank, and 34 other statistical measures.
   b. stores certain results automatically. (See table B)
   c. output is automatic. (See figures 5-1 and 5-2)
   d. uses unit weights when no weight column is given.
   e. SSTATIS suppresses the printout.

4. FIT COL ++, WEIGHTS IN COL ++, VECTORS IN ++, ++, ...
   a. provides a least squares fit of the type

   $$y = f(x) = Ax_1 + Bx_2 + Cx_3 + Dx_4 + \dots$$

   b. $f(x)$ is defined by the vectors $x_1, x_2 \dots x_n$ in the designated columns ($n \leq 23$).
   c. the output is automatic (see figures 5-6 and 5-7) and certain results are stored in columns 44, 45, 46 (see table B).
   d. the deviations, Δy_i, stored in column 46, may be used with the original y_i to obtain computed values from the relation y_i comp. $= y_i$ obs. $- \Delta y_i$.
   e. the standard deviation of the fit, σ, and the coefficients A, B, C, D, ... are stored in column 45 (see table B).

f. the standard deviations of the coefficients, δA, δB, δC, etc., stored starting in row 2 of column 44, are the square roots of the diagonal terms of figure 5-6).
g. SFIT etc., suppresses the automatic printout.
h. these results are obtained by a subroutine called ORTHO° which uses the Gram-Schmidt Orthonormalization process °° .

5. POLYFIT COL ++, WEIGHTS IN ++, X IN ++, USE ,, DEGREE
a. fits a polynomial of degree n so that

$$y = a_0 + a_1 x + a_2 x^2 + \dots a_n x^n \quad ; \quad 23 \geq n$$

is satisfied in the least squares sense.
b. the program generates the vectors appropriate to the degree from the tabulated values of x.
c. output is automatic (see figures 5-6 and 5-7).
d. deviations are stored in column 46 and may be subtracted from the original y values to get computed values.
e. the standard deviation, σ, of the fit, and the coefficients a_0, a_1, a_2, ... are stored in column 45 (see table B).
f. the standard deviations of the coefficients, δa_0, δa_1, δa_2, etc., stored starting in row 2 column 44, are the square roots of the diagonal terms of the variance covariance matrix (see table B).
g. for many reasons polynomials of high degree should be used with extreme caution.
h. SPOLYFIT suppresses the automatic printout.

° Walsh, P., Ortho, Comm A. C. M., 5 511-13 (1962).

°° Davis, P. J. Orthonormalizing Codes in Numerical Analysis in Survey of Numerical Analysis, John Todd, ed. pp. 347-379 McGraw-Hill Book Company, New York, 1962.

Davis, P. J. and Rabinowitz, P., Advances in Orthonormalizing Computation in Advances in Computers, Vol. 2, F. L. Alt, ed. pp. 55-133 Academic Press, New York, 1961.

Davis, P. J., and Rabinowitz, A Multiple Purpose Orthonormalizing Code and its Uses, J. Assoc. Comp. Mach. 1, 183-191, 1954.

6. SOLVE FOR COL ++, COEFFICIENTS IN ++, ++, ++, ...

a. provides for a solution of a system of n equations in m unknowns,

$$y_i = \sum_{j=1}^{m} a_{ij} x_j, \; i = 1 \ldots n,$$

for y_i and the coefficients a_{ij} stored in the designated columns.

b. the equations are solved exactly for n = m and in the least squares sense for n > m.

c. the limit for m and n are $2 \leq m \leq n$; $n \leq 101$; $m \leq 23$.

d. output is automatic.

e. results are stored in columns 44, 45, 46 as in items 4 and 5 above.

7. GQUAD WITH ** POINTS, A = ** B = ** STORE X IN ++, WEIGHTS IN ++

a. generates the n abscissae, x_j, and the weight coefficients W_j for the Gaussian quadrature formula

$$\int_a^b f(x)\,dx = \sum_{j=1}^{n} W_j \, f(x_j)$$

b. the region of integration is divided into n/4 intervals, in each of which 4-point Gaussian quadrature coefficients and weights are computed.

c. the number of points, n, can be any multiple of 4 from 4 to 100.

d. the abscissae, x_i, are stored in the first indicated column and corresponding weights, W_i, in the second.

8. MAXMIN X IN ++,Y IN ++, STORE MAX IN ++, ++, MIN IN ++, ++

a. finds the maxima and minima in a function defined by its tabulated values. (Locates hills and valleys)

b. validity depends on the adequacy of the interval of tabulation and monotonic arrangement of x for single valued functions.

c. the program identifies the extreme value in each run° of the data and fits a parabola, $y = ax^2 + bx + c$, through the extreme point and one on each side.

d. determines $x_{m_i} = -b/2a$ from the derivative.

° A "run" is sequence of values which are monotonic.

e. lists x_{m_i} and $f(\bar{x}_{m_i})$ from the parabolic fit and indicates if it is a MAX or a MIN.
f. the results are presented in an automatic printout (see problem 5-6).

9. °INTERPOLATE X IN ++, Y IN ++, L = ,, FOR THE FIRST ,, VALUES OF X IN COL ++, USE ,, POINTS, STORE IN ++
a. provides n-point Lagrangian interpolation for y = f(x) for the specified number of x's in the designated column.
b. the number L denotes the length of the original table indicated in the first two column designations.
c. the original table must be monotonic and increasing in the independent variable but need not be uniformly spaced.
d. since the program permits, but does not limit the range of extrapolation beyond the ends of the table, extreme extensions will provide answers which may be unreliable.
e. the maximum number of points is 12 and the minimum (corresponding to linear interpolation) is 2. An even number of points gives better results.

10. °DERIVATIVE X IN ++, Y IN ++, L =,, FOR THE FIRST ,, ENTRIES IN ++, USE ,, POINTS, STORE IN ++
a. provides n-point Lagrangian derivatives for the specified number of X's in the designated column.
b. the number L denotes the length of the original table indicated in the first two column designations.
c. the original table must be monotonic and increasing in the independent variable but need not be uniformly spaced.
d. since the program permits, but does not limit the range of extrapolation beyond the ends of the table, extreme extensions will provide answers which will be unrealiable.
e. the maximum number of points is 12 and the minimum (corresponding to parabolic approximation) is 3. An odd number of points gives smoother derivatives.

11. °LAGINTEGRATION COL ++, USE ,, POINTS, H = **, STORE IN ++
a. provides for 3,4,5,6,7,8, or 9-point Lagrangian integration.
b. the program computes the integral numerically for each value in the argument column.
c. arguments must be uniformly spaced.

° This like all instructions, must be punched on a single card.

d. the integration is achieved through the formula:

$$\int_{x_0}^{x_0 + (m+1)h} f(x)dx = h \sum_k [F_k^n(m)(f(a_k))]$$

where the $f(a_k)$ are functional values stored in the designated column, h is the indicated tabular interval, and F_k^n (m) are the Lagrangian integration coefficients taken from "Tables of Lagrangian Interpolation Coefficients," Columbia University Press, New York (1944).

12.° HARMONIC ANALYSIS OF COL ++, FOR ,, ORDINATES, STORE COEF IN ++

a. determines the coefficients of the equation

$$Y = a_0 + \sum_{k=1}^{n} a_k \cos kx + \sum_{k=1}^{n-1} b_k \sin kx$$

which fits the designated ordinates corresponding to the abscissae, x, which are distributed uniformly between 0 and 360 degrees, or between 0 and 2π radians.

b. the number of ordinates must be even and cannot exceed 48.

c. for 2n points, the coefficients are stored as follows $a_0, a_1, \ldots a_n, b_1, b_2, \ldots b_{n-1}$

13. CENSOR COL ++ FOR $$,REPLACING BY $$,STORE IN COL ++

a. compares each line of the designated column with a fixed value or with a column of values and replaces those values which are ≤ than the indicated ones by values in the third designated column or by a constant, and stores the revised values as indicated.

b. to censor values larger than a particular value, change the sign of the values, censor for the negative of the desired value, and change the sign back.

14.° SELECT IN COL ++ VALUES APPROXIMATING COL ++ TO WITHIN **, STORE IN COL ++

a. searches in the first designated column and selects values approximating any of those in the second column to the designated tolerance and stores as indicated.

b. if more than one value is found approximating a given value to within the tolerance, the closest one is stored as indicated.

c. if no value is found, a zero is stored on the corresponding line.

° Instructions must be punched on a single card.

d. in addition columns 40–46 contain the values satisfying the criterion as well as all further values – up to a maximum of six – which obey the tolerance.

15. SEARCH IN COL ++ FOR NUMBERS IN ++, TRANSFER CORRESP VALUES FOR ++ INTO ++,++ INTO ++,++ INTO ++,...

a. provides for a search of a multicolumned table to select rows corresponding to values in the first named column identical with values in the second named column.

b. the column on which the search is made must be monotonically increasing.

c. the search is terminated on encountering three consecutive zeros in either of the argument columns.

d. if a value is not found, a zero is placed in the corresponding row.

16. CHOP X IN COL ++ STORE [X] IN COL ++

a. computes [X], the largest integer less than or equal to X, for the values of X in each row of the designated column up to NRMAX and stores the result in the specified column.

b. difficulties may be encountered if X is very large.

c. regardless of the sign of X, CHOP merely removes the fractional part of the number, i.e. −3.5 becomes −3.0.

5·2 Statistical Analysis

The instruction STATIS (see page 112) represents one of the more powerful commands in OMNITAB· It provides an automatic printout of a frequency distribution (in deciles), rank ordering and 34 statistical properties relating to a set of measurements· The results of STATIS are stored in columns 43–46 in accord with the table on the next page· The notations in the last column refer to references listed below·

REFERENCES

BNLE = BROWNLEE, K· A· (1960) Statistical Theory and Methodology in Science and Engineering, Wiley, New York·

D/M = DIXON, W· J·, Massey, F· J· Jr· (1957) INTRODUCTION TO STATISTICAL ANALYSIS, McGraw-Hill, New York·

Duncan = DUNCAN, A· J· (1959) Quality Control and Industrial Statistics, 2nd· ed·, Irwin, Homewood, Illinois·

Hald = HALD, A· (1952) Statistical Theory with Engineering Applications, Wiley, New York·

H–91 = NATRELLA, M· G· (1963) Experimental Statistics, NBS Handbook 91, U·S· Government Printing Office, Washington, D·C·

5.2. Statistical Analysis

Frequency Distribution — count of no. of values in 10 equal intervals between minimum and maximum

The following are computed from the X array

		(Row, Col)	Reference Page Numbers
n = NRMAX		(1, 43)	
Non Zero Weights = NZW		(2, 43)	
Sum of Weights	$\sum_{i=1}^{n} W_i$	(3, 43)	
Sum of Values	$\sum_{i=1}^{n} X_i$	(4, 43)	
Weighted Mean	$\frac{\sum_{i=1}^{n} (W_i X_i)}{\sum_{i=1}^{n} W_i} = \bar{X}$	(5, 43)	BNLE 72 to 74; Hald 243 to 245
Unweighted Mean	$\sum_{i=1}^{n} X_i/n$	(6, 43)	D/M 14 to 17, 112 to 130; H-91 1-10, Ch. 3; Hald 67 to 72
Minimum Value	$X_{min.}$	(7, 43)	D/M 275 to 278, 293 to 294; H-91 Ch. 7; Hald 329 to 336; Duncan 113 to 114
Median	for n odd, $X_{(n+1)/2}$ for n even, $\frac{X_{(n/2)} + X_{(n+1)/2}}{2}$	(8, 43)	D/M 70 to 75, 295 to 297
Maximum Value	$X_{max.}$	(9, 43)	D/M 275 to 278, 293 to 294; H-91 Ch. 7; Hald 329 to 336; Duncan 113 to 114
Weighted Sum of Squares	$\sum_{i=1}^{n} (W_i X_i^{\ 2})$	(10, 43)	
Variance	$\frac{\sum_{i=1}^{n} (dev_i)^2 W_i}{NZW - 1} = S^2$	(11, 43)	D/M 18 to 21, 80, 102 to 110; H-91 1-10; Hald 72 to 77, Ch. 11; BNLE 218 to 223
Standard Deviation	S	(12, 43)	D/M 18 to 21, 80, 102 to 110; H-91 1-10; Hald 72 to 77, Ch. 11; BNLE 218 to 223
S.D. of Mean	$S/\sqrt{\Sigma W_i}$	(13, 43)	D/M 36 to 42, 50 to 52; H-91 2-1 to 2-5; Hald 199 to 202; Duncan 98 to 103
Range	$X_{max.} - X_{min.}$	(14, 43)	Hald 319 to 329; H-91 2-6; Duncan 44, 111 to 113
Coeff. of Var.	$100S/\bar{X}$	(15, 43)	Hald 77, 301
Student's t	$\bar{X}(\sqrt{\Sigma W_i})/S$	(16, 43)	H-91 3-3 to 3-8; D/M 115 to 119; BNLE 227 to 230
Mean Sq. Successive Diff.	$D = \frac{\sum_{i=1}^{n-1} (X_{i+1} - X_i)^2}{n - 1}$	(17, 43)	Hald 257 to 258
Mean Sq. Succ. Diff./Var.	D/S^2	(18, 43)	D/M 299; Hald 357 to 359; BNLE 161 to 164
No. of Runs Up and Down	RUNS	(19, 43)	
Expected No. of Runs (ENR)	$(2n - 1)/3$	(20, 43)	
S.D. of No. of Runs (SDR)	$[(16n-29)/90]^{1/2}$	(21, 43)	

Diff/S.D. of Runs	$\frac{\lvert RUNS - ENR \rvert}{SDR}$	(22, 43)	Hald 353 to 357; Duncan 117 to 122

The following are computed from the deviations from the mean

No. of + signs in dev	m_1	(23, 43)	
No. of − signs in dev	m_2	(24, 43)	
(No. of sign changes in dev) + 1 = DRUNS		(25, 43)	
Expected No. of Runs (ENRD)	$1 + (2m_1m_2/n)$	(26, 43)	
S.D. of No. of Runs (SDRD)	$\left[\frac{2m_1m_2(2m_1m_2-m_1-m_2)}{(m_1+m_2)^2(n-1)}\right]^{1/2}$	(27, 43)	
Diff/S.D. of Runs	$\frac{\lvert DRUNS - ENRD \rvert}{SDRD}$	(28, 43)	Duncan 117 to 122; Hald 342 to 353; D/M 288 to 289; BNLE 170 to 171
Trend (Linear)	$T = \frac{12 \sum_i^n (i(dev_i))}{n(n^2-1)}$	(29, 43)	
S.D. of Trend	$\frac{1}{\sqrt{n-2}}\left[\frac{12 \sum_i^n (dev_i)^2}{n(n^2-1)} - T^2\right]^{1/2}$	(30, 43)	
Trend/S.D. of Trend		(31, 43)	D/M 191 to 196; H-91 5-19; Hald 540; BNLE 280 to 283
Beta One	$\frac{\left[\frac{1}{n} \sum_i^n (dev_i)^3\right]^2}{\left[\frac{n-1}{n} \cdot S^2\right]^3}$	(32, 43)	Duncan 49 to 51, 496 to 501
Beta Two	$\frac{\frac{1}{n} \sum (dev_i)^4}{\left[\frac{n-1}{n} \cdot S^2\right]^2}$	(33, 43)	Duncan 49 to 51, 496 to 501
Mean Deviation	$\frac{\sum \lvert dev_i \rvert}{n}$	(34, 43)	Duncan 44 to 46

Column 44 contains the ranks

Column 45 contains the ordered X_i

Column 46 contains $X_i - \bar{X} = dev_i$

On this and the following pages we illustrate the output which is achieved automatically by the instruction

STATISTICAL ANALYSIS OF COL 6

```
PAGE   1      OMNITAB     CORRECTION TO NBS STANDARD 100 GRAM WEIGHT
                                               STATISTICAL ANALYSIS OF COL  6

                     FREQUENCY DISTRIBUTION    6   6  10  22  20  12  13   5   3   3

                COMPUTATIONS ON THE X ARRAY            COMPUTATIONS ON DEVIATIONS

                          N= 100                          NO OF + SIGNS=  46
               NON ZERO WEIGHTS= 100                      NO OF - SIGNS=  54
                 SUM OF WEIGHTS=  1.000000E 02               NO OF RUNS=  40
                  SUM OF VALUES=  9.910961E-01     EXPECTED NO OF RUNS=  5.068000E 01
                  WEIGHTED MEAN=  9.910961E-03           S.D. OF RUNS=  4.942520E 00
                UNWEIGHTED MEAN=  9.910961E-03      DIFF./S.D. OF RUNS=  2.160841E 00
                  MINIMUM VALUE= -8.795980E-03                  TREND= -9.915385E-05
                         MEDIAN=  8.974615E-03          S.D. OF TREND=  2.942139E-06
                  MAXIMUM VALUE=  3.358555E-02     TREND/S.D. OF TREND= -3.370128E 01
             WTD SUM OF SQUARES=  1.756906E-02               BETA ONE=  6.178769E-02
                       VARIANCE=  7.824594E-05               BETA TWO=  2.992088E 00
                          S.D.=  8.845673E-03          MEAN DEVIATION=  6.887437E-03
                  S.D. OF MEAN=  8.845673E-04
                          RANGE=  4.238153E-02
                 COEFF. OF VAR.=  8.925142E 01
                    STUDENT*S T=  1.120430E 01
       MEAN SQ SUCCESSIVE DIFF.=  9.534669E-05
       MEAN SQ SUCC. DIFF./VAR.=  1.218551E 00
         NO OF RUNS UP AND DOWN=  61
           EXPECTED NO OF RUNS=  6.633333E 01
             S.D. OF NO OF RUNS=  4.177985E 00
             DIFF./S.D. OF RUNS=  1.276532E 00
```

Figure 5-1. Automatic Printout for the Command STATIS. Part I.

CORRECTION TO NBS STANDARD 100 GRAM WEIGHT

N	X(N)	W(N)	RANK	DEV(N)	N(S)	ORDERED X	DELTA
1	6.580590E-03	1.0	36.0	-3.330371E-03	82	-8.795980E-03	2.657200E-04
2	8.235690E-03	1.0	45.0	-1.675271E-03	70	-8.530260E-03	5.849600E-04
3	6.410480E-03	1.0	34.0	-3.500481E-03	83	-7.945300E-03	8.531800E-04
4	1.966143E-02	1.0	87.0	9.750469E-03	53	-7.092120E-03	1.468180E-03
5	1.403487E-02	1.0	74.0	4.123909E-03	85	-5.623940E-03	9.249500E-04
6	2.520990E-02	1.0	95.0	1.529894E-02	81	-4.698990E-03	1.804590E-03
7	9.583470E-03	1.0	52.0	-3.274906E-04	58	-2.894400E-03	1.602200E-04
8	1.715016E-02	1.0	78.0	7.239199E-03	97	-2.734180E-03	9.399600E-04
9	1.465738E-02	1.0	75.0	4.746419E-03	50	-1.794220E-03	7.808000E-05
10	6.026150E-03	1.0	33.0	-3.884811E-03	54	-1.716140E-03	2.459300E-04
11	1.072621E-02	1.0	58.0	8.152494E-04	63	-1.470210E-03	1.067640E-03
12	2.059352E-02	1.0	89.0	1.068256E-02	51	-4.025700E-04	7.412400E-04
13	8.106350E-03	1.0	43.0	-1.804611E-03	22	3.386700E-04	5.476500E-04
14	1.624460E-03	1.0	16.0	-8.286500E-03	62	8.863200E-04	1.533000E-04
15	1.379478E-02	1.0	72.0	3.883819E-03	64	1.039620E-03	5.848400E-04
16	8.756880E-03	1.0	49.0	-1.154081E-03	14	1.624460E-03	6.677000E-04
17	1.235771E-02	1.0	64.0	2.446749E-03	30	2.292160E-03	3.283000E-04
18	1.892900E-02	1.0	84.0	9.018039E-03	80	2.620460E-03	5.697000E-04
19	1.725161E-02	1.0	79.0	7.340649E-03	61	3.190160E-03	2.842500E-04
20	1.558733E-02	1.0	76.0	5.676369E-03	66	3.474410E-03	1.975900E-04
21	1.864457E-02	1.0	82.0	8.733609E-03	75	3.672000E-03	2.395000E-04
22	3.386700E-04	1.0	13.0	-9.572290E-03	74	3.911500E-03	4.899001E-05
23	2.049446E-02	1.0	88.0	1.058350E-02	60	3.960490E-03	2.895100E-04
24	2.667689E-02	1.0	96.0	1.676593E-02	89	4.250000E-03	9.016995E-05
25	2.131724E-02	1.0	92.0	1.140628E-02	77	4.340170E-03	1.682100E-04
26	9.792920E-03	1.0	53.0	-1.180406E-04	39	4.508380E-03	2.486700E-04
27	1.913440E-02	1.0	86.0	9.223439E-03	31	4.757050E-03	2.034900E-04
28	2.900016E-02	1.0	97.0	1.908920E-02	48	4.960540E-03	1.200400E-04
29	1.165688E-02	1.0	62.0	1.745919E-03	47	5.080580E-03	5.257002E-05
30	2.292160E-03	1.0	17.0	-7.618801E-03	95	5.133150E-03	1.275600E-04
31	4.757050E-03	1.0	27.0	-5.153911E-03	88	5.260710E-03	9.118998E-05
32	1.198924E-02	1.0	63.0	2.078279E-03	73	5.351900E-03	6.742500E-04
33	7.070780E-03	1.0	40.0	-2.840181E-03	10	6.026150E-03	3.843300E-04
34	1.402235E-02	1.0	73.0	4.111389E-03	3	6.410480E-03	4.268001E-05
35	1.853967E-02	1.0	81.0	8.628709E-03	71	6.453160E-03	1.274300E-04
36	2.087438E-02	1.0	90.0	1.096342E-02	1	6.580590E-03	8.761999E-05
37	3.043199E-02	1.0	99.0	2.052103E-02	65	6.668210E-03	1.543800E-04
38	3.358555E-02	1.0	100.0	2.367459E-02	98	6.822590E-03	2.183900E-04
39	4.508380E-03	1.0	26.0	-5.402581E-03	69	7.040980E-03	2.979999E-05
40	1.099741E-02	1.0	60.0	1.086449E-03	33	7.070780E-03	2.477200E-04
41	2.130365E-02	1.0	91.0	1.139269E-02	79	7.318500E-03	2.037300E-04
42	1.044822E-02	1.0	57.0	5.372594E-04	55	7.522230E-03	5.841200E-04
43	1.318228E-02	1.0	69.0	3.271319E-03	13	8.106350E-03	2.265000E-05
44	1.241148E-02	1.0	65.0	2.500519E-03	76	8.129000E-03	1.066900E-04
45	2.393603E-02	1.0	94.0	1.402507E-02	2	8.235690E-03	1.764996E-05
46	2.943015E-02	1.0	98.0	1.951919E-02	93	8.253340E-03	2.731000E-04
47	5.080580E-03	1.0	29.0	-4.830381E-03	59	8.526440E-03	1.968200E-04
48	4.960540E-03	1.0	28.0	-4.950421E-03	84	8.723260E-03	3.362005E-05
49	1.354527E-02	1.0	71.0	3.634309E-03	16	8.756880E-03	1.658200E-04
50	-1.794220E-03	1.0	9.0	-1.170518E-02	52	8.922700E-03	1.038300E-04

Figure 5-2. Automatic Printout from the Command STATIS.
Part II Showing: Rank, Deviations, etc.

COLUMN 43	COLUMN 44	COLUMN 45	COLUMN 46
100.0000000	36.0000000	-0.0087960	-0.0033304
100.0000000	45.0000000	-0.0085303	-0.0016753
100.0000000	34.0000000	-0.0079453	-0.0035005
0.9910961	87.0000000	-0.0070921	0.0097505
0.0099110	74.0000000	-0.0056239	0.0041239
0.0099110	95.0000000	-0.0046990	0.0152989
-0.0087960	52.0000000	-0.0028944	-0.0003275
0.0089746	78.0000000	-0.0027342	0.0072392
0.0335855	75.0000000	-0.0017942	0.0047464
0.0175691	33.0000000	-0.0017161	-0.0038848
0.0000782	58.0000000	-0.0014702	0.0008152
0.0088457	89.0000000	-0.0004026	0.0106826
0.0008846	43.0000000	0.0003387	-0.0018046
0.0423815	16.0000000	0.0008863	-0.0082865
89.2514248	72.0000000	0.0010396	0.0038838
11.2043028	49.0000000	0.0016245	-0.0011541
0.0000953	64.0000000	0.0022922	0.0024467
1.2185512	84.0000000	0.0026205	0.0090180
61.0000000	79.0000000	0.0031902	0.0073406
66.3333330	76.0000000	0.0034744	0.0056764
4.1779846	82.0000000	0.0036720	0.0087336
1.2765325	13.0000000	0.0039115	-0.0095723
46.0000000	88.0000000	0.0039605	0.0105835
54.0000000	96.0000000	0.0042500	0.0167659
40.0000000	92.0000000	0.0043402	0.0114063
50.6799994	53.0000000	0.0045084	-0.0001180
4.9425205	86.0000000	0.0047570	0.0092234
2.1608407	97.0000000	0.0049605	0.0190892
-0.0000992	62.0000000	0.0050806	0.0017459
0.0000029	17.0000000	0.0051331	-0.0076188
-33.7012758	27.0000000	0.0052607	-0.0051539
0.0617877	63.0000000	0.0053519	0.0020783
2.9920876	40.0000000	0.0060261	-0.0028402
0.0068874	73.0000000	0.0064105	0.0041114
0.	81.0000000	0.0064532	0.0086287
0.	90.0000000	0.0065806	0.0109634
0.	99.0000000	0.0066682	0.0205210

Figure 5-3. Printout of results stored automatically by the instructions STATIS and SSTATIS. This printout is not automatic.

5.3. Least-Squares Curve Fitting

The curve fitting in OMNITAB is accomplished with the aid of a versatile subroutine which is called ORTHO — short for Orthonormalization. This subroutine, referred to briefly in section 5.1, provides for fitting in the least squares sense or in the exact sense, and can be used for the following related operations:

a. fitting of data to polynomial functions or to any of the algebraic or transcendental functions in the OMNITAB repertoire.

b. fitting of empirical data in two or more dimensions as a function of several independent variables.

c. solutions of systems of linear functional equations.

d. expansion of a given set of functional values in series of orthogonal functions — trigonometric functions, Legendre polynomials, Tchebychev polynomials.

The above operations are only a few of the mathematical applications of the orthonomalization technique for which the reader is referred to the series of papers by Davis and Rabinowitz cited in section 5.1.

The almost universal use of polynomial approximations in certain segments of the scientific literature leaves the casual reader with a distorted view of the superiority of polynomials over other functional forms of approximation. That this influence has extended to machine programmers may be inferred from the preponderance of programs for least-squares fitting which are specialized to polynomials.

In view of the generality of the ORTHO subroutine, we illustrate its use first in the more general sense. We consider first a matrix of measurements composed of a column vector v_i representing the dependent variable and three column vectors x_i, y_i, z_i representing three independent variables. The numbers are arranged as follows:

$$
\begin{matrix}
v_1, & x_1, & y_1, & z_1 \\
v_2, & x_2, & y_2, & z_2 \\
v_3, & x_3, & y_3, & z_3 \\
\cdot & & & \cdot \\
\cdot & & & \cdot \\
\cdot & & & \cdot \\
v_n, & x_n, & y_n, & z_n
\end{matrix}
$$

If the functional relationship between these measurements is considered to be a linear combination of the variables

$$v_i = ax_i + by_i + cz_i ,$$

the OMNITAB program requires simply to be told the location of the vectors v_i, x_i, y_i, and z_i and the location of a column of weights w_i. In case the curve does not go through the origin, and the relationship is still of the same form:

$$v_i = ax_i + by_i + cz_i + d,$$

it is necessary to supply also a column vector of 1's to accommodate the constant term d. The order in which the vectors are specified is not important for the overall fit, but does influence the sum of the squared residuals which gives the amount by which the sum of the squared residuals are decreased by each of the terms of the equation.

If the measurements are believed to obey the relationship

$$v_i = a_1x_i + a_2x_i^2 + b_1y_i + b_2y_i^2 + c_1z_i + c_2z_i^2$$
$$+ dx_iy_i + ex_iz_i + fy_iz_i + g ,$$

it will be necessary to generate from the original measurements x_i, y_i, z_i, the vectors x_i^2, y_i^2, z_i^2, x_iy_i, x_iz_i, y_iz_i and a vector of 1's.

OMNITAB can handle the least-squares fitting of as many as 23 vectors. The instruction is:

FIT COL ++, WEIGHTS IN COL ++, VECTORS IN ++, ++, etc

The same instruction can be used regardless of the nature of the functional forms of the vectors provided only that the unknowns are in the coefficients. Thus, if a set of measurements yield the vectors, h_i,θ_i and a fit is desired of the form

$$h = a_0 + a_1 \sin \theta_i + a_2 \sin 2\theta_i + \cdots b_1 \cos \theta_i + b_2 \cos 2\theta_i + \cdots;$$

it is necessary only to generate the unit vector and the vectors

$$\sin \theta_i, \quad \sin 2\theta_i \cdots, \quad \cos \theta_i, \quad \cos 2\theta_i \cdots$$

Having the vectors for one or more independent variables, it is possible in OMNITAB to fit a variety of functions represented by the equations

$$y_i = a_0 + a_1 \varphi_1(x_{1i}) + a_2 \varphi_2(x_{2i}) + \cdots$$

or

$$y_i = a_0 + \sum_j a_j \varphi_j(x_{ji}) \quad .$$

The φ_j may represent polynomial functions, trigonometric, exponential, hyperbolic functions, etc.

The need to generate powers or other functions for certain of the experimental vectors becomes obvious when one recalls how such calculations are carried out on a desk calculator. It should also be clear that after the vectors are supplied, the algebraic method of solution is identical for all functional forms. In the case of a polynomial fit of the form

$$y_i = a_0 + a_1 x_i + a_2 x_i^2 + a_3 x_i^3 + \cdots ,$$

we have introduced a special command which obviates the need to supply the vectors x^2, x^3, or the vectors of 1's. The program generates these automatically for the designated column of x_i on the special command :

POLYFIT ++, WEIGHTS IN ++, X IN ++, USE 4TH DEGREE

This command can only be used when the functional form is a polynomial of the above type with no terms missing. If terms are missing, the vectors must be generated and the instruction is FIT, etc., instead of POLYFIT.

A similar problem arises when we wish to place a restraint on the fitting function by specifying one or more of the coefficients. For example, if we should desire to fit compressibility data to a virial equation in pressure of the form

$$Z = 1 + BP + CP^2 + DP^3 ,$$

we cannot use POLYFIT. We must fit $(Z - 1) = BP + CP^2 + DP^3$ in which case we must generate the vectors $Z - 1$, P^2, P^3 from the original vectors Z, P. In OMNITAB the generation of these vectors poses no particular problem.

In the course of curve-fitting it is often desired to compute the function for a set of values of X other than those at which the data are tabulated. Normally such calculations would be carried out using the least-square coefficients obtained from the curve-fitting routine. It is, however, not necessary to make two passes. The values of the function for a second set of arguments can be obtained at the same time as the curve-fitting is done, simply by augmenting the matrix of input data by a suitably arranged matrix containing a vector of the new values of X and zeros in the corresponding rows of the y vector and the vector of weights.

Thus if x_i, y_i, and w_i represent vectors corresponding to the abscissae, ordinates, and weights of the data to be fitted, and values are desired for a set of abscissae $X_1, X_2, X_3, \cdots, X_n$, the augmentation of the above three vectors as follows:

$$\left.\begin{matrix} x_1, & y_1, & w_1 \\ x_2, & y_2, & w_2 \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ x_n, & y_n, & w_n \end{matrix}\right\} \text{original data array}$$

$$\left.\begin{matrix} X_1, & 0, & 0 \\ X_2, & 0, & 0 \\ X_3, & 0, & 0 \\ \cdot & & \cdot \\ \cdot & & \cdot \\ X_n, & 0, & 0 \end{matrix}\right\} \text{augmented array}$$

will produce the desired result.

The values for $f(X_i)$ will appear with the opposite sign in the column headed DEVIATIONS. These are subtracted from the given values of y to yield deviations. Since the Y_n were all zeros, the deviation is the negative of the desired Y_n values. The zeros in the weight column insure that these points are not included in the least-squares fitting.

In the example which follows, 36 points are fitted to a polynomial of the second degree. The weights are not included with the data. They are computed in the ADD instruction preceding the READ statement. These will serve as the weights for the experimental points. The 36 points have, however, been augmented by 14 values of X extending from 0. to .325 in intervals of .025. The READ statement resets the number of lines from 36 to 50 so that the subsequent operations are carried out on the augmented matrix. Since column 3, which contains the weights, is blank below the 36th line, the weights for the last 14 points are zero; hence these points do not influence the fit.

The printout for the POLYFIT and the FIT instruction (see figures 5-6 and 5-7) gives: the coefficients, the standard deviation, the covariance matrix, the square root of the covariance diagonal, the variance-covariance matrix, the square root of the variance-covariance matrix diagonal, the Gram determinant, the Fourier coefficients, the squared Fourier coefficients, the sum of the squared residuals, the residuals, the A_{ij} matrix, the Gram factors, the vector norms, and the deviations.

```
CMNITAB                  11/8/62
TITLE2  X IS BLADE-TO JET-SPEED RATIO,Y IS TOTAL-TO-STATIC
NCSUMMARY
GENERATE 1.(1.)36. 30
ADC       1. TC 0.,3
READ 1,2
3.86 .126
4.20 .138
4.23 .140
4.31 .144
4.55 .157
4.55 .159
4.69 .162
4.74 .164
4.75 .166
4.79 .168
4.91 .173
4.97 .178
5.20 .188
5.33 .194
5.42 .203
5.37 .208
5.62 .217
5.66 .219
5.74 .230
5.79 .230
5.75 .237
5.95 .243
5.97 .244
5.96 .247
6.06 .258
6.09 .260
6.02 .266
6.15 .269
6.22 .270
6.24 .270
6.16 .273
6.25 .277
6.19 .288
6.23 .288
6.07 .294
6.11 .297
0. 0.
0. .025
0. .05
0. .075
0. .10
0. .125
0. .15
0. .175
0. .20
0. .225
0. .25
0. .275
0. .30
0. .325
MULT CCL 1 BY .1 STORE IN COL 1
POLYFIT Y IN 1 WTS IN 3 X IN 2 2ND DEGREE
TITLE1  Y IN 1 WTS IN 3 X  IN 2 2ND DEGREE
SUB   46 FRCM CCL 1 STORE IN COL 4
ADC   46 TO 0. 5
HEAD 1/  Y
HEAD 2/  X
HEAD 4/COMPUTEC
HEAD 5/CEVIATICN
FIXEC 4 DECIMALS
PRINT 2 1 5 4
       STCP
```

Figure 5-4. Data and instructions for a polynomial fit. See Figure 5-5 for results.

```
PAGE    3      OMNITAB                    11/8/62
  Y IN 1 WTS IN 3 X  IN 2 2ND DEGREE
  X IS BLADE-TO JET-SPEED RATIO,Y IS TOTAL-TO-STATIC
   X                Y                DEVIATION      COMPUTED
```

X	Y	DEVIATION	COMPUTED
0.1730	0.4910	0.0020	0.4890
0.1780	0.4970	-0.0012	0.4982
0.1880	0.5200	0.0042	0.5158
0.1940	0.5330	0.0073	0.5257
0.2030	0.5420	0.0021	0.5399
0.2080	0.5370	-0.0104	0.5474
0.2170	0.5620	0.0018	0.5602
0.2190	0.5660	0.0031	0.5629
0.2300	0.5740	-0.0030	0.5770
0.2300	0.5790	0.0020	0.5770
0.2370	0.5750	-0.0103	0.5853
0.2430	0.5950	0.0031	0.5919
0.2440	0.5970	0.0040	0.5930
0.2470	0.5960	-0.0001	0.5961
0.2580	0.6060	-0.0007	0.6067
0.2600	0.6090	0.0005	0.6085
0.2660	0.6020	-0.0116	0.6136
0.2690	0.6150	-0.0009	0.6159
0.2700	0.6220	0.0053	0.6167
0.2700	0.6240	0.0073	0.6167
0.2730	0.6160	-0.0030	0.6190
0.2770	0.6250	0.0032	0.6218
0.2880	0.6190	-0.0097	0.6287
0.2880	0.6230	-0.0057	0.6287
0.2940	0.6070	-0.0248	0.6318
0.2970	0.6110	-0.0223	0.6333
0.	0.	0.0037	-0.0037
0.0250	0.	-0.0885	0.0885
0.0500	0.	-0.1736	0.1736
0.0750	0.	-0.2516	0.2516
0.1000	0.	-0.3226	0.3226
0.1250	0.	-0.3864	0.3864
0.1500	0.	-0.4431	0.4431
0.1750	0.	-0.4928	0.4928
0.2000	0.	-0.5353	0.5353
0.2250	0.	-0.5708	0.5708
0.2500	0.	-0.5991	0.5991
0.2750	0.	-0.6204	0.6204
0.3000	0.	-0.6346	0.6346
0.3250	0.	-0.6417	0.6417

Figure 5-5. Results of a polynomial fit to data shown in Figure 5-4. Note that the function is computed for the lower values whose weights are set to zero.

5.4 Problems in Statistical and Numerical Analysis

Problem 5 - 1

Given 34 experimental points representing the data of state of boron trifluoride at 0°C, fit polynomials of 4, 5, 6, and 7th degree, print, plot and save the deviations to permit an easy comparison and especially to determine the suitability of any of the polynomial representations indicated above.

The data are read into the machine with the following instruction set: (Because the READ instruction asks only for 2 columns, the last two are ignored.)

```
OMNITAB BF3 0 DEGREE PV VS  D IN AMAGAT UNIT 10/16/64
TITLE1  Z VALUE AT 0 DEG 1 ATM IS 0.99409
READ   1  2
9.98416   .93770     336  5
11.94314   .92478    335  5
14.24323   .90932    337  4
16.61896   .89300    336  4
19.67139   .87149    335  4
23.16199   .84606    337  3
26.65774   .81959    336  3
30.98029   .78529    335  3
35.67938   .74568    337  2
40.11188   .70560    336  2
45.19485   .65547    335  2
50.19874   .60029    337  1
54.41398   .54766    336  1
58.66507   .48683    335  1
61.76115   .43666    345  2
64.00868   .39926    345  2
65.34411   .37831    346  2
65.88513   .37059    343  2
66.59998   .36103    338  2
67.73169   .34719    342  2
68.63512   .33791    347  2
68.83938   .33628    341  2
69.63908   .32967    340  2
70.31823   .32499    339  2
75.74023   .30640    344  1
86.78898   .30976    345  1
99.98467   .33122    346  1
107.31114   .34537   343  1
119.52989   .37076   338  1
146.49848   .4297     342  1
174.65717   .49207     347  1
181.14272   .50637   341  1
210.80997   .57110    340  1
239.10488   .63242    339  1
```

```
ACC     1.  0.  3
MULT    2  1.00595  4
DIV     1  4  5
 POLYFIT 4  3  5  4TH DEGREE
EXCHANGE 44 11 45 15 46 19
 POLYFIT 4  3  5  5TH DEGREE
EXCHANGE 44 12 45 16 46 20
 POLYFIT 4  3  5  6TH DEGREE
EXCHANGE 44 13 45 17 46 21
 POLYFIT 4  3  5  7TH DEGREE
EXCHANGE 44 14 45 18 46 22
SPACE 5
NOTE                           STANDARD DEVIATIONS
SPACE 1
NOTE       4TH              5TH              6TH               7TH
SPACE 1
ABRIDGE ROW 1 OF COLS 15 16 17 18
SPACE 1
NOTE                            COEFFICIENTS
SPACE 1
APRINT 2 15  8 4
SPACE 1
NOTE                           UNCERTAINTIES
SPACE 1
APRINT ROW 2 COL 11  8  BY 4
FIXED  6
NOSUMMARY
HEAD   1/   P(ATM)
HEAD   2/     Z
HEAD   4/     PV(AMAG)
HEAD   5/    D (AMAG)
PRINT  1  2  4  5
PLOT   4  5       0. 600. .3 1.0
HEAD COL  19/      4TH
HEAD COL 20/       5TH
HEAD COL 21/       6TH
HEAD COL 22/      7TH
TITLE2                      DEVIATIONS
PRINT 19 20 21 22
PLOT  19 20 21 AGAINST 5
NOTE                                        DEVIATION PLOT
PLOT 22 21 20 AGAINST 5
NOTE                                        DEVIATION PLOT
PLOT 22 AGAINST 5
NOTE                                         DEVIATION PLOT
       STOP
```

° The array starting in row 2, col 15 having dimensions 8 by 4 . See chapter 8 for a discussion of this command.

The following eight pages are an abridgment of the printout for this problem. Brief comments on the results are given in the following figures.

```
PAGE   7      CMNITAB BF3 0 DEGREE PV VS  D IN AMAGAT UNIT 10/16/64

 NUMBER OF NCN-ZERC WEIGHTS =     3.4000000E 01

 POLYNOMIAL FIT CF CEGREE    7 TO FUNCTION IN COLUMN    4
 USING WEIGHTS IN CCLUMN    3


 COEFFICIENTS ANC THEIR STANDARD CEVIATIONS

 1.C074366E 00  4.8398630E-04
-6.2508308E-03  4.4390433E-05
 1.9830401E-05  1.3033790E-06
-6.2270507E-C8  1.7255221E-08
 4.7483951E-10  1.1776473E-10
-2.3975730E-12  4.2975670E-13
 5.4957886E-15  7.9628235E-16
-4.2497746E-18  5.8841590E-19

 STANDARD DEVIATICN    2.5610186E-04

 COVARIANCE MATRIX

 3.57142E 00
-3.14663E-01  3.00437E-02
 8.67543E-03 -8.65539E-04  2.59010E-05
-1.07720E-04  1.10389E-05 -3.39320E-07  4.53958E-09
 6.92704E-07 -7.22662E-08  2.26458E-09 -3.07856E-11  2.11449E-13
-2.39528E-09  2.53083E-10 -8.04523E-12  1.10738E-13 -7.68395E-16
 2.81592E-18
 4.22646E-12 -4.50818E-13  1.449CCE-14 -2.01432E-16  1.40942E-18
-5.20171E-21  9.66738E-24
-2.98625E-15  3.20882E-16 -1.04044E-17  1.45809E-19 -1.02736E-21
 3.81463E-24 -7.12707E-27  5.27890E-30

 SQUARE RCOT OF CCVARIANCE MATRIX DIAGONALS

 1.88982E 00  1.73331E-01  5.C8930E-03  6.73764E-05  4.59835E-07
 1.67807E-09  3.10924E-12  2.29759E-15

 VARIANCE CCVARIANCE MATRIX

 2.34243E-07
-2.06382E-08  1.97051E-09
 5.69005E-10 -5.67691E-11  1.69880E-12
-7.06518E-12  7.24019E-13 -2.22554E-14  2.97743E-16
 4.54332E-14 -4.73981E-15  1.48530E-16 -2.01917E-18  1.38685E-20
-1.57102E-16  1.65992E-17 -5.27672E-19  7.26313E-21 -5.03976E-23
 1.84691E-25
 2.77205E-19 -2.95683E-20  9.50375E-22 -1.32116E-23  9.24413E-26
-3.41170E-28  6.34066E-31
-1.95863E-22  2.10460E-23 -6.82406E-25  9.56331E-27 -6.73829E-29
 2.50194E-31 -4.67451E-34  3.46233E-37
```

Figure 5-6. The first page of the automatic output from the POLYFIT instruction.

```
SQUARE ROOT OF VAR. COVARIANCE MATRIX DIAGONALS

4.83986E-04  4.43904E-05  1.30338E-06  1.72552E-08  1.17765E-10
4.29757E-13  7.96282E-16  5.88416E-19

GRAM DETERMINANT    2.3770645E-27

FOURIER COEFFICIENTS

3.21676E 00 -8.94292E-01  8.78644E-01  7.62594E-02  8.47799E-02
1.88659E-02 -1.13897E-03 -1.84967E-03  1.19327E 01

SQUARED FOURIER COEF.

1.03475E 01  7.99758E-01  7.72015E-01  5.81549E-03  7.18762E-03
3.55924E-04  1.29724E-06  3.42128E-06

SUM OF SQUARED RESIDUALS

1.58514E 00  7.85382E-01  1.33665E-02  7.55101E-03  3.63382E-04
7.45793E-06  6.16068E-06  2.73941E-06

RESIDUALS

2.19168E-01  1.56663E-01  2.07648E-02  1.58651E-02  3.53983E-03
5.16096E-04  4.77675E-04  3.24595E-04

A(I,J) MATRIX

 1.71499E-01
-2.41749E-01  1.42871E-03
 2.71022E-01 -5.25085E-03  1.43461E-05
-3.78187E-01  1.48641E-02 -1.02650E-04  1.81670E-07
 4.96924E-01 -2.72032E-02  3.21180E-04 -1.31542E-06  1.73209E-09
-7.00074E-01  5.17966E-02 -9.63444E-04  6.79741E-06 -2.00939E-08
 2.10348E-11
 9.16917E-01 -8.28566E-02  2.08623E-03 -2.15460E-05  1.05354E-07
-2.42808E-10  2.12344E-13
-1.29974E 00  1.39660E-01 -4.52841E-03  6.34616E-05 -4.47149E-07
 1.66028E-09 -3.10198E-12  2.29759E-15

GRAM FACTORS

5.83095E 00  6.99934E 02  6.97051E 04  5.50449E 06  5.77338E 08
4.75404E 10  4.70934E 12  4.35240E 14  1.30587E-03

VECTOR NORMS

5.83095E 00  1.20970E 03  3.65340E 05  1.20814E 08  4.15454E 10
1.45745E 13  5.17453E 15  1.85233E 18
```

Figure 5-7. The second page of the automatic output from the POLYFIT instruction. The deviations follow on the next page.

DEVIATIONS

-1.4875084E-04	-5.3592026E-05	2.6464462E-05	7.4729323E-05	1.1043251E-04
1.1128932E-04	7.2903931E-05	3.7401915E-06	-7.0504844E-05	-1.1721253E-04
-1.2475997E-04	-6.7420304E-05	5.3986907E-05	1.9038841E-04	8.9846551E-05
9.1902912E-06	-4.0780753E-05	-7.9721212E-06	2.8893352E-05	-6.3952059E-05
-1.2253970E-04	-4.4107437E-05	-6.4533204E-05	-4.0702522E-05	9.5386058E-05
2.8879568E-04	3.4866855E-04	-1.8158928E-04	-5.8098882E-04	-3.3542886E-04
4.8597530E-04	5.5939704E-04	-5.2925199E-04	4.7057867E-05	

STANDARD DEVIATIONS

4TH	5TH	6TH	7TH
3.534788E-03	4.789794E-04	4.357434E-04	2.561019E-04

COEFFICIENTS

1.019284E 00	1.006077E 00	1.005033E 00	1.007437E 00
-7.064067E-03	-6.086876E-03	-5.992505E-03	-6.250831E-03
3.200677E-05	1.383049E-05	1.145434E-05	1.983040E-05
-9.766715E-08	3.057235E-08	5.511257E-08	-6.227051E-08
1.468462E-10	-2.322450E-10	-3.522392E-10	4.748395E-10
0.	3.968405E-13	6.733909E-13	-2.397573E-12
0.	0.	-2.418529E-16	5.495789E-15
0.	0.	0.	-4.249775E-18

UNCERTAINTIES

2.624484E-03	4.887867E-04	5.977959E-04	4.839863E-04
1.112514E-04	2.903046E-05	4.473256E-05	4.439043E-05
1.192957E-06	4.889636E-07	1.012060E-06	1.303379E-06
4.693868E-09	3.317364E-09	9.861674E-09	1.725522E-08
6.122565E-12	9.660277E-12	4.674077E-11	1.177647E-10
0.	1.007522E-14	1.061983E-13	4.297567E-13
0.	0.	9.252756E-17	7.962823E-16
0.	0.	0.	5.884159E-19

Figure 5-8. Printout achieved by using the SPACE, NOTE, ABRIDGE, and APRINT following the automatic printout from POLYFIT.

PAGE 9 OMNITAB BF3 0 DEGREE PV VS D IN AMAGAT UNIT 10/16/64
Z VALUE AT 0 DEG 1 ATM IS 0.99409

P(ATM)	Z	PV(AMAG)	D (AMAG)
9.984160	0.937700	0.943279	10.584521
11.943140	0.924780	0.930282	12.838187
14.243230	0.909320	0.914730	15.570959
16.618960	0.893000	0.898313	18.500181
19.671390	0.871490	0.876675	22.438626
23.161990	0.846060	0.851094	27.214372
26.657740	0.819590	0.824467	32.333319
30.980290	0.785290	0.789962	39.217421
35.679380	0.745680	0.750117	47.565100
40.111880	0.705600	0.709798	56.511659
45.194850	0.655470	0.659370	68.542468
50.198740	0.600290	0.603862	83.129529
54.413980	0.547660	0.550919	98.769552
58.665070	0.486830	0.489727	119.791462
61.761150	0.436660	0.439258	140.603319
64.008679	0.399260	0.401636	159.370037
65.344110	0.378310	0.380561	171.704720
65.885130	0.370590	0.372795	176.732868
66.599979	0.361030	0.363178	183.381035
67.731689	0.347190	0.349256	193.931482
68.635119	0.337910	0.339921	201.915178
68.839379	0.336280	0.338281	203.497705
69.639079	0.329670	0.331632	209.989321
70.318230	0.324990	0.326924	215.090656
75.740230	0.306400	0.308223	245.731861
86.788980	0.309760	0.311603	278.524155
99.984670	0.331220	0.333191	300.082367
107.311139	0.345370	0.347425	308.875748
119.529889	0.370760	0.372966	320.484665
146.498480	0.429700	0.432257	338.915459
174.657169	0.492070	0.494998	352.844326
181.142719	0.506370	0.509383	355.612099
210.809969	0.571100	0.574498	366.946369
239.104879	0.632420	0.636183	375.842991

Figure 5-9. Results obtained with a PRINT instruction preceded by FIXED 6.

PAGE 10 OMNITAB BF3 0 DEGREE PV VS D IN AMAGAT UNIT 10/16/64
Z VALUE AT 0 DEG 1 ATM IS 0.99409

ABSCISSA - COLUMN 5
ORDINATES - COLUMN 4 (.),

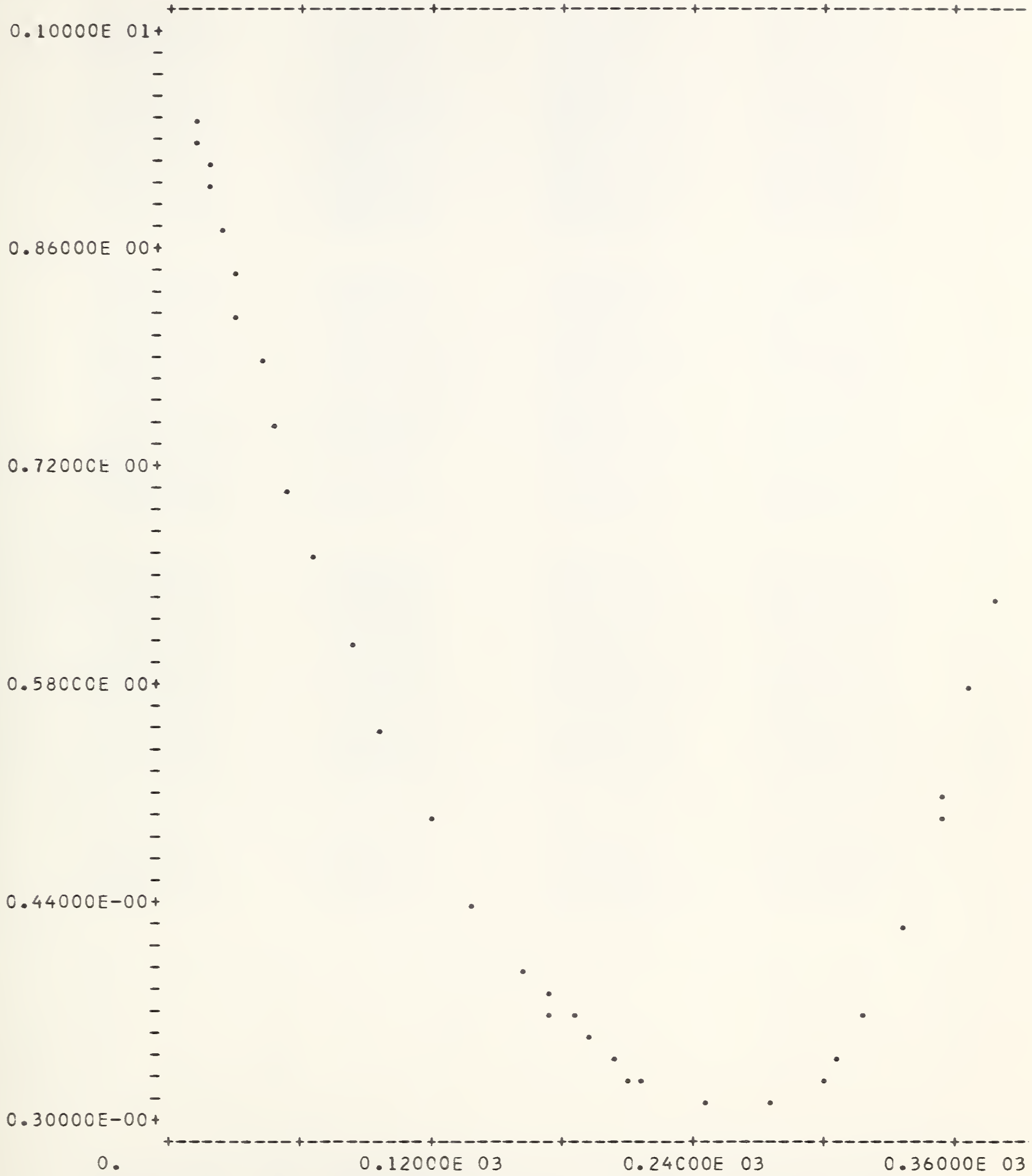


Figure 5-10. A typical OMNITAB plot except that the curve was squeezed to the left of the page by putting the right scale limit at 600.

Z VALUE AT 0 DEG 1 ATM IS 0.99409

DEVIATIONS

4TH	5TH	6TH	7TH
-0.004707	0.000046	0.000330	-0.000149
-0.003385	0.000012	0.000188	-0.000054
-0.001960	-0.000023	0.000042	0.000026
-0.000638	-0.000056	-0.000086	0.000075
0.000850	-0.000073	-0.000198	0.000110
0.002237	-0.000070	-0.000267	0.000111
0.003266	-0.000054	-0.000285	0.000073
0.004029	-0.000006	-0.000229	0.000004
0.004181	0.000074	-0.000085	-0.000071
0.003629	0.000155	0.000098	-0.000117
0.002113	0.000207	0.000299	-0.000125
-0.000280	0.000161	0.000395	-0.000067
-0.002760	0.000032	0.000335	0.000054
-0.004988	-0.000181	0.000074	0.000190
-0.005459	-0.000422	-0.000316	0.000090
-0.004174	-0.000383	-0.000422	0.000009
-0.002652	-0.000251	-0.000361	-0.000041
-0.001876	-0.000130	-0.000259	-0.000008
-0.000801	0.000028	-0.000118	0.000029
0.000801	0.000121	-0.000025	-0.000064
0.001986	0.000183	0.000057	-0.000123
0.002301	0.000282	0.000161	-0.000044
0.003199	0.000332	0.000242	-0.000065
0.003871	0.000393	0.000334	-0.000041
0.005882	0.000355	0.000545	0.000095
0.003431	-0.000128	0.000221	0.000289
-0.000058	-0.000279	-0.000033	0.000349
-0.002050	-0.000741	-0.000590	-0.000182
-0.004044	-0.000888	-0.000896	-0.000581
-0.004669	0.000017	-0.000237	-0.000335
-0.002304	0.001194	0.000893	0.000486
-0.001631	0.001271	0.000990	0.000559
0.000905	-0.000277	-0.000310	-0.000529
0.005756	-0.000895	-0.000482	0.000047

Figure 5-11. Results from accumulating the deviations. Note that the deviations are not random even for the 7th degree polynomial. This fact is confirmed in the following two figures.

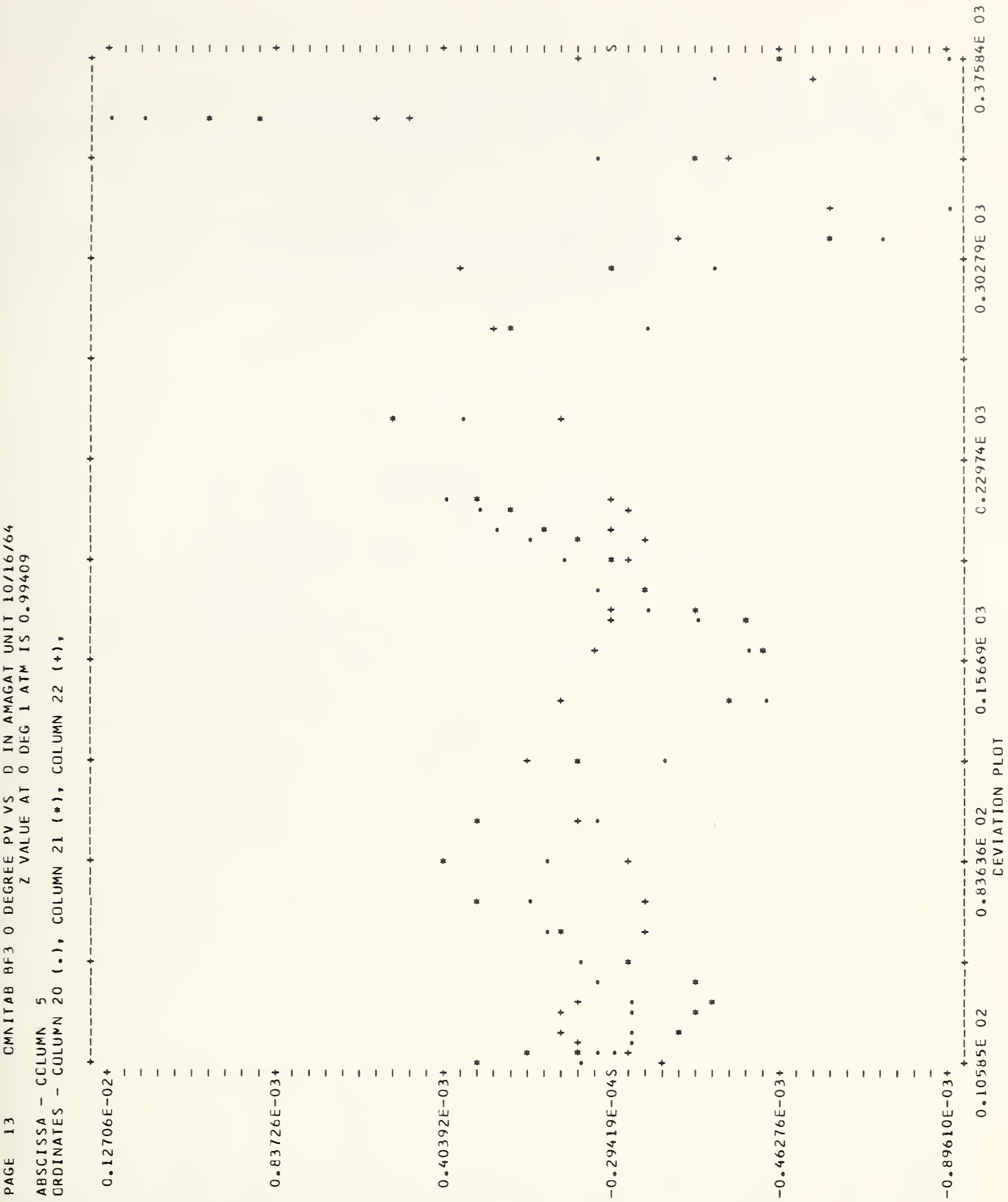


Figure 5-12. Comparison of deviations from the fit to a 5th, 6th, and 7th degree polynomial. Note the larger deviations at the higher pressures.

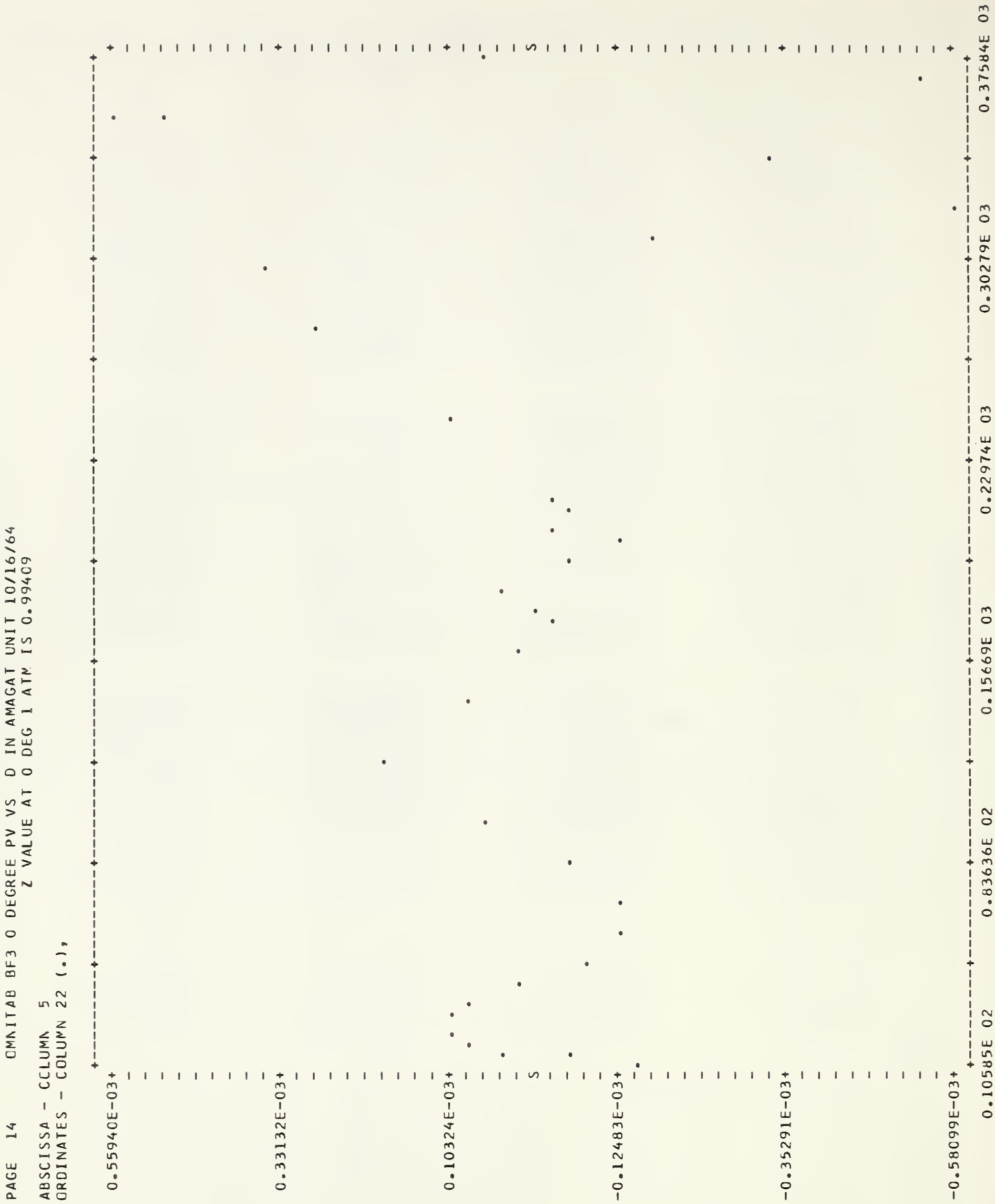


Figure 5—13. Plot of the deviations of the data from a 7th degree polynomial. Note how the plot dramatizes the remaining systematic deviations at the lower end and the large deviations at the higher end.

Problem 5 – 2

Generate a table of sin θ for 0° to 40° in 2° intervals and difference the table so as to determine a suitable integration formula.

The OMNITAB instructions are as follows:

```
OMNITAB    PROBLEM 5 - 2
GENERATE 0. 2. 40.  STORE IN 1
SIND OF COL 1 , STORE 2
DIFFERENCE COL 2 FOR X IN COL 1
FIXED 6
NOSUMMARY
PRINT 1 2 41 42 43 44 45 46
       STOP
```

The results of this calculation are given in the pages immediately following. The first is the automatic printout and the page following is the same results printed in fixed format. Note how much easier it is to assess the behavior of the differences in fixed format.

X	Y	1ST	2ND	3RD	4TH	5TH	6TH
0.	0.						
		3.4899496E-02					
2.0000000E 00	3.4899496E-02		-4.2520463E-05				
		3.4856975E-02		-4.2468309E-05			
4.0000000E 00	6.9756470E-02		-8.4988773E-05		1.1175871E-07		
		3.4771986E-02		-4.2356551E-05		2.2351742E-08	
6.0000000E 00	1.0452846E-01		-1.2734532E-04		1.3411045E-07		7.4505806E-08
		3.4644641E-02		-4.2222440E-05		9.6857548E-08	
8.0000000E 00	1.3917310E-01		-1.6956776E-04		2.3096800E-07		-8.9406967E-08
		3.4475073E-02		-4.1991472E-05		7.4505806E-09	
1.0000000E 01	1.7364817E-01		-2.1155924E-04		2.3841858E-07		8.1956387E-08
		3.4263514E-02		-4.1753054E-05		8.9406967E-08	
1.2000000E 01	2.0791169E-01		-2.5331229E-04		3.2782555E-07		-8.1956387E-08
		3.4010202E-02		-4.1425228E-05		7.4505806E-09	
1.4000000E 01	2.4192189E-01		-2.9473752E-04		3.3527613E-07		8.9406967E-08
		3.3715464E-02		-4.1089952E-05		9.6857548E-08	
1.6000000E 01	2.7563735E-01		-3.3582747E-04		4.3213367E-07		-8.9406967E-08
		3.3379637E-02		-4.0657818E-05		7.4505806E-09	
1.8000000E 01	3.0901699E-01		-3.7648529E-04		4.3958426E-07		7.4505806E-08
		3.3003151E-02		-4.0218234E-05		8.1956387E-08	
2.0000000E 01	3.4202014E-01		-4.1670352E-04		5.2154064E-07		-4.4703484E-08
		3.2586448E-02		-3.9696693E-05		3.7252903E-08	
2.2000000E 01	3.7460659E-01		-4.5640022E-04		5.5879354E-07		-1.4901161E-08
		3.2130048E-02		-3.9137900E-05		2.2351742E-08	
2.4000000E 01	4.0673663E-01		-4.9553812E-04		5.8114529E-07		6.7055225E-08
		3.1634510E-02		-3.8556755E-05		8.9406967E-08	
2.6000000E 01	4.3837114E-01		-5.3409487E-04		6.7055225E-07		-5.9604645E-08
		3.1100415E-02		-3.7886202E-05		2.9802322E-08	
2.8000000E 01	4.6947156E-01		-5.7198107E-04		7.0035458E-07		-7.4505806E-09
		3.0528434E-02		-3.7185848E-05		2.2351742E-08	
3.0000000E 01	4.9999999E-01		-6.0916692E-04		7.2270632E-07		5.9604645E-08
		2.9919267E-02		-3.6463141E-05		8.1956387E-08	
3.2000000E 01	5.2991925E-01		-6.4563006E-04		8.0466270E-07		-6.7055225E-08
		2.9273637E-02		-3.5658479E-05		1.4901161E-08	
3.4000000E 01	5.5919290E-01		-6.8128854E-04		8.1956387E-07		5.2154064E-08
		2.8592348E-02		-3.4838915E-05		6.7055225E-08	
3.6000000E 01	5.8778524E-01		-7.1612746E-04		8.1956387E-07		
		2.7876221E-02		-3.3952296E-05			
3.8000000E 01	6.1566146E-01		-7.5007975E-04				
		2.7126141E-02					
4.0000000E 01	6.4278761E-01						
ABSOLUTE VALUES OF DIFFERENCES							
		3.2139380E-02	4.0912393E-04	3.9308849E-05	5.0094197E-07	4.8428774E-08	6.3578287E-08

Figure 5-14. Automatic printout from the DIFFER instruction. The last line is the means of the absolute values.

PAGE 1 CMNITAB PROBLEM 5 - 2

COLUMN 1	COLUMN 2	COLUMN 41	COLUMN 42	COLUMN 43	COLUMN 44	COLUMN 45	COLUMN 46
0.	0.	0.034899	-0.000043	-0.000042	0.000000	0.000000	0.000000
2.000000	0.034899	0.034857	-0.000085	-0.000042	0.000000	0.000000	-0.000000
4.000000	0.069756	0.034772	-0.000127	-0.000042	0.000000	0.000000	0.000000
6.000000	0.104528	0.034645	-0.000170	-0.000042	0.000000	0.000000	-0.000000
8.000000	0.139173	0.034475	-0.000212	-0.000042	0.000000	0.000000	0.000000
10.000000	0.173648	0.034264	-0.000253	-0.000041	0.000000	0.000000	-0.000000
12.000000	0.207912	0.034010	-0.000295	-0.000041	0.000000	0.000000	0.000000
14.000000	0.241922	0.033715	-0.000336	-0.000041	0.000000	0.000000	-0.000000
16.000000	0.275637	0.033380	-0.000376	-0.000040	0.000001	0.000000	-0.000000
18.000000	0.309017	0.033003	-0.000417	-0.000040	0.000001	0.000000	0.000000
20.000000	0.342020	0.032586	-0.000456	-0.000039	0.000001	0.000000	-0.000000
22.000000	0.374607	0.032130	-0.000496	-0.000039	0.000001	0.000000	-0.000000
24.000000	0.406737	0.031635	-0.000534	-0.000038	0.000001	0.000000	0.000000
26.000000	0.438371	0.031100	-0.000572	-0.000037	0.000001	0.000000	-0.000000
28.000000	0.469472	0.030528	-0.000609	-0.000036	0.000001	0.000000	0.000000
30.000000	0.500000	0.029919	-0.000646	-0.000036	0.000001	0.000000	0.
32.000000	0.529919	0.029274	-0.000681	-0.000035	0.000001	0.	0.
34.000000	0.559193	0.028592	-0.000716	-0.000034	0.	0.	0.
36.000000	0.587785	0.027876	-0.000750	0.	0.	0.	0.
38.000000	0.615661	0.027126	0.	0.	0.	0.	0.
40.000000	0.642788	0.	0.	0.	0.	0.	0.

Figure 5-15. Results from printing the stored differences via the instruction FIXED 6 and PRINT 1, 2, 41, 42, 43, 44, 45, 46.

Problem 5 - 3

Generate a table of values of $y = x^5$ for even values of x between 0 and 50 and interpolate for integral values using a linear, cubic, quartic, and quintic approximation.

The OMNITAB instructions are as follows:

```
OMNITAB    PROBLEM 5 - 3
TITLE1  INTERPCLATION CHECK    7/8/64
NOSUMMARY
GENERATE 2.(2.)50.  IN COL 1
RAISE  1 5. 2
GENERATE 1.(1.)50. 3
RAISE 3 5. 4
INTERP 1 2 25 50 3 2 5
INTERP 1 2 25 50 3 4 6
INTERP 1 2 25 50 3 5 7
INTERP 1 2 25 50 3 6 8
HEAD CCL1/    X
HEAD CCL2/    Y
HEAD CCL3/   NEW X
HEAD CCL4/ CCMPUTED Y
HEAD CCL 5/ LINEAR
HEAD CCL 6/  CUBIC
HEAD CCL 7/ QUARTIC
HEAD CCL 8/ QUINTIC
PRINT 1 2 3 4 8
PRINT 3 4 5 6 7
       STCP
```

From the following page it should be obvious that except for end effects, the quintic approximation is exact as, indeed, should be. The next page shows the reduction in accuracy on going to lower order interpolation. Note should be taken that even with linear interpolation the answers for even values of x are quite correct. This is a characteristic of the program, namely that it does not interpolate for a point which is already tabulated.

PAGE 1 OMNITAB PROBLEM 5 - 3
INTERPOLATION CHECK 7/8/64

X	Y	NEW X	COMPUTED Y	QUINTIC
2.000000E 00	3.200000E 01	1.000000E 00	1.000000E 00	9.916992E-01
4.000000E 00	1.024000E 03	2.000000E 00	3.200000E 01	3.200000E 01
6.000000E 00	7.775999E 03	3.000000E 00	2.430000E 02	2.430013E 02
8.000000E 00	3.276799E 04	4.000000E 00	1.024000E 03	1.024000E 03
1.000000E 01	9.999997E 04	5.000000E 00	3.124999E 03	3.124999E 03
1.200000E 01	2.488319E 05	6.000000E 00	7.775999E 03	7.775999E 03
1.400000E 01	5.378237E 05	7.000000E 00	1.680699E 04	1.680700E 04
1.600000E 01	1.048576E 06	8.000000E 00	3.276799E 04	3.276799E 04
1.800000E 01	1.889567E 06	9.000000E 00	5.904899E 04	5.904898E 04
2.000000E 01	3.199999E 06	1.000000E 01	9.999997E 04	9.999997E 04
2.200000E 01	5.153630E 06	1.100000E 01	1.610510E 05	1.610510E 05
2.400000E 01	7.962621E 06	1.200000E 01	2.488319E 05	2.488319E 05
2.600000E 01	1.188137E 07	1.300000E 01	3.712929E 05	3.712928E 05
2.800000E 01	1.721036E 07	1.400000E 01	5.378237E 05	5.378237E 05
3.000000E 01	2.429999E 07	1.500000E 01	7.593748E 05	7.593747E 05
3.200000E 01	3.355442E 07	1.600000E 01	1.048576E 06	1.048576E 06
3.400000E 01	4.543540E 07	1.700000E 01	1.419856E 06	1.419857E 06
3.600000E 01	6.046615E 07	1.800000E 01	1.889567E 06	1.889567E 06
3.800000E 01	7.923513E 07	1.900000E 01	2.476098E 06	2.476098E 06
4.000000E 01	1.023999E 08	2.000000E 01	3.199999E 06	3.199999E 06
4.200000E 01	1.306912E 08	2.100000E 01	4.084099E 06	4.084099E 06
4.400000E 01	1.649162E 08	2.200000E 01	5.153630E 06	5.153630E 06
4.600000E 01	2.059629E 08	2.300000E 01	6.436341E 06	6.436341E 06
4.800000E 01	2.548039E 08	2.400000E 01	7.962621E 06	7.962621E 06
5.000000E 01	3.124999E 08	2.500000E 01	9.765621E 06	9.765621E 06
0.	0.	2.600000E 01	1.188137E 07	1.188137E 07
0.	0.	2.700000E 01	1.434890E 07	1.434890E 07
0.	0.	2.800000E 01	1.721036E 07	1.721036E 07
0.	0.	2.900000E 01	2.051114E 07	2.051114E 07
0.	0.	3.000000E 01	2.429999E 07	2.429999E 07
0.	0.	3.100000E 01	2.862914E 07	2.862914E 07
0.	0.	3.200000E 01	3.355442E 07	3.355442E 07
0.	0.	3.300000E 01	3.913537E 07	3.913537E 07
0.	0.	3.400000E 01	4.543540E 07	4.543540E 07
0.	0.	3.500000E 01	5.252185E 07	5.252185E 07
0.	0.	3.600000E 01	6.046615E 07	6.046615E 07
0.	0.	3.700000E 01	6.934393E 07	6.934392E 07
0.	0.	3.800000E 01	7.923513E 07	7.923513E 07
0.	0.	3.900000E 01	9.022417E 07	9.022415E 07
0.	0.	4.000000E 01	1.023999E 08	1.023999E 08
0.	0.	4.100000E 01	1.158561E 08	1.158561E 08
0.	0.	4.200000E 01	1.306912E 08	1.306912E 08
0.	0.	4.300000E 01	1.470084E 08	1.470084E 08
0.	0.	4.400000E 01	1.649162E 08	1.649162E 08
0.	0.	4.500000E 01	1.845280E 08	1.845280E 08
0.	0.	4.600000E 01	2.059629E 08	2.059629E 08
0.	0.	4.700000E 01	2.293449E 08	2.293449E 08
0.	0.	4.800000E 01	2.548039E 08	2.548039E 08
0.	0.	4.900000E 01	2.824752E 08	2.824752E 08
0.	0.	5.000000E 01	3.124999E 08	3.124999E 08

NEW X	COMPUTED Y	LINEAR	CUBIC	QUARTIC
1.000000E 00	1.000000E 00	-4.639999E 02	-2.203999E 03	9.459993E 02
2.000000E 00	3.200000E 01	3.200000E 01	3.200000E 01	3.200000E 01
3.000000E 00	2.430000E 02	5.279999E 02	5.879998E 02	1.380001E 02
4.000000E 00	1.024000E 03	1.024000E 03	1.024000E 03	1.024000E 03
5.000000E 00	3.124999E 03	4.399999E 03	2.900000E 03	3.170000E 03
6.000000E 00	7.775999E 03	7.775999E 03	7.775999E 03	7.775999E 03
7.000000E 00	1.680699E 04	2.027200E 04	1.649200E 04	1.685200E 04
8.000000E 00	3.276799E 04	3.276799E 04	3.276799E 04	3.276799E 04
9.000000E 00	5.904899E 04	6.638398E 04	5.864398E 04	5.909398E 04
1.000000E 01	9.999997E 04	9.999997E 04	9.999997E 04	9.999997E 04
1.100000E 01	1.610510E 05	1.744160E 05	1.605560E 05	1.610960E 05
1.200000E 01	2.488319E 05	2.488319E 05	2.488319E 05	2.488319E 05
1.300000E 01	3.712929E 05	3.933278E 05	3.707078E 05	3.713378E 05
1.400000E 01	5.378237E 05	5.378237E 05	5.378237E 05	5.378237E 05
1.500000E 01	7.593748E 05	7.931997E 05	7.586997E 05	7.594197E 05
1.600000E 01	1.048576E 06	1.048576E 06	1.048576E 06	1.048576E 06
1.700000E 01	1.419856E 06	1.469072E 06	1.419092E 06	1.419902E 06
1.800000E 01	1.889567E 06	1.889567E 06	1.889567E 06	1.889567E 06
1.900000E 01	2.476098E 06	2.544783E 06	2.475243E 06	2.476143E 06
2.000000E 01	3.199999E 06	3.199999E 06	3.199999E 06	3.199999E 06
2.100000E 01	4.084099E 06	4.176815E 06	4.083155E 06	4.084145E 06
2.200000E 01	5.153630E 06	5.153630E 06	5.153630E 06	5.153630E 06
2.300000E 01	6.436341E 06	6.558126E 06	6.435306E 06	6.436386E 06
2.400000E 01	7.962621E 06	7.962621E 06	7.962621E 06	7.962621E 06
2.500000E 01	9.765621E 06	9.921996E 06	9.764496E 06	9.765666E 06
2.600000E 01	1.188137E 07	1.188137E 07	1.188137E 07	1.188137E 07
2.700000E 01	1.434890E 07	1.454586E 07	1.434768E 07	1.434894E 07
2.800000E 01	1.721036E 07	1.721036E 07	1.721036E 07	1.721036E 07
2.900000E 01	2.051114E 07	2.075517E 07	2.050983E 07	2.051118E 07
3.000000E 01	2.429999E 07	2.429999E 07	2.429999E 07	2.429999E 07
3.100000E 01	2.862914E 07	2.892720E 07	2.862774E 07	2.862918E 07
3.200000E 01	3.355442E 07	3.355442E 07	3.355442E 07	3.355442E 07
3.300000E 01	3.913537E 07	3.949491E 07	3.913389E 07	3.913542E 07
3.400000E 01	4.543540E 07	4.543540E 07	4.543540E 07	4.543540E 07
3.500000E 01	5.252185E 07	5.295077E 07	5.252027E 07	5.252189E 07
3.600000E 01	6.046615E 07	6.046615E 07	6.046615E 07	6.046615E 07
3.700000E 01	6.934393E 07	6.985064E 07	6.934226E 07	6.934397E 07
3.800000E 01	7.923513E 07	7.923513E 07	7.923513E 07	7.923513E 07
3.900000E 01	9.022417E 07	9.081754E 07	9.022240E 07	9.022420E 07
4.000000E 01	1.023999E 08	1.023999E 08	1.023999E 08	1.023999E 08
4.100000E 01	1.158561E 08	1.165456E 08	1.158543E 08	1.158562E 08
4.200000E 01	1.306912E 08	1.306912E 08	1.306912E 08	1.306912E 08
4.300000E 01	1.470084E 08	1.478037E 08	1.470064E 08	1.470084E 08
4.400000E 01	1.649162E 08	1.649162E 08	1.649162E 08	1.649162E 08
4.500000E 01	1.845280E 08	1.854395E 08	1.845260E 08	1.845281E 08
4.600000E 01	2.059629E 08	2.059629E 08	2.059629E 08	2.059629E 08
4.700000E 01	2.293449E 08	2.303834E 08	2.293428E 08	2.293449E 08
4.800000E 01	2.548039E 08	2.548039E 08	2.548039E 08	2.548039E 08
4.900000E 01	2.824752E 08	2.836519E 08	2.824787E 08	2.824752E 08
5.000000E 01	3.124999E 08	3.124999E 08	3.124999E 08	3.124999E 08

Problem 5 — 4

Compute the derivatives of $y = x^4$ for 50 values of x using a 5 point formula.

The OMNITAB instructions are as follows:

```
OMNITAB       PROBLEM 5 - 4
NOSUMMARY
GENERATE 1.(1.)10. IN 1
RAISE 1 TO 4. POWER STORE IN 3
GENERATE .2(.2)10. IN COL 2
RAISE 2  TO 3. POWER ,MULT BY 4. ADD TO 4
DERIVATIVE OF 1 3 10 BY 50 2 5 5
HEAD 1/        X
HEAD 3/   X TO 4
HEAD 2/   NEW  X
HEAD 4/   4X TO 3
PRINT 1 3 2 4 5
       STOP
```

The following page shows the results. The column headed 4X TO 3 represents the correct value and column 5 contains the computed values.

X	X TO 4	NEW X	4X TO 3	COLUMN 5
1.000000E 00	1.000000E 00	2.000000E-01	3.200000E-02	3.212738E-02
2.000000E 00	1.600000E 01	4.000000E-01	2.560000E-01	2.560959E-01
3.000000E 00	8.100000E 01	6.000000E-01	8.640000E-01	8.640823E-01
4.000000E 00	2.560000E 02	8.000000E-01	2.048000E 00	2.048038E 00
5.000000E 00	6.249999E 02	10.000000E-01	4.000000E 00	4.000029E 00
6.000000E 00	1.296000E 03	1.200000E 00	6.912000E 00	6.912006E 00
7.000000E 00	2.400999E 03	1.400000E 00	1.097600E 01	1.097600E 01
8.000000E 00	4.096000E 03	1.600000E 00	1.638400E 01	1.638399E 01
9.000000E 00	6.560999E 03	1.800000E 00	2.332800E 01	2.332799E 01
1.000000E 01	9.999997E 03	2.000000E 00	3.199999E 01	3.199999E 01
0.	0.	2.200000E 00	4.259200E 01	4.259199E 01
0.	0.	2.400000E 00	5.529599E 01	5.529599E 01
0.	0.	2.600000E 00	7.030399E 01	7.030399E 01
0.	0.	2.800000E 00	8.780799E 01	8.780799E 01
0.	0.	3.000000E 00	1.080000E 02	1.080000E 02
0.	0.	3.200000E 00	1.310720E 02	1.310720E 02
0.	0.	3.400000E 00	1.572160E 02	1.572160E 02
0.	0.	3.600000E 00	1.866240E 02	1.866240E 02
0.	0.	3.800000E 00	2.194880E 02	2.194879E 02
0.	0.	4.000000E 00	2.560000E 02	2.559999E 02
0.	0.	4.200000E 00	2.963519E 02	2.963518E 02
0.	0.	4.400000E 00	3.407360E 02	3.407358E 02
0.	0.	4.600000E 00	3.893440E 02	3.893439E 02
0.	0.	4.800000E 00	4.423679E 02	4.423679E 02
0.	0.	5.000000E 00	4.999999E 02	5.000000E 02
0.	0.	5.200000E 00	5.624319E 02	5.624323E 02
0.	0.	5.400000E 00	6.298559E 02	6.298561E 02
0.	0.	5.600000E 00	7.024639E 02	7.024639E 02
0.	0.	5.800000E 00	7.804479E 02	7.804478E 02
0.	0.	6.000000E 00	8.639999E 02	8.639997E 02
0.	0.	6.200000E 00	9.533119E 02	9.533111E 02
0.	0.	6.400000E 00	1.048576E 03	1.048575E 03
0.	0.	6.600000E 00	1.149984E 03	1.149983E 03
0.	0.	6.800000E 00	1.257728E 03	1.257728E 03
0.	0.	7.000000E 00	1.372000E 03	1.372000E 03
0.	0.	7.200000E 00	1.492992E 03	1.492992E 03
0.	0.	7.400000E 00	1.620896E 03	1.620896E 03
0.	0.	7.600000E 00	1.755904E 03	1.755904E 03
0.	0.	7.800000E 00	1.898208E 03	1.898208E 03
0.	0.	8.000000E 00	2.047999E 03	2.048001E 03
0.	0.	8.200000E 00	2.205472E 03	2.205472E 03
0.	0.	8.400000E 00	2.370816E 03	2.370816E 03
0.	0.	8.600000E 00	2.544224E 03	2.544224E 03
0.	0.	8.800000E 00	2.725887E 03	2.725887E 03
0.	0.	9.000000E 00	2.916000E 03	2.916000E 03
0.	0.	9.200000E 00	3.114751E 03	3.114750E 03
0.	0.	9.400000E 00	3.322335E 03	3.322334E 03
0.	0.	9.600000E 00	3.538943E 03	3.538942E 03
0.	0.	9.800000E 00	3.764767E 03	3.764765E 03
0.	0.	1.000000E 01	3.999999E 03	3.999996E 03

Problem 5 - 5

Prepare a check on the ability of the program to compute the integral of $y = 5x^4$ for tabulated values of $x = 1.(1.)50.$

The OMNITAB instructions are as follows:

```
OMNITAB     PRCBLEM 5 - 5
NOSUMMARY
GENERATE 1. 1. 50. 1
RAISE 1 4. 5. 2
RAISE 1 5. 3
HEAD CCL 1/     X
HEAD CCL 2/ 5X TO 4
HEAD CCL 3/ X TO 5
LAGINT 2 3 1. 4
LAGINT 2 6 1. 5
PRINT 1 2 3 4
PRINT 1 2 3 4 5
       STOP
```

Note the results on the following page. These show the quadratic approximation to be inaccurate and the quintic approximation to be exact.

X	5X TO 4	X TO 5	COLUMN 4	COLUMN 5
1.000000E 00	5.000000E 00	1.000000E 00	0.	0.
2.000000E 00	8.000000E 01	3.200000E 01	2.166667E 01	3.100006E 01
3.000000E 00	4.050000E 02	2.430000E 02	2.433333E 02	2.420000E 02
4.000000E 00	1.280000E 03	1.024000E 03	1.040000E 03	1.023000E 03
5.000000E 00	3.124999E 03	3.124999E 03	3.161666E 03	3.124000E 03
6.000000E 00	6.480000E 03	7.775999E 03	7.838332E 03	7.774999E 03
7.000000E 00	1.200500E 04	1.680699E 04	1.690000E 04	1.680600E 04
8.000000E 00	2.048000E 04	3.276799E 04	3.289666E 04	3.276699E 04
9.000000E 00	3.280500E 04	5.904899E 04	5.921832E 04	5.904799E 04
1.000000E 01	4.999999E 04	9.999997E 04	1.002150E 05	9.999898E 04
1.100000E 01	7.320498E 04	1.610510E 05	1.613166E 05	1.610500E 05
1.200000E 01	1.036800E 05	2.488319E 05	2.491533E 05	2.488309E 05
1.300000E 01	1.428050E 05	3.712929E 05	3.716749E 05	3.712919E 05
1.400000E 01	1.920799E 05	5.378237E 05	5.382715E 05	5.378229E 05
1.500000E 01	2.531250E 05	7.593748E 05	7.598931E 05	7.593738E 05
1.600000E 01	3.276800E 05	1.048576E 06	1.049170E 06	1.048575E 06
1.700000E 01	4.176049E 05	1.419856E 06	1.420531E 06	1.419856E 06
1.800000E 01	5.248799E 05	1.889567E 06	1.890328E 06	1.889566E 06
1.900000E 01	6.516049E 05	2.476098E 06	2.476949E 06	2.476097E 06
2.000000E 01	7.999997E 05	3.199999E 06	3.200946E 06	3.199998E 06
2.100000E 01	9.724046E 05	4.084099E 06	4.085147E 06	4.084099E 06
2.200000E 01	1.171280E 06	5.153630E 06	5.154783E 06	5.153629E 06
2.300000E 01	1.399205E 06	6.436341E 06	6.437605E 06	6.436340E 06
2.400000E 01	1.658880E 06	7.962621E 06	7.964001E 06	7.962621E 06
2.500000E 01	1.953125E 06	9.765621E 06	9.767122E 06	9.765621E 06
2.600000E 01	2.284879E 06	1.188137E 07	1.188300E 07	1.188137E 07
2.700000E 01	2.657204E 06	1.434890E 07	1.435066E 07	1.434890E 07
2.800000E 01	3.073279E 06	1.721036E 07	1.721226E 07	1.721036E 07
2.900000E 01	3.536404E 06	2.051114E 07	2.051318E 07	2.051114E 07
3.000000E 01	4.049999E 06	2.429999E 07	2.430217E 07	2.429999E 07
3.100000E 01	4.617604E 06	2.862914E 07	2.863147E 07	2.862914E 07
3.200000E 01	5.242879E 06	3.355442E 07	3.355691E 07	3.355442E 07
3.300000E 01	5.929603E 06	3.913537E 07	3.913803E 07	3.913538E 07
3.400000E 01	6.681678E 06	4.543540E 07	4.543822E 07	4.543541E 07
3.500000E 01	7.503123E 06	5.252185E 07	5.252484E 07	5.252186E 07
3.600000E 01	8.398077E 06	6.046615E 07	6.046932E 07	6.046616E 07
3.700000E 01	9.370803E 06	6.934393E 07	6.934727E 07	6.934393E 07
3.800000E 01	1.042568E 07	7.923513E 07	7.923867E 07	7.923514E 07
3.900000E 01	1.156720E 07	9.022417E 07	9.022789E 07	9.022417E 07
4.000000E 01	1.280000E 07	1.023999E 08	1.024039E 08	1.024000E 08
4.100000E 01	1.412880E 07	1.158561E 08	1.158603E 08	1.158562E 08
4.200000E 01	1.555847E 07	1.306912E 08	1.306955E 08	1.306912E 08
4.300000E 01	1.709400E 07	1.470084E 08	1.470129E 08	1.470084E 08
4.400000E 01	1.874048E 07	1.649162E 08	1.649209E 08	1.649162E 08
4.500000E 01	2.050312E 07	1.845280E 08	1.845330E 08	1.845281E 08
4.600000E 01	2.238728E 07	2.059629E 08	2.059681E 08	2.059629E 08
4.700000E 01	2.439840E 07	2.293449E 08	2.293503E 08	2.293449E 08
4.800000E 01	2.654207E 07	2.548039E 08	2.548095E 08	2.548039E 08
4.900000E 01	2.882400E 07	2.824752E 08	2.824811E 08	2.824752E 08
5.000000E 01	3.124999E 07	3.124999E 08	3.125060E 08	3.124999E 08

Problem 5 — 6

Find the maxima and minima, if any, of the K given by the following equation.

$$K = -.008 + .00004462T + 5.784 \cdot 10^{-8} T^2 + \frac{2.441 \cdot 10^{11}}{(T^{4.184} \cdot B^2)}$$

$$+ \frac{(-.173 + .001229T - 7.493 \cdot 10^{-7} T^2)}{B}$$

* * * * * * * * * *

```
CMNITAB       PROBLEM 5 - 6
NOSUMMARY
TITLE1  MAXIMA AND MINIMA IN A THERMAL CONDUCTIVITY OF STEAM CORRELATION
READ  TEMP IN 1, B IN 3
  0  0.9904
 50  1.0033
100  1.0336
150  1.0782
200  1.1391
250  1.2251
300  1.360
350  1.665
375  7.68
400  9.95
425 11.47
450 12.71
475 13.79
500 14.78
550 16.55
600 18.16
650 19.67
700 21.11
750 22.50
800 23.85
DIVIDE 1. 3 3
ADD     273.15 TO 1 STORE IN 2
SUBTRACT  .008 FROM 4 STORE IN 4
MULT  4.462E-5 BY 2, BY 1. AND ADD TO 4
MULT     5.784E-8 BY 2, BY 2, AND ADD TO 4
SUBT   .173 FROM 5 STORE 5
MULT    1.229E-3 BY 1, BY 1. AND ADD TO 5
MULT  -7.493E-7 BY 1, BY 1, AND ADD TO 5
MULT   3 BY 5, BY 1. AND ADD TO 4
MULT   2.441E11 BY 3, BY 3 AND ADD TO 6
ADD    4.184 TO 7, STORE IN 7
RAISE  1 TO 7, STORE IN 8
DIV   6 BY 8, MULT BY 1., AND ADD TO 4
TITLE2                           PRESSURE 200 BAR
HEAD COL 1/ T- THETA
HEAD COL 4/    K
PRINT 1 4
PLOT  4 AGAINST 1
MAXMIN X IN 1,Y IN 4, STORE XMAX IN 9,YMAX IN 10,X MIN IN 11,Y MIN IN 12
HEAD 9/ T - THETA
HEAD 10/MAXIMA OF 5
HEAD COL 11/ T - THETA
HEAD COL 12/MINIMA OF K
PRINT 9 10 11 12
      STOP
```

* * * RESULTS * * *

PAGE 1 OMNITAB PROBLEM 5 - 6

MAXIMA AND MINIMA IN A THERMAL CONDUCTIVITY OF STEAM CORRELATION

PRESSURE 200 BAR

T- THETA	K
0.	-1.661734E-01
5.000000E 01	1.888920E 04
1.000000E 02	9.791420E 02
1.500000E 02	1.649856E 02
2.000000E 02	4.441788E 01
2.500000E 02	1.517707E 01
3.000000E 02	5.835350E 00
3.500000E 02	2.138493E 00
3.750000E 02	1.393058E-01
4.000000E 02	1.001979E-01
4.250000E 02	8.867436E-02
4.500000E 02	8.445160E-02
4.750000E 02	8.339784E-02
5.000000E 02	8.396775E-02
5.500000E 02	8.766397E-02
6.000000E 02	9.304213E-02
6.500000E 02	9.927883E-02
7.000000E 02	1.060465E-01
7.500000E 02	1.131980E-01
8.000000E 02	1.206653E-01

PAGE 3 OMNITAB PROBLEM 5 - 6

MAXIMUM AND MINIMUM OF COLUMN 4

X= 5.066530E 01 Y= 1.889246E 04 IS A MAXIMUM

X= 4.787250E 02 Y= 8.337982E-02 IS A MINIMUM

PAGE 4 OMNITAB PROBLEM 5 - 6

MAXIMA AND MINIMA IN A THERMAL CONDUCTIVITY OF STEAM CORRELATION

PRESSURE 200 BAR

T - THETA	MAXIMA OF 5	T - THETA	MINIMA OF K
5.066530E 01	1.889246E 04	4.787250E 02	8.337982E-02
0.	0.	0.	0.
0.	0.	0.	0.
0.	0.	0.	0.
0.	0.	0.	0.
0.	0.	0.	0.
0.	0.	0.	0.
0.	0.	0.	0.
0.	0.	0.	0.
0.	0.	0.	0.

Problem 5 − 7

Perform a harmonic analysis on the following values spaced evenly between 0° and 360°: 149, 137, 128, 126, 128, 135, 159, 178, 189, 191, 189, 187, 178, 170, 177, 183, 181, 179, 179, 185, 182, 176, 166, 160.

```
OMNITAB   PROBLEM  5 - 7
TITLE1  PROBLEM FROM SCARBOROUGH PAGE 512  4TH EDITION
TITLE2          OR PAGE 572  5TH EDITION
NOSUMMARY
SET IN COL 1
149 137 128 126 128 135 159 178 189      191 189 187 178
170 177 183 181 179 179 185 182 176 166 160
HEAD COL 1/  Y-DATA
HEAD COL 2/ X COEFF
HARMON 1 24 2
FIXED 6
PRINT 1 2
     STOP
```

* * * RESULTS * * *

```
PAGE   1      OMNITAB   PROBLEM  5 - 7
  PROBLEM FROM SCARBOROUGH PAGE 512  4TH EDITION
       OR PAGE 572  5TH EDITION
    Y-DATA        X COEFF

    149.000000     167.166666
    137.000000     -19.983130
    128.000000      -3.409897
    126.000000       5.470667
    128.000000      -1.291681
    135.000000       0.249746
    159.000000       0.749984
    178.000000       0.309207
    189.000000       0.458333
    191.000000      -0.304022

    189.000000      -0.090113
    187.000000      -0.242531
    178.000000      -0.083333
    170.000000     -12.779235
    177.000000     -16.624566
    183.000000      -0.323218
    181.000000       1.515551
    179.000000       1.461703
    179.000000      -2.583315
    185.000000       0.322179

    182.000000      -0.216497
    176.000000       0.676798
    166.000000      -0.458756
    160.000000      -0.639697
```

The order of the coefficients is a_0, a_1,... a_n, b_1, b_2... b_{n-1}

6. THE REPEAT MODE

In the ordinary operating mode discussed in the foregoing sections, each command (a card containing a valid vocabulary word and a series of numbers) is executed as it is read and then dumped on an output tape to be printed later with the answers. This is called a single-pass system.

In order to be able to repeat operations more efficiently it is necessary to have the instructions stored in the machine just as the program is. Such a scheme can be achieved in a number of ways with varying cost in storage space, execution time, or tape operations. The method adopted in OMNITAB to permit a repeat mode has been designed, we believe, to minimize all of these.

The instructions which are not to be repeated are handled in the usual way, but those instructions following the word BEGIN and ending just before the word FINISH are stored in the core and are available for subsequent use. A maximum of 150 instructions can be so stored. These are automatically numbered by the program.

The stored instructions can be modified in a number of ways, to be discussed below, and used over and over again with or without intervening computations in the ordinary mode. The stored instructions can be replaced by another set of instructions preceded by the word BEGIN and terminated by the word FINISH. It should be emphasized that the stored instructions are not executed at all unless and until the REPEAT instruction is given. Thus the instruction

REPEAT 1, 10, 5 TIMES

carries out the first 10 stored instructions, 5 times. This might perhaps be followed by the instruction

REPEAT INSTRUCTION 11, 15 1 TIME

or even

REPEAT 5, 15 5 TIMES

if such an operation were meaningful. It will be obvious that the operations are performed in a consecutive order from the top down. If it were desired to execute instructions 1, 3, and 5 one time each (there being good reason to separate them so), this could be achieved somewhat less efficiently by the following instructions:

```
REPEAT INST  1,  1,  1 TIME
REPEAT   3,  3,  1
REPEAT   5,  5,  1
```

The arrangement of the program precludes the use of FORMAT, TITLE, NOTE, READ, SET, or HEAD instructions inside the stored set. Their presence will either be ignored or result in an error diagnostic. Where it makes sense to do so, PRINT and PUNCH instructions may be used in the stored set.

Where identical operations are to be performed on a number of independent sets of data, each set is preceded by an appropriate READ instruction, and then followed by an appropriate REPEAT instruction. This assumes, of course, that the computation instructions had previously been entered between the BEGIN and FINISH instructions.

A repeat mode would have limited utility if there were not also a facile way of modifying one or more of the instructions after each pass. This feature is provided by the instructions:

```
INDEX INSTRUCTIONS  ,, ,, ,, ,, BY ,,
INCREMENT INSTRUCTION ,, BY  $$, $$, $$ . . . .
```

The INDEX instruction changes only the last number on each of the designated instructions by the designated value. The INCREMENT instruction modifies only a single instruction but has provision for incrementing any or all of the numbers in the instruction. The INCREMENT instruction must contain as many increment numbers as the instruction which it modifies. Whether they carry a decimal point or not depends upon whether or not the numbers which they modify do so. They may be incremented either positively or negatively. The instruction numbers are assigned automatically by the program starting with number 0 for the BEGIN and then sequentially through the set. Neither the BEGIN nor FINISH instruction are considered part of the stored set.

Before discussing the other instructions and the solution of problems via the repeat mode, we summarize the operating characteristics and logic of the repeat mode.

a. With normal input, OMNITAB executes each command as it is read. On reading the word BEGIN, the program goes out of the execute mode and stores the instructions following BEGIN up to the word FINISH.

b. On encountering the word FINISH the program goes back into the execute mode and is ready to handle subsequent instructions in the usual way.

c. The commands following the FINISH instruction are executed but not saved until another BEGIN instruction is encountered at which time the new set replaces the old one.

d. As long as the machine is in the execute mode it will carry out an INDEX or an INCREMENT instruction in the usual fashion so that the instructions being saved in core can be modified for subsequent re-entry. Even the INDEX and INCREMENT instructions can be incremented.

e. Entry and re-entry into previously stored instructions is achieved by the instruction REPEAT. This word halts the reading of new commands until the REPEAT instruction is satisfied.

f. When a stored set of instructions is no longer required, it can be overwritten by inserting another BEGIN instruction followed by suitable commands, terminated by FINISH.

g. Arguments may be generated within the stored instructions. The READ and other input instructions cannot be accommodated within the repeat mode. They can easily be inserted either prior to entering the repeat mode or between repeats.

h. It should be noted that the INDEX and INCREMENT instructions can be used both in the stored instructions and in the ordinary instructions.

i. Instructions to be stored cannot contain more than 10 numbers each.

j. Neither FORMAT nor HEAD instructions can be used inside the stored instructions. They can however be used in the normal way.

k. Experienced users of the repeat mode find it advantageous, when using a long list of stored instructions, to intersperse a number of innocuous instructions to facilitate later changes without need for renumbering the REPEAT instructions. The cheapest instruction for this purpose is SUMMARIZE or NOSUMMARY.

6.1. Repeat Mode Commands

1. BEGIN STORING INSTRUCTIONS
   a. puts the program into the nonexecute mode under which it reads and stores the subsequent instructions but does not carry them out.
   b. under this mode the program stores up to 150 commands and numbers them so that they may be referenced in subsequent REPEAT instructions.
   c. once the word BEGIN is encountered, the instructions which follow it are stored until the word FINISH is encountered.

2. INDEX INSTRUCTIONS ,, ,, ,, ,, ... BY ,,
   a. provides for incrementing the storage location (the last number on the card) of the designated instructions by a constant value.

3. INCREMENT INSTRUCTION ,, BY $$,$$,$$, ...
   a. provides for incrementing the arguments of the designated instruction as indicated.
   b. these must be a one to one correspondence between the numbers on the original instruction and on the INCREMENT instruction.
   c. where the number on the original instruction carried a decimal, the corresponding number on the INCREMENT instruction must also carry a decimal point.
   d. a zero must be used where an argument is not to be incremented.

4. COMPARE COL ++ AND ++ TO A TOLERANCE OF $$
   a. this instruction is used only in the repeat mode and must be one of the stored instructions.
   b. when the comparison is satisfied for each value in the two columns, the program exits from the repeat mode and prints out how many cycles of the REPEAT instruction have been executed.
   c. the tolerance is defined as $|(C_1-C_2)/C_2|$ where the C's are the values in the designated columns.
   d. if the comparison is not satisfied after carrying out the designated REPEAT cycle, a diagnostic statement to that effect is printed and the

designated columns as well as the tolerance are printed out for diagnostic use.

5. ISOLATE X IN ++, FOR Y IN COL ++ = ** STORE IN ++

or

ISOLATE X IN ++ FOR COL ++ = **, USE ,, POINTS STORE IN ++

a. locates the values of X between which the desired value (**) lies, and inserts between them either 3 points or the designated number of points uniformly distributed, and stores all these values in the last designated column.

b. used in the application of the principle of false position to locating the zeros of functions given explicitly in terms of one independent variable which must be monotonic in the first named column.

c. also stores the average of the bracketing values in column 45, (see table B).

d. three points are inserted when no points are specified.

e. at the end of this operation the program increases or decreases NRMAX to coincide with the exact length of the results generated in the storage column indicated in the instruction.

f. when more points are inserted than can be accommodated by the length of the work sheet, the excess values are stored in column 46. If overflow is expected, the extra points should be moved to a blank column and operated on as was the original column of X's.

g. if the value (**) is outside of the range of the dependent variable in the second named column, an appropriate diagnostic statement is printed.

h. if the desired value is located exactly, the corresponding X value is transferred to the storage column and to column 45.

6. ISETUP X IN ++, Y IN ++, DESIRED Y IN ++, STORE IN ++

a. locates the values of X between which the desired values of Y lie, and inserts between them three points, and stores all five values of X in the designated column.

b. stores the average of the bracketing X's in column 45, the average of the Y values corresponding to the bracketing X's in column 44, and each of the successfully bracketing Y's in column 46.(See the solution to Problem 6-4.)

c. stores the necessary information for the first 20 points located from the desired list and gives a notice if additional values were located.

d. computation is stopped if no points are found and if the columns specified do not contain enough values to allow iteration.

e. if the desired Y occurs more than once, only the first occurrence will be bracketed.
f. this instruction operates like the ISOLATE instruction, except that it isolates the arguments which bracket each of a list of different values.
g. if the desired value is located exactly, the corresponding X value is transferred to the storage column and columns 44, 45, and 46 are treated as usual.
h. this instruction generates blocks of numbers for each of the desired functional values so that the ITERATE instruction may operate to find desired inverse values.

7. ITERATE X IN ++, Y IN ++, DESIRED Y IN ++,STORE IN ++
a. assumes that ISETUP has generated blocks of X's for which the values of the function were subsequently computed and stored in the first two columns indicated.
b. isolates within each block of 5 X's the values which bracket one or more of the desired Y values.
c. can be used with the COMPARE instruction in the repeat mode to compute inverse functions to a given tolerance.
d. items 6 b through 6 g apply here as well.

8. FINISH STORING INSTRUCTIONS
a. terminates the storing of instructions and changes the program to the execute mode for the instructions which follow.
b. at this point the instructions which preceded the word FINISH have not been executed, and will not be until called for by the REPEAT instruction.
c. any illegal instruction preceding the word FINISH causes the program to regard FINISH as an illegal instruction, and when the original mistake is corrected, this will correct itself also.

9. REPEAT INSTRUCTIONS ,, THRU ,, ,, TIMES
a. provides for the execution of one or more of the previously stored instructions.
b. this instruction cannot be stored; it must be used outside of the stored set.
c. the stored instructions may be entered at any place and called for either one at a time or in consecutive increasing order.
d. stored instructions may be used any number of times or even ignored.

6.2. The Generation of Ad Hoc Subroutines Written in the OMNITAB Language

The point has been made earlier that OMNITAB is a general - purpose program. That is not to say it is an all- purpose program. However it is considerably more general than it appears to be because it can easily be extended by the novice to encompass operations not presently incorporated. Undoubtedly each reader will find different deficiencies and will probably wish that his own often used operations had been anticipated. Some omissions may indeed limit the application of the program in certain areas. We hold no brief for the use of this program in areas for which it is not suited, but resist the temptation to define more explicitely the "proper" areas of application - leaving it to the reader to bridge the gap between the computation problems illustrated and his own. This point of view has been forged by numerous instances where colleagues have successfully applied the program to problems more involved and complex than we were prepared to accept on cursory examination.

Having thus disposed of the "large" problems we return now to a brief discussion of a number of simple operations which have not been programmed explicitly but which can nevertheless be performed with considerable dispatch.

This section has been deferred until now in order to illustrate an important feature of the OMNITAB arrangement which permits one to write special ad hoc operations and to handle them as little subroutines in the repeat mode.

The harmonic mean of a column of numbers stored in column 1 can be achieved via the instructions:

```
DIVIDE 1. BY COL 1, STORE IN 1
AVERAGE COL 1, STORE IN COL 2
DIVIDE 1. BY COL 2, STORE IN 2
```

The geometric mean can be achieved almost as easily via logarithms as follows:

```
LOGTEN 1, 1
AVERAGE 1, 1
ANTILOG 1, 1
```

The generation of the geometric mean without resorting to logarithms requires a few more instructions. We leave it to the reader to perform this exercise.

Consider the problem of dropping off the decimal part of a number and retaining only the integer part. Although such a command is useful, there is no instruction in OMNITAB at present to do this. Let us see if this can be achieved via the existing instructions.

The instruction RAISE performs the operation A^b. If b is an integer the operation proceeds via b multiplications, with due regard for the sign when A is negative. If b is not an integer and A is positive, the program multiplies the logarithm of A by b and takes the antilog of the product. If A is negative, however, the program drops off the decimal portion of the power and raises the base to an integral power by multiplication. This mode of operation can now be used to recover the integral part as follows:

Let us assume that the numbers whose integral portions are required are contained in column 1. The following instructions

```
ADD -10., 0., STORE IN 2
RAISE COL 2 TO COL 1, STORE IN COL 3
ABSOLUTE VALUE OF 3, STORE IN 3
ANTILOG 3 STORE IN  3
```

will provide in column 3 only the integral part of the number. From this point the calculations can continue as necessary. If the above operations were required many times during a calculation, the above 4 instructions could be stored in the core and used under the repeat mode.

Let us now assume that each of the above three operations will be required many times in a particular calculation as for example on 2 or more sets of data. In particular let us assume that the truncation of a set or sets of data are desired before taking the harmonic and geometric mean. We will however leave the instructions in the order in which they are illustrated. The OMNITAB calculations, to accomplish this in the repeat mode, might be as follows:

```
OMNITAB 10/20/64
BEGIN STORING INSTRUCTIONS
DIV 1., 1, 11
AVERAGE 11, 2
DIV 1., 2, 2
LOGE 1, 12
AVERAGE 12, 12
EXP 12, 12
PRINT 10, 1, 12, 13
ADD -10., 0., 13
RAISE 13, 10, 13
ABSOLUTE 13, 13
LOGTEN  13, 1
FINISH STORING INSTRUCTIONS

SET IN COL 10 DATA TO FOLLOW
 - - - - - - - - - -
   (data follows)

REPEAT INST 8 THRU 11, 1 TIME
REPEAT INST 1 THRU 7, 1 TIME
SET 10
   - - - - - - - -
   (data follows)

REPEAT 8, 11, 1
REPEAT 1, 7, 1
    etc.
```

6.3. Problems Solved in the Repeat Mode

We now consider the solution of a number of problems in the Repeat mode.

Problem 6 — 1

Re—do Problem 3—5 using the Repeat mode.

The OMNITAB instructions are as follows:

```
CMNITAB    PRCBLEM 6 - 1
TITLE1  RECIPRCCAL TABLE, 1000./T -  T = 100.0 TO 200.0 BY 0.1
GENERATE 100.(1.)200. ,1
ACC  1 0. 20
BEGIN
CIVICE 1000. BY 20 ,2
ACC .1 20 20
INCEX 1 BY 1
FINISH
REPEAT 1 TO 3 ,10 TIMES
FCRMAT(1H F9.0,10F10.5)
FPRINT 1,2,3,4,5,6
FPRINT 1,7,8,9,10,11
      STCP
```

* * * RESULTS * * *

```
PAGE   1      OMNITAB    PROBLEM 6 - 1
  RECIPRCCAL TABLE, 1000./T -  T = 1C0.0 TO 200.C BY 0.1
```

100.	10.00000	9.99001	9.98C04	9.97009	9.96016
101.	9.90099	9.89120	9.88142	9.87167	9.86193
102.	9.80392	9.79432	9.78474	9.77517	9.76563
103.	9.70874	9.69932	9.68992	9.68054	9.67118
104.	9.61538	9.60615	9.59693	9.58773	9.57854
105.	9.52381	9.51475	9.50570	9.49668	9.48767
106.	9.43396	9.42507	9.41620	9.40734	9.39850
107.	9.34579	9.33707	9.32836	9.31966	9.31099
108.	9.25926	9.25C69	9.24214	9.23361	9.22509
109.	9.17431	9.16590	9.15751	9.14913	9.14077
110.	9.09091	9.08265	9.07441	9.06618	9.05797
111.	9.00901	9.00090	8.99281	8.98473	8.97666
112.	8.92857	8.92061	8.91266	8.90472	8.89680

Problem 6 − 2

Compute the table of Problem 3−6 in the Repeat mode with the addition of a PLOT and a SPLOT.

The OMNITAB instructions are as follows:

```
OMNITAB        PROBLEM 6 - 2
TITLE1     TABLE OF VERTICAL HEIGHTS
NOSUMMARY
FIXED 6
GENERATE  0. .01 .5 1
BEGIN
TAND 10. 1 2
INCREMENT 1 BY 10. 0 1
FINISH
REPEAT 1 2 8
HEAD 1/        M
HEAD 2/  THETA = 10
HEAD 3/  THETA = 20
HEAD 4/  THETA = 30
HEAD 5/  THETA = 40
HEAD 6/  THETA = 50
HEAD 7/  THETA = 60
HEAD 8/  THETA = 70
HEAD 9/  THETA = 80
PRINT 1 2 3 4 5
PRINT 1 6 7 8 9
PLOT  2 4 6 8 AGAINST 1      0. 1. 0. 1.0
SPLOT 2 3 4 5 6 7 8 9 AGAINST 1,X FROM  0. TO .6,Y FROM 0. 1.0
       STOP
```

PAGE 5 OMNITAB PROBLEM 6 - 2
TABLE OF VERTICAL HEIGHTS
ABSCISSA - COLUMN 1
ORDINATES - COLUMN 8 (.), COLUMN 6 (*), COLUMN 4 (+), COLUMN 2 (,),

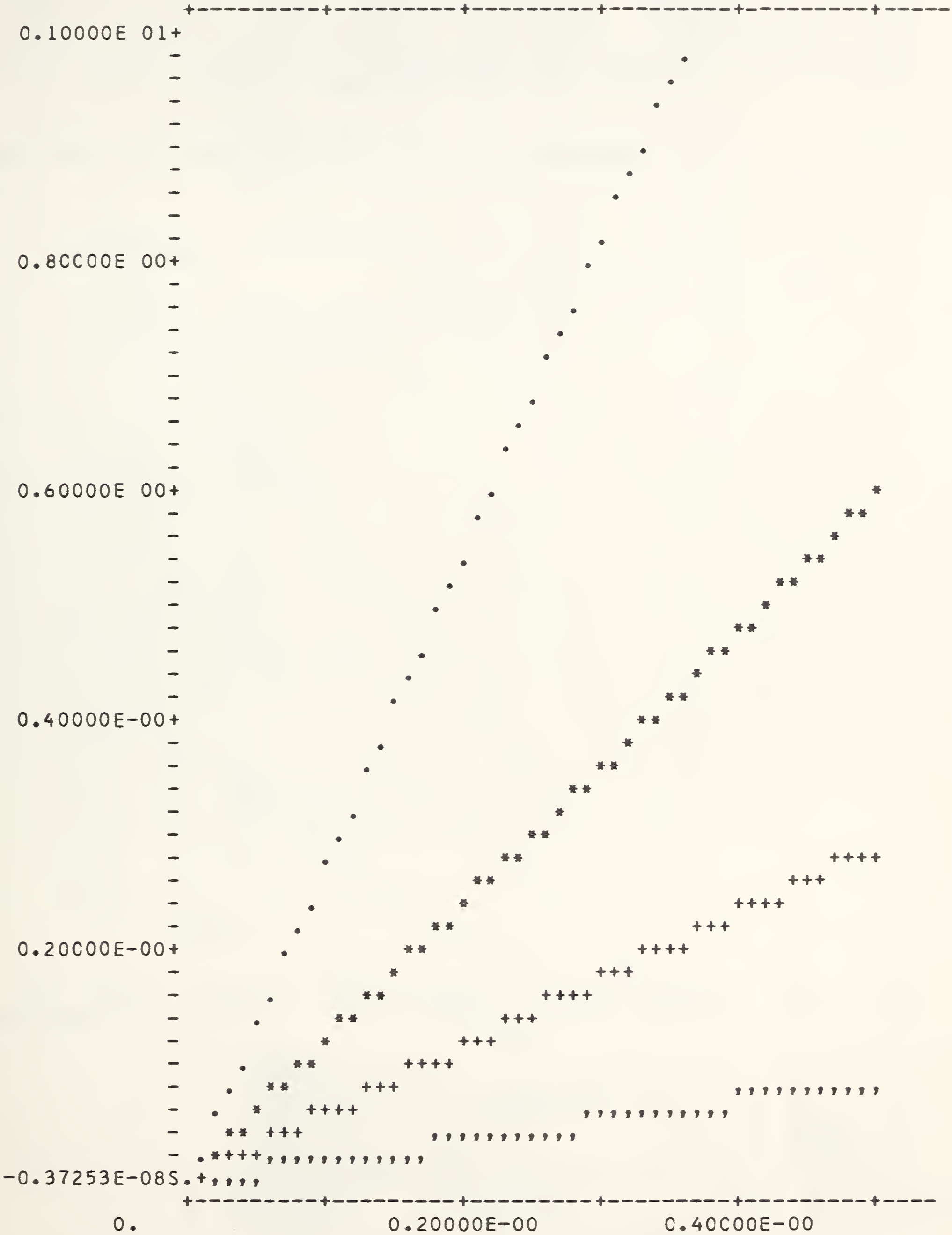
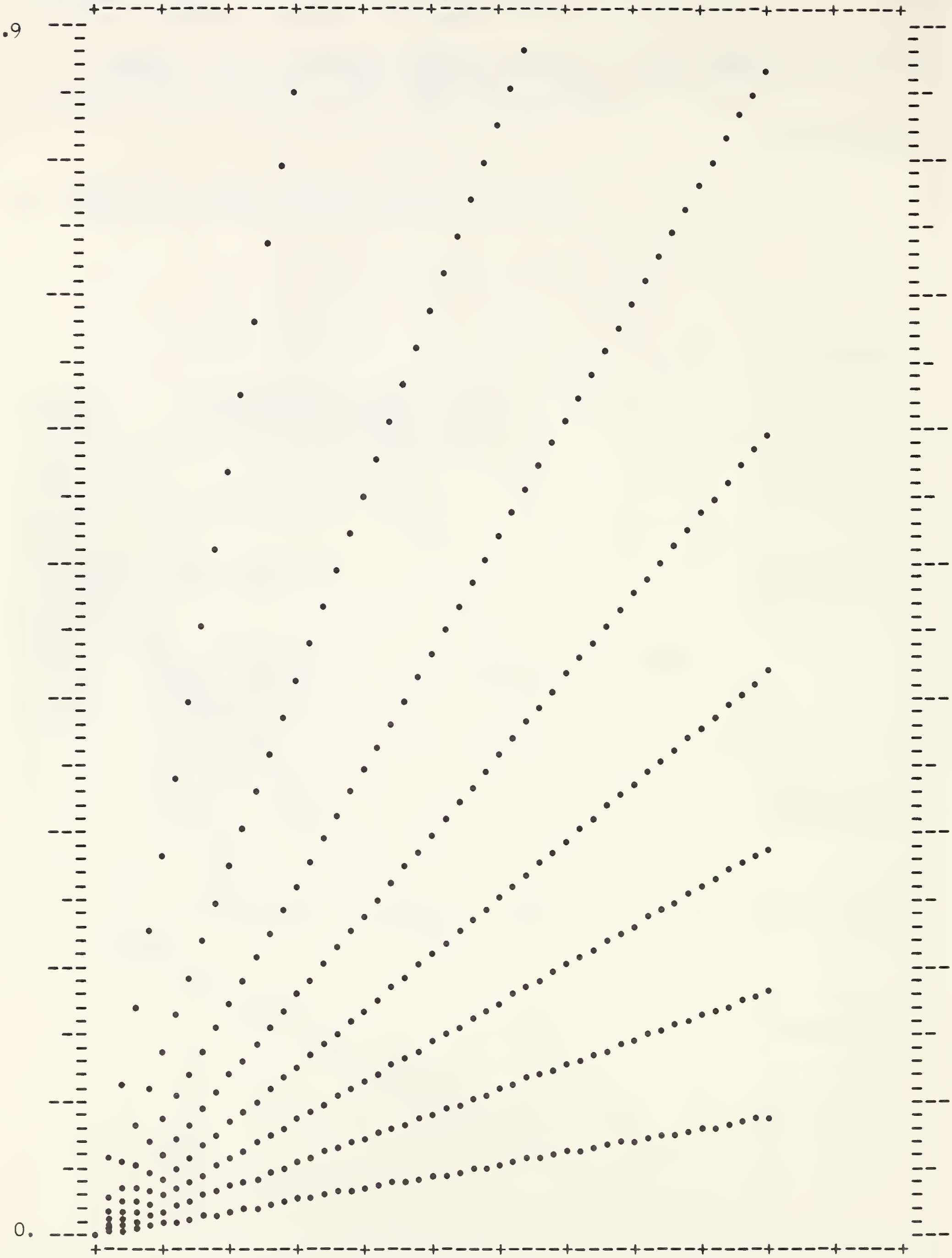


Figure 6-1. A plot for Problem 6-2 using the PLOT instruction. See Figure 6-2 for a more precise plot of the same data.

TABLE OF VERTICAL HEIGHTS



MINIMUM X = 0. MAXIMUM X = 0.6000000E 00
MINIMUM Y = 0. MAXIMUM Y = 0.1000000E 01

Figure 6-2. A portion of a plot for Problem 6-2 using the SPLOT instruction.

Problem 6 – 3

Compute $J_1(x)$ for values of x corresponding to the zeros of $J_0(x)$. First find the zeros between $0 \leq x \leq 60$ using the instruction ISOLATE. Illustrate the convergence of $J_0(x)$ toward zero by storing results for each cycle and printing them side by side.

The OMNITAB instructions are as follows:

```
OMNITAB      PROBLEM 6 - 3
NOSUMMARY
TITLE1                 ZEROS  OF  BESSEL  J  ZERO
GENERATE .6(.6) 60.  IN COL 1
BEJZERO OF 1 STORE 2
ISOLATE 1 2 0. 3
BEJZERO 3 4
ISOLATE 3 4 0. 5   °
BEJZERO  5 6
ISOLATE 5 6 0. 7
BEJZERO 7 8
ISOLATE 7 8 0. 9
BEJZERO 9 10
ISOLATE 9 10 0. 11
BEJZERO 11 12
ISOLATE 11 12 0. 13
BEJZERO 45  24
BEJONE 45 25
FIXED 6
HEAD COL 45/      X
HEAD 24/  JZERO(X)
HEAD COL 25/  JONE(X)
PRINT  45  24  25
PRINT  1 2 3 4 5 6
PRINT 7 8 9 10 11 12
       STOP
```

° In the Repeat mode these instructions could be replaced by:

```
BEGIN
BEJZERO OF 1, STORE IN 2
ISOLATE 1,2,0.,3
INCREMENT 1 BY 2,2
INCREMENT 2 BY 2,2,0.,2
FINISH
REPEAT INST 1 TO 4, 6 TIMES
```

Results after Six Iterations for the Zeros of $J_0(x)$

PAGE 1 OMNITAB PROBLEM 6 - 3
ZEROS OF BESSEL J ZERO

X	JZERO(X)	JONE(X)
2.404980	-0.000080	0.519114
5.519824	-0.000087	-0.340280
8.654003	-0.000075	0.271444
11.791698	0.000038	-0.232457
14.931152	-0.000048	0.206543
18.071190	0.000024	-0.187727
21.211815	-0.000031	0.173264
24.352440	-0.000005	-0.161702
27.493651	-0.000026	0.152180
30.634862	0.000037	-0.144165
33.776072	-0.000035	0.137296
36.917283	0.000024	-0.131324
40.058493	-0.000008	0.126069
43.199706	-0.000010	-0.121399
46.340916	0.000032	0.117212
49.482712	0.000012	-0.113429
52.623923	0.000014	0.109991
55.765720	0.000022	-0.106847
58.906930	0.000006	0.103960
0.	1.000000	0.
0.	1.000000	0.
0.	1.000000	0.
0.	1.000000	0.
0.	1.000000	0.
0.	1.000000	0.
0.	1.000000	0.
0.	1.000000	0.
0.	1.000000	0.
0.	1.000000	0.
0.	1.000000	0.

Results of successive applications of the ISOLATE instruction to obtain zeros of $J_0(X)$ are shown on the next page.

PAGE 3 OMNITAB PROBLEM 6 - 3
ZEROS OF BESSEL J ZERO

COLUMN 1	COLUMN 2	COLUMN 3	COLUMN 4	COLUMN 5	COLUMN 6
0.600000	0.912005	2.400000	0.002508	2.400000	0.002508
1.200000	0.671133	2.550000	-0.072923	2.437500	-0.016846
1.800000	0.339986	2.700000	-0.142449	2.475000	-0.035880
2.400000	0.002508	2.850000	-0.205102	2.512500	-0.054578
3.000000	-0.260052	3.000000	-0.260052	2.550000	-0.072923
3.600000	-0.391769	5.400000	-0.041210	5.400000	-0.041210
4.200000	-0.376557	5.550000	0.010152	5.437500	-0.028279
4.800000	-0.240425	5.700000	0.059920	5.475000	-0.015396
5.400000	-0.041210	5.850000	0.107066	5.512500	-0.002580
6.000000	0.150645	6.000000	0.150645	5.550000	0.010152
6.600000	0.274043	8.400000	0.069157	8.550000	0.028277
7.200000	0.295071	8.550000	0.028277	8.587500	0.018034
7.800000	0.215408	8.700000	-0.012523	8.625000	0.007810
8.400000	0.069157	8.850000	-0.052346	8.662500	-0.002380
9.000000	-0.090334	9.000000	-0.090334	8.700000	-0.012523
9.600000	-0.208979	11.400000	-0.090215	11.700000	-0.021331
10.200000	-0.249617	11.550000	-0.056180	11.737500	-0.012584

PAGE 5 OMNITAB PROBLEM 6 - 3
ZEROS OF BESSEL J ZERO

COLUMN 7	COLUMN 8	COLUMN 9	COLUMN 10	COLUMN 11	COLUMN 12
2.400000	0.002508	2.400000	0.002508	2.404687	0.000072
2.409375	-0.002360	2.402344	0.001289	2.405273	-0.000232
2.418750	-0.007208	2.404687	0.000072	2.405859	-0.000537
2.428125	-0.012036	2.407031	-0.001145	2.406445	-0.000841
2.437500	-0.016846	2.409375	-0.002360	2.407031	-0.001145
5.512500	-0.002580	5.512500	-0.002580	5.519531	-0.000186
5.521875	0.000611	5.514844	-0.001782	5.520117	0.000013
5.531250	0.003797	5.517187	-0.000984	5.520703	0.000213
5.540625	0.006978	5.519531	-0.000186	5.521289	0.000412
5.550000	0.010152	5.521875	0.000611	5.521875	0.000611
8.625000	0.007810	8.653125	0.000164	8.653125	0.000164
8.634375	0.005259	8.655468	-0.000472	8.653710	0.000005
8.643750	0.002710	8.657812	-0.001108	8.654296	-0.000154
8.653125	0.000164	8.660156	-0.001744	8.654882	-0.000313
8.662500	-0.002380	8.662500	-0.002380	8.655468	-0.000472
11.774999	-0.003846	11.784374	-0.001665	11.791405	-0.000030
11.784374	-0.001665	11.786718	-0.001120	11.791991	0.000106
11.793749	0.000515	11.789062	-0.000575	11.792577	0.000242
11.803124	0.002693	11.791405	-0.000030	11.793163	0.000379
11.812499	0.004869	11.793749	0.000515	11.793749	0.000515
14.925000	0.001223	14.925000	0.001223	14.929687	0.000254
14.934375	-0.000714	14.927343	0.000738	14.930273	0.000133
14.943750	-0.002649	14.929687	0.000254	14.930859	0.000012
14.953125	-0.004583	14.932031	-0.000230	14.931445	-0.000109
14.962500	-0.006515	14.934375	-0.000714	14.932031	-0.000230
18.037499	-0.006306	18.065624	-0.001021	18.070311	-0.000141
18.046874	-0.004544	18.067968	-0.000581	18.070897	-0.000031
18.056249	-0.002782	18.070311	-0.000141	18.071483	0.000079
18.065624	-0.001021	18.072655	0.000299	18.072069	0.000189
18.074999	0.000739	18.074999	0.000739	18.072655	0.000299
21.187499	0.004184	21.206249	0.000934	21.210936	0.000121
21.196874	0.002559	21.208593	0.000528	21.211522	0.000020
21.206249	0.000934	21.210936	0.000121	21.212108	-0.000082
21.215624	-0.000691	21.213280	-0.000285	21.212694	-0.000183
21.224999	-0.002314	21.215624	-0.000691	21.213280	-0.000285
24.337499	-0.002422	24.346874	-0.000905	24.351561	-0.000147
24.346874	-0.000905	24.349218	-0.000526	24.352147	-0.000053
24.356249	0.000611	24.351561	-0.000147	24.352733	0.000042
24.365624	0.002126	24.353905	0.000232	24.353319	0.000137
24.374999	0.003641	24.356249	0.000611	24.353905	0.000232

22.3457	1299.7479	0.	0.	0.
22.3458	1299.8774	0.	0.	0.
22.3459	1300.0065	0.	0.	0.
22.3460	1300.1358	0.	0.	0.
22.3461	1300.2650	0.	0.	0.
22.6402	1399.5785	0.	0.	0.
22.6403	1399.7146	0.	0.	0.
22.6404	1399.8510	0.	0.	0.
22.6405	1399.9871	0.	0.	0.
22.6406	1400.1235	0.	0.	0.
22.9195	1499.5981	0.	0.	0.
22.9196	1499.7414	0.	0.	0.
22.9197	1499.8844	0.	0.	0.
22.9198	1500.0276	0.	0.	0.
22.9199	1500.1707	0.	0.	0.
21.5000	1040.8707	0.	0.	0.
21.6000	1069.3777	0.	0.	0.
21.7000	1098.4398	0.	0.	0.
21.8000	1128.0631	0.	0.	0.
21.9000	1158.2537	0.	0.	0.
22.0000	1189.0179	0.	0.	0.
22.1000	1220.3622	0.	0.	0.
22.2000	1252.2923	0.	0.	0.
22.3000	1284.8148	0.	0.	0.
22.4000	1317.9357	0.	0.	0.
22.5000	1351.6614	0.	0.	0.
22.6000	1385.9985	0.	0.	0.
22.7000	1420.9526	0.	0.	0.
22.8000	1456.5306	0.	0.	0.
22.9000	1492.7386	0.	0.	0.
23.0000	1529.5830	0.	0.	0.

PAGE 4 OMNITAB PROBLEM 6 - 4
VAPOR PRESSURE OF LIQUID NORMAL HYDROGEN

COLUMN 44	COLUMN 45	COLUMN 46
99.9996	15.1014	100.0000
199.9896	16.6080	200.0000
300.0213	17.6193	300.0000
400.0089	18.4041	400.0000
500.0059	19.0557	500.0000
600.0292	19.6182	600.0000
699.9803	20.1158	700.0000
799.9827	20.5646	800.0000
900.0219	20.9748	900.0000
1000.0479	21.3533	1000.0000
1099.9876	21.7053	1100.0000
1200.0318	22.0354	1200.0000
1299.9419	22.3459	1300.0000
1400.0553	22.6404	1400.0000
1499.9560	22.9197	1500.0000

Problem 6 — 5

Locate the maxima and minima of the function in Problem 3—10 making sure to tabulate the function at values which do not coincide with the integers.

The OMNITAB instructions are as follows:

```
OMNITAB       PROBELM 6 - 5      TEST OF MAXMIN AND ISOLATE
TITLE1                           TEST OF MAXMIN
NOSUMMARY
GENERATE  -0.35(.2)6.35 IN COL 1
BEGIN STORING INSTRUCTIONS
SUBTRACT 1. FRCM COL 1 , MULT BY 1 ADD TO COL 9
SUB   2. FRCM COL 1,STORE 2
SUB      3. FRCM 1 MULT BY 2, ADD TO COL 3
SUB   4. ,1,2
SUB  5. FRCM 2 MULT 2 ADD TO COL 4
SUBTRACT  6. FROM COL 1, MULT BY COL 4, ADD TO COL 5
MULT CCL 5 BY COL 3 MULT BY COL 9 ADD TO COL 6
PLCT CCL 6 AGAINST COL 1,-1.,6.,-100.,100.
ISCLATE X IN CCL 1,Y IN COL 6,ABOUT 0.,STORE X IN COL 1
ERASE CCL 2 3 4 5 6 9
ADC COL 45 TO COL 19 STORE IN COL 21
INCREMENT INST 11 BY 0 0 1
                    FINISH STORING INSTRUCTIONS
REPEAT INSTRUCTION 1 THRU 8, 1 TIME
MAXMIN X IN 1, Y IN 6, STORE IN 11 12 13 14
FIXEC 6
PRINT 1 6 11 12 13 14
REPEAT INST 1 THRU 12 5 TIMES
TITLE1          TEST OF THE ISOLATE INSTRUCTION
PRINT CCLS 21 22 23 24 25
GENERATE 0.(.1)6. IN COL 1
TITLE1                    TEST OF MAXMIN ON  A FINER GRID
REPEAT INST 1 THRU 8 1 TIME
ERASE 11 12 13 14
MAXMIN X IN 1 Y IN 6 ,STORE IN 11 12 13 14
PRINT 1 6 11 12 13 14
       STOP
```

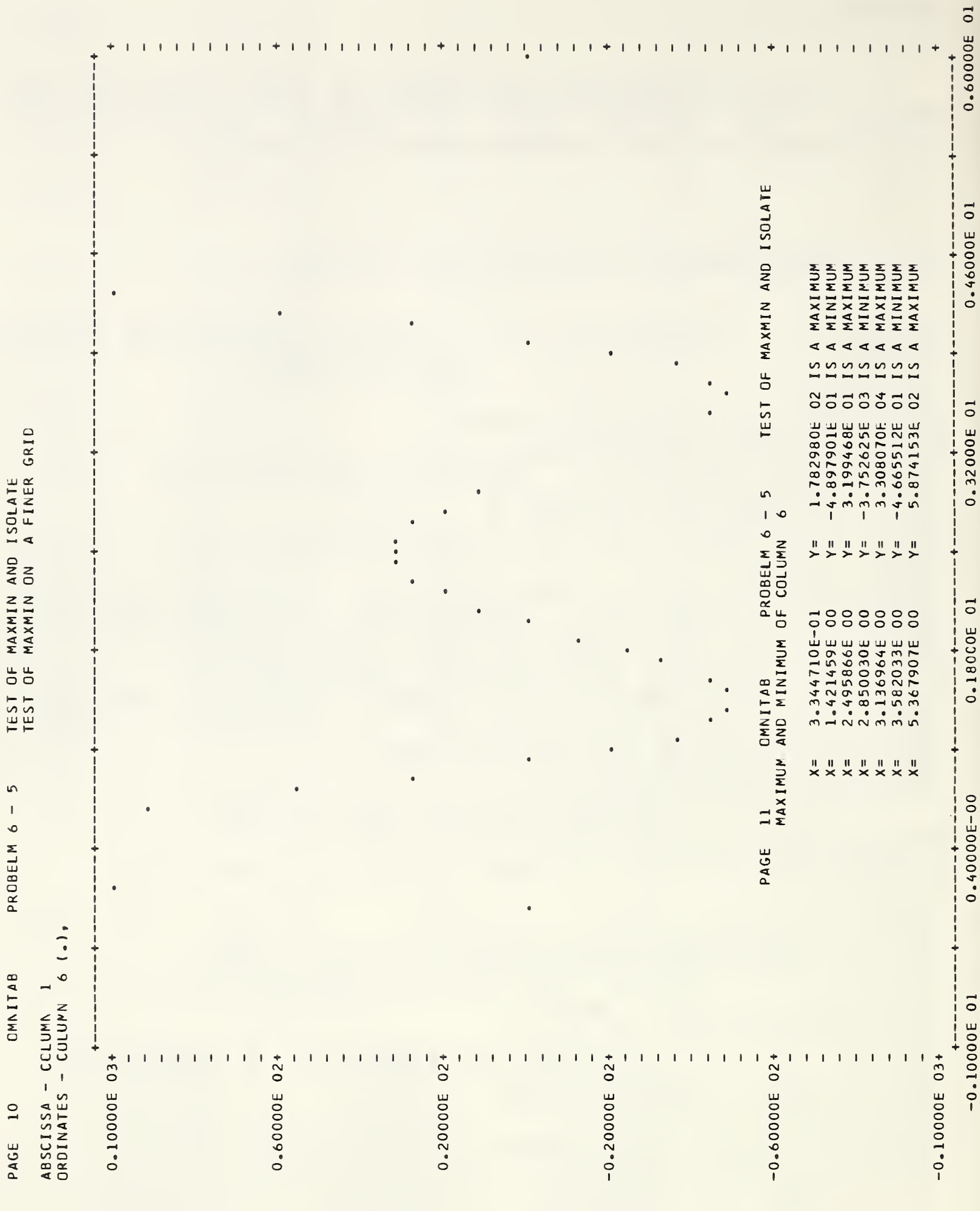


Figure 6–3. A plot of the function generated in Problem 6–5 with the automatic printout from MAXMIN superimposed upon it.

```
OMNITAB      10/6/64
NOSUMMARY
TITLE1  JAHNKE AND EMDE   TABLES OF FUNCTIONS WITH FORMULAE AND CURVES
READ  1 2 3 4 5 6
0.5 0.9659 0.9330 0.9013 0.8706 0.8409
0.6 0.9748 0.9502 0.9262 0.9029 0.8801
0.7 0.9823 0.9650 0.9497 0.9312 0.9147
0.8 0.9889 0.9779 0.9673 0.9551 0.9457
0.9 0.9948 0.9895 0.9843 0.9792 0.9740
1.0 1.0000 1.0000 1.0000 1.0000 1.0000
1.2 1.0092 1.0184 1.0277 1.0371 1.0466
1.4 1.0170 1.0342 1.0518 1.0696 1.0878
1.6 1.0238 1.0481 1.0730 1.0986 1.1247
1.8 1.0298 1.0605 1.0922 1.1247 1.1583
2.0 1.0353 1.0718 1.1096 1.1487 1.1892
2.2 1.0402 1.0820 1.1255 1.1708 1.2179
2.4 1.0447 1.0915 1.1403 1.1914 1.2447
2.6 1.0489 1.1003 1.1541 1.2106 1.2698
2.8 1.0528 1.1084 1.1670 1.2287 1.2936
3.0 1.0565 1.1161 1.1791 1.2457 1.3161
3.5 1.0646 1.1335 1.2067 1.2847 1.3678
4.0 1.0718 1.1487 1.2311 1.3195 1.4142
4.5 1.0781 1.1623 1.2531 1.3510 1.4565
5.0 1.0838 1.1746 1.2731 1.3797 1.4953
10. 1.1220 1.2589 1.4125 1.5849 1.7783
BEGIN
DIVDIF   OF COL 2 X IN COL 1
INCREMENT INST 1 BY 1,0
RAISE COL 1 TO .05 POWER STORE IN COL 7
INCREMENT INST 3 BY 0,.05,1
SUBTRACT COL 2 FROM COL 7 STORE IN COL 21
INCREMENT INST 5 BY 1,1,1
FINISH
REPEAT INST 1 TO 2  ,  5 TIMES
REPEAT INST 3 TO 4 ,  20 TIMES
TITLE2OMNITAB CALCULATIONS FOR POWERS  GIVEN ON PAGES 8 AND 9 OF ADDENDA
FORMAT (1F11.1,10F9.5)
FPRINT 1,7,8,9,10,11,12,13,14,15,16
FPRINT  1,17,18,19,20,21,22,23,24,25,26
TITLE2      I.   TABLES OF POWERS  PAGE 8 OF ADDENDA
FIXED   4  DECIMALS
HEAD COL 1/          X
HEAD COL 2/  X TO 0.05
HEAD COL 3/  X TO 0.10
HEAD COL 4/  X TO 0.15
HEAD COL 5/  X TO 0.20
HEAD COL 6/  X TO 0.25
PRINT 1 2 3 4 5 6
REPEAT   5  TO  6  ,  5 TIMES
TITLE2DIFFERENCE OF PAGE 8 OF ADDENDA FROM VALUES CALCULATED ON OMNITAB
PRINT  1,21,22,23,24,25
      STOP
```

Figure 6-4. Application of the Repeat mode to the calculation of a table of fractional powers of certain numbers.

Problem 6 - 6

Compute the function:

$$\Gamma\left[\frac{6T - S}{12}\right]$$

for T = 0, 1, 2, ... 40 and S = -3, -2, -1, 0, 1, 2, 3

The OMNITAB instructions are as follows:

```
OMNITAB   PROBLEM 6 - 6
TITLE1         GAMMA((6T-S)/(12))
GENERATE  0. IN STEPS OF 1. TO 40. STORE 1
HEAD 1/        T
MULT COL 1 BY 6. STORE 2
BEGIN
SUBTRACT -3. FROM COL 2 MULTIPLY BY .08333333 ADD TO COL 11
GAMMA OF 11 STORE 11
INCREMENT 1 BY 1. 0 0. 1
INCREMENT 2 BY 1 1
FINISH
REPEAT INSTRUCTIONS 1 TO 4 , 7 TIMES
HEAD  11/ S = -3
HEAD  12/ S = -2.
HEAD  13/ S = -1.
HEAD 14/ S= 0.
HEAD 15/ S= 1.
HEAD COL 16/ S = 2.
HEAD 17/ S = 3.
NOSUMMARY
PRINT 1 11 12 13 14
PRINT  1 15 16 17
      STOP
```

Problem 6 - 7

Using the relation 1 inch = 25.4 millimeter, compute a table to convert in either direction for 50 values in unit steps. Tabulate the values in each of the units equivalent to integral values of the other from 1 to 50. Print the results in six columns with the numbers 1 to 25 in column 2 and 26 to 50 in column 5. Arrange for printing in blocks of five instead of the usual blocks of ten provided by the normal OMNITAB print instruction.

The OMNITAB instructions are as follows:

```
OMNITAB   PROBLEM 6 - 7
GENERATE 1. 1. 50. 2
MULT 2 25.4 3
DIV  2 25.4 1
FORMAT(1X,1F10.5,1F5.0,1F8.1,1F12.5,1F5.0,1F8.1)
BLOCKTRANSFER 26,1,25,3 INTO 1,4 (SEE FOOTNOTE)
NEWPAGE
NOTE
NOTE     INCHES              MM        INCHES              MM
NOTE   +---------+----+----------+----------+----+----------+
NOTE
BEGIN
FABRIDGE ROW 1 COLS 1 2 3  4 5 6
INCREMENT 1 BY 1 0 0 0 0 0 0
FINISH
REPEAT 1 2 5
SPACE 1
REPEAT 1 2 5
SPACE 1
REPEAT 1 2 5
SPACE 1
REPEAT 1 2 5
SPACE 1
REPEAT 1 2 5
NOTE   +---------+----+----------+----------+----+----------+
      STOP
```

° See section 8.1 for instructions on BLOCKTRANSFER.

INCHES		MM	INCHES		MM
0.03937	1.	25.4	1.02362	26.	660.4
0.07874	2.	50.8	1.06299	27.	685.8
0.11811	3.	76.2	1.10236	28.	711.2
0.15748	4.	101.6	1.14173	29.	736.6
0.19685	5.	127.0	1.18110	30.	762.0
0.23622	6.	152.4	1.22047	31.	787.4
0.27559	7.	177.8	1.25984	32.	812.8
0.31496	8.	203.2	1.29921	33.	838.2
0.35433	9.	228.6	1.33858	34.	863.6
0.39370	10.	254.0	1.37795	35.	889.0
0.43307	11.	279.4	1.41732	36.	914.4
0.47244	12.	304.8	1.45669	37.	939.8
0.51181	13.	330.2	1.49606	38.	965.2
0.55118	14.	355.6	1.53543	39.	990.6
0.59055	15.	381.0	1.57480	40.	1016.0
0.62992	16.	406.4	1.61417	41.	1041.4
0.66929	17.	431.8	1.65354	42.	1066.8
0.70866	18.	457.2	1.69291	43.	1092.2
0.74803	19.	482.6	1.73228	44.	1117.6
0.78740	20.	508.0	1.77165	45.	1143.0
0.82677	21.	533.4	1.81102	46.	1168.4
0.86614	22.	558.8	1.85039	47.	1193.8
0.90551	23.	584.2	1.88976	48.	1219.2
0.94488	24.	609.6	1.92913	49.	1244.6
0.98425	25.	635.0	1.96850	50.	1270.0

Results of Problem 6 - 7.

Problem 6 - 8

Compute log Y for 50 values between 1.01 and 1.50 from the series:

$$\log(1 + x) = x - \frac{x^2}{2} + \frac{x^3}{3} - \frac{x^4}{4} + \frac{x^5}{5} - \frac{x^6}{6} + \ldots.$$

and compare with results from the instruction LOGE.

The OMNITAB instructions are as follows:

```
OMNITAB    PROBLEM 6 - 8
NOSUMMARY
GENERATE .01 .01 .5 1
ADD 1. 1 2
LOGE 2 10
ADD 1. 0. 5
BEGIN
ADD 1. 3 3
DIVIDE 1. 3 5 4
RAISE 1 TO 3 MULT BY 4 ADD 6
MULT 5 -1. 5
ERASE 4
COMPARE COL 6 TO COL 10 TO 1.E-4
FINISH
REPEAT 1 TO 6 2 TIMES
REPEAT 1 TO 6  30 TIMES
HEAD 2/    Y
HEAD 6/EXPANSION
HEAD 10/ FUNCTION
PRINT 2 6 10
      STOP
```

The output from this problem is in three parts. (See the next page.) The first is a statement that the comparison was not satisfied on 2 passes as requested in the first REPEAT instruction. This was followed by three columns of results representing the status of the columns being compared and the computed tolerance which exceeded the desired tolerance in some cases.

The second part is a statement that the comparison was indeed satisfied on the 8th pass resulting from the second REPEAT instruction which would have permitted 30 iterations.

The third portion is in response to the PRINT and is not given automatically as were the first two parts.

See problem 9-3 for a solution to this problem without recourse to the repeat mode.

INSTRUCTICNS HAVE BEEN REPEATED THE NUMBER OF TIMES
INDICATED. THE COMPARE INSTRUCTION HAS NOT BEEN SATISFIED.

9.950000E-03	9.950319E-03	3.199856E-05
1.980000E-02	1.980262E-02	1.323079E-04
2.955000E-02	2.955878E-02	2.971155E-04
3.920000E-02	3.922068E-02	5.273564E-04
4.875000E-02	4.879016E-02	8.230219E-04
5.820000E-02	5.826890E-02	1.182348E-03
6.755000E-02	6.765863E-02	1.605580E-03
7.680000E-02	7.696105E-02	2.092628E-03
8.595000E-02	8.617770E-02	2.642171E-03
9.500000E-02	9.531018E-02	3.254444E-03
1.039500E-01	1.043600E-01	3.928831E-03
1.128000E-01	1.133287E-01	4.665196E-03
1.215500E-01	1.222176E-01	5.462782E-03
1.302000E-01	1.310283E-01	6.321278E-03
1.387500E-01	1.397[illegible]	[illegible]
1.472000E-01	[illegible]	5.377581E-[illegible]
[illegible]	[illegible]576744E-01	5.626472E-02
3.432000E-01	3.646431E-01	5.880577E-02
3.487500E-01	3.715635E-01	6.139880E-02
3.542000E-01	3.784364E-01	6.404358E-02
3.595500E-01	3.852624E-01	6.673998E-02
3.648000E-01	3.920421E-01	6.948763E-02
3.699500E-01	3.987761E-01	7.228645E-02
3.750000E-01	4.054651E-01	7.513622E-02

COMPARE SATISFIED, CONTROL FORCED FROM REPEAT MODE AFTER 8 TIMES

* * * * * * * * * *

PAGE 1 OMNITAB PROBLEM 6 - 8

Y	EXPANSION	FUNCTION
1.010000E 00	9.950331E-03	9.950319E-03
1.020000E 00	1.980263E-02	1.980262E-02
1.030000E 00	2.955880E-02	2.955878E-02
1.040000E 00	3.922071E-02	3.922068E-02
1.050000E 00	4.879016E-02	4.879016E-02
1.060000E 00	5.826891E-02	5.826890E-02
1.070000E 00	6.765864E-02	6.765863E-02
1.080000E 00	7.696104E-02	7.696105E-02
1.090000E 00	8.617770E-02	8.617770E-02
1.100000E 00	9.531018E-02	9.531018E-02

```
OMNITAB                   DATA  135       AUGUST 1963
      TITLE1 BURK     IBM2
      READ 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20
2 1235   0 099 071   0   0 125 095 325 270 210 028 730 460   0
2 1255 225 106 088 255 220 106 082 290 255 220 028 731 450   (
2 1313 250   0   0 180 100 117 080   0   0 220 031 730 475   (
2 1335   0   0   0   0   0   0   0   0   0 215 029 729 470 413
2 1345 250 091 082 190 120 123 088 330 305 230 027 728 465 415
2 1400   0 103 074 205 130 129 091 295 270 225 027 730 460 406
2 1415 210 091 069 210 160 128 089 260 210 250 025 727 475 41
2 1430 208 090 068 260 180 133 094 240 190 250 025 731 480 41
2 1445 198 066 050 260 180 130 096 230 170 265 022 735 470 45
2 1500 155 057 041 260 200 125 090 225 165 265 019 736 450 44
2 1515   0 053 037 250 190 109 089 220 160 260 023 743 455 44
2 1530 175 050 035 230 180 121 087 245 165 260 022 745 440 4
2 1545   0 050 036 220 180 120 087 240 175 270 018 745 440 4
2 1600 155 045 031 200 160 115 082 310 180 255 018 745 445 4
2 1615   0 046 030 200 160 117 085 265 160 260 020 745 445 4
2 1630 150 046 034 180 150 108 075 240 155 250 018 739 445
2 1645 190 050 036 180 150 107 075 235 155 260 022 740 435
2 1700 155   0   0   0   0   0   0   0   0 240
2 1718 155 053 036 190 150 121 086 225 155
2 1732 155 055 039 190 150 119 084 220
      READ 1,2,21
2 1235   0
2 1255   0
2 1313   0
2 1335   0
2 1345   0
2 14
        0
  1700 070
 2 1718   0
 2 1732 084
      BEGIN
      PLOT COLUMN3AGAINSTCOLUMN4
      INCREMENT 1 BY0,BY1
      FINISH
      REPEAT INSTRUCTIONS 1THROUGH2 18TIMES
      INCREMENT INSTRUCTION1 BY 1 -17.
      REPEAT INSTRUCTIONS 1THROUGH2 17TIMES
      NOLIST
      STOP
```

Figure 6-5. Instructions in the Repeat mode for preparing 35 plots.

7. MATRIX OPERATIONS

By this time it should be quite clear that OMNITAB operates on columns of numbers in a fairly automatic and routine fashion, and that in most instances the length of a column is of no particular concern to the user since the program has its own way of keeping track of the length of the column or, more correctly, the length (NRMAX) of the longest column. The program derives some of its power and simplicity from the fact that once the length of a column is determined by the machine the instruction set does not need to supply this information. It will serve a purpose later for us to point out now that up to this point all operations started at the top of the designated column and worked down to NRMAX. Furthermore, all results are stored the same way and all normal printout starts at the top of the column and extends to the current value of NRMAX.

The extension of the program to handle matrix calculations more directly depends on a somewhat more complicated sentence structure since the specification of the size and location of a matrix requires at least four parameters. As soon as it becomes clear that there is no alternative to specifying the exact location of a matrix and that the sentence structure is still fairly manageable even when three matrices are mentioned in a single sentence, certain other features become apparent. The most important difference between the vector operations discussed earlier and the matrix operations is that it is no longer necessary to restrict operations to the top rows of the work sheet. Since the specification of the coordinates of the first element and the number of rows and columns is required anyway, the matrix can be located anywhere on the work sheet. Thus, operations can be performed on <u>any</u> array of numbers and stored in <u>any</u> location on the work sheet.

An important consideration in deciding on the sentence structure for the matrix operations was the desire to keep the number of parameters in a single instruction at or below 10 so that these instructions could be used in the repeat mode. Fortunately, this was easily achieved since in all of the operations the dimension of the resulting matrix is invariably defined by the operation and the original matrix or matrices. Thus, it is necessary to specify only the coordinates of the first element of the matrix in storing the result. For example, if a 5 x 5 matrix is stored starting in row 51, column 2, it will extend to row 55 and to column 6.

The sentences for 20 major matrix operations are given in the next section. Since the extent and location of the matrices are always given, these instructions are not subject to the control of NRMAX nor do they modify NRMAX in any way. But, NRMAX must still be reckoned with if results are to be printed via the normal print instructions discussed earlier. This is a problem only when results are generated and stored on the work sheet by matrix operations below the value of NRMAX.

The advantage of storing results below NRMAX and hence out of reach of the normal operations was deemed so attractive as to incorporate a new print instruction (called APRINT) which prints a suitably defined array of numbers regardless of their location on the work sheet. This and other array operations are discussed in the next chapter.

Since most of the matrix operations store results directly in the work sheet element by element, care must be taken in storing the results not to write over some portion of the original matrix which may still be needed to complete the operation. Thus, the sum of two matrices could properly be stored exactly on top of one of the addends but not in partially overlapping manner. In the case of matrix multiplication, the result must obviously be stored in an area completely distinct from that occupied by either of the original matrices. The instructions INVERT, EIGENVALUES, EIGENVECTOR, and LINEAR are not subject to the above restriction since the results are accumulated in a scratch area of the subroutine and transferred in a block at the end of the operation.

7.1. Matrix Operation Commands°

1. MADD [A] IN ,, ++, R = ,, C = ,, TO [B] IN ,, ++ START STORING IN ,, ++
   a. computes $[a_{ij}] + [b_{ij}] = [a_{ij} + b_{ij}]$
2. MSUB [A] IN ,, ++, R = ,, C = ,, FROM [B] IN ,, ++, START STORING IN ,, ++
   a. computes $[b_{ij}] - [a_{ij}] = [b_{ij} - a_{ij}]$
3. SCALAR PRODUCT OF [A] IN ,, ++, R = ,, C = ,, BY **, START STORING IN ,, ++
   a. computes $\alpha[A] = [\alpha a_{ij}]$ where α = **
4. MMULT [A] IN ,, ++, R = ,, C = ,, BY [B] IN ,, ++ WITH ,, COLUMNS, START STORING IN ,, ++
   a. computes $[a_{ij}][b_{jk}] = [c_{ik}]$, where

$$c_{ik} = \sum_{j=1}^{n} a_{ij} b_{jk}$$

5. DIAGVEC OF [A] IN ,, ++ R = ,, STORE IN COL ++
   a. stores the principal diagonal a_{ii} in the designated column.

6. MTRANS OF [A] IN ,, ++ R = ,, C = ,, START STORING IN ,, ++
   a. generates the transpose of the given R by C into a C by R matrix.

7. VECMAT [A] IN ,, ++, R= ,, C = ,, START STORING IN ,, ++
   a. generates a column vector as follows:

$a_{11}, a_{12}, a_{13} \cdots a_{1C}, a_{21} \cdots a_{2C} \cdots$
$a_{r1}, a_{r2} \cdots a_{rc}$

8. RESTMAT FROM ,, ++ INTO R = ,, C = ,, STARTING IN ,, ++
   a. restores a column matrix into rectangular form.
   b. this operation is the reverse of VECMAT.

9. TRACE OF [A] IN ,, ++ R = ,, STORE IN COL ++
   a. stores the sum of the elements, $\sum_{i=1}^{R} a_{ii}$ of the principal diagonal NRMAX times.

° In this chapter R is the number of rows and C is the number of columns in a matrix. When a column is indicated in an instruction without a row designation, the column is operated upon starting with row one.

10. COLNORM [A] IN ,, ++ R = ,, C = ,,
START STORING IN ,, ++

a. normalizes the matrix by columns.

b. divides the elements in the jth column by the square root of the sum of squares of the elements in the jth column.

$$b_{ij} = a_{ij} / \sqrt{\sum_i a_{ij}^2}$$

11. ROWNORM [A] IN ,, ++ R = ,, C = ,,
START STORING IN ,, ++

a. normalizes the matrix by rows.

b. divides the elements in the ith row by the square root of the sum of squares of the elements in the ith row.

$$b_{ij} = a_{ij} / \sqrt{\sum_j a_{ij}^2}$$

12. MRAISE [A] IN ,, ++ R = ,, C = ,, TO **,
START STORING IN ,, ++

a. raises the elements of the matrix to a constant power(**).

13. TRANSFORM [A] IN ,, ++ R = ,, ; R = ,, ROWS OF [U]
IN ,, ++ START STORING IN ,, ++

a. provides for the transformation $B = UAU^T$ where

$$[B] = [b_{ij}] = \sum_{k=1}^{R} \sum_{m=1}^{R} U_{ik} A_{km} U_{jm}$$

14. BACKTRANS [A] IN ,, ++ R = ,, ; R = ,, ROWS OF [U]
IN ,, ++ START STORING IN ,, ++

a. provides for the back transformation $C = U^T AU$ where

$$[C] = [c_{ij}] = \sum_{k=1}^{R} \sum_{m=1}^{R} U_{ki} A_{km} U_{mj}$$

15. SYMUTM [A] IN ,, ++, R = ,,

a. fills out an R by R symmetric matrix from the designated upper triangular matrix.

16. SYMLTM [A] IN ,, ++, R = ,,

a. fills out an R by R symmetric matrix from the designated lower triangular matrix.

17. EIGENVALUES OF [A] IN ,, ++ R = ,,
STORE ROOTS IN COL ++

a. stores the roots of a symmetric matrix in the designated column.

b. results are stored starting in row one regardless of the location of the original matrix.

18. EIGENVECTORS OF [A] IN ,, ++ R = ,, PUT ROOTS IN COL ++ AND VECTORS STARTING IN ,, ++

a. stores the roots starting in row 1 and the eigenvectors of a symmetric matrix as indicated.

19. LINEAR EQ IN ,, ++ R = ,, RIGHT HAND SIDE IN COL ++,
STORE SOLUTIONS IN COL ++

a. solves a system of n linear equations in n unknowns ($2 \leq n \leq 45$).

b. solves algebraically and not in the least squares sense as does the instruction SOLVE.

c. the value of the determinant of the coefficients is stored in row one, column 46.

20. INVERT [A] IN ,, ++ R = ,, START STORING IN ,, ++

a. inverts a matrix as large as 46 by 46.

b. computes a matrix [B] such that [A] [B] = [I] , where [I] is a unit matrix.

c. the value of the determinant of the coefficients is stored in row one, column 46.

7.2. Problems Solved via Matrix Operations

```
PAGE   1        OMNITAB   PROBLEM 7 - 1


   CONSIDER THE MATRIX


    1.0000    0.        0.        0.        0.
    0.0294    0.9996    0.        0.        0.
    0.4922    0.2023    0.8466    0.        0.
    0.5722    0.6218   -0.0344    0.5336    0.
    0.5032    0.6533    0.3764   -0.0671    0.4619


   AND ITS INVERSE AS GIVEN IN A WELL KNOWN TEXT AS


    1.00000   0.        0.        0.        0.
   -0.02940   1.00040   0.        0.        0.
   -0.57440  -0.23916   1.18120   0.        0.
   -1.07510  -1.18120   0.07610   1.87410   0.
   -0.81540  -1.54190  -1.05420   0.30160   2.39870


   CHECK THE ABOVE INVERSE, THEN INVERT DIRECTLY AND BY A
   BRUTE FORCE METHOD. PRINT THE RESULTS IN SUCH A WAY AS TO
   DISPLAY THE QUALITY OF THE RESULTS ON A SINGLE PAGE.
```

Aside from illustrating certain matrix operations this and the two pages immediately following provide a preview to the report writing capabilities of OMNITAB. The above problem statement was prepared via the computer by appropriate use of NOTE, SPACE, FORMAT, and TPRINT instructions. How this is done can be seen from the list of commands following the results. This feature is discussed more fully in chapter 11.

Returning to the matrix problem, note should be taken of the following: a) the published inverse suffers from either rounding or other errors and how this is confirmed by multiplying the matrix by its inverse; b) although the two methods employed in OMNITAB give the same result to six decimals, the direct method is more precise — an indication of this is given by how the zero is printed in the unit matrix. Here a 0. is a zero while a 0.0000000 is only some number smaller than 10E−7.

PAGE 2 OMNITAB PROBLEM 7 - 1

THE MATRICES BELOW ARE THE RESULT OF MULTIPLYING A BY ITS INVERSE. THEY SHOULD BE UNIT MATRICES.

1.0000	0.	0.	0.	0.
0.0294	0.9996	0.	0.	0.
0.4922	0.2023	0.8466	0.	0.
0.5722	0.6218	-0.0344	0.5336	0.
0.5032	0.6533	0.3764	-0.0671	0.4619

THE ABCVE IS THE ORIGINAL MATRIX A

1.00000	0.	0.	0.	0.	1.000000	-0.	-0.	0.	0.
-0.02940	1.00040	0.	0.	0.	0.000012	1.000000	-0.	0.	0.
-0.57440	-0.23916	1.18120	0.	0.	-0.000035	-0.000092	1.000004	0.	0.
-1.07510	-1.18120	0.07610	1.87410	0.	0.000005	-0.000013	-0.000026	1.000020	0.
-0.81540	-1.54190	-1.05420	0.30160	2.39870	-0.036705	-0.069404	-0.047438	0.013557	1.107960

THE ABCVE IS THE INVERSE GIVEN IN THE TEXT.

1.000000	0.000000	-0.000000	0.000000	0.000000	1.000000	0.000000	-0.000000	0.000000	0.000000
-0.029412	1.000400	-0.	-0.000000	-0.000000	0.000000	1.000000	-0.000000	-0.000000	-0.000000
-0.574356	-0.239051	1.181195	-0.000000	0.000000	-0.	-0.000000	1.000000	0.000000	0.000000
-1.075093	-1.181170	0.076149	1.874063	-0.000000	-0.000000	-0.000000	-0.000000	1.000000	-0.000000
-0.735952	-1.391728	-0.951488	0.272244	2.164971	-0.	-0.000000	-0.000000	0.	1.000000

THE ABCVE IS THE OMNITAB SOLUTION VIA SOLVE (BRUTE FORCE)

1.000000	0.	0.	-0.	-0.	1.000000	-0.	-0.	0.	0.
-0.029412	1.000400	-0.	0.	0.	0.000000	1.000000	-0.	0.	0.
-0.574356	-0.239051	1.181195	0.	0.	0.000000	0.000000	1.000000	0.	0.
-1.075093	-1.181170	0.076149	1.874063	-0.	0.000000	0.000000	-0.000000	1.000000	0.
-0.735952	-1.391727	-0.951488	0.272244	2.164971	0.000000	0.000000	0.000000	-0.000000	1.000000

THE ABCVE IS THE OMNITAB SOLUTION VIA INVERT.

```
OMNITAB   PROBLEM 7 - 1
READ 1 2 3 4 5
1.0 0. 0. 0. 0.
.0294 .9996 0. 0. 0.
.4922 .2023 .8466 0. 0.
.5722 .6218 -.0344 .5336 0.
.5032 .6533 .3764 -.0671 .4619
READ 11 12 13 14 15
1.0 0. 0. 0. 0.
-.0294 1.0004 0. 0. 0.
-.5744 -.23916 1.1812 0. 0.
-1.0751 -1.1812 .0761 1.8741 0.
-.8154 -1.5419 -1.0542 .3016 2.3987
NEWPAGE
SPACE 3
NOTE      CONSICER THE MATRIX
SPACE 2
FORMAT(1X,5F10.4)
TPRINT 1 2  3 4 5
SPACE 2
NOTE      ANC ITS INVERSE AS GIVEN IN A WELL KNOWN TEXT AS
SPACE 2
FORMAT(2X,5F10.5)
TPRINT 11 12 13 14 15
SPACE 2
NOTE      CHECK THE ABOVE INVERSE, THEN INVERT DIRECTLY AND BY A BRUTE
NOTE      FORCE METHOC. PRINT THE RESULTS IN SUCH A WAY AS TO DISPLAY
NOTE      THE QUALITY OF THE RESULTS ON A SINGLE PAGE.
ACC  1. 0.  42
MMULT 1 1 5 5 BY 1 11 5 1 21
FORMAT(1X,5F10.4)
TITLE1                                                     THE MATRICES BELO
TITLE3W ARE THE RESULT OF MULTIPLYING
TITLE2                                                     A BY ITS INVERSE.
TITLE4 THEY SHCULD BE UNIT MATRICES.
FPRINT 1 2 3 4 5
SPACE 1
NOTE      THE ABOVE IS THE ORIGINAL MATRIX  A
SPACE 2
FORMAT(2X,5F10.5, 2X,5F10.6)
TPRINT 11 12 13 14 15      21 22 23 24 25
SPACE 1
NOTE      THE ABOVE IS THE INVERSE GIVEN IN THE TEXT.
SPACE 2
DIAGCNALIZE 1. 31 35
BEGIN
SFIT   31 42 1 2 3 4 5
INCREMENT 1 BY 1 0 0 0 0 0  0
RESET 6
PRCMCTE 1 45 21
INCREMENT 4  BY 0 0 1
RESET 5
FINISH
REPEAT 1 6 5
MMULT  1 1 5 5 1 21 5 1 31
FORMAT(3X,5F10.6,1X,5F10.6)
TPRINT 21 22 23 24 25      31 32 33 34 35
SPACE 1
NOTE      THE ABOVE IS THE OMNITAB SOLUTION VIA SOLVE (BRUTE FORCE)
SPACE 2
INVERT 1 1 5 1 21
MMULT  1 1 5 5 1 21 5 1 31
TPRINT 21 22 23 24 25      31 32 33 34 35
SPACE 1
NOTE      THE ABOVE IS THE OMNITAB SOLUTION VIA INVERT.
       STOP
```

8. ARRAY OPERATIONS

Experience gained in incorporating the matrix operations into OMNITAB pointed clearly to the advantage to be gained in being able to handle entire arrays of numbers as easily as column vectors. In many calculations in thermodynamics and other field, operations must be performed on all the elements of an array. It seemed natural therefore to add the instruction ALOGE to take the logarithm of each element of an array. It also seemed natural to extend certain of the operations on matrices to a more general form. Thus instead of simply multiplying a matrix by a scalar factor or by a vector (a set of scalars) we have a provision for multiplying each element of a matrix by its corresponding element in another matrix.

Here also the requirement to define clearly the location of the array permits us to operate on any array and to store the result in any location of the work sheet. This feature is a great help in those problems in which the columns are short but which require more than the 46 columns; in effect provides an extension to the equivalent of 96 or even over 200 columns of appropriate length. Obviously the storage of many short columns throughout the work sheet puts more stringent requirements on the bookkeeping aspects of a calculation.

Special attention should be drawn to the instruction BLOCKTRANSFER which permits moving an entire block of data. One important application of this instruction is to move a block of data up to the top of the work sheet just prior to printing via the normal PRINT or FPRINT instruction. Note should also be taken of the APRINT instruction which prints the specified array without moving it.

8.1. Commands for Operating on Arrays°

1. ARAISE [A] IN ,, ++ R = ,, C = ,, TO [B] IN ,, ++
   START STORING IN ,, ++
   a. raises each element of the first matrix to the corresponding element of the second matrix.

   ARAISE [A] IN ,, ++, R = ,, C = ,, TO COL ++,
   START STORING IN ,, ++
   a. raises each row of the matrix to the power in the correspondong row of the designated column.

2. ALOGE OF [A] IN ,, ++, R = ,, C = ,,
   START STORING IN ,, ++
   a. takes the natural logarithm of each element of the array.

3. ALOGTEN OF [A] IN ,, ++, R = ,, C = ,,
   START STORING IN ,, ++
   a. takes the common logarithm of each element of the array.

4. ADIVIDE [A] IN ,, ++ R = ,, C = ,, BY $$,
   START STORING IN ,, ++
   a. divides each element of the array by a constant if $$ carries a decimal point.
   b. divides each row of the array by the value in the corresponding row of the designated column if $$ is a column number.

5. AMULT [A] IN ,, ++ R = ,, C = ,, BY [B] IN ,, ++,
   START STORING IN ,, ++
   a. multiplies each element of [A] by the corresponding element of [B].

6. MINEL OF [A] IN ,, ++ R = ,, C = ,, STORE IN COL ++
   a. finds the minimum element of the matrix, and stores this quantity as a vector in the designated column.
   b. stores the coordinates of the minimum value in row 1 of columns 45 and 46 (see table A).

7. MAXEL OF [A] IN ,, ++ R = ,, C = ,, STORE IN COL ++
   a. finds the maximum element of the matrix, and stores this quantity as a vector in the designated column.

- - - - - - - - - - -

° When a column is indicated in an instruction without a row designation, the column is operated upon starting with row one.

b. stores the coordinates of the minimum value in row 1 of columns 45 and 46 (see table A).

8. BLOCKTRANSFER [A] IN ,, ++ R = ,, C = ,, TO ,, ++
a. moves the matrix to the designated new location.
b. leave original matrix unchanged.

9. APRINT [A] IN ,, ++, R = ,, C = ,,
a. obeys the built-in format if C is less than or equal to 8 columns.
b. if C exceeds 8 columns the matrix is printed out by rows with 8 numbers per line of output until the row is finished.
c. each row of the matrix starts a new line in the printout.
d. does not start a new page as does PRINT and FPRINT.

10. VECMULT [A] IN ,, ++ R=,, C=,, BY $$ STORE IN ,,++
a. multiplies each element of the array by a constant if $$ carries a decimal point.
b. multiplies each row of the array by the value in the corresponding row of the column if $$ is a column number.

11. COALESCE ON FIRST COL OF [A] IN ,, ++ R = ,, C = ,, START STORING IN ,, ++
a. adds together the elements in each column of the array corresponding to identical elements in the first column.
b. unless the location of the first element of the result coincides with the location of the first element of the original array, the two arrays must not overlap.
c. the number of rows in the coalesced array is equal to the number of different values contained in the first column of the matrix.

12. COALESCE ON ** IN FIRST COL OF [A] IN ,, ++ R = ,, C=,, START STORING IN ,, ++
a. adds together the elements in each column of the array corresponding to ** in the first column.
b. if the designated value is not found, a row of zeros is stored in the designated location.
c. here the number of rows in the result is one.

13. ARRAYAVERAGE ON FIRST COL OF [A] IN ,, ++, R =,, C=,, START STORING IN ,, ++
a. averages the elements in each column of the array corresponding to identical elements in the first column.

b. unless the location of the first element of the result coincides with the location of the first element of the original array, the two arrays must not overlap.
c. the number of rows in the averaged array is equal to the number of different values contained in the first column of the matrix.

14. ARRAYAVERAGE ON ** IN FIRST COL OF [A] IN ,, ++
R = ,, C = ,, START STORING IN ,, ++
a. averages the elements in each column of the array corresponding to ** in the first column.
b. if the designated value is not found, a row of zeros is placed in the designated location.
c. here the number of rows in the result is one.

15. DUPLICATE ,, TIMES THE ARRAY IN ,, ++
R = ,, C = ,, START STORING IN ,, ++
a. duplicates an array of numbers as indicated vertically downward.
b. the storage location should not overlap the original array.
c. increases NRMAX accordingly.

8.2. Problems Involving Array Operations

Problem 8 − 1

Compute the tables of compressibility factors for deuterium from the relations:

$$Z = PV/RT = \exp[B(T)\rho + C(T)\rho^2]$$

where

$$B(T) = 0.0055298T^{-1/4} - 0.036040T^{-3/4} - 0.25878T^{-5/4}$$

and

$$C(T) = 0.00580\ T^{-3/2} - 0.0565\ T^{-2}$$

for $T = 20(2)50(5)100(10)250°K$

and $\rho = 20(20)100$ Amagats.

The OMNITAB instructions are as follows:

```
OMNITAB PROBLEM 8 - 1 COMPRESSIBILITY FACTOR OF DEUTERIUM
NOSUMMARY
GENERATE 20.(2.)50.(5.)100.(10.)250. IN 11
RAISE 11 TO -.25,MULT BY .0055298, STORE IN 12
RAISE 11 TO -.75,MULT BY -.036040, STORE IN 12
RAISE 11 TO -1.25, MULT BY -.25878, ADD TO 12
RAISE 11 TO -1.5 , MULT BY  .00580, ADD TO 13
RAISE 11 TO -2. POWER,MULT-.0565,13
RESET 0
READ 1,2,3,4,5
20. 40. 60. 80. 100.
2.7182818 2.7182818 2.7182818 2.7182818 2.7182818
DUPLICATE 49 TIMES 1,1,1,5 INTO 2,1
DUPLICATE 50 TIMES 2,1,1,5 INTO 51,1
RESET 50
SCALAR PROD 1,1,50,5, BY COL 12, START STORING 1,21
ARAISE 1,1,50,5 TO 2. ,STORE 1,1
SCALAR PROD 1,1,50,5 BY COL 13, STORE 1,1
MADD 1,1,50,5 TO 1,21, STORE 1,31
ARAISE 51,1,50,5  TO 1,31, 1,1
HEAD COL 11/    T DEG K
HEAD COL 12/         B
HEAD COL 13/          C
PRINT 11,12,13
FIXED 5
PRINT  11 1 2 3 4 5
       STOP
```

Problem 8 - 2

In certain operations it is necessary to merge two matrices so as to form a single matrix built-up of alternating columns from each of the original matrices. Let [A] be in columns 1 - 4 and [B] in columns 5 - 8. Assume the matrices are 20 rows long. Arrange this to end up with the desired array in columns 1 - 8.

The OMNITAB instructions to do this are:

Method 1.

```
PROMOTE 0 ROWS 1,11,2,13,3,15,4,17
PROMOTE 0 ROWS 5,12,6,14,7,16,8,18
BLOCKTRANS  1,11,20,8, INTO 1,1
```

Method 2.

```
DEMOTE BY 50 ROWS 1,1,2,3,3,5,4,7
RESET 20
DEMOTE BY 50 ROWS 5,2,6,4,7,6,8,8
PROMOTE ARRAY 50 ROWS
```

Method 3.

```
EXCHANGE COLS 1,11,2,13,3,15,4,17,5,12,6,14,7,16,8,18
BLOCKTRANSFER 1,11,20,8 INTO 1,1
```

If the restriction to having the result in column 1 - 8 were removed, the single exchange instruction above, would have solved the problem. Furthermore even with the above restriction, the operation can be carried out even more succinctly as follows:

```
EXCHANGE  2,5,5,3,4,6,6,7
```

9. MATHEMATICAL AND SPECIAL OPERATORS

Early attempts on our part to design a general-purpose computer program, dating back to 1959, centered around a series of specialized operators called adders, transformers, and compilers. These took one or more columns of numbers and operated on them via an incidence matrix of coefficients in the case of the function adder and transformer, and an incidence matrix of both coefficients and exponents in the case of the compiler.

Although the operators proved quite successful, they were discarded in the later stages of the development of OMNIFORM I program (see reference to NBS Technical Note 125 on page 1) for a single operator which seemed more attractive in view of the then limited core storage and programming capability.

The arrangement of OMNITAB subroutines into a system under IBSYS has removed the necessity of choosing between one or another method of handling a particular type of problem. Under the present system two or more methods can be made available especially if each method has its own domain of problems which it handles with singular efficiency. The nature and advantage of such operators can be appreciated better, perhaps, by considering the following example from thermodynamics.

The concentration, C_i, of species – molecules, for example – in a gaseous system under equilibrium conditions can be written in terms of the concentration α_j of a set of reference species – atoms – in the following way (at a fixed temperature and pressure):

$$C_i = k_i \prod_j \alpha_j^{r_{ij}}$$

where k_i is a given vector of constants, α_j is a vector of concentrations of the reference species and r_{ij} is an incidence matrix of exponents. It is most efficient if the vectors and matrices for such problems are read in as data rather than incorporated in the instruction set.

The solution of this problem for 25 values of k_i, employing operators, would go as follows:

```
OMNITAB
SET IN COL 20 THE FOLLOWING VALUES OF KI
     k1 k2 ............one or more cards follow
READ COL 1, 2, 3, 4, 5
      α1, α2, α3....... 1 card
READ COL 11, 12, 13, 14, 15
     r11   r12   r13   r14   r15  \
     r21    .                 .    |
      .           .           .    |
      .                 .     .    > rij matrix 25 rows
      .                       .    |
      .                       .    |
     r25,1  .     .     .  r25,5  /

DUPLICATE 24 TIMES 1,1,1,5 INTO 2,1
ARAISE 1,1,25,5 TO R IN 1,11, STORE IN 1,21
PRODUCT COLS 20 THRU 25, STORE IN 26
PRINT COL 26,20
STOP
```

The power of the method should be evident after observing that once the data are entered, the entire calculation is achieved in only three instructions.

The operator EXPAND, which raises a vector of values to a succession of powers in uniform increments, is of value in generating the vectors for curve fitting when the functional form is a polynomial with some of the terms missing.

Finally the evaluation of series expansions in terms of the variables x, or x log x, or x sin x, or even in terms containing $x^b e^{-cx}$ is greatly facilitated by suitably defined operators. In the next section we give 15 such operators. The first of these evaluates

$$y_i = a_j \sum_j X_i^{b_j}$$

from the single instruction

```
FORMULA  1, X IN COL 1, PARAMETERS FOR 10 TERMS IN COL 2,
                    STORE IN COL  3
```

This instruction assumes that the parameters (coefficients and exponents) are stored in the designated columns as follows.

$a_1, b_1, a_2, b_2, \quad \ldots \quad a_n, b_n$

There are no restrictions on the values of a_j or b_j except that the resultant values must be real.

9.1. Mathematical Operators

1. FORMULA ,, OF $$, PARAMETERS FOR ,, TERMS IN ++, STORE IN COL ++
   a. this instruction substitutes the parameters (coefficients and exponents) a_j, b_j, c_j, d_j, into the designated formula having the designated number of terms.
   b. the parameters are stored in the order a_1, b_1, a_2, b_2, etc or a_1, b_1, c_1, d_1, a_2, b_2, c_2, d_2, a_3, etc as applicable.
   c. the formulas are as follows:

$$\text{Formula 1} = \sum_j a_j X^{b_j}$$

$$\text{Formula 2} = \sum_j a_j X^{b_j} \log X$$

$$\text{Formula 3} = \sum_j a_j X^{b_j} \ln X$$

$$\text{Formula 4} = \sum_j a_j X^{b_j} \exp(c_j X^{d_j})$$

$$\text{Formula 5} = \sum_j (a_j/b_j) \; X^{c_j} (\sin X)^{d_j}$$

$$\text{Formula 6} = \sum_j a_j X^{b_j} (\sin c_j X)^{d_j}$$

$$\text{Formula 7} = \sum_j (a_j/b_j) \; X^{c_j} (\cos X)^{d_j}$$

Formula 8 $= \sum_j a_j X^{b_j} (\cos c_j X)^{d_j}$

Formula 9 $= \sum_j (a_j / b_j) \; X^{c_j} (\tan X)^{d_j}$

Formula 10 $= \sum_j a_j X^{b_j} (\tan c_j X)^{d_j}$

Formula 11 $= \sum_j a_j (\sinh X)^{b_j}$

Formula 12 $= \sum_j a_j (\cosh X)^{b_j}$

Formula 13 $= \sum_j a_j (\tanh X)^{b_j}$

Formula 14 $= \sum_j a_j \; \pi/2 + b_j \arcsin X$

Formula 15 $= \sum_j a_j \; \pi/2 + b_j \arctan X$

2. EXPAND $$ TO ,, POWER IN INTERVALS OF ,, START STORING IN COL ++
   a. provides for exponentiation of a column of numbers to integral values of the exponent in uniform integral steps.
   b. for example EXPAND COL 1, TO 8, 2, 5 generates the 2nd, 4th, 6th, and 8th powers of the numbers in column 1 and stores these results in column 5 et seq.
   c. the exponentiation is carried out by multiplication.

3. EXPAND $$ TO ** POWER IN INTERVALS OF ** START STORING IN COL ++
   a. same as above but for nonintegral powers.
   b. for example EXPAND COL 1, TO 4.5, 0.5, 5 generates the 0.5, 1.0, 1.5 ..... 4.5 powers of the values in column 1 and stores these in column 5 et seq.

4. PRODUCT OF COL ++ TO ++, STORE IN ++
   a. provides for the product (row by row) of the elements in each row the contiguous designated columns and stores the resulting vector as indicated.

$$b_i = \prod_j a_{ij}$$

5. PRODUCT OF COL ++, ++, ++, ... STORE IN ++
   a. provides for the product (row by row) of the elements in the designated columns and storing of the resulting vector.
   b. there must be 3 or more columns in the product.

9.2. The Utility of Special Operators

The operators discussed earlier in this chapter, while very useful and efficient tools, are not really the ultimate that can be achieved in a problem-oriented set of instructions. Their principal merit is that since they are mathematical, they have a fairly wide application. In this section we illustrate the utility of a series of instructions which we consider to be the ultimate point to which one can with profit push a problem-oriented instruction set (We resist the temptation to use the word "language".). Before discussing the macro instructions which have been programmed in OMNITAB to handle certain recurring calculations in thermodynamics, we think it profitable to illustrate the motivation for such macro instructions with a more familiar problem from elementary mathematics.

Consider an array of numbers arrayed in 3 columns which give on successive rows the length of the sides of a number of triangles. It is required to find the areas of each triangle from the well-known formula

$$A = \sqrt{s(s-a)(s-b)(s-c)}$$

where s is the semiperimeter. We leave it to the reader to confirm the fact that the problem can be solved in OMNITAB via about 9 simple instructions exclusive of the input and output.

Can one conclude that the program as presently arranged is quite suitable to handle problems of this kind? Our view is that it is indeed suitable if the problem arises only rarely. If the problem is a recurring one as it would be in some problems in surveying, then it is a waste of valuable time to be forced to solve the problem in detail with 9 or so instructions. It makes more sense to incorporate in the program a short routine which responds to the instruction

AREA OF TRIANGLES WHOSE SIDES ARE IN COLS 1,2,3,
STORE IN COL 4

The program COGO discussed in chapter 2 is composed of approximately 100 such commands applicable to the solution of coordinate geometry problems related to civil engineering. It is, in our view, an excellent model of the successful application of a problem-oriented interpretive program which has been pushed to an ultimate level. We are convinced that in such intelligent extensions into other specialized areas lies the key to more widespread and efficient use of computers in the future. This is especially true, we feel, since such programs as OMNITAB and COGO are ideally suited for remote operations via teletype or other telecommunication channels.

In the following section, we describe a half dozen macro instructions in thermodynamics which have been finished in time for inclusion in this Handbook. They represent only a part of what is planned for a truly pushbutton capability in thermodynamics. They should, however, suffice to point the way in other areas of pure and applied science.

9.3. Special Operators for Thermodynamics

1. CTOF OF $$, STORE IN COL ++
   a. converts temperatures from the Celsius scale to the Fahrenheit scale.

   $$^\circ F = 1.8^\circ C + 32$$

2. FTOC OF $$, STORE IN COL ++
   a. converts temperatures from the Fahrenheit scale to the Celsius scale.

   $$^\circ C = (^\circ F - 32)/1.8$$

3. ATOMIC MASS TABLE STORE IN ++
   a. stores as designated, the atomic masses of the elements in order of Z number from 1 for Hydrogen to 101 for Mendelevium.
   b. changes NRMAX to 101.
   c. atomic weights are taken from T. Batuecas and J. Gueron, Preliminary Report of the Commission on Atomic Weights. Reprint from Information Bulletin No. 14b, International Union of Pure and Applied Chemistry.

4. MOLWT Z=,, AMOUNT =,, Z=,, AMOUNT =,, ...
   STORE SUM IN COL ++
   a. computes molecular weight of the molecule indicated. For example water (H_2O):

   MOLWT 1,2,8,1 STORE IN COL 5

b. all atomic masses are included up to and including Nobelium (Z=102).
c. atomic weights are taken from T. Batuecas and J. Gueron, Preliminary Report of the Commission on Atomic Weights.

5. EINSTEIN,TEMP IN $$,VIBRATIONAL FREQ(IN WAVE NO) IN $$
or
EINSTEIN,TEMP IN $$,FREQUENCY IN $$,GAS CONSTANT R=,,
a. computes and stores automatically as indicated the contributions to the thermodynamic properties of a harmonic oscillator in one degree of freedom for desired temperatures and vibratonal frequencies as follows:

	Storage Column
(E) wave numbers	40
(T) temperatures	41
$-(F^\circ - E_0^\circ)/RT = -\ln(1-\exp(-x))$	42
$(H^\circ - E_0^\circ)/RT = x\exp(-x)/(1-\exp(-x))$	43
$S^\circ/R = -(F^\circ - E_0^\circ)/RT + (H^\circ - E_0^\circ)/RT$	44
$C_p^\circ/R = x^2\exp(-x)/(1-\exp(-x))^2$	45
$(H^\circ - E_0^\circ)/R$	46

where $x = hcE/kT$ and $hc/k = 1.43879$

b. if a value of R is given, the stored thermal functions are multiplied by R.
c. zeros are entered for all of the functions when a zero is encountered in the temperature column.
d. negative temperatures and wave numbers are not allowed.
e. input can come from any column.

6. PFTRANS TEMP IN $$, MOLECULAR WEIGHT (M) IN $$
a. computes and stores automatically as indicated the translational contributations to the thermal functions.

	Storage Column
(T) temperatures	41
$-(F^\circ - E_0^\circ)/RT = 2.5 \ln T + 1.5 \ln M - 3.66495$	42
$(H^\circ - E_0^\circ)/RT = 2.5$	43
$S^\circ/R = (H^\circ - E_0^\circ)/RT - (F^\circ - E_0^\circ)/RT$	44
$C_p^\circ/R = 2.5$	45
$(H^\circ - E_0^\circ)/R$	46

b. zeros are entered for all of the functions when a zero is encountered in the temperature column.
c. negative temperatures are not allowed.
d. input can come from any column.

PFATOM TEMP IN $$, MOLECULAR WEIGHT IN $$, WAVE NO IN ++, G IN ++
a. computes and stores automatically as indicated the contributations to the thermal functions.

	Storage Column
(T) temperatures	41
$-(F^\circ - E_0^\circ)/RT = 2.5 \ln T + 1.5 \ln M - 3.66495 + \ln Q^0$	42
$(H^\circ - E_0^\circ)/RT = 2.5 + Q^1/Q^0$	43
$S^\circ/R = (H^\circ - E_0^\circ)/RT - (F^\circ - E_0^\circ)/RT$	44
$C_p^\circ/R = 2.5 + Q^2/Q^0 - (Q^1/Q^0)^2$	45
$(H^\circ - E_0^\circ)/R$	46

where

$$Q^0 = \sum_i g_i \exp(-hcE_i/kT)$$

After the numbers are entered in the columns as indicated, the calculation is carried out as before by the two instructions

```
SIN 1, 3
COS 2, 3, 4
```

We address ourselves now to the problem of how to enter the arrangement of variables as indicated above. The generation of the required values of φ is achieved in instructions

```
GENERATE 0.1(0.1)1.0 IN 1
DUPLICATE 9 TIMES, ARRAY IN 1,1,10,1
     START STORING IN  11,1
```

The generation of the values of φ presents a number of alternatives. We discuss a few of these to illustrate some of the features of the OMNITAB commands.

Method A via repeat mode:

```
RESET LENGTH TO 100
ADD 2.0, 0., STORE IN 1
BEGIN STORING INSTRUCTIONS
RESET LENGTH TO 90
INDEX INST 1 BY -10
SUB 0.2, 1, 1
FINISH STORING INST
REPEAT INST 1 THRU 3, 9 TIMES
```

Method B via matrix operations:

```
READ 1,2,3,4,5,6,7,8,9,10,
    .2,.2,.2,.2,.2,.2,.2,.2,.2,.2
    .4,.4,.4,.4,.4,.4,.4,.4,.4,.4
    .6  .                    .6
    .8     .
   1.0        .
   1.2           .
    .               .
   1.8                    .
   2.0  .  .  .  .  .  .  .  .  2.0
VECMAT IN 1,1,10,10 INTO COL 2
```

One advantage of using method B is that a third variable could now be handled using the repeat mode. Thus the function of three variables

$$y = \omega \sin \theta \cos \varphi$$

could be handled as follows:

```
CMNITAB        10/1/64
GENERATE 0.1(0.1)1.0 IN 40
DUPLICATE 9 TIMES ARRAY IN 1,40,10, BY 1, STORE IN 11,40
READ 1,2,3,4,5,6,7,8,9,10
.1 .1 .1 .1 .1 .1 .1 .1 .1 .1
.2 .2 .2 .2 .2 .2 .2 .2 .2 .2
.3 .3 .3 .3 .3 .3 .3 .3 .3 .3
.4 .4 .4 .4 .4 .4 .4 .4 .4 .4
.5 .5 .5 .5 .5 .5 .5 .5 .5 .5
.6 .6 .6 .6 .6 .6 .6 .6 .6 .6
.7 .7 .7 .7 .7 .7 .7 .7 .7 .7
.8 .8 .8 .8 .8 .8 .8 .8 .8 .8
1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
VECMAT 1,1,10,10, STORE IN 1   12
SIN 40,41
COS 12,13
MULT 41,13,14
      BEGIN STORING INSTRUCTIONS
ADD 0.1 TO COL 31, STORE IN 31     (OMEGA)
MULT 31,14,21
INDEX INST 2 BY 1
     FINISH STORING INSTRUCTIONS
REPEAT INST 1,3,10 TIMES
FORMAT (1X,2F5.1,5F9.5)
FPRINT 40,12,21,22,23,24,25
FPRINT 40,12,26,27,28,29,30
       STOP
```

The results for this calculation are given on the following page.

We leave it to the reader to devise a set of instructions for this problem using the MTRANS and VECMULT instructions without recourse to the repeat mode.

PAGE 1 OMNITAB 10/1/64

θ	φ	ω = .1	ω = .2	ω = .3	ω = .4	ω = .5
0.1	0.1	0.00993	0.01987	0.02980	0.03973	0.04967
0.2	0.1	0.01977	0.03954	0.05930	0.07907	0.09884
[illegible]	[illegible]	0.02940	0.05881	0.08821	0.11762	0.14702
			[illegible]	0.11624	0.15499	0.19374
						[illegible]

PAGE 2 OMNITAB 10/1/64

θ	φ	ω = .1	ω = .2	ω = .3	ω = .4	ω = .5
0.1	0.6	0.00824	0.01648	0.02472	0.03296	0.04120
0.2	0.6	0.01640	0.03279	0.04919	0.06559	0.08198
0.3	0.6	0.02439	0.04878	0.07317	0.09756	0.12195
0.4	0.6	0.03214	0.06428	0.09642	[illegible]	
0.5	0.6	0.03957	0.07914	[illegible]		
0.6	0.6	[illegible]				

PAGE 3 OMNITAB 10/1/64

θ	φ	ω = .6	ω = .7	ω = .8	ω = .9	ω = 1.
				0.07947	0.08940	0.09933
		0.05960	0.06953	0.15814	0.17791	0.19768
0.1	0.1	0.11861	0.13837	0.23524	0.26464	0.29404
0.2	0.1	0.17643	0.20583	0.30998	0.34873	0.38747
0.3	0.1	[illegible]	0.27123	[illegible]	0.42933	0.47703

PAGE 4 OMNITAB 10/1/64

θ	φ	ω = .6	ω = .7	ω = .8	ω = .9	ω = 1.
0.1	0.6	0.04944	0.05768	0.06592	0.07416	0.08240
0.2	0.6	0.09838	0.11478	0.13118	0.14757	0.16397
0.3	0.6	0.14634	0.17073	0.19512	0.21951	0.24390
0.4	0.6	0.19284	0.22498	0.25712	0.28926	0.32140
0.5	0.6	0.23741	0.27698	0.31655	0.35612	0.39569
0.6	0.6	0.27961	0.32621	0.37282	0.41942	0.46602
0.7	0.6	0.31902	0.37219	0.42536	0.47853	0.53170
0.8	0.6	0.35524	0.41444	0.47365	0.53285	0.59206
0.9	0.6	0.38790	0.45256	0.51721	0.58186	0.64651
1.0	0.6	0.41670	0.48615	0.55560	0.62505	0.69450
0.1	0.7	0.04581	0.05345	0.06109	0.06872	0.07636
0.2	0.7	0.09117	0.10637	0.12156	0.13676	0.15195
0.3	0.7	0.13562	0.15822	0.18082	0.20342	0.22603
0.4	0.7	0.17871	0.20849	0.23827	0.26806	0.29784
0.5	0.7	0.22001	0.25668	0.29335	0.33002	0.36668

Figure 9-1. Results of three argument calculation. The headings have been inserted by hand.

9.5. Problems Solved Via Operators

Problem 9 – 1

A dramatic example of the power of certain operators is afforded by the solution of a problem arising in certain crystal calculations where it is necessary to evaluate 22 functions of the following form:

Ref No.	s
1	$2a + a^2$
2	$2b + b^2$
11	$2b + a^2 + 2b^2 - 2ab$
14	$2c + .5a^2 + 1.5c^2 - ac$
16	$-2a + 4d + a^2 + 4d^2 - 4ad$
19	$-a + 2e + f + .5a^2 + 2e^2 + .5f^2 - 2ae$
38	$2c + 2b^2 + 3c^2 - 4bc$
41	$2d + b^2 + 2d^2 - 2bd$
44	$-2b + 4e + 2b^2 + 4e^2 + f^2 - 4be - 2bf$
52	$-b + 2g + h + b^2 + 2g^2 + h^2 - 2bg - bh$
59	$-2b + 4i + b^2 + 4i^2 - 4bi$
72	$2e + 1.5c^2 + 2e^2 + .5f^2 - 2ce - cf$
76	$-2c + 4g + 3c^2 + 4g^2 + 2h^2 - 4cg - 4ch$
85	$-2c + 1.5c^2$
90	$2f + 4d^2 + 4e^2 + f^2 - 8de$
95	$2h + 2d^2 + 2g^2 + h^2 - 4dg$
102	$-4d + 6j + 4d^2 + 9j^2 - 12dj$
105	$-2d + 2d^2$
106	$2f + f^2$
107	$4e - 2f + 4e^2 + f^2 - 4ef$
121	$2h + 4e^2 + f^2 + 4g^2 + 2h^2 - 8eg - 2fh$
141	$-f + 2e^2 + .5f^2$

The evaluation of the 22 functions for a set of values of the 10 variables a, b, c .... j. can be achieved in OMNITAB with only four instructions (exclusive of the data input and print instructions). To achieve this brevity of commands it is necessary to recognize that each term of each equation is a special case of a general term having 11 factors, one for the coefficient and the others for the 10 variables, each with its own exponent. The variables which are missing from any term would have zero for an exponent, while those which are present would have a suitable nonzero exponent. Symbolically the i equations can be represented by

$$F_i = \sum_j \alpha_{ij} \prod_{k=1}^{10} X_k^{r_{ijk}}$$

We simplify the problem somewhat by calculating each term of the entire set separately and identify each term with a formula number which is later used to sort out those belonging together. In this case the terms can be represented simply by

$$\prod_{k=1}^{10} X_k^{r_{jk}} ,$$

where the X_k are the values a, b, c .... j, and the r_{jk} are 10 exponents for each of the j terms in the set.

The problem is specified as follows:

a. by reading in a matrix of 12 columns containing for each term an identification number tying the term to a particular line, its coefficient, and 10 exponents, r_k, representing the incidence of each of the variables in the term.

b. by reading in a row of values for the 10 variables X_k = a, b, c .... j.

The actual calculations are achieved by:

a. duplicating the X_k row 100 times and making it available on each row of the work sheet.

b. raising each of the values in the X_k matrix to its corresponding exponent.

c. taking the product of the 11 columns comprising the resulting matrix and the vector of the coefficients.

d. using the identifier to coalesce (add together) terms belonging to the same equation.

The OMNITAB instructions are as follows:

```
OMNITAB PROBLEM  9 - 1
READ 19 20 1 2 3 4 5 6 7 8 9 10
1.  2. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
1. 1. 2.
2 2 0 1
2 1 0 2
11 2 0 1
11 1 2  0
11 2 0 2
 11 -2 1 1
14 2 0 0 1
14 .5 2     0 0
14 1.5 0 0 2
14 -1 1 0 1
16 -2 1 0 0
16 4 0 0 0 1
16 1 2 0 0 0
16 4 0 0 0 2
16 -4 1 0  0 1
19 -1 1 0 0 0
19 2 0 0 0 0 1
19  1 0 0 0 0 0 1
19 .5 2 0 0 0 0 0
19 2 0 0 0 0 2
19 .5 0 0 0 0 0 2
19 -2 1 0 0 0 1 0
38 2 0 0 1 0 0
38 2 0 2 0
38 3 0 0 2
38 -4 0 1 1
41 2 0 0 0  1
41 2 0 0 0 2
41 1 0 2 0 0
41 -2 0 1 0 1
44 -2 0 1 0 0
44 4 0 0 0 0 1
44 2 0 2 0 0 0
44 4 0 0 0 0 2
44 1 0 0 0 0 0 2
44 -4 0 1 0 0 1 0
44 -2 0 1 0 0 0 1
 .
 .
 .
```

```
141 -1 0 0 0 0 0 1 0 0
141 2 0 0 0 0 0 2 0
141 .5 0 0 0 0 0 2

READ 21 22 23 24 25
0.05616 0.01391 -0.00179 0.00705 0.00362

READ 26 27 28 29 30
0.00667 0.00107 0.00011 0.00193 0.00072

DUPLICATE 100 TIMES ROW 1 COL 21 R=1 C=10 START STORING IN ROW 2 COL 21
ARAISE ROW 1 CCL 21 R=101 C=10 TO ROW 1 COL 1 START STORING IN 1,21

PRCDUCTS OF COLS 20 THRU 30, STORE IN COL 20
COALESCE CN THE FIRST COL THE MATRIX IN 1,19, R=101,C=2,STORE IN 1,32
NOSUMMARY
RESET 22
HEAD 32/ REF NC.
HEAD CCL 33/     S
PRINT 32 33
      STOP
```

* * * **RESULTS** * * *

```
PAGE   1      OMNITAB PROBLEM  9 - 1

  REF NO.              S

 1.000000E 00   1.154739E-01
 2.000000E 00   2.801349E-02
 1.100000E 01   2.979855E-02
 1.400000E 01  -1.897694E-03
 1.600000E 01  -8.235096E-02
 1.900000E 01  -4.103117E-02
 3.800000E 01  -3.083816E-03
 4.100000E 01   1.419676E-02
 4.400000E 01  -1.324309E-02
 5.200000E 01  -1.149551E-02

 5.900000E 01  -1.999900E-02
 7.200000E 01   7.318159E-03
 7.600000E 01   7.882665E-03
 8.500000E 01   3.584806E-03
 9.000000E 01   1.343155E-02
 9.500000E 01   2.915329E-04
 1.020000E 02  -2.373744E-02
 1.050000E 02  -1.400060E-02
 1.060000E 02   1.338449E-02
 1.070000E 02   1.140326E-03

 1.210000E 02   2.890557E-04
 1.410000E 02  -6.621547E-03
```

Problem 9 – 2

Compute tables of compressibility factors for hydrogen from the relations:

$$Z = 1 + B \quad + [(1/2)B + C]\rho^2 + [(1/6)B^3 + BC]\rho^3$$

$$+[(1/24)B^4 + (1/2)C^2 + (1/2)B^2C]\rho^4 .$$

where $B = 0.0055478\ T^{-1/4} - 0.036877T^{-3/4} - 0.22004T^{-5/4}$

and $C = 0.004788\ T^{-3/2} - 0.04053\ T^{-2}$

for $T = 210°(10°)600°K$

and $\rho = 100(100)500$ Amagats.

The OMNITAB instructions are as follows:

```
OMNITAB PROBLEM 9 - 2 COMPRESSIBILITY FACTORS FOR HYDROGEN
NOSUMMARY
GENERATE 210.(10.)600. IN 1
RAISE 1 TO -.25 .0055478 ,2
RAISE 1 -.75 -.036877 2
RAISE 1 -1.25 -0.22004 2
RAISE 1 -1.5 .004788 3
RAISE 1 -2. -0.04053 3
READ 11 12 13 14 15
100.,200.,300.,400.,500.
DUPLICATE 49 1 11 1 5  INTO 2 11
ADD 0. 1. 41
ADD 0. 2 42
ADD 3 0. 43
MULT 2 BY 2 BY .5     43
MULT 2 BY 3 44
RAISE 2 TO 3. POWER MULT BY .16666666 ADD TO 44
MULT .5 3 4
MULT 3 BY 4 45
RAISE 2 TO 4. POWER MULT BY .04166667 ADD 45
MULT 2 2 4 45
ADD 1. 0. 20
BEGIN
EXPAND COL 11 TO 4TH POWER IN STEPS OF 1 START STORING IN 21
INCREMENT 1 BY 1 0 0 0
AMULT 1 20 50 5 BY 1 41      1 26
ROWSUM COLS 26 27 28 29 30 STORE 31
INDEX    4 BY 1
FINISH
REPEAT 1 5 5
HEAD COL 1/      T
HEAD 31/      100.
HEAD 32/      200.
HEAD 33/      300.
HEAD 34/      400.
HEAD 35/      500.
FIXED 5
PRINT 1 31 32 33 34 35 1
       STOP
```

PAGE 1 OMNITAB PROBLEM 9 - 2 COMPRESSIBILITY FACTORS FOR HYDROGEN

T	100.	200.	300.	400.	500.	T
210.00000	1.05961	1.13753	1.23699	1.36196	1.51715	210.00000
220.00000	1.06165	1.14134	1.24231	1.36849	1.52450	220.00000
230.00000	1.06345	1.14468	1.24691	1.37399	1.53049	230.00000
240.00000	1.06505	1.14762	1.25087	1.37863	1.53535	240.00000
250.00000	1.06647	1.15020	1.25431	1.38255	1.53928	250.00000
260.00000	1.06775	1.15249	1.25730	1.38585	1.54244	260.00000
270.00000	1.06889	1.15452	1.25990	1.38865	1.54495	270.00000
280.00000	1.06992	1.15632	1.26216	1.39100	1.54691	280.00000
290.00000	1.07085	1.15792	1.26414	1.39297	1.54842	290.00000
300.00000	1.07168	1.15935	1.26587	1.39462	1.54953	300.00000
310.00000	1.07244	1.16063	1.26737	1.39599	1.55032	310.00000
320.00000	1.07313	1.16177	1.26868	1.39712	1.55082	320.00000
330.00000	1.07375	1.16279	1.26982	1.39804	1.55109	330.00000
340.00000	1.07431	1.16370	1.27081	1.39878	1.55114	340.00000
350.00000	1.07483	1.16451	1.27167	1.39935	1.55103	350.00000
360.00000	1.07529	1.16524	1.27241	1.39978	1.55076	360.00000
370.00000	1.07572	1.16590	1.27305	1.40009	1.55036	370.00000
380.00000	1.07610	1.16648	1.27359	1.40029	1.54984	380.00000
390.00000	1.07646	1.16700	1.27404	1.40039	1.54924	390.00000
400.00000	1.07678	1.16746	1.27443	1.40041	1.54854	400.00000
410.00000	1.07707	1.16787	1.27474	1.40036	1.54778	410.00000
420.00000	1.07734	1.16824	1.27499	1.40023	1.54696	420.00000
430.00000	1.07758	1.16856	1.27519	1.40005	1.54608	430.00000
440.00000	1.07780	1.16884	1.27534	1.39982	1.54516	440.00000
450.00000	1.07800	1.16909	1.27544	1.39954	1.54419	450.00000
460.00000	1.07818	1.16931	1.27550	1.39922	1.54320	460.00000
470.00000	1.07834	1.16949	1.27553	1.39886	1.54218	470.00000
480.00000	1.07849	1.16965	1.27552	1.39846	1.54113	480.00000
490.00000	1.07863	1.16978	1.27548	1.39804	1.54007	490.00000
500.00000	1.07875	1.16989	1.27542	1.39760	1.53898	500.00000

Problem 9 - 3

Compute the log Y from the formula given in problem 6-8 using operators.

The OMNITAB instructions are as follows:

```
OMNITAB   PROBLEM 9 - 3
NOSUMMARY
GENERATE 1. 1. 20. IN 2
DIVIDE 1. BY 2 STORE 3
SET IN COL 4
1. -1. 1 -1. 1 -1. 1 -1. 1 -1. 1 -1. 1 -1. 1 -1. 1 -1. 1 -1.
MULTIPLY 4 3 4
INSERT IN 4 FROM 2 EVERY 2ND ROW STARTING AS 2ND ROW STORE 5
PRINT 2 4 5
GENERATE .01 .01 0.5 IN 2
ADD 1., 2, 1
FORMULA 1, X IN 2, PARAMETERS FOR 20 TERMS IN 5, STORE 6
LOGE 1 7
PRINT 2 5 6 7
      STOP
```

We leave it as an exercise for the reader to devise other schemes for generating the coefficients and their powers.

Problem 9 − 4

Compute the thermodynamic properties for the proton and for the hydrogen atom, given a set of energy levels and their statistical weights for the atom.

The OMNITAB instructions are as follows:

```
OMNITAB  PROBLEM 9 - 4
NOSUMMARY
SET 10
       DEC 040,0.,2.,82259.,8.,97492.,18.
       DEC 102823.,32.,105291.05,50.,106632.,72.
       DEC 107440.432,98.,107965.039,128.,108324.710,162.
       DEC 108581.979,200.,108772.329,242.,108917.109,288.
       DEC 109029.779,338.,109119.180,392.,109191.300,450.
       DEC 109250.329,512.,109299.250,578.,109340.250,648.
       DEC 109374.939,722.,109404.569,800.,109430.060,882.
       DEC 109452.149,968.,109471.428,1058.,109488.346,1152.
       DEC 109503.273,1250.,109516.513,1352.,109528.309,1458.
       DEC 109538.862,1568.,109548.345,1682.,109556.894,1800.
       DEC 109564.629,1922.,109571.650,2048.,109578.044,2178.
       DEC 109583.881,2312.,109589.225,2450.,109594.130,2592.
       DEC 109598.643,2738.,109602.804,2888.,109606.648,3042.
       DEC 109610.210,3200.
SEPARATE IN 10 EVERY 2ND ROW STARTING WITH THE 2ND ROW STORE IN 11
SEPARATE IN 10 EVERY 2ND ROW STARTING WITH THE 3RD ROW STORE IN 12
GENERATE 100. 100. 10000. IN 3
PFTRANS 3  1.0080

PRINT 41 42 43 44 45 46
PFATOM TEMP IN 3,MOLWT IS 1.0080 FREQUENCES IN 11, G IN 12
PRINT 41 42 43 44 45 46
      STOP
```

The SEPARATE instructions are used because PFATOM calls for the energy levels and their statistical weights in separate columns and the data had been previously punched in the form n, E_1, g_1, E_2, g_2, E_3, g_3 ··· The special program, for which these data represent the input, required a notification of the number of energy levels to be used (here n=40). This is not necessary in OMNITAB. The DEC required by the special program is ignored in the OMNITAB scan of the data.

Problem 9 − 5

```
CMNITAB   PROBLEM 9 - 5
TITLE1  CALCULATION OF EINSTEIN FUNCTIONS USING THE
TITLE2                 OPERATOR INSTRUCTION
NOSUMMARY
GENERATE 1.01(.01)1.5 IN COL 1
ADD  0. TO 1.43879 STORE IN COL 2
EINSTEIN TEMP IN 2, FREQUENCY IN 1
FIXED   5
HEAD COL 1/      X
HEAD COL 42/    -G
HEAD COL 43/     H
HEAD COL 44/     S
HEAD COL 45/  CSUBP
PRINT 1,42,43,44,45
      STOP
```

The results from the above are identical with those given for problem 3−1 in section 3.9. Normally the instruction EINSTEIN carries out the calculation for pairs of temperatures and frequencies, rather than for the single valued argument $x = hcE/kT$. The instruction ADD 0. TO 1.43879 STORE IN COL 2 was introduced above in order to reproduce the results of problem 3−1.

10. FUNDAMENTAL PHYSICAL CONSTANTS

Until now it has been emphasized that the OMNITAB program scans each card and extracts from it the first word and the integers or numbers on the card – ignoring the intervening words. Actually the scan routine can extract more than the first word from the card, provided that the other word is preceded by an asterisk * and followed by an asterisk.

The purpose of this provision is to enable the inclusion in the program of certain fundamental physical constants, so that the user need only ask for them by name. Thus the quotient h/k (h = Plancks Constant, 6.6256E–34; k=Boltzman constant = 1.38054E–23) can be obtained by the following OMNITAB instruction:

DIVIDE *PLANCK* BY *BOLTZMAN*, STORE IN COL 2

This provision enables the user to employ an up-to-date set of constants for his calculation simply by calling for them by name. It is, however, necessary to become familiar with the OMNITAB acronyms for the various constants. The acronyms are given below for the fundamental constants and for certain conversion factors given in Tables 2 and 3 of the recent NBS release entitled "New Values for the Physical Constant ." (NBS Technical News Bulletin, Vol 47, October 1963.)

Unless otherwise specified the program will use the constants in the MKSA system. If CGS units are desired, they can be had by using the instruction CGS prior to asking for a constant. These units will be supplied in the system until the end of the problem or until the command MKSA is introduced. The use of either of these instructions will provide a one page printout (See section 10.2) of the above cited physical constants unless such a listing has been suppressed by the NOSUMMARY instruction.

Because the program sees only the first six characters, it will accept as legal such words as *AVOGADRO*, *THOMSON*, *BOHRADIUS*, *BOLTZMAN*, etc.

10.1. Acronyms for the Physical Constants

SOLIVA	Speed of Light in Vacuum
ELECHA	Elementary Charge
AVOGAD	Avagadro Constant
ELECMA	Electron Rest Mass
PROMAS	Proton Rest Mass
NEUMAS	Neutron Rest Mass
FARADA	Faraday Constant
PLANCK	Planck Constant
FISTCO	Fine Structure Constant
EOVERM	Charge to Mass Ratio For Electron
HOVERE	Quantum-Charge Ratio
COWAEL	Compton Wavelength of Electron
COWAPR	Compton Wavelength of Proton
RYDBER	Rydberg Constant
BOHRAD	Bohr Radius
ELECRA	Electron Radius
THOMSO	Thomson Cross Section
GY/MAP	Gyromagnetic Ratio of Proton
BOHRMA	Bohr Magneton
NUCMAG	Nuclear Magneton
PROMOM	Proton Moment
ZEEMAN	Zeeman Splitting Constant
RJ/KMO	Gas Constant R in J K^{-1} $mole^{-1}$
RC/KMO	Gas Constant R in Calories K^{-1} $mole^{-1}$
VSUBZE	Normal Volume Perfect Gas
BOLTZM	Boltzmann Constant
CONE	First Radiation Constant
CTWO	Second Radiation Constant
WIEN	Wien Displacement Constant
STEFBO	Stefan-Boltzmann Constant
GRAVCO	Gravitational Constant

10.2. Acronyms for Energy Conversion Factors

J/EV	Electron-Volt
EV/U	Energy Associated with Unified Atomic Mass Unit
EV/PRO	Proton Mass
EV/NEU	Neutron Mass
EV/ELE	Electron Mass
HZ/EV	Cycle
WAVELE	Wavelength
WAVENO	Wave Number
K/EV	K

TABLE OF GENERAL PHYSICAL CONSTANTS

OMNITAB NAME	CONSTANT NAME	EST. ERROR LIMIT	(MKSA) VALUE	(CGS) VALUE
(SOLIVA)	SPEED OF LIGHT IN VACUUM	3	2.9979250E 08	2.9979250E 10
(ELECHA)	ELEMENTARY CHARGE	7	1.6021000E-19	1.6021000E-20
(AVOGAD)	AVAGADRO CONSTANT	28	6.0225199E 23	6.0225199E 23
(ELECMA)	ELECTRON REST MASS	4	9.1090999E-31	9.1091000E-28
(PROMAS)	PROTON REST MASS	8	1.6725200E-27	1.6725200E-24
(NEUMAS)	NEUTRON REST MASS	8	1.6748200E-27	1.6748200E-24
(FARADA)	FARADAY CONSTANT	16	9.6487000E 04	9.6486999E 03
(PLANCK)	PLANCK CONSTANT	5	6.6255999E-34	6.6255999E-27
(FISTCO)	FINE STRUCTURE CONSTANT	10	7.2972000E-03	7.2972000E-03
(EOVERM)	CHARGE TO MASS RATIO FOR ELECTRON	19	1.7587960E 11	1.7587960E 07
(HOVERE)	QUANTUM-CHARGE RATIO	12	4.1355600E-15	4.1355599E-07
(COWAEL)	COMPTON WAVELENGTH OF ELECTRON	6	2.4262100E-12	2.4262100E-10
(COWAPR)	COMPTON WAVELENGTH OF PROTON	4	1.3214000E-15	1.3214000E-13
(RYDBER)	RYDBERG CONSTANT	3	1.0973730E 07	1.0973731E 05
(BOHRAD)	BOHR RADIUS	7	5.2916700E-11	5.2916699E-09
(ELECRA)	ELECTRON RADIUS	11	2.8177700E-15	2.8177700E-13
(THOMSO)	THOMSON CROSS SECTION	5	6.6515999E-29	6.6515999E-25
(GY/MAP)	GYROMAGNETIC RATIO OF PROTON	2	2.6751900E 08	2.6751900E 04
(BOHRMA)	BOHR MAGNETON	6	9.2731999E-24	9.2731999E-21
(NUCMAG)	NUCLEAR MAGNETON	4	5.0505000E-27	5.0504999E-24
(PROMOM)	PROTON MOMENT	13	1.4104900E-26	1.4104900E-23
(ZEEMAN)	ZEEMAN SPLITTING CONSTANT	4	4.6685799E 01	4.6685799E-05
(RJ/KMO)	GAS CONSTANT(RJ/MOLE)	12	8.3142999E 00	8.3142999E 07
(RC/KMO)	GAS CONSTANT(RC/MOLE)		1.9871700E 00	1.9871700E 00
(VSUBZE)	NORMAL VOLUME PERFECT GAS	30	2.2413599E-02	2.2413600E 04
(BOLTZM)	BOLTZMANN CONSTANT	18	1.3805400E-23	1.3805400E-16
(CONE)	FIRST RADIATION CONSTANT	3	3.7405000E-16	3.7404999E-05
(CTWO)	SECOND RADIATION CONSTANT	19	1.4387900E-02	1.4387900E 00
(WIEN)	WEIN DISPLACEMENT CONSTANT	4	2.8978000E-03	2.8978000E-01
(STEFBO)	STEFAN-BOLTZMANN CONSTANT	29	5.6696999E-08	5.6697000E-05
(GRAVCO)	GRAVITATIONAL CONSTANT	15	6.6700000E-11	6.6699999E-08
(J/EV)	ELECTRON-VOLT	7	1.6021000E-19	1.6021000E-12
ENERGY ASSOCIATED WITH				
(EV/U)	UNIFIED ATOMIC MASS UNIT	15	9.3147800E 08	9.3147800E 08
(EV/PRO)	PROTON MASS	15	9.3825600E 08	9.3825600E 08
(EV/NEU)	NEUTRON MASS	15	9.3954999E 08	9.3954999E 08
(EV/ELE)	ELECTRON MASS	5	5.1100600E 05	5.1100600E 05
(HZ/EV)	CYCLE	7	2.4180400E 14	2.4180400E 14
(WAVELE)	WAVELENGTH	4	1.2398100E-06	1.2398100E-04
(WAVENO)	WAVE NUMBER	23	8.0657300E 05	8.0657300E 03
(K/EV)	K	16	1.1604900E 04	1.1604900E 04

Problem 10 - 1

Compute the values of the thermodynamic constant stated below in units of calories mole^{-1} degree K^{-1} from the Physical Constants stored in OMNITAB.

$$K = R \log_e \frac{64(10^{-24})\pi^5 k^4}{N^3 h^6 P_{1\ atm}}$$

$$P_{1\ atm} = 1.01325 \cdot 10^6 \frac{\text{dyne}}{\text{cm}^2}$$

The OMNITAB instructions are as follows:

```
OMNITAB   PROBLEM 10 - 1
NOSUMMARY
CGS
RESET 1
MULT *AVOGADRO*  BY  *PLANCK* , STORE 1
MULT  COL 1 BY 1 BY1  ADD TO 2
DIVIDE *BOLTZMAN* BY *PLANCK* MULT BY *BOLTZMAN*  ADD TO 3
DIVIDE 3 BY *PLANCK*   , MULT BY 64.E-24 ADD TO 4
RAISE 3.14159267 TO 5. , MULT 4 ADD TO 5
DIVIDE 5 BY 2 MULT BY 3 ADD TO 6
DIVIDE 6 BY 1.01325E6 STORE 6
LOGE 6 MULT BY *RC/KMO* ADD TO 7
PRINT 6 7
      STOP
```

* * * **RESULTS** * * *

```
PAGE   1      OMNITAB   PROBLEM 10 - 1


   COLUMN  6       COLUMN  7

   3.799467E-04   -1.564992E 01
```

Note that the powers of k, h, and N were not computed directly as they would have exceeded the machine range of 10E$\pm$38 .

11. DOCUMENTATION OF AND WITH OMNITAB

One of the recent trends in the documentation of computer programming systems has been toward so called self-documentation. We mean by this that the program carried its own full description. A notable example of this is the MAD[°] system developed at the University of Michigan. It is our understanding that the tape containing the system carries also the manual for its operation and use.

An extension of this development is to put into the hands of the user a facility to mix text with computation so as that the computer output may be a final document containing both text and results. Such a useful system has recently been provided for programmers by a program called MOIST[°°] developed at the University of Maryland.

OMNITAB has both of the features discussed above. The documentation of the program is in two main parts. First, there is the word COMMANDS which results in an automatic printout of the command structure (the vocabulary and sentence structure) of section 2.2 of this Handbook. The other is an abridgment of section 3.1 through 3.8, 5.1, 6.1, 7.1, 8.1, 9.1, 9.3 wherein the commands are defined. This 16 page text results from the use of the word MANUAL. In addition, provision has been made to respond to the instruction WATSNU by supplying a brief account of the changes and additions to the program since the latest edition of this Handbook. Figures 11-1 and 11-2 illustrate the type of information which the program supplies in response to the words COMMANDS and MANUAL.

° Arden, B. W., Galler, B. A., and Graham, R. M., MAD at Michigan, Datamation, Dec. 1961.

°° Berne, G. M., MOIST: Macro Output Input System for the IBM 7090 Technical Report TR-64-5, University of Maryland Computer Science Center (Jan. 1964).

```
OUTPUT INSTRUCTICNS

 NOTE''         (AS APPROPRIATE)
 FOCTNOTE'''' (AS APPROPRIATE)
 TITLE1'   (66 CHARACTERS IF REQUIRED)
 TITLE2'   (66 CHARACTERS IF REQUIRED)
 TITLE3'   (53 CHARACTERS IF REQUIRED)
 TITLE4'   (53 CHARACTERS IF REQUIRED)
 FIXED WITH ,, DECIMALS
 FLCATING WITH ,, DECIMALS
 NOSUMMARY    (SUPPRESSES SUMMARY)
 SUMMARIZE    (RESTCRES SUMMARY)
 FORMAT       (AS DESIRED ACCORDING WITH FORTRAN RULES)
 PRINT COL ++,++,++,ETC   (8 COL LIMIT)
 FPRINT CCL ++,++,++,ETC (AS ALLOWED BY FORMAT)
 TPRINT  COL ++,++,++,++ .... (AS ALLOWED BY FORMAT BUT DOES NOT PAGE)
 PUNCH COL ++,++,++,++ (4 COL LIMIT)
 FPUNCH CCL ++,++,++, ... (AS ALLOWEC BY FORMAT)
 PLCT CCLS ++,++,++, AGAINST ++  (5 COL LIMIT)
```

Figure 11-1. A portion of the printout from the COMMAND instruction.

```
1. PRINT COL ++,++,++,....(8 COL LIMIT).
  A. 'NRMAX' VALUES IN THE SPECIFIED COLUMNS ARE PRINTED FIXED
     OR FLOATING POINT DEPENDING ON 2 OR 3 BELOW.
  B. TITLES AND COLUMN HEACING ARE ALSO PRINTED.,
  C. WHEN NO VALUES WERE ENTERED IN A COLUMN OR A PORTION OF IT,
     ZEROS WILL BE PRINTED.
2. FIXEC WITH ,, DECIMALS
  A. PRCVIDES FIXED FIELD FORMAT WITH THE SPECIFIED DECIMAL PLACES.
  B. MAXIMUM DECIMAL FIELD IS 7.
3. FLOATING WITH ,, DECIMALS
  A. PRCVIDES FLOATING FIELD FORMAT (NORMAL) WITH THE SPECIFIED
     NUMBER CF DECIMAL PLACES.
  B. MAXIMUM DECIMAL FIELD IS 7.
4. PUNCH COL ++,++,++,(4 COL LIMIT)
  A. PRCVIDES 'NRMAX' CARD OUTPUT IN ACCORD WITH THE BUILT-IN
     'FCRMAT'.
  B. PUNCHING IS LIMITED TO 60 CHARACTERS INCLUDING SPACES.
5. PLCT CCLS ++,++,++,AGAINST COL ++
  A. PLCTS CF ONE TC FIVE FUNCTIONS ARE PROVIDED.
  B. THE FIVE SYMBOLS USED ARE  .  +  *  ,  0
```

Figure 11-2. A portion of the printout from the MANUAL instruction.

11.1. Report Writing via OMNITAB

Provision has been made in the program to accommodate textual material via the use of the instructions NOTE (and certain synonyms), FOOTNOTE, SPACE, NEWPAGE, PAGE, APRINT, and TPRINT. These instructions make it possible for the user to have considerable flexibility in formating, and mixing the text with computed results and, in fact, mixing text writing interchangeably with computation. In the pages following, we give examples illustrative of the results which can be achieved in this way. The arrangement of the instruction cards follow the results.

Experience with handling many small pieces of data, especially those culled from the scientific literature, pointed up the need for a system of identification — authors, journal, title, subject headings, table units, etc. — which would make the data self explanatory. Furthermore, we consider it important that the program which carries out computations on these data be able to handle the entire pack—text, notes, titles, etc. — in an intelligent way. By intelligent we mean having the ability to differentiate between data and bibliographic or editorial comment. Furthermore, it should not ignore the bibliographic information completely but should rather print the information out as a preamble to the calculations, either with or without any ad hoc comments which the user may see fit to make in the course of the calculations.

In order to achieve this seemingly large, order it was necessary to do two things. First, to adopt a simple code to identify the various non—data cards and secondly to introduce this list of code words as synonyms for the word NOTE. Thus, any card read into OMNITAB which has the following words starting in column 1 of the card will be treated as a NOTE card:

SU	for subject
PRO	for property
AUT	for author
TIT	for title
JOU	for journal
CAP	for caption
HEA	for heading
COM	for comment
IND	for index

Problem 11 – 1

The material below shows the result of reading in a portion of a data file for hydrogen and printing the data immediately below certain statements via the TPRINT instruction. How this is achieved is indicated on the two pages following:

PAGE 1 OMNITAB DATA FILE OF H2 AND D2
HYDROGEN,DEUTERIUM
P,PVT,RHO
MICHELS,DE GRAAFF,WASSENAAR,LEVELT,LOUWERE
COMPRESSIBILITY ISOTHERMS OF HYDROGEN AND DEUTERIUM AT TEMPERATURE
BETWEEN-175 C AND +150 C(AT DENSITIES UP TO 960 AMAGAT)
PHYSICA 25,25,1959
TABLE 1
T,P,PV, AMAGAT

-25.00000	1.00000	7.17550	0.91215	7.86660
-25.00000	2.00000	9.31740	0.91327	10.20220
-25.00000	23.00000	11.00250	0.91439	12.03260
-25.00000	3.00000	13.13280	0.91546	14.34550
-25.00000	4.00000	16.91820	0.91788	18.43180
-25.00000	5.00000	22.05850	0.92089	23.95360
-25.00000	6.00000	28.48020	0.92472	30.79860
-25.00000	7.00000	35.87870	0.92933	38.60720
-25.00000	8.00000	48.18070	0.93690	51.42590
-25.00000	9.00000	67.25160	0.94898	70.86720
-25.00000	10.00000	88.66680	0.96276	92.09630
-25.00000	11.00000	123.45700	0.98554	125.26810
-25.00000	12.00000	153.26480	1.00566	152.40280
-25.00000	13.00000	186.62280	1.02854	[illegible]
-25.00000	14.00000	235.15570	[illegible]	[illegible]
-25.00000	15.00000	281.01420	[illegible]	[illegible]
-25.00000	16.00000	[illegible]	[illegible]	224.59890
-25.00000	[illegible]	[illegible]	0.86220	260.35430
[illegible]	[illegible]	267.82700	0.89295	299.93510
[illegible]	17.00000	340.08700	0.94557	359.66350
[illegible]	18.00000	409.10870	0.99681	410.41790
-75.00000	19.00000	488.10520	1.05596	462.23830
-75.00000	20.00000	570.92799	1.11819	510.58230
-75.00000	21.00000	681.21940	1.20074	567.33299
-75.00000	22.00000	814.64359	1.29950	626.89000
-75.00000	13.00000	149.44000	0.81138	6792.00000

A COMPARISON IS MADE WITH NBS RP1932
SMOOTHED TABLES ARE GIVEN OF PV AT INTEGRAL VALUES OF RHO AND P
COEFFICIENTS ARE GIVEN OF DENSITY EXPANSION OF PV ISOTHERMS
LEVELT IS NOW SENGERS-LEVELT

THE FOLLOWING WORDS OR PHRASES ARE INDEXING OR DESCRIPTOR TERMS.

DEUTERIUM
GAS
HIGH PRESSURE
LOW TEMPERATURE
MEDIUM TEMPERATURE
EXPERIMENTAL
VAN DER WAALS LABORATORY

The OMNITAB instructions are as follows:

```
OMNITAB        DATA FILE OF H2 AND D2
FORMAT (8F12.5)
NEWPAGE
SU    HYDROGEN,DEUTERIUM
PRC   P,PVT,RHO
AUT   MICHELS,DE GRAAFF,WASSENAAR,LEVELT,LOUWERE
TIT   COMPRESSIBILITY ISOTHERMS OF HYDROGEN AND DEUTERIUM AT TEMPERATURE
TIT   BETWEEN-175 C AND +150 C(AT DENSITIES UP TO 960 AMAGAT)
JOU   PHYSICA 25,25,1959
CAP   TABLE 1
HEA   T,P,PV, AMAGAT
READ 1,2,3,4,5
D      -25.C  1  7.1755  0.91215  7.8666
D      -25.C  2  9.3174  0.91327  10.2022
D      -25.C  23  11.0025  0.91439  12.0326
D      -25.C  3  13.1328  0.91546  14.3455
D      -25.C 4  16.9182  0.91788  18.4318
D      -25.C  5  22.0585  0.92089  23.9536
D      -25.C  6  28.4802  0.92472  30.7986
D      -25.C  7  35.8787  0.92933  38.6072
D      -25.C  8  48.1807  0.93690  51.4259
D      -25.C  9  67.2516  0.94898  70.8672
D      -25.C  10  88.6668  0.96276  92.0963
D      -25.C  11  123.4570  0.98554  125.2681
D      -25.C  12  153.2648  1.00566  152.4028
D      -25.C  13  186.6228  1.02854  181.4440
D      -25.C  14  235.1557  1.06264  221.2947
D      -25.C  15  281.0142  1.09529  256.5660
D      -25.C  16  335.4776  1.13478  295.6312
D      -25.C  17  425.9170  1.20102  354.6306
D      -25.C  18  511.8284  1.26433  404.8231
D      -25.C  19  609.5024  1.33626  456.1253
D      -25.C  20  711.1191  1.41084  504.0413
D      -25.C  21  845.1927  1.50832  560.3550
D      -25.C  22  1005.6475  1.62324  619.5310
D      -75.C  1  5.8140  0.72781  7.9883
D      -75.C  2  7.5491  0.72872  10.3594
D      -75.C  23  8.9114  0.72940  12.2174
D      -75.C  3  10.6370  0.73021  14.5671
D      -75.C  4  13.6951  0.73171  18.7165
D      -75.C  5  17.8486  0.73379  24.3239
D      -75.C  6  23.0308  0.73640  31.2749
D      -75.C  7  28.9941  0.73957  39.2040
D      -75.C  8  38.8960  0.74484  52.2206
D      -75.C  9  54.2029  0.75330  71.9540
D      -75.C  10  71.3544  0.76308  93.5084
D      -75.C  11  99.1456  0.77959  127.1766
D      -75.C  12  122.9006  0.79437  154.7145
D      -75.C  14  188.0162  0.83712  224.5989
D      -75.C  15  224.4775  0.86220  260.3543
```

```
D     -75.C  16  267.8270  0.89295  299.9351
D     -75.C  17  340.0870  0.94557  359.6635
D     -75.C  18  409.1087  0.99681  410.4179
D     -75.C  19  488.1052  1.05596  462.2383
D     -75.C  20  570.9280  1.11819  510.5823
D     -75.C  21  681.2194  1.20074  567.3330
D     -75.C  22  814.6436  1.29950  626.8900
D     -75.C  13  149.4400  0.81138  1841800
TPRINT 1 2 3 4 5
COM   A COMPARISON IS MADE WITH NBS RP1932
COM   SMOOTHED TABLES ARE GIVEN OF PV AT INTEGRAL VALUES OF RHO AND P
COM   COEFFICIENTS ARE GIVEN OF DENSITY EXPANSION OF PV ISOTHERMS
COM   LEVELT IS NOW SENGERS-LEVELT
COM
COM   THE FOLLOWING WORDS OR PHRASES ARE INDEXING OR DESCRIPTOR TERMS.
COM
INC   DEUTERIUM
INC   GAS
INC   HIGH PRESSURE
INC   LOW TEMPERATURE
INC   MEDIUM TEMPERATURE
INC   EXPERIMENTAL
INC   VAN DER WAALS LABORATORY
        STOP
```

Problem 11 - 2

The material below shows the result of incorporating the problem statement as a preamble to the instruction set.

```
PAGE   1      OMNITAB  GEOMETRIC PARAMETERS FOR TRIATOMIC MOLECULES
THE FOLLOWING CALCULATIONS SOLVE FOR THE SIDE AND THE SINE AND
CCSINE CF TWC ANGLES OF A TRIANGLE FORMED BY A TRIATOMIC
MCLECULE, CCNTAINING ATOMS LABELED 0, 1, 2,WHERE THE ZERO
DENCTES THE VERTEX CF THE KNOWN ANGLE, PHI.

THE TABLE IS GIVEN IN TERMS OF THE RATIO OF THE BOND DISTANCES
RC1 OVER RC2 WHERE RC1 IS GREATER THAN RO2.
TABLE 3 GIVES, AS A FUNCTICN OF PHI AND RO1 OVER RO2, THE
FCLLOWING PARAMETERS.....
Q, THE CISTANCE BETWEEN ATOMS 1 AND 2
S1, THE CCS PHI 012
S2 THE CCS PHI 021
T1 THE SIN PHI 312
T2 THE SIN PHI 021
AS WELL AS THEIR SQUARES AND PRODUCTS S1S2,T1T2, ETC.
THE CALCULATIONS ARE CARRIED OUT IN THE REPEAT MODE FOR A
VECTOR CF 50 VALUES OF RO1 OVER RO2 FOR EACH OF 91 CYCLES
CORRESPCNDING TO VALUES OF ANGLE PHI FROM 90 DEGREES TO
180 DEGREES IN UNIT INTERVALS.
```

PAGE 2 OMNITAB GEOMETRIC PARAMETERS FOR TRIATOMIC MOLECULES

PHI	R	Q	S1S1	S2S2	T1T1	T2T2	S1S2
90.	1.00	1.414214	0.500000	0.500000	0.5C0000	0.500000	0.500000
90.	0.99	1.407160	0.505025	0.494975	0.494975	C.505025	0.499975
90.	0.98	1.400143	0.510100	0.489900	0.489900	C.510100	0.499898
90.	0.97	1.393162	0.515225	0.484775	0.484775	C.515225	0.499768
90.	0.96	1.386218	0.520400	0.479600	0.479600	0.520400	0.499584
90.	0.95	1.379311	0.525624	0.474376	0.474376	C.525624	0.499343
90.	0.94	1.372443	0.530898	0.469102	0.469102	0.530898	0.499044
90.	0.93	1.365613	0.536222	C.463778	0.463778	C.536222	0.498686
90.	0.92	1.358823	0.541594	0.458406	0.458406	C.541594	0.498267
90.	0.91	1.352072	0.547016	0.452984	0.452984	C.547016	0.497785
90.	0.90	1.345362	0.552486	0.447514	0.447514	C.552486	0.497238
90.	0.89	1.338693	0.558005	0.441995	0.441995	0.558005	0.496624
90.	0.88	1.332066	0.563571	0.436429	0.436429	C.563571	0.495942
90.	0.87	1.325481	0.569184	0.430816	0.430816	C.569184	0.495190
90.	0.86	1.318939	0.574845	0.425155	0.425155	C.574845	0.494367
90.	0.85	1.312440	0.580552	0.419448	0.419448	0.580552	0.493469
90.	0.84	1.305986	0.586304	0.413696	0.413696	0.586304	0.492495
90.	0.83	1.299577	0.592101	0.407899	0.407899	C.592101	0.491444
90.	0.82	1.293213	0.597943	0.402057	0.402057	C.597943	0.490313
90.	0.81	1.286895	0.603828	0.396172	0.396172	C.603828	0.489101

Problem 11 - 2

The OMNITAB instructions are as follows:

```
OMNITAB  GEOMETRIC PARAMETERS FOR TRIATOMIC MOLECULES   JULY 31,1964
TITLE2PHI    R        Q        S1S1       S2S2      T1T1      T2T2       S1
TITLE4S2     T1T2      S1T2      S2T1
NEWPAGE
NOTE  THE FOLLOWING CALCULATIONS SOLVE FOR THE SIDE AND THE SINE AND
NOTE  COSINE OF TWO ANGLES OF A TRIANGLE FORMED BY A TRIATOMIC
NOTE  MOLECULE, CONTAINING ATOMS LABELED 0, 1, 2 WHERE THE ZERO
NOTE  DENOTES THE VERTEX OF THE KNOWN ANGLE, PHI.
SPACE 1
NOTE  THE TABLE IS GIVEN IN TERMS OF THE RATIO OF THE BOND DISTANCES
NOTE  RO1 OVER RO2 WHERE RO1 IS GREATER THAN RO2.
NOTE  TABLE 3 GIVES, AS A FUNCTION OF PHI AND RO1 OVER RO2, THE
NOTE  FOLLOWING PARAMETERS.....
NOTE  Q, THE DISTANCE BETWEEN ATOMS 1 AND 2
NOTE  S1, THE COS PHI 012
NOTE  S2 THE COS PHI 021
NOTE  T1 THE SIN PHI 312
NOTE  T2 THE SIN PHI 021
NOTE  AS WELL AS THEIR SQUARES AND PRODUCTS S1S2,T1T2, ETC.
NOTE  THE CALCULATIONS ARE CARRIED OUT IN THE REPEAT MODE FOR A
NOTE  VECTOR OF 50 VALUES OF RO1 OVER RO2 FOR EACH OF 91 CYCLES
NOTE  CORRESPONDING TO VALUES OF ANGLE PHI FROM 90 DEGREES TO
NOTE  180 DEGREES IN UNIT INTERVALS.

GENERATE   90., 1., 180. COL 40
RESET      0
GENERATE    1.,   -0.01, 0.51  COL 41
BEGIN
DEFINE     1, 40, 1
INCREMENT INST 1, BY 1, 0, 0
SIND       1, 3
COSD       1, 4
SUBTRACT   4, 41, 5
MULT       41, 4, 6
SUB        6, 1., 6
MULT       6, 1., 7
MULT       5, 41, 1., 7
SQRT       7, 8
DIVIDE     6, 8, 9
DIVIDE     5, 8, 10
DIVIDE     3, 8, 12
MULT       12, 41, 11
MULT       9  9  19
MULT     10  10  20
MULT     11  11  21
MULT     12  12  22
MULT     9  10  23
MULT     11  12  24
MULT     9  12  25
MULT       10  11  26
FINISH
FORMAT  (1X, 1F4.0, 1F6.2, 9F10.6)
REPEAT 1 22 1
FPRINT  1  41  8  19  20  21 22  23 24 25 26
REPEAT 1 22 1
FPRINT  1  41  8  19  20  21 22  23 24 25 26
      STOP
```

Problem 11 - 3

```
   1                                                              1
    FINC THE ROOTS BETWEEN ZERO AND 30 OF THE TRANSCENDENTAL
EQUATICN
                         TANX = 3X / (3 - X.X).
ADAMS (SMITHSCNIAN TABLES VOL74,NO 1,PAGE 85) GIVES THE FIRST
SEVEN RCOTS AS FCLLCWS
     0, 1.8346  ,2.8950  ,3.9225  ,4.9385  ,5.9489  ,6.9563  .
THE OMNITAB INSTRUCTIONS TO SOLVE THIS PROBLEM ARE AS FOLLOWS.

   CMNITAB   11 - 3
   GENERATE 0.  .3  30.  IN COL 1
   PAGE  STARTING WITH 1 IN  COL 55

   BEGIN STCRING INSTRUCTICNS
   MULT   1  1  2 GIVES X SQUARED
   SUB COL 2 FROM 3.  STORE IN 2
   MULT    1  BY 3.  STORE IN 3
   TAN    1  4
   CIV  3  BY  2  STCRE IN  5
   SUB    5  FROM 4  STORE IN  6
   ISOLATE IN COL 1 FOR COL 6 EQUAL C. STORE IN COL 1
   COMPARE CCL 43 ANC 45 TO A TOLERANCE OF .00001
   CIV   45 BY 3.1415926  STORE IN  44
   EXCHANGE  COL 43 AND  45
   FINISH STCRING INSTRUCTIONS
   REPEAT INSTR 1  THRU 10  9 TIMES
   HEAD CCL 1/   RCCTS
   HEAD CCL 44/  ROCT / PI
   HEAD CCL 43/    RCOTS
   HEAD CCL  6/     F(X)
   FIXEC  WITH 6 CECIMALS
   PRINT    43  44   1   6
   STCP


UPON EXECUTICN OF THE ABOVE COMMANDS, THE MACHINE PRINTS A LINE
INDICATING WHETHER CR NOT THE COMPARE WAS SATISFIED.  IN THIS
CASE IT IS AS FOLLOWS.


 COMPARE SATISFIEC, CONTROL FORCED FROM REPEAT MODE AFTER       8 TIMES
```

THE RESULTS FROM THE OMNITAB CALCULATION ARE AS FOLLOWS

ROOTS	ROOTS/PI	X	F(X)
0.	0.	0.	0.
4.712365	1.499992	4.712384	16574.683594
5.763464	1.834568	4.712388	23787.610596
7.854016	2.500011	4.712393	42039.723145
9.095031	2.895038	4.712397	182535.781250
10.995592	3.500006	4.712402	-76944.686523
12.322961	3.922520	5.763446	-0.000039
14.137170	4.500001	5.763450	-0.000016
15.514635	4.938462	5.763455	0.000006
17.278746	5.499996	5.763459	0.000028
18.689024	5.948901	5.763464	0.000050
20.420324	6.499991	7.853979	379871.320313
21.853892	6.956310	7.853984	-63541.136719
23.561974	7.500009	7.853988	-29429.145508
25.012829	7.961831	7.853993	-19142.119141
26.703552	8.500005	7.853997	-14166.927368
28.167808	8.966092	9.094995	-0.000018
29.845129	9.500000	9.094999	0.000002
0.	0.	9.095004	0.000021
0.	0.	9.095008	0.000041

THE ABCVE PAGE WAS PRINTED PARTIALLY BEFORE AND PARTIALLY AFTER THE CALCULATIONS BUT BEFORE THE NORMAL PRINTOUT WAS REQUESTED. THE TEXTUAL PORTION WAS PUNCHED ON CARDS WHICH CARRIED THE WORD NOTE IN THE FIRST SIX COLUMNS. EXCEPT FOR THE HEADINGS,THE RESULTS WERE OBTAINED VIA THE TPRINT INSTRUCTION. NOTE SHOULD ALSO BE TAKEN OF THE LOCATICN CF THE PAGE NUMBER WHICH IS CONTROLLED BY THE INSTRUCTICN

PAGE STARTING WITH 1 IN LOCATION 55

The extraneous values are not roots. They correspond to infinite discontinuities in the function. They show up here since the ISOLATE instruction searches for a change of sign of the function and interposes a finer mesh between the two abscissae for which the function exhibits a change of sign. This turns out to be a boon rather than a problem since it shows how the instruction can be used to isolate certain types of infinite discontinuities as well as roots. We leave it to the reader to devise a few instructions which will isolate cusp type discontinuities.

Pages 234 and 235 give the instruction set to achieve the above results.

RCOTS	RCOT / PI	ROOTS	F(X)
0.	0.	0.	0.
4.712365	1.499992	4.712384	16574.683594
5.763464	1.834568	4.712388	23787.610596
7.854016	2.500011	4.712393	42039.723145
9.095031	2.895038	4.712397	182535.781250
10.995592	3.500006	4.712402	-76944.686523
12.322961	3.922520	5.763446	-0.000039
14.137170	4.500001	5.763450	-0.000016
15.514635	4.938462	5.763455	0.000006
17.278746	5.499996	5.763459	0.000028
18.689024	5.948901	5.763464	0.000050
20.420324	6.499991	7.853979	379871.320313
21.853892	6.956310	7.853984	-63541.136719
23.561974	7.500009	7.853988	-29429.145508
25.012829	7.961831	7.853993	-19142.119141
26.703552	8.500005	7.853997	-14166.927368
28.167808	8.966092	9.094995	-0.000018
29.845129	9.500000	9.094999	0.000002
0.	0.	9.095004	0.000021
0.	0.	9.095008	0.000041
0.	0.	9.095013	0.000061
0.	0.	10.995574	53672.940430
0.	0.	10.995579	2702931.812500
0.	0.	10.995583	-55952.775879
0.	0.	10.995588	-27748.279297
0.	0.	10.995592	-18320.543945
0.	0.	12.322924	-0.000018
0.	0.	12.322929	0.000001
0.	0.	12.322933	0.000020
0.	0.	12.322938	0.000039
0.	0.	12.322942	0.000059
0.	0.	14.137151	29334.310059
0.	0.	14.137156	64315.842773
0.	0.	14.137161	-356130.003906
0.	0.	14.137165	-47292.640137
0.	0.	14.137170	-25303.915039
0.	0.	15.514599	-0.000004
0.	0.	15.514603	0.000014
0.	0.	15.514608	0.000033
0.	0.	15.514612	0.000052
0.	0.	15.514617	0.000071
0.	0.	17.278746	19853.452637
0.	0.	17.278750	31044.850830
0.	0.	17.278755	71201.400391
0.	0.	17.278759	-254010.958984
0.	0.	17.278764	-43777.187012
0.	0.	18.689024	-0.000050
0.	0.	18.689029	-0.000031
0.	0.	18.689033	-0.000013
0.	0.	18.689038	0.000006

The OMNITAB instructions are as follows:

```
OMNITAB       11 - 3
NOSUMMARY
         PAGE  STARTING WITH 1 IN  COL 55
NOTE      FIND THE ROOTS BETWEEN ZERO AND 30 OF THE TRANSCENDENTAL
NOTE  EQUATION
NOTE                              TANX = 3X / (3 - X.X).
NOTE  ADAMS (SMITHSONIAN TABLES VOL74,NO 1,PAGE 85) GIVES THE FIRST
NOTE  SEVEN ROOTS AS FOLLOWS
NOTE       0, 1.8346  ,2.8950  ,3.9225  ,4.9385  ,5.9489  ,6.9563  .
NOTE  THE OMNITAB INSTRUCTIONS TO SOLVE THIS PROBLEM ARE AS FOLLOWS.
SPACE   1
NOTE     OMNITAB   11 - 3
NOTE     GENERATE 0.  .3  30.  IN COL 1
NOTE     PAGE  STARTING WITH 1 IN  COL 55
SPACE  1
NOTE     BEGIN STORING INSTRUCTIONS
NOTE     MULT   1  1  2 GIVES X SQUARED
NOTE     SUB COL 2 FROM 3.  STORE IN 2
NOTE     MULT    1  BY 3.  STORE IN 3
NOTE     TAN    1  4
NOTE     DIV  3  BY  2  STORE IN  5
NOTE     SUB    5  FROM 4  STORE IN  6
NOTE     ISOLATE IN COL 1 FOR COL 6 EQUAL 0. STORE IN COL 1
NOTE     COMPARE COL 43 AND 45 TO A TOLERANCE OF .00001
NOTE     DIV   45 BY 3.1415926  STORE IN  44
NOTE     EXCHANGE  COL 43 AND  45
NOTE     FINISH STORING INSTRUCTIONS
NOTE     REPEAT INSTR 1  THRU 10  9 TIMES
NOTE     HEAD COL 1/   ROOTS
NOTE     HEAD COL 44/  ROOT / PI
NOTE     HEAD COL 43/    ROOTS
NOTE     HEAD COL  6/     F(X)
NOTE     FIXED  WITH 6 DECIMALS
NOTE     PRINT    43  44   1   6
NOTE     STOP
SPACE  1
         GENERATE 0.  .3  30.  IN COL 1

         BEGIN STORING INSTRUCTIONS
         MULT   1  1  2 GIVES X SQUARED
         SUB COL 2 FROM 3.  STORE IN 2
         MULT    1  BY 3.  STORE IN 3
         TAN    1  4
         DIV  3  BY  2  STORE IN  5
SUB    5  FROM 4  STORE IN  6
         ISOLATE IN COL 1 FOR COL 6 EQUAL 0. STORE IN COL 1
         COMPARE COL 43 AND 45 TO A TOLERANCE OF .00001
         DIV   45 BY 3.1415926  STORE IN  44
         EXCHANGE  COL 43 AND  45
         FINISH STORING INSTRUCTIONS
```

```
SPACE 1
NOTE  UPCN EXECUTICN OF THE ABOVE COMMANDS, THE MACHINE PRINTS A LINE
NOTE  INDICATING WHETHER OR NOT THE COMPARE WAS SATISFIED.  IN THIS
NOTE  CASE IT IS AS FOLLOWS.
SPACE 1
        REPEAT INSTR 1  THRU 1C  9 TIMES
SPACE  1
NOTE    THE RESULTS FROM THE OMNITAB CALCULATION ARE AS FOLLOWS
SPACE  1
NOTE        RCOTS       ROOTS/PI            X           F(X)
FORMAT  (1X,4F15.6)
RESET 20
SPACE 1
TPRINT 43 44 1 6
SPACE  1
NOTE    THE ABOVE  PAGE WAS PRINTED PARTIALLY BEFORE AND PARTIALLY
NOTE    AFTER THE CALCULATIONS BUT BEFORE THE NORMAL PRINTOUT
NOTE    WAS REQUESTEC. THE TEXTUAL PORTION WAS PUNCHED ON CARDS
NOTE    WHICH CARRIEC THE WORD NOTE IN THE FIRST SIX COLUMNS.
NOTE    EXCEPT FOR THE HEACINGS,THE RESULTS WERE OBTAINED VIA
NOTE    THE TPRINT INSTRUCTION. NOTE SHOULD ALSO BE TAKEN OF THE
NOTE    LOCATION OF THE PAGE NUMBER WHICH IS CONTROLLED BY THE
NOTE    INSTRUCTION
SPACE 1
NOTE            PAGE STARTING WITH 1 IN LOCATION 55
        HEAD COL 1/    ROOTS
        HEAD COL 44/   ROOT / PI
        HEAD COL 43/     ROOTS
        HEAD COL  6/      F(X)
        FIXED  WITH 6 DECIMALS
        RESET 50
        PRINT     43  44   1   6
        STCP
```

11.2. Sources and Accuracy of the OMNITAB Subroutines

With the exception of the function subroutines discussed below, the subroutines in OMNITAB were written at the National Bureau of Standards. The symbolic listings for these (approximately 12,000 lines) will be incorporated in a separate report. For the present work it seems sufficient to extract from the detailed program and its description such information as will: give the user confidence in the results; emphasize the consequences of certain errors in the commands; and help identify the causes of the error diagnostics. Accordingly appendix II lists the causes and consequences for those error diagnostics which are not self-explanatory. The other diagnostics which are not listed there are self-explanatory.

As indicated above, some of the subroutines were obtained from sources outside of NBS. Of these, the following were obtained from the Bell Telephone Laboratories and modified only superficially to fit the OMNITAB framework: the sine integral, cosine integral, exponential integral, and the complete and incomplete elliptic integrals.

The function subroutine for SIN, COS, EXP, NEGEXP, LOG, LOGTEN, TAN, COTAN, ARCSIN, ARCCOS, ARCTAN, ARCCOT, SINH, COSH, TANH, and GAMMA are those recently written at the Computation Center, University of Chicago.° They represent a significant improvement in accuracy over comparable subroutines available under the IBM Fortran II library or under the IBJOB library.°° The accuracy and limitations of these subroutines are summarized in table C below.

The matrix diagonalization subroutine used to calculate EIGENVALUES and EIGENVECTORS was obtained from the SHARE library.

- - - - - - - - - - -

° Hirando Kuki, Analysis for University of Chicago, Math Library University of Chicago, Computation Center (August 1964).

°°H. Kuki and C. C. J. Roothaan, Tests on University of Chicago Library Function Subprograms, February 1964 as Revised March 1964.

TABLE C.

Characteristics of Certain Function Subroutines in OMNITAB

COMMAND	RELATIVE ERROR	LIMITATION	NOTE
SQRT	$\leq$10E−8	for positive numbers only	
EXP	$\leq$10E−8	$-88.02969 \leq X \leq 88.02969$	a,b
LOGE	$\leq$10E−8 <10E−7.6 (for X near 1)	for $0 < X$	c
SIN	$\leq$10E−8 for $-\pi/2\ X < \pi/2$	for $X < 2^{25}$	b,d
COS	$\leq$10E−8 for $0 < X < \pi$	for $X < 2^{25}$	b,d,e
ATAN TANH	$\leq$10E−8	10E−38 < X < 10E+38	
TAN COT	<10E−8 for $-\pi/4 < X < \pi/4$	$X < 2^{20}$	f
ASIN ACOS	$\leq 2^{-26}$	for $X \leq 1$	a
SINH COSH	$\leq 2^{-26}$	for $X < 88.028$	a
ERROR	<10E−8	10E−38 < X < 10E+38	
GAMMA	$\leq 2^{-26}$ for $0 < X < 4$	$2^{-127} < X < 34.843$	g

a. if the argument exceeds the range, the calculation is terminated.
b. if the argument is below the range, a zero is given as the answer and the calculations continue.
c. if the argument is zero or negative the calculation is terminated.
d. rounding errors in reducing the argument to the principal range, introduce much larger errors.
e. except near $\pi/2$.
f. if the argument is close to a multiple of $\pi/2$, rounding may introduce larger errors.
g. for larger arguments the error is of the order of the rounding uncertainty.

WE WISH TC CHECK THE QUALITY OF THE RESULTS (ROUNDED TO SIX DECIMALS) OBTAIN FRCM OMNITAB FOR THE FOLLOWING FUNCTIONS LNX EXP(X), EXP(-X), LOG(X) , 10 TO X, 1/X, SIN(X) FOR THE FOLLOWING VALUES OF X. 0.0(0.002)0.1, 0.(1.0)50. ALSO LOG X FOR THE FOLLOWING VALUES OF X. -50.(2.)50.

THE INSTRUCTIONS ARE AS FOLLOWS

```
CMNITAB  PRCBLEM 11 -  4
CMNITAB       TEST PROBLEM 2
GENERATE 0.0(0.002)0.1 STORE IN COL 1
BEGIN
LOGE OF 1 STORE 2
EXP COL 1, STORE IN COL 3
NEGEXP CCL 1, STORE 4
LOGTEN CCL 1, STCRE IN COL 5
RAISE 10.0 TO CCL 1, STORE IN COL 6
DIVIDE 1.0 BY CCL 1, STORE IN COL 7
SIN OF 1, STORE 8
FINISH
HEAD 1/      X
HEAD 2/    LN X
HEAD 3/  EXP X
HEAD 4/  EXP-X
HEAD 5/  LCG X
HEAD 6/TEN TO THE X
HEAD 7/     1/X
HEAD 8/   SIN X
REPEAT 1 THRU 7,  1 TIME
PRINT COL 1,2,3,4,5,6,7,8
FIXED 6     DECIMALS
PRINT  1,2,3,4,5,6,7,8
ERASE 1 2 3 4 5 6 7 8
RESET 1
GENERATE 0.(1.0)50. IN COL 1
REPEAT 1 TC 4 1 TIME
RESET TO 37  THIS IS DCNE SO THE MACHINE WILL NOT TRY TO RAISE
REPEAT  5 TC 5 1 TIME    TEN TO THE FIFTYTH POWER
RESET 51
REPEAT 6 TC 7 1 TIME
FLCATING WITH 6 DECIMALS
PRINT 1,2,3,4,5,6,7,8
RESET 1
GENERATE -50.(2.)50. IN CCL 1
LOGE OF CCL 1, STORE 2
LOGTEN OF CCL 1, STORE IN COL 5
PRINT 1 2 5
STCP
```

X	LN X	EXP X	EXP-X	LOG X	TEN TO THE X	1/X	SIN X
0.	1.807522E-08	1.000000E 00	1.000000E 00	7.849969E-09	1.000000E 00	0.	0.
2.000000E-03	-6.214608E 00	1.002002E 00	9.980020E-01	-2.698970E 00	1.004616E 00	5.000000E 02	1.999999E-03
4.000000E-03	-5.521461E 00	1.004008E 00	9.960080E-01	-2.397940E 00	1.009253E 00	2.500000E 02	3.999989E-03
6.000000E-03	-5.115996E 00	1.006018E 00	9.940180E-01	-2.221849E 00	1.013911E 00	1.666667E 02	5.999964E-03
8.000000E-03	-4.828314E 00	1.008032E 00	9.920319E-01	-2.096910E 00	1.018591E 00	1.250000E 02	7.999914E-03
10.000000E-03	-4.605170E 00	1.010050E 00	9.900498E-01	-2.000000E 00	1.023293E 00	1.000000E 02	9.999833E-03
1.200000E-02	-4.422849E 00	1.012072E 00	9.880717E-01	-1.920819E 00	1.028016E 00	8.333333E 01	1.199971E-02
1.400000E-02	-4.268698E 00	1.014098E 00	9.860975E-01	-1.853872E 00	1.032761E 00	7.142857E 01	1.399954E-02
1.600000E-02	-4.135167E 00	1.016129E 00	9.841273E-01	-1.795880E 00	1.037528E 00	6.250000E 01	1.599931E-02
1.800000E-02	-4.017383E 00	1.018163E 00	9.821610E-01	-1.744727E 00	1.042317E 00	5.555556E 01	1.799902E-02
2.000000E-02	-3.912023E 00	1.020201E 00	9.801987E-01	-1.698970E 00	1.047129E 00	5.000000E 01	1.999866E-02
2.200000E-02	-3.816713E 00	1.022244E 00	9.782402E-01	-1.657577E 00	1.051962E 00	4.545455E 01	2.199822E-02
2.400000E-02	-3.729701E 00	1.024290E 00	9.762857E-01	-1.619789E 00	1.056818E 00	4.166667E 01	2.399769E-02
2.600000E-02	-3.649659E 00	1.026341E 00	9.743351E-01	-1.585027E 00	1.061696E 00	3.846154E 01	2.599707E-02
2.800000E-02	-3.575551E 00	1.028396E 00	9.723884E-01	-1.552842E 00	1.066596E 00	3.571429E 01	2.799634E-02
3.000000E-02	-3.506558E 00	1.030455E 00	9.704455E-01	-1.522879E 00	1.071519E 00	3.333333E 01	2.999549E-02
3.200000E-02	-3.442019E 00	1.032517E 00	9.685066E-01	-1.494850E 00	1.076465E 00	3.125000E 01	3.199454E-02
3.400000E-02	-3.381395E 00	1.034585E 00	9.665715E-01	-1.468521E 00	1.081434E 00	2.941176E 01	3.399345E-02
3.600000E-02	-3.324236E 00	1.036656E 00	9.646403E-01	-1.443697E 00	1.086426E 00	2.777778E 01	3.599222E-02
3.800000E-02	-3.270169E 00	1.038731E 00	9.627129E-01	-1.420216E 00	1.091440E 00	2.631579E 01	3.799085E-02
4.000000E-02	-3.218876E 00	1.040811E 00	9.607894E-01	-1.397940E 00	1.096478E 00	2.500000E 01	3.998933E-02
4.200000E-02	-3.170086E 00	1.042894E 00	9.588698E-01	-1.376751E 00	1.101539E 00	2.380952E 01	4.198765E-02
4.400000E-02	-3.123566E 00	1.044982E 00	9.569540E-01	-1.356547E 00	1.106624E 00	2.272727E 01	4.398580E-02
4.600000E-02	-3.079114E 00	1.047074E 00	9.550420E-01	-1.337242E 00	1.111732E 00	2.173913E 01	4.598378E-02
4.800000E-02	-3.036554E 00	1.049171E 00	9.531338E-01	-1.318759E 00	1.116863E 00	2.083333E 01	4.798157E-02
5.000000E-02	-2.995732E 00	1.051271E 00	9.512294E-01	-1.301030E 00	1.122018E 00	2.000000E 01	4.997917E-02
5.200000E-02	-2.956512E 00	1.053376E 00	9.493289E-01	-1.283997E 00	1.127197E 00	1.923077E 01	5.197656E-02
5.400000E-02	-2.918771E 00	1.055485E 00	9.474321E-01	-1.267606E 00	1.132400E 00	1.851852E 01	5.397376E-02
5.600000E-02	-2.882404E 00	1.057598E 00	9.455391E-01	-1.251812E 00	1.137627E 00	1.785714E 01	5.597073E-02
5.800000E-02	-2.847312E 00	1.059715E 00	9.436499E-01	-1.236572E 00	1.142878E 00	1.724138E 01	5.796748E-02
6.000000E-02	-2.813411E 00	1.061837E 00	9.417645E-01	-1.221849E 00	1.148154E 00	1.666667E 01	5.996400E-02
6.200000E-02	-2.780621E 00	1.063962E 00	9.398829E-01	-1.207608E 00	1.153453E 00	1.612903E 01	6.196028E-02
6.400000E-02	-2.748872E 00	1.066092E 00	9.380050E-01	-1.193820E 00	1.158777E 00	1.562500E 01	6.395631E-02
6.600000E-02	-2.718101E 00	1.068227E 00	9.361309E-01	-1.180456E 00	1.164126E 00	1.515152E 01	6.595208E-02
6.800000E-02	-2.688248E 00	1.070365E 00	9.342605E-01	-1.167491E 00	1.169499E 00	1.470588E 01	6.794760E-02
7.000000E-02	-2.659260E 00	1.072508E 00	9.323938E-01	-1.154902E 00	1.174898E 00	1.428571E 01	6.994284E-02
7.200000E-02	-2.631089E 00	1.074655E 00	9.305309E-01	-1.142667E 00	1.180321E 00	1.388889E 01	7.193780E-02
7.400000E-02	-2.603690E 00	1.076807E 00	9.286717E-01	-1.130768E 00	1.185769E 00	1.351351E 01	7.393248E-02
7.600000E-02	-2.577022E 00	1.078963E 00	9.268162E-01	-1.119186E 00	1.191242E 00	1.315789E 01	7.592686E-02
7.800000E-02	-2.551046E 00	1.081123E 00	9.249644E-01	-1.107905E 00	1.196741E 00	1.282051E 01	7.792093E-02

PAGE 4 OMNITAB PROBLEM 11 - 4

X	LN X	EXP X	EXP-X	LOG X	TEN TO THE X	1/X	SIN X
0.	0.000000	1.000000	1.000000	0.000000	1.000000	0.	0.
0.002000	-6.214608	1.002002	0.998002	-2.698970	1.004616	500.000004	0.002000
0.004000	-5.521461	1.004008	0.996008	-2.397940	1.009253	250.000002	0.004000
0.006000	-5.115996	1.006018	0.994018	-2.221849	1.013911	166.666668	0.006000
0.008000	-4.828314	1.008032	0.992032	-2.096910	1.018591	125.000001	0.008000
0.010000	-4.605170	1.010050	0.990050	-2.000000	1.023293	100.000001	0.010000
0.012000	-4.422849	1.012072	0.988072	-1.920819	1.028016	83.333334	0.012000
0.014000	-4.268698	1.014098	0.986098	-1.853872	1.032761	71.428572	0.014000
0.016000	-4.135167	1.016129	0.984127	-1.795880	1.037528	62.500000	0.015999
0.018000	-4.017383	1.018163	0.982161	-1.744727	1.042317	55.555556	0.017999
0.020000	-3.912023	1.020201	0.980199	-1.698970	1.047129	50.000000	0.019999
0.022000	-3.816713	1.022244	0.978240	-1.657577	1.051962	45.454546	0.021998
0.024000	-3.729701	1.024290	0.976286	-1.619789	1.056818	41.666667	0.023998
0.026000	-3.649659	1.026341	0.974335	-1.585027	1.061696	38.461539	0.025997
0.028000	-3.575551	1.028396	0.972388	-1.552842	1.066596	35.714286	0.027996
0.030000	-3.506558	1.030455	0.970446	-1.522879	1.071519	33.333333	0.029995
0.032000	-3.442019	1.032517	0.968507	-1.494850	1.076465	31.250000	0.031995
0.034000	-3.381395	1.034585	0.966572	-1.468521	1.081434	29.411765	0.033993
0.036000	-3.324236	1.036656	0.964640	-1.443697	1.086426	27.777778	0.035992
0.038000	-3.270169	1.038731	0.962713	-1.420216	1.091440	26.315790	0.037991
0.040000	-3.218876	1.040811	0.960789	-1.397940	1.096478	25.000000	0.039989
0.042000	-3.170086	1.042894	0.958870	-1.376751	1.101539	23.809524	0.041988
0.044000	-3.123566	1.044982	0.956954	-1.356547	1.106624	22.727273	0.043986
0.046000	-3.079114	1.047074	0.955042	-1.337242	1.111732	21.739131	0.045984
0.048000	-3.036554	1.049171	0.953134	-1.318759	1.116863	20.833333	0.047982
0.050000	-2.995732	1.051271	0.951229	-1.301030	1.122018	20.000000	0.049979
0.052000	-2.956512	1.053376	0.949329	-1.283997	1.127197	19.230769	0.051977
0.054000	-2.918771	1.055485	0.947432	-1.267606	1.132400	18.518519	0.053974
0.056000	-2.882404	1.057598	0.945539	-1.251812	1.137627	17.857143	0.055971
0.058000	-2.847312	1.059715	0.943650	-1.236572	1.142878	17.241379	0.057967
0.060000	-2.813411	1.061837	0.941765	-1.221849	1.148154	16.666667	0.059964
0.062000	-2.780621	1.063962	0.939883	-1.207608	1.153453	16.129032	0.061960
0.064000	-2.748872	1.066092	0.938005	-1.193820	1.158777	15.625000	0.063956
0.066000	-2.718101	1.068227	0.936131	-1.180456	1.164126	15.151515	0.065952
0.068000	-2.688248	1.070365	0.934260	-1.167491	1.169499	14.705883	0.067948
0.070000	-2.659260	1.072508	0.932394	-1.154902	1.174898	14.285715	0.069943
0.072000	-2.631089	1.074655	0.930531	-1.142667	1.180321	13.888889	0.071938
0.074000	-2.603690	1.076807	0.928672	-1.130768	1.185769	13.513514	0.073932
0.076000	-2.577022	1.078963	0.926816	-1.119186	1.191242	13.157895	0.075927
0.078000	-2.551046	1.081123	0.924964	-1.107905	1.196741	12.820513	0.077921

X	LN X	EXP X	EXP-X	LOG X	TEN TO THE X	1/X	SIN X
0.	1.807522E-08	1.000000E 00	1.000000E 00	7.849969E-09	1.000000E 00	0.	0.
1.000000E 00	1.807522E-08	2.718282E 00	3.678794E-01	7.849969E-09	9.999999E 00	1.000000E 00	8.414710E-01
2.000000E 00	6.931472E-01	7.389056E 00	1.353353E-01	3.010300E-01	9.999999E 01	5.000000E-01	9.092974E-01
3.000000E 00	1.098612E 00	2.008554E 01	4.978707E-02	4.771212E-01	9.999998E 02	3.333333E-01	1.411200E-01
4.000000E 00	1.386294E 00	5.459815E 01	1.831564E-02	6.020600E-01	9.999997E 03	2.500000E-01	-7.568025E-01
5.000000E 00	1.609438E 00	1.484132E 02	6.737947E-03	6.989700E-01	9.999997E 04	2.000000E-01	-9.589243E-01
6.000000E 00	1.791759E 00	4.034288E 02	2.478752E-03	7.781512E-01	9.999996E 05	1.666667E-01	-2.794155E-01
7.000000E 00	1.945910E 00	1.096633E 03	9.118820E-04	8.450980E-01	9.999995E 06	1.428571E-01	6.569866E-01
8.000000E 00	2.079442E 00	2.980958E 03	3.354626E-04	9.030900E-01	9.999995E 07	1.250000E-01	9.893582E-01
9.000000E 00	2.197225E 00	8.103084E 03	1.234098E-04	9.542425E-01	9.999993E 08	1.111111E-01	4.121186E-01
1.000000E 01	2.302585E 00	2.202647E 04	4.539993E-05	10.000000E-01	9.999994E 09	1.000000E-01	-5.440211E-01
1.100000E 01	2.397895E 00	5.987414E 04	1.670170E-05	1.041393E 00	9.999992E 10	9.090909E-02	-9.999902E-01
1.200000E 01	2.484907E 00	1.627548E 05	6.144212E-06	1.079181E 00	9.999992E 11	8.333333E-02	-5.365729E-01
1.300000E 01	2.564949E 00	4.424134E 05	2.260329E-06	1.113943E 00	9.999992E 12	7.692308E-02	4.201670E-01
1.400000E 01	2.639057E 00	1.202604E 06	8.315287E-07	1.146128E 00	9.999990E 13	7.142857E-02	9.906073E-01
1.500000E 01	2.708050E 00	3.269017E 06	3.059023E-07	1.176091E 00	9.999990E 14	6.666667E-02	6.502879E-01
1.600000E 01	2.772589E 00	8.886110E 06	1.125352E-07	1.204120E 00	9.999991E 15	6.250000E-02	-2.879032E-01
1.700000E 01	2.833213E 00	2.415495E 07	4.139938E-08	1.230449E 00	9.999986E 16	5.882353E-02	-9.613975E-01
1.800000E 01	2.890372E 00	6.565997E 07	1.522998E-08	1.255272E 00	9.999987E 17	5.555556E-02	-7.509874E-01
1.900000E 01	2.944439E 00	1.784823E 08	5.602796E-09	1.278754E 00	9.999987E 18	5.263158E-02	1.498771E-01
2.000000E 01	2.995732E 00	4.851652E 08	2.061154E-09	1.301030E 00	9.999987E 19	5.000000E-02	9.129452E-01
2.100000E 01	3.044522E 00	1.318816E 09	7.582560E-10	1.322219E 00	9.999988E 20	4.761905E-02	8.366556E-01
2.200000E 01	3.091042E 00	3.584913E 09	2.789468E-10	1.342423E 00	9.999983E 21	4.545455E-02	-8.851171E-03
2.300000E 01	3.135494E 00	9.744803E 09	1.026188E-10	1.361728E 00	9.999983E 22	4.347826E-02	-8.462204E-01
2.400000E 01	3.178054E 00	2.648912E 10	3.775135E-11	1.380211E 00	9.999984E 23	4.166667E-02	-9.055784E-01
2.500000E 01	3.218876E 00	7.200490E 10	1.388794E-11	1.397940E 00	9.999984E 24	4.000000E-02	-1.323519E-01
2.600000E 01	3.258096E 00	1.957296E 11	5.109089E-12	1.414973E 00	9.999984E 25	3.846154E-02	7.625584E-01
2.700000E 01	3.295837E 00	5.320482E 11	1.879529E-12	1.431364E 00	9.999980E 26	3.703704E-02	9.563759E-01
2.800000E 01	3.332204E 00	1.446257E 12	6.914400E-13	1.447158E 00	9.999980E 27	3.571429E-02	2.709058E-01
2.900000E 01	3.367296E 00	3.931334E 12	2.543666E-13	1.462398E 00	9.999976E 28	3.448276E-02	-6.636336E-01
3.000000E 01	3.401197E 00	1.068647E 13	9.357623E-14	1.477121E 00	9.999981E 29	3.333333E-02	-9.880317E-01
3.100000E 01	3.433987E 00	2.904885E 13	3.442477E-14	1.491362E 00	9.999976E 30	3.225806E-02	-4.040378E-01
3.200000E 01	3.465736E 00	7.896296E 13	1.266417E-14	1.505150E 00	9.999981E 31	3.125000E-02	5.514265E-01
3.300000E 01	3.496507E 00	2.146436E 14	4.658886E-15	1.518514E 00	9.999977E 32	3.030303E-02	9.999119E-01
3.400000E 01	3.526360E 00	5.834617E 14	1.713908E-15	1.531479E 00	9.999973E 33	2.941176E-02	5.290827E-01
3.500000E 01	3.555348E 00	1.586013E 15	6.305117E-16	1.544068E 00	9.999978E 34	2.857143E-02	-4.281827E-01
3.600000E 01	3.583519E 00	4.311231E 15	2.319523E-16	1.556302E 00	9.999973E 35	2.777778E-02	-9.917788E-01
3.700000E 01	3.610918E 00	1.171914E 16	8.533048E-17	1.568202E 00	0.	2.702703E-02	-6.435383E-01
3.800000E 01	3.637586E 00	3.185593E 16	3.139133E-17	1.579784E 00	0.	2.631579E-02	2.963684E-01
3.900000E 01	3.663562E 00	8.659340E 16	1.154822E-17	1.591065E 00	0.	2.564103E-02	9.637953E-01

X	LN X	LOG X
-5.000000E 01	3.912023E 00	1.698970E 00
-4.800000E 01	3.871201E 00	1.681241E 00
-4.600000E 01	3.828641E 00	1.662758E 00
-4.400000E 01	3.784190E 00	1.643453E 00
-4.200000E 01	3.737670E 00	1.623249E 00
-4.000000E 01	3.688879E 00	1.602060E 00
-3.800000E 01	3.637586E 00	1.579784E 00
-3.600000E 01	3.583519E 00	1.556302E 00
-3.400000E 01	3.526360E 00	1.531479E 00
-3.200000E 01	3.465736E 00	1.505150E 00
-3.000000E 01	3.401197E 00	1.477121E 00
-2.800000E 01	3.332204E 00	1.447158E 00
-2.600000E 01	3.258096E 00	1.414973E 00
-2.400000E 01	3.178054E 00	1.380211E 00
-2.200000E 01	3.091042E 00	1.342423E 00
-2.000000E 01	2.995732E 00	1.301030E 00
-1.800000E 01	2.890372E 00	1.255272E 00
-1.600000E 01	2.772589E 00	1.204120E 00
-1.400000E 01	2.639057E 00	1.146128E 00
-1.200000E 01	2.484907E 00	1.079181E 00
-1.000000E 01	2.302585E 00	10.000000E-01
-8.000000E 00	2.079442E 00	9.030900E-01
-6.000000E 00	1.791759E 00	7.781512E-01
-4.000000E 00	1.386294E 00	6.020600E-01
-2.000000E 00	6.931472E-01	3.010300E-01
0.	1.807522E-08	7.849969E-09
2.000000E 00	6.931472E-01	3.010300E-01
4.000000E 00	1.386294E 00	6.020600E-01
6.000000E 00	1.791759E 00	7.781512E-01
8.000000E 00	2.079442E 00	9.030900E-01
1.000000E 01	2.302585E 00	10.000000E-01
1.200000E 01	2.484907E 00	1.079181E 00
1.400000E 01	2.639057E 00	1.146128E 00
1.600000E 01	2.772589E 00	1.204120E 00
1.800000E 01	2.890372E 00	1.255272E 00
2.000000E 01	2.995732E 00	1.301030E 00
2.200000E 01	3.091042E 00	1.342423E 00
2.400000E 01	3.178054E 00	1.380211E 00
2.600000E 01	3.258096E 00	1.414973E 00
2.800000E 01	3.332204E 00	1.447158E 00

11.3. Operating Efficiency

In various places in earlier chapters we have, by direction and implication, encouraged the reader to experiment, to calculate more rather than less, and to print more rather than less. Such advice is based on conviction — reinforced by particular experience both with OMNITAB and more conventional machine usage — that a few more microseconds, even seconds, of computing, and a few extra pages of output is cheaper by many orders of magnitude than the cost of returning to the machine at a later time. But even returning to the machine a few times to modify or polish an earlier run need not be costly if one has other business to transact there anyway. This is especially so if the machine is fast and the program is efficient, and if the jobs can be run in batches without requiring each job to set up the system.

An indication of the speed with which problems of the type illustrated here are handled on the OMNITAB system can be gotten from the following results of an experimental run on 45 of the problems used as illustrations in this Handbook. The problems were run in tandem on a single $JOB card. The total elapsed time was six minutes, the total printout was 12300 lines and the total punchout was 715 cards. Thus, on the average each problem took 8 seconds. The instruction set and data for the 45 problems came to about 2000 cards. A further experiment was run to separate out the computing time from the time needed to dump the results on tape. The six minutes were distributed as follows:

To load the program into the core from the system tape.	0.5 min.
To read the control cards and data (2000 cards in all) and write them on the scratch tape.	1.0 min.
To carry out the calculations.	2.5 min.
To dump the answers and instructions (12,300 lines and 715 cards)	2.0 min.

It should be clear from the above results that, if run in tandem, the problems illustrated require somewhat less than 4 seconds on the average for computation, and another 4 seconds for input and output. The proportion between computation and input/output varies considerably from problem to problem so the 4 to 4 ratio cannot be considered universal. Regardless of the exact ratio it should be obvious that the OMNITAB user ordinarily pays little if any price for the extra output which OMNITAB "imposes" on him; or for the interpretive mode; or for the length of the instruction set; or, except in rare instances, for the need to keep some subroutines on tape files because there is no room for all of them in the core.

In problems of the type illustrated here and even in larger ones it can be said quite categorically that they should not run longer than a minute. A problem which runs two minutes on OMNITAB is an exceedingly long one. It is, however, possible to set up a problem in the repeat mode which may burn up a lot of time if not set up wisely. If a choice presents itself, the calculations should be so arranged as to minimize the number of times instructions are used alternately from different files. We can illustrate this point best by describing a situation which existed when the instruction SEARCH was not in the same file with the TRANSPOSE instruction and the matrix operations.

> Consider a calibration table containing 36 weights and their identification and 59 sets of numbers (approximately 16 per set) representing weight combinations for 59 experimental points. The calibration table is to be searched and summed for the weights contained in each of the 59 sets and the sums are to be stored in a single column.

This seemingly straightforward problem can be done either in an efficient way or in an exceedingly inefficient way. Indeed it can be done so inefficiently as to use six minutes of machine time rather than one minute. This time is taken in running back and forth on the system tape 100 or more times to pull in subroutines which are in the transient files. The following two figures show how a simple change in the arrangement of the data and the instructions can cut the machine time from six minutes to one minute. Because of this possible pitfall, there is included in tables D and E the distribution of the OMNITAB Commands among the system files. These tables will enable a user to determine how many times his problem will call upon the system to read transient files during the calculation. The swapping of transient files only takes 0.05 minutes, and is really of no consequence in most calculations. The swapping of files is of real consequence when it is executed hundreds of times, as could well be the case in the repeat mode.

While the present arrangement of the commands resolves the sample problem, it will obviously not resolve all. The user is therefore cautioned to be careful when calling on several files in the repeat mode.

```
OMNITAB CONTROLLED CLEARANCE AIR GAGE DATA REDUCTION  11/10/64
NOSUMMARY
READ    1  2  2  3
3.     2.547861   6.364843   63.643294
4.     2.027874   5.065857   50.654483
 21.    20.001400   49.965747   499.617126
 22.    20.000100   49.962499   499.584652
 23.    20.001200   49.965248   499.612129
 24.    20.001200   49.965248   499.612129
 25.    20.000700   49.963998   499.599640
 26.    20.000500   49.963499   499.594643
                        .
                        .
                        .
 1.   0.002205   0.005508   0.055079
 .500    0.001102   0.002754   0.027534
 .200    0.000441   0.001102   0.011016
 .100    0.000221   0.000552   0.005520
 .050    0.000110   0.000275   0.002748
 .020    0.000044   0.000110   0.001099
 .010    0.000022   0.000055   0.000550
35. 1. 256.18758  2548.68016
151.  1.          9.99356     99.92752
SORT 1 2 3
READ   4 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19
145.11 35. 26. 27. 51. 12. 79. 100. 10. 2. .2 .2 9. 9. 9. 9.
145.12 35. 26. 27. 51. 12. 79. 100. 5. 2. 2. .5 9. 9. 9. 9.
145.13 35. 26. 27. 51. 12. 79. 100. 10. .5 .2 .1 9. 9. 9. 9.
145.21 3. 21. 22. 23. 24. 151. 76. 50. 10. .2 .2 9. 9. 9. 9.
145.22 3. 21. 22. 23. 24. 151. 76. 50. 10. .5 9. 9. 9. 9. 9.
145.23 3. 21. 22. 23. 24. 151. 76. 50. 10. .2 .1 9. 9. 9. 9.
145.31 3. 21. 22. 151. 76. 20. 10. 5. 1. .5 .2 9. 9. 9. 9.
145.32 3. 21. 22. 151. 76. 20. 10. 5. 1. .5 9. 9. 9. 9. 9.
145.33 3. 21. 22. 151. 76. 20. 10. 5. 1. .5 .1 9. 9. 9. 9.
                                   .
                                   .
                                   .
148.31 3. 21. 51. 12. 79. 50. 10. 5. 2. 2. .5 9. 9. 9. 9.
148.32 3. 21. 51. 12. 79. 50. 10. 5. 2. 2. .5 9. 9. 9. 9.
148.33 3. 21. 51. 12. 79. 50. 10. 5. 2. 2. .2 .2 9. 9. 9.
BEGIN
 TRANSPOSE 1 21
 PROMOTE 3 21 22
 SEARCH OLD X IN 1 NEW X 22 OLD Y 2 NEW Y 23
 SUM   23 24
 DEFINE 1 24 1 40
 ERASE 21 22 23 24
 INCREMENT 5 0 0 1 0
 INCREMENT 1 1 0
FINISH
REPEAT 1 8 59 TIMES
HEAD 40/ AO-PSI
FIXED   5
PRINT 40
       STOP
```

Figure 11-3. An example of a very inefficient solution of a problem requiring commands from different OMNITAB files. Since TRANSPOSE and SEARCH are in different transient files, these files are swapped 118 times during the course of this calculation. See figure 11-4 for an efficient solution.

```
OMNITAB CONTROLLED CLEARANCE AIR GAGE DATA REDUCTION  11/10/64
NOSUMMARY
READ   4 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19
145.11 35. 26. 27. 51. 12. 79. 100. 10. 2. .2 .2 9. 9. 9. 9.
145.12 35. 26. 27. 51. 12. 79. 100. 5. 2. 2. .5 9. 9. 9. 9.
145.13 35. 26. 27. 51. 12. 79. 100. 10. .5 .2 .1 9. 9. 9. 9.
145.21 3. 21. 22. 23. 24. 151. 76. 50. 10. .2 .2 9. 9. 9. 9.
145.22 3. 21. 22. 23. 24. 151. 76. 50. 10. .5 9. 9. 9. 9. 9.
145.23 3. 21. 22. 23. 24. 151. 76. 50. 10. .2 .1 9. 9. 9. 9.
145.31 3. 21. 22. 151. 76. 20. 10. 5. 1. .5 .2 9. 9. 9. 9.
145.32 3. 21. 22. 151. 76. 20. 10. 5. 1. .5 9. 9. 9. 9. 9.
145.33 3. 21. 22. 151. 76. 20. 10. 5. 1. .5 .1 9. 9. 9. 9.
                                .
                                .
                                .
148.31 3. 21. 51. 12. 79. 50. 10. 5. 2. 2. .5 9. 9. 9. 9.
148.32 3. 21. 51. 12. 79. 50. 10. 5. 2. 2. .5 9. 9. 9. 9.
148.33 3. 21. 51. 12. 79. 50. 10. 5. 2. 2. .2 .2 9. 9. 9.
DEMOTE  36 LINES
RESET  0
READ   1  2  2  3
3.    2.547861   6.364843   63.643294
4.    2.027874   5.065857   50.654483
 21.   20.001400   49.965747   499.617126
 22.   20.000100   49.962499   499.584652
 23.   20.001200   49.965248   499.612129
                                .
                                .
                                .
151.  1.          9.99356      99.92752
SORT   1  2  3
SCALAR 1,4,36,16,0.,1,4
MTRANS ROW 37 ,4,30 ROWS BY 16 INTO 1,4
BEGIN
 SEARCH OLD X 1 NEW X 4 ,OLD Y 2 , NEW Y  38
 SUM   38, 39
 DEFINE 1  39  1  40
 INCREMENT 1, 0 ,1,0,0
 INCREMENT 3,0,0,1,0
FINISH
REPEAT 1,5 30 TIMES
MTRANS 67,4,29,16,1,4
 INCREMENT 1 BY 0,-30,0,0
REPEAT  1,5,29
HEAD   40/  A0-PSI
FIXED   5
PRINT       40
STOP
```

Figure 11-4. A more efficient solution to a search problem requiring commands from different OMNITAB files. First, more operations are done outside of the repeat mode using matrix operations. Next, the repeat mode contains instructions from a single transient file which is read in only once for the 30 repeats and once again for the 29 repeats. Here the transient files are swapped only 4 times instead of 118 times.

TABLE D

OMNITAB Commands in the Basic Package*

ABRIDGE	EXP	PRO
ABSOLUTE	EXPONENT	PROMOTE
ACOSD	FABRIDGE	PUNCH
ACOSH	FIXED	RAISE
ACOTD	FLOATING	READ
ACOTH	FOOTNOTE	REPEAT
ADD	FORMAT	RESET
ANTILOG	FPRINT	RMS
ARCCOS	FPUNCH	ROWSUM
ARCCOT	FREAD	SCRATCH
ARCSIN	GENERATE	SET
ARCTAN	HEA	SIN
ASIND	HEAD	SIND
ASINH	INCREMENT	SINH
ARAN	INDEX	SINTEGRAL
ATAND	IFEQ	SPACE
ATANH	IFGR	SQRT
AUT	IFLS	STOP
BEGIN	JOU	SU
CAP	LOGE	SUBTRACT
CGS	LOGTEN	SUM
CHANGE	MAXIMUN	SUMMARIZE
COM	MINIMUM	TAB
COMPARE	MKSA	TAN
COS	MULTIPLY	TAND
COSD	NEGEXP	TANH
COSH	NEWPAGE	TIT
COT	NOLIST	TITLE1
COTD	NOSUMMARY	TITLE2
COTH	NOTE	TITLE3
DEFINE	OMNITAB	TITLE4
DEMOTE	PAGE	TPRINT
DIVIDE	PARPRODUCT	FINISH
END	PARSUM	
ERASE	PLOT	
EXCHANGE	PRINT	

* The program to execute these commands is always in the core.

TABLE E

Distribution of OMNITAB Commands in the Transient Files*

FILE 1

ADIVIDE
ALOGE
ALOGTEN
AMULT
APRINT
ARAISE
ARRAYAVERAGE
AVERAGE
BACKTRANSFORM
BLOCKTRANSFER
CENSOR
CLOSEUP
COALESCE
COUNT
COLNORMALIZE
DIAGONALIZE
DIAGVECTOR
DELETE
DUPLICATE
FLIP
INSERT
ISETUP
ITERATE
LENGTH
MADD
MAXEL
MINEL
MMULT
MRAISE
MSUB
MTRANSPOSE
NRMAX
ORDER
CHOP
RESTMATRIX
ROWNORMALIZE
SCALAR
SEARCH
SELECT
SHORTEN
SORT
SEPARATE
SYMLTM
SYMUTM
TRACE
TRANSFORM
TRANSPOSE
VECMATRIX
VECMULT

FILE 2

CADD
CDIV
CMULT
CSUB
EXPAND
FIT
POLYFIT
PRODUCTS
SFIT
SOLVE
SPOLYFIT

FILE 3

COMPLETE
COSINTEGRAL
EINT
FORMULA
INCOMPLETE
LAGINTEGRAL
SININTEGRAL
SPLOT

FILE 4

EIGENVALUES
EIGENVECTORS
LINEAR
INVERT

FILE 5

DERIVATIVE
DIFFERENCE
DIVDIFFERENCE
INTERPOLATE
ISOLATE
MAXMIN
SDIFFERENCE
SDIVDIFFERENCE
STATISTICAL
SSTATISTICAL

FILE 6

ATOMIC
BEIONE
BEIZERO
BEJONE
BEJZERO
BEKZERO
BEKONE
BERBEI
BEYONE
BEYZERO
CERF
CTOF
DBERBEI
DKERKEI
EINSTEIN
ERROR
FTOC
GAMMA
GQUAD
HARMON
HSUB
KERKEI
LSUB
MOLWT
PARTFUNCTION
PFATOM
PFTRANSLATION
PSUB
STRUVE
TSUB
USUB

FILE 7

COMMANDS
MANUAL
WATSNU

*Words not listed are in the basic package (See Table D) which is always in the core.

APPENDIX I

CAUSES AND CONSEQUENCES OF ERROR DIAGNOSTICS

ARGUMENT IN SENTENCE INCORRECT

a. too few or too many arguments on the instruction card.

b. an argument is in the wrong form (i.e., fixed or floating instead of integer or vice versa).

c. calls INVLD.°

ARGUMENTS OF PRODUCTS MUST BE COLUMN NUMBERS

a. argument in instruction has the wrong form.

b. calls INVLD.

CHECK SEQUENCE OF X VALUES

a. the SEARCH requires that the first column specified contain numbers in increasing order.

b. calls INVLD.

CHECK SEQUENCING OF X VALUES

a. INTERPOLATE and DERIVATIVE require algebrically increasing X.

b. calls INVLD.

COLUMNS SPECIFIED IN ITERATE STATEMENT DO NOT CONTAIN ENOUGH VALUES TO ALLOW ITERATION

a. calls INVLD

CONTROL FORCED FROM REPEAT MODE

a. something illegal was encountered while in the REPEAT mode.

b. calls INVLD.

DEMOTE EXCEEDS ROW LIMIT, DEMOTION TERMINATES AT END OF COLUMN

a. when demote would give more than 101 lines only 101 are stored.

b. this notice is given the first 4 times under any one OMNITAB run.

DEMOTED OR PROMOTED ARRAY OFF THE WORK SHEET

a. promoting or demoting by more than 101 rows not allowed.

b. calls INVLD.

EIGENV HAS ,, ARGUMENTS

a. too few arguments on EIGENVECTOR.

b. calls INVLD.

° INVLD terminates the computation, reads the rest of the instructions, and flags illegal words and goes on to the next OMNITAB instruction, where computation resumes.

EQUATION SOLVE HAS ,, ARGUMENTS
a. too few arguments on LINEAR.
b. calls INVLD.

ERROR IN DEFINE INSTRUCTION
a. too few arguments in instruction.
b. calls INVLD.

ERROR IN REPEAT MODE
a. calls INVLD.

FRACTIONAL POWERS OF NEGATIVE NUMBERS NOT PERMITTED
POWER OF THE ABSOLUTE VALUE IS GIVEN
a. EXPAND does not allow imaginary numbers.
b. the program continues.

IMPROPER ISOLATE INSTRUCTION
a. too few arguments on card, or the form of the fourth number is wrong.
b. calls INVLD.

IMPROPER PLOT INSTRUCTION
a. too many arguments on card.
b. if last number is floating point all four last numbers must be floating point. If limits for plot are given, all four must be given X min, X max, Y min, Y max.
c. calls INVLD.

INSTRUCTION IS DESTROYING ORIGINAL DATA TO BE DUPLICATED, COALESCED OR ARRAYAVERAGED, CHANGE STORAGE LOCATION
a. calls INVLD.

INVALID INPUT TO POLYFIT, FIT OR SOLVE
a. too few points for order specified.
b. or the weights may all be zero.
c. or too few arguments are given.
d. calls INVLD.

INVERT HAS ,, ARGUMENTS
a. too few arguments on INVERT.
b. calls INVLD.

ISETUP OR ITERATE HAS FOUND ,, VALUES
a. if no values are found INVLD is called.
b. if more than 100 rows (i.e. 20 values) are found by them NRMAX is set to 100 and 100 values are stored.

MATRIX IS SINGULAR
a. a singular matrix cannot be solved.
b. calls INVLD.

MAXIMUM OF 10 ARGUMENTS ALLOWED IN REPEAT MODE
   a. to conserve storage space only ten arguments (numbers) are allowed on each instruction inside the REPEAT MODE.
   b. calls INVLD.

NO FURTHER WARNING ON DEMOTE WILL BE GIVEN
   a. when 5 DEMOTEs under one OMNITAB run exceed 101 this printout is given.
   b. only 101 values are stored without further notice.

NO MAXIMUM OR MINIMUM FOUND WITH MAXMIN
   a. the program continues.

ONLY 101 VALUES HAVE BEEN SET INTO COLUMN
   a. under SET, only 101 arguments are stored even if more are given.

PLEASE REMOVE DATA FROM COLS ++ THRU ++
   a. SELECT does not allow any storage in or input from columns 41 - 46.
   b. calls INVLD.

PRODUCTS MUST HAVE AT LEAST THREE ARGUMENTS
   a. too few arguments on the instruction PRODUCTS.
   b. calls INVLD.

SUBROUTINE IS NOT IN DECK
   a. INTEGRAL has not been written.
   b. calls INVLD.

THE CORRECT SPELLING OF THE INSTRUCTION IS SEPARATE
   a. warning is printed and computation continues as SEPARATE.

THE PRODUCT OF N=,, M=,, MUST NOT EXCEED 101
   a. one cannot make a vector longer than 101 row out of matrix with VECMAT.
   b. calls INVLD.

THE PROGRAM GENERATED ONLY 101 VALUES
   a. no matter what the range under GENERATE, only a maximum of 101 will be generated.
   b. the program continues.

THE PROGRAM READ ONLY 101 DATA CARDS
   a. under READ, only 101 data cards are stored even if more are given.
   b. the program continues.

TOO MANY INSTRUCTIONS IN REPEAT MODE

a. more than 150 instructions were placed between BEGIN and FINISH.

b. calls INVLD.

VALUE ** NOT FOUND FOR SHORTEN

a. transfers designated columns and does not reset NRMAX.

b. the program continues.

VALUE ,, NOT FOUND IN ISOLATE OR MAXMIN INSTRUCTION

a. the program continues.

VALUE ** WAS NOT FOUND BY COALESCE OR ARRAYAVERAGE, ZEROS RETURNED

a. the program continues.

101 VALUES OF INSERT STORED

a. only 101 values are stored and NRMAX is set to 101.

X IS DUPLICATED IN A TRIAD

a. in MAXMIN three adjacent points are examined; if two of the three X's are identical, this is printed out and no calculation is done for those three points. Program continues for next three points.

YOU FORGOT THE DATE ON THE OMNITAB CARD

a. the program expects at least two numbers on the OMNITAB card.

b. the program continues.

ILLEGAL CONSTANT ILLEGAL CONSTANT

a. an asterisk,*, cannot be used except when calling for one of the fundamental constants.

b. calls INVLD.

ILLEGAL INSTRUCTIONXXXXXX***ILLEGAL INSTRUCTION***

a. given whenever INVLD is called.

b. computation is stopped until an OMNITAB card is encountered.

*** LOGARITHM OR BASE OF POWER NEGATIVE OR ZERO ***

a. a negative number is raised to a floating point number, the power is truncated to an integer and program continues.

APPENDIX II

BRANCHING INSTRUCTIONS

In Section 6·1, one finds a description of the instruction COMPARE which is useful in exiting from a repeat loop after all of the values in two designated columns match row by row to within the stated tolerance. This command is useful in iterating each of the elements in a vector to a fixed set of values or until the series of iterants settle down to a set of stable values. In the latter case, the result of the nth iteration is compared with those of the (n−1)st. The two columns are interchanged after each iteration, until the comparison is satisfied.

The COMPARE instruction has its most important application in computing the inverse of a function as is illustrated in Problem 6–4 (page 170), as well as in the evaluation of power series as in Problem 6–8 (page 179). It should be noted that when used, the COMPARE instruction must be imbedded in the stored set.

When applied to a problem in chemistry where it is important to check a mass balance of 6 or more atoms, a single compare instruction takes the place of many separate IF statements required in ordinary programming techniques.

It is interesting that during the four or more years that OMNITAB was being designed, programmed, and applied, the need for more conventional (by programming standards) branching capabilities did not arise. Only at the very end, when OMNITAB was applied to the solution of a complex problem in chemical equilibrium of air at high temperatures involving the solution of large sets of non-linear equations, did it become clear that additional conditional branching instructions would be useful. These were programmed forthwith, added to the basic package, and then ignored in the text. Their presence in the system was not disclosed because of the reluctance of the authors at that time to have OMNITAB viewed as still another programming language.

Three branching instructions are provided in OMNITAB: IFEQ compares two columns of numbers and if each element of the first is equal to the corresponding element of the second, certain contiguous stored instructions are executed (repeated) a specified number of times. If the equality is not satisfied for each of the numbers in the column, an alternate set of instructions is excuted the number of times designated. Thus:

IFEQ 2, 3, 1, 10, 4, 11, 18, 5

causes instructions 1 through 10 of the stored set to be repeated four times if values in column 2 equal those in column 3. If this equality is not satisfied, the calculation proceeds to instruction 11 through 18 which are repeated 5 times. Unlike the COMPARE instructions, the IFEQ instruction is always outside of the stored set or as is now the accepted jargon of OMNITAB users ''outside of the repeat mode.''

The above operating characteristics of the instruction IFEQ apply to the instruction IFLS (if less) and IFGR (if greater).

APPENDIX III

A Statistical Consultant's View of OMNITAB

by

David Hogben

II·1 Examples of the Exploitation of the OMNITAB Commands

Since OMNITAB became fully operational some three years ago, it has been used extensively by the staff of the Statistical Engineering Laboratory to solve a variety of problems which have arisen while providing consultative services and also in basic research· Some of this experience is passed on now in the anticipation that it may be of use to others· The notes and thoughts are based on the experience of too many users to mention, but fruitful discussions with Brian L· Joiner and Joan R· Rosenblatt have been appreciated·

The structure and features of OMNITAB provide a tool far more powerful than is evident to the casual reader· A belief, common among those who have had experience writing programs in other languages, is that OMNITAB is fine for doing simple calculations, but is inadequate for more sophisticated problems· Two objections have been that OMNITAB does not have the capability of logical branching and does not have adequate capability for modifying instructions· Now it is realized that the observations of the casual observer are virtually unfounded· OMNITAB is a general-purpose program rather than an all-purpose program, but the user who has had some experience with OMNITAB shortly realizes the generality is far beyond that which can be gleaned from a mere reading of this manual·

In addition to the features discussed fully in the beginning chapters, such as freedom from formating, simple and liberal rules, etc·; there are other features of considerable advantage such as flexibility, reliability and the power of numerous manipulative commands· These are discussed more fully in what follows·

OMNITAB permits a consultant to be productive without facility with FORTRAN and without the ready assistance of professional programmers· Applied statistics requires extensive and varied computation, so OMNITAB has been used often· While in the beginning a number of us approched OMNITAB with some reservation, now the consensus is that it would be difficult to get along without OMNITAB·

Sometimes all that has been necessary is a STATISTICAL ANALYSIS or FIT· Usually these programs* are written and punched in minutes· At other times it has been necessary to analyze a partially balanced incomplete block design or solve a problem in non-linear regression· Programs for these problems have been written free of errors in less than a half hour·

Writing a program is one thing and debugging it is another· After a little experience it is common for OMNITAB users to write programs free of programming errors· The errors that do occur are usually easily detected and corrected· The following numbered examples illustrate some of the features which make OMNITAB such a useful tool·

- - - - - - - - - - -

*Here and in what follows the word program is used to denote a set of OMNITAB instructions·

1. It is hard to overemphasize the importance and usefulness of self-teaching, described in section 4.6. One can learn the basic rules of OMNITAB very quickly, say within two hours, and then proceed on his own. This is very valuable for those of us who have difficulty reading manuals carefully.

Many times little programs have been added (at virtually no cost) at the end of a program to clarify a particular instruction. For example, it is not completely clear whether the command RMS described on page 38 stores

$$\sqrt{\Sigma x_i^2/n}\ ,\ \sqrt{\Sigma(x_i-\bar{x})^2/n}\ \text{ or } \sqrt{\Sigma(x_i-\bar{x})^2/(n-1)}\,.$$

However, the question is readily answered by using the following instructions:

```
OMNITAB
GENERATE 1. (1.) 5. STORE 1
RMS 1 STORE 2
PRINT 1 2
STOP
```

(Note, Σx_i^2 for integers is easily obtained from the formula $1^1 + 2^2 + \ldots + n^2 = n(n+1)(2n+1)/6$ so that $\Sigma x_i^2/n = (n+1)(2n+1)/6$.) It would be easy to write many programs like this in a short period of time and very quickly become an ''expert'' user. The ease with which this can be done is sometimes overlooked or unappreciated. The technique of self-teaching has been used effectively in class instruction.

2. An obstacle faced by non-programmers is that with some languages, such as FORTRAN, it is almost necessary to be an expert programmer to do even the simplest calculation. To do anything in FORTRAN, it is necessary to understand some of the most troublesome statements of the language such as FORMAT, DIMENSION and WRITE. An OMNITAB user with no computing experience can being writing programs to perform non-trivial computations with less than two hours' study. By self-teaching he can proceed at his own speed to become an ''expert.''

3. The very nature of OMNITAB is such as to make logical branching unnecessary in most instances. In others it may be possible with a little ingenuity. For example, the following instructions provide for replacing $\log_{10}(a_i/b_i)$ by the constant c whenever $a_i \leq 0$. Capital letters A, B, D, and F represent column numbers; corresponding lower case letters represent elements in the columns. We assume $b_i > 0$.

	$a \leq 0$	$a > 0$
CENSOR A FOR 0.0 REPLACE B STORE D	$d = b$	$d = a$
DIVIDE D BY B STORE F	$f=b/b=1$	$f=a/b$
LOGTEN F STORE F	$f = 0$	$f=\log_{10}(a/b)$
SUBTRACT A FROM D STORE D	$d = b-a$	$d = 0$
DIVIDE D BY D STORE D	$d = 1$	$d = 0$
MULTIPLY D BY c. BY 1.0 ADD TO F	$f = c$	$f=\log_{10}(a/b)$

4. A problem that occurred in a program for non-linear estimation was to find a way to restore an instruction which was incremented in the repeat mode. Since COMPARE was one of the stored instructions, the actual number of cycles excuted was unknown. The relevant part of the program was

```
   Instr.      Command

               BEGIN
     .           .
     .           .
     .           .
    14         BLOCKTRANSFER  ,, ++   ,, ,,   [1],++
    15         INCREMENT  14  0,0   0,0   1,0
     .           .
     .           .
     .           .
    34         COMPARE  4  30  5.E-8
    35           .
               FINISH
               REPEAT  1  35  20
```

The solution is given below. The asterisk denotes instructions added. Columns 45 and 46 were arbitrarily chosen.

```
   Instr.      Command

     .           .
     .           .
     .           .
    14         BLOCKTRANSFER  ,, ++   ,, ,,   1  ,,
    15         INCREMENT  14  0,0   0,0   1,0
   *16         ADD    1.  45  45
     .           .
     .           .
     .           .
    35         COMPARE  4  30  5.E-8
    36           .
   *37         INCREMENT 14  0  0  0  0  -1  0
   *38         SUBTRACT  1.  45  45
   *39         COMPARE  46  45  to .005
               FINISH
               REPEAT  1  36  20
              *ADD      0.  0.  46
              *REPEAT  37  39  20
```

5. The user of OMNITAB is not required to use subroutines, many of which have rigid, complicated instructions for their use. Nevertheless, in the analysis of data certain methods are used frequently and it is sometimes useful to write a code once, so that it can be used repeatedly rather than be rewritten each time. This is particularly true for methods which are lengthy to program. A very informal library of OMNITAB programs has been established in The Statistical Engineer Laboratory for such analyses. Included, for example, in the library are Exponential Probability Plot for Censored Samples (B. L. Joiner), Analysis of Residuals (D. Hogben) and the Kolmogorov Goodness-of-Fit Test (J. R. Rosenblatt).

6. The package of matrix and array commands can be used in obvious ways, and with a little imagination they can be used to construct and manipulate two dimensional arrays. Adding a constant ''a'' to the two-way array B can be expressed in matrix notation as $a11' + B = C$, where 1 is a column vector of ones. An example of constructing a design matrix is given in section 4.

7. A nicety that has been used occasionally is to insert blank cards in a program to separate data from executable commands and to subdivide the commands into logical units. The blank cards are ignored by the computer, but produce a blank line when the list of commands is printed.

8. It has been helpful to indicate in the margin of this Handbook the number of arguments for each matrix and array instruction on pages 184-186 and 191-193. A check of punched cards against these numbers significantly reduces the number of programming errors.

9. On the surface it seems inefficient and costly for scientists to punch their own cards. But let's examine this more closely. Since the language of OMNITAB is close to technical English and the number of instructions in a program is usually small, the time to punch cards is often of the order of a few minutes. This time can also be spent checking the program and correcting silly mistakes. The time taken is usually considerably less than that required to doublecheck the commands, rewrite the command with appropriate instructions for punching, see that they reach the appropriate card punch operator and do whatever repunching is necessary. Moreover, there is something to be said for ''getting your hands dirty.'' The statistician R. A. Fisher is reputed to have spent two to three hours a day at a desk calculator. Some people believe this helped the development of his amazing intuitive power or at least helped focus his attention on ''real'' problems.

10. Input to OMNITAB is in free-field format. This is certainly a desirable flexibility. Those of us who have been annoyed by having to punch cards so that the information starts in a specific card column such as column 7, are pleased to be able to start in any column including column 1. There are indeed a few exceptions. The commands FORMAT, TITLE, and OMNITAB must start in card column 1. Indeed most users start their instructions in column 1. The rule - when in doubt start in column 1 - is always worth following.

11. Some of the instructions may (or do) change the current value of NRMAX. It has proved helpful to collect the names of these instructions in one place. The instructions that have been found so far are:

Input: OMNITAB, READ, FREAD, GENERATE, SET

Manipulative: ERASE, DEMOTE, RESET, SHORTEN, INSERT

Statistical and Numerical Analysis: GQUAD

Repeat: ISOLATE, ITERATE, ISETUP

Array: DUPLICATE

12. Although flow charts are unnecessary for the use of OMNITAB, it has sometimes been helpful in more complex problems to keep track of where data and intermediate results are stored. The following chart, reduced from 8 1/2 x 11, illustrates the book-keeping for a particular problem.

13. Many users have a virtually unshakeable trust in computers and computer programs. Others who have had the experience of running exactly the same program on two different computers (or even different models of the same computer) and getting different answers, are more cautious. The careful user will require evidence that the computer program is reliable. Many competent scientists working within their own specialty, have contributed to OMNITAB. A certain reliability has thus been achieved which probably would not have been possible if the program had been constructed by only one person.

```
OMNITAB        12/15/67        NON-LINEAR LEAST SQUARES ESTIMATION
SET CODED Y MEASUREMENTS IN COL 1
-6 -1 1 4 8 11 9 14 16 17 15 18 19 20 21 22 26 22 27
ADD 200.0 TO COL 1 AND STORE DECODED MEASUREMENTS IN COL 40
GENERATE INDEPENDENT VARIABLE X EQUAL TO 0.(5.)70.(10.)110. IN COL 3
ADD 228.0 TO 0.0 STORE STARTING VALUE OF THETA ONE IN COL 21
ADD -1.83 TO 0.0 STORE STARTING VALUE OF THETA TWO IN COL 22
ADD -.0257 TO 0.0 STORE STARTING VALUE OF THETA THREE IN COL 23
ADD 1.0 TO 0.0 STORE WEIGHTS IN COL 2
BEGIN ITERATIVE PROCESS TO OBTAIN ESTIMATES AND SUM OF SQUARES
ADD 0.0 TO COL 7 AND STORE SUM OF SQUARES IN COL 8
SFIT Z IN 6 WTS IN 2 TO FONE IN 11 FTWO IN 12 AND FTHREE IN 13
MTRANSPOSE  2,45 WITH 3,1 STORE DELTA THETA IN 2,21
MADD 1,21 WITH 1,3 TO 2,21  STORE 1,21
DUPLICATE 18 TIMES THE ARRAY IN 1,21   1,3 AND STORE NEW THETAS IN 2,21
MULTIPLY X IN COL 3 BY THETA THREE IN COL 23 AND STORE IN COL 30
ADD COL 22 TO COL 30 AND STORE IN COL 30
EXPONENT OF COL 30 STORE IN COL 30
MULTIPLY COL 30 BY COL 21 AND STORE IN COL 12
MULTIPLY COL 12 BY -1.0 AND STORE F TWO IN COL 12
MULTIPLY COL 12 BY X IN COL 3 AND STORE F THREE IN COL 13
SUBTRACT COL 30 FROM 1.0 AND STORE F ONE IN COL 11
MULTIPLY FONE IN COL 11 BY THETA ONE IN COL 21 AND STORE F IN COL 5
SUBTRACT F IN COL 5 FROM Y IN COL 40 AND STORE Z IN COL 6
MULTIPLY Z IN COL 6 BY Z IN COL 6 AND STORE Z SQUARED IN COL 7
SUM 7 AND STORE SUM OF Z SQUARED IN COL 7
DEFINE 0.0 AS ROW 1 OF COL 4 THIS IS ITERATION NUMBER
ABRIDGE ROW 1 OF 4 7 21 22 23
INCREMENT INSTRUCTION 17 BY 1.0     0.0
COMPARE NEW SS IN COL 7 WITH OLD SS IN COL 8 TO TOLERANCE OF 5E-7
FINISH
NEWPAGE
SPACE 2
REPEAT 6 19 1 TO OBTAIN INITIAL VALUES OF F, FONE, FTWO, FTHREE AND SS
REPEAT INSTRUCTIONS 1 THRU 20  100 TIMES
STOP
```

OMNITAB CODING CHART

NON-LINEAR LEAST SQUARES ESTIMATION

TITLE: $f(\theta_1, \theta_2, \theta_3; x) = \theta_1\left(1 - e^{\theta_2 + \theta_3 x}\right)$ DATE: 12/8/67

COL.	CONTENTS	COL.	CONTENTS
1	y = measurements	24	
2	weights $\equiv 1.0$	25	
3	x = 0(5) 70 (10) 110	26	
4	iteration = 0, 1, 2, ...	27	
5	$f = f(\theta_1, \theta_2, \theta_3; x)$	28	
6	$z = y - f$	29	
7	$z^2 \longrightarrow \sum_{i=1}^{n} z_i^2$ = Residual sum of squares	30	$\hat{\theta}_3 x \longrightarrow \hat{\theta}_2 + \hat{\theta}_3 x \longrightarrow e^{\hat{\theta}_2 + \hat{\theta}_3 x}$
8	Previous $\sum z_i^2$ for COMPARE	31	
9		32	
10		33	
11	$f_1 = \frac{\partial f}{\partial \theta_1} = 1 - e^{\theta_2 + \theta_3 x}$	34	
12	$f_2 = -\theta_1 e^{\theta_2 + \theta_3 x}$	35	
13	$f_3 = x f_2$	36	
14		37	
15		38	
16		39	
17		40	
18		41	
19		42	
20		43	
21	$\hat{\theta}_1$ } Replaced by new values	44	
22	$\hat{\theta}_2$ } after each iteration	45	Coeff's from SFIT = $\Delta\theta$'s
23	$\hat{\theta}_3$ }	46	Residuals

COMMENTS: Method: Linear Gauss

new θ = old $\theta + \Delta\theta$

$\Delta\theta$'s are coefficients from fitting z to f_1, f_2 and f_3.

II·2 Notes on Some Specific Commands

No user's manual can be complete. Often certain subtleties are observed only after extensive experience. The notes given here on specific instructions are based on the experience of many users. A lot of the information was obtained from self-teaching. The commands are listed in order of page number on which they are defined.

1. SQRT OF $$, STORE IN ++ (p. 20).

INVLD is called and computation stopped if any number in the column is negative. Mathematical zero can be either +0. or −0. in the computer. In the latter case computation would halt.

If some of the numbers in a column may be negative, but it is nevertheless desirable to continue computations for the positive numbers, precede the SQRT command by PRINT (or some other device for recording the location of negative numbers) and then use ABSOLUTE.

2. READ COL ++, ++, ++, ++, ... (p. 28).
SET IN COL ++, THE ARG ON THE FOLLOWING CARDS (p. 29).

Data is read and the command ''... is terminated when an executable statement is encountered.'' Extra care should be taken to ensure that the first executable command following is spelled correctly or the card will be interpreted as a data card. For example, the cards

```
SET 1
X 1.2 Y 3.4  Z 5.6
ORDIR 1
```

will cause 1.2, 3.4, 5.6, and 1.0 to be read into column 1. ORDIR, a misspelling of ORDER, is not an executable command. Consequently, ORDIR will be treated as a comment like X, Y and Z. (If ORDIR followed an excutable command, INVLD would be called, computation stopped and an error diagnostic given.)

3. PLOT COLS ++, ++, AGAINST ++, ABSC FROM ** TO ** ORD FROM ** TO ** (p.30).

a. It is permissible to provide ranges which give a scale in decreasing order, e.g.

PLOT COL 46 AGAINST 23 ABSC FROM 100. TO 0. ORD FROM 0. TO 100.

b. There is room for one line of title (using TITLE1 and TITLE3) at the top and one line at the bottom on which numeric information can be printed using ABRIDGE or alpha and/or numeric information using FABRIDGE. The instructions TITLE2, TITLE4 and FOOTNOTE are ignored. The instruction NOTE will cause the information to be printed on the next page.

4· FORMAT (as appropriate) (p· 32)·

a· The I and A formats may be used to input, ORDER, SORT, BLOCKTRANSFER, DUPLICATE and output data in the same type format· However, the E or F format must be used for any type of arithmetic operation· The X and H formats may be used as in FORTRAN·

b· Specifying more fields in the FORMAT statement than columns in the FPRINT instruction causes no difficulty, (only the first necessary fields are used)· If there are more columns indicated on the FPRINT card than fields on the FORMAT card, a given worksheet row is printed on one line until all fields in the FORMAT have been used, then the printing of the row continues on a new line with the FORMAT being reused and so on until the entire worksheet row has been printed· The printing of the next row starts on a new line with the beginning of the FORMAT· For example, the instructions: FORMAT (X,F15·5) and FPRINT 45 46 will cause the numbers in (1,45); (1,46); (2,45); (2,46); ·····; (NRMAX,45); (NRMAX,46) to be printed consecutively in one column·

c· There are occasions when one may want to print row titles· This can be done easily by reading in and printing the titles using the A format and alloting one column for each six or less characters (including blanks)· For example, the following instructions could be used to print row titles for the contents of column 45 after SFIT with three vectors:

```
FORMAT   (3A6)
FREAD  4  CARDS INTO 31  32  33
STANDARD DEVIATION
ESTIMATE OF A
ESTIMATE OF B
ESTIMATE OF C
FORMAT  (1H ,3A6, E15·6)
RESET  4
FPRINT  31  32  33  45
```

5· SPACE ,, (p· 33)·

SPACE will be ignored if used just before PRINT since PRINT calls for a new page·

6· DEFINE ** AS ROW ,, COL ++ (p· 36)·

DEFINE 1·0 AS ROW 105 COL 31 is interpreted as DEFINE 1·0 AS ROW 4 COL 32, hence care is needed when incrementing this instruction in the repeat mode· Of course, there may be cases where this interpretation is desirable·

7· PARSUM COL ++, STORE IN COL ++ (p· 39)·

This instruction requires two column numbers and no diagnostic is given if only one column number is on the card· The computer seeks a second argument and the results are hard to predict·

8. RESET ,, ++ (p. 36).

This instruction makes several problems easy which would be difficult without it. Often one wants to rest NRMAX to its unknown, original value after it has been changed, either within or outside the repeat mode. The original value of NRMAX can be stored in the worksheet by using LENGTH C and then NRMAX can be restored when necessary by using RESET 1, C.

9. COUNT LENGTH OF COL ++, STORE IN COL ++ (p. 40).

This instruction should be used with care. If x_1, x_2, and x_3 are non-zero numbers and NRMAX is 5 then the results stored in the following examples will be 1, 5 and 2 respectively:

(i) $x_1, 0, 0, 0, x_2$
(ii) $x_1, x_2, x_3, 0, 0$
(iii) $x_1, x_2, 0, 0, 0$.

10. ERROR OF $$, STORE IN COL ++ (p. 44).

The normal probability integral

$$\mathrm{Gau}(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-\frac{1}{2}t^2} dt$$

can be obtained from the ERROR function using the relation

$$\mathrm{Gau}(x) = 0.5\,[1 + \mathrm{erf}(x/\sqrt{2})].$$

11. FIT COL ++, WEIGHTS IN COL ++, VECTORS IN ++, ++, ... (p. 112).

The degrees of freedom for the standard deviation is equal to the number of non-zero weights (NZW) minus the number of vectors (n) fitted, i.e.

$$s = \sqrt{\Sigma(\Delta y_i)^2/(NZW - n)}.$$

12. COMPARE COL ++ AND ++ TO A TOLERANCE OF $$ (p. 157).

a. If one's interest is in relative error (significant digits) of column X compared to Y, let the tolerance $$ be a constant K, e.g. K = 1.E-8 (1×10^{-8}). Then the instruction COMPARE X Y K causes the program to exit from the repeat mode if $|X-Y| < K$. In iterative procedures, K = 1.E-m gives X accurate to m significant digits, but this will be a conservative use if the first significant digit of X is small. If the tolerance is 1.E-m, the error will be in the (m+1)st significant digit.

b. If interest is in absolute error (decimal places), let $$ be a column, say Z, and precede the instruction COMPARE X Y Z by DIVIDE K Y Z. The program will exit from the repeat mode if |X−Y| < K.

13. MMULT [A] IN ,, ++ R= ,, C= ,, BY [B] IN ,, ++ WITH ,, COLUMNS, START STORING IN ,, ++ (p. 184).

For computing $A_{30x20}\ B_{20x40} = C_{30x40}$, the instruction

MMULT 1,41 30,20 81,1 40 1,1 ,

which should read

MMULT 41,1 30,20 81,1 40 1,1 ;

puts the matrix outside the worksheet and havoc may result. In one instance, it caused the program to go into an infinite loop.

14. EIGENVECTORS OF [A] IN ,, ++ R= ,, PUT ROOTS IN COL ++ AND VECTORS STARTING IN ,, ++ (p. 186).

a. Roots are stored in decreasing order.

b. Mathematically zero roots may be far from zero in the computer due to round-off error.

c. The eigenvectors are normalized, i.e., $\Sigma x_i^2 = 1$ and are stored as an array of column vectors in order corresponding to the order of the roots.

d. Subject to (b), the number of non-zero roots gives the rank of a normal matrix (i.e. AA′ = A′A).

15. INVERT [A] IN ,, ++ R= ,, START STORING IN ,, ++ (p. 186).

The determinant of the specified matrix is automatically stored in row 1 of column 46.

16. BLOCKTRANSFER A IN ,, ++ R= ,, C= ,, TO ,, ++ (p. 192).

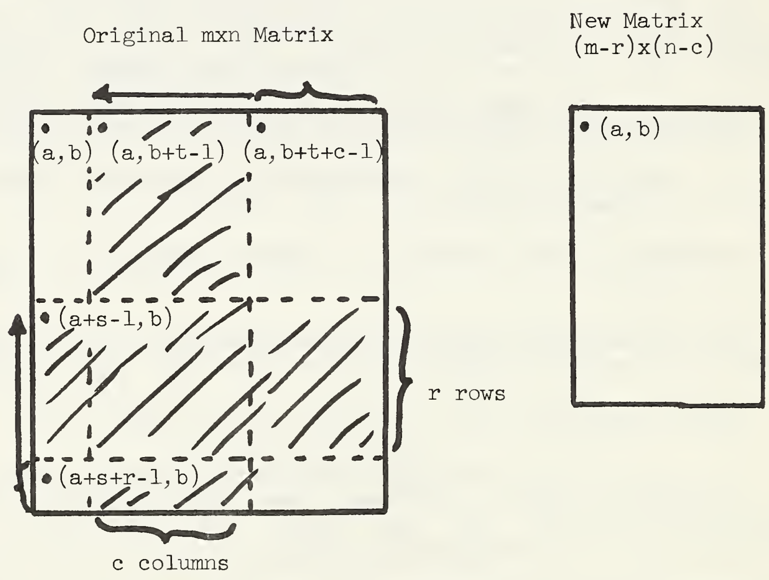
The instruction can be used when the new location overlaps the original array provided the direction of movement in the worksheet is up or to the left. It will not work if the direction is to the right or down.

Note, it is illegal to use R=0 in a BLOCKTRANSFER instruction. INVLD is called and computation stopped.

Often it is desirable to eliminate some rows and columns of a matrix and close up the resulting spaces, as illustrated in the sketch below. To eliminate r rows starting with the s th. row and c columns starting with the t th. column of an m x n matrix stored starting in (a,b) and have the resulting (m−r)x(n−c) matrix stored starting in (a,b), one can use the following instructions. The first instruction eliminates the c columns.

BLOCKTRANSFER (a,b+t+c−1) DIMENSIONS (m,n−t−c+1) TO (a,b+t−1)

BLOCKTRANSFER (a+s+r−1,b) DIMENSIONS (m−s−r+1,n) TO (a+s−1,b)



The instruction STATISTICAL ANALYSIS gives a frequency distribution in the automatic printout. The distribution is not stored and the number of cells, 10, is fixed. This is usually a minor disadvantage, but there are times when a frequency distribution is needed in the worksheet having a prechosen number of cells. A method of constructing frequency distributions is given below preceded by a technique for computing the number of values in a column less (or greater) than a specified constant.

Although frequency distributions are useful in many areas, the purpose here is more to illustrate certain facets of OMNITAB. Finding a method of constructing frequency distributions in a computer is more elusive than one would suspect at first glance. But I contend that it is considerably easier for a wide class of scientists to think and solve problems in OMNITAB than in a language more foreign to technical English. The wide variety of manipulative commands such as SHORTEN, CENSOR and COALESCE often reduces the size of a problem so that it is more easily comprehended and solved. Also, it is interesting to observe that just a few commands can represent a fair amount of logic and computing.

a. Suppose we have a column of numbers, X, a constant, k, and we want to calculate in OMNITAB the number of X values, for example, less than k which will be denoted by $n(X < k)$. This problem arises frequently, in one form or another.

Compute $Y = (X - k.)/|X - k.|$, then:

To find:	Perform operations successively on column Y
$n(X > k)$	(i) Censor for 0.0 replace by 0.0 (ii) Sum
$n(X \geq k)$	(i) Change sign (ii) Censor for 0.0 replace by 2.0 (iii) Subtract 1.0 (iv) Sum
$n(X = k)$	(i) Censor for −1.0 replace by 1.0 (ii) Subtract 1.0 (iii) Change sign (iv) Sum
$n(X < k)$	(i) Change sign (ii) Censor for 0.0 replace by 0.0 (iii) Sum
$n(X \leq k)$	(i) Censor for 0.0 replace by 2.0 (ii) Subtract 1.0 (iii) Sum

The following instructions will compute $n(X < k)$, $n(X = k)$ and $n(X > k)$ simultaneously, the results being stored in rows 1, 2 and 3 of column (R+1). D is any storage column.

```
SUB       k.  X  D
ABSOLUTE  D  Y
DIVIDE  D  Y  Y
ADD   0.0  1.0  (Y+1)
COALESCE   1,Y   101,2   1,R
SORT  R  CARRY ALONG  (R+1)
```

b. The following instructions divide x values into k cells of width c and compute the number of x values in each cell. The lower boundary of the first cell is denoted by a. The numbers k, c and a are to be specified. The program is such that the cells are non-overlapping and each x value falls into one and only one uniquely defined cell. Columns M, M+1, M+2, and M+3 are used for storage.

```
SUBTRACT   a  X  M
DIVIDE     M  c  M
CHOP       M  M
ADD  1.0   M  M
COALESCE ON FIRST COLUMN OF 1,M  101,2 STORE (M+2)
DIVIDE    (M+3)  (M+2)  (M+3)
SORT      (M+2)  CARRY ALONG  (M+3)
```

The cell number i = 1, 2, ...k is in column (M+2) and f_i, the frequency of x values in cell i, is in column (M+3), where f_i is the number of x's such that $a \leq x < a + ic$. This program does not give i or f_i for those cells where $f_i = 0$. This mild defect is overcome below.

To obtain i and f_i for those cells where $f_i = 0$, eliminate the SORT instruction in (a) and replace the COALESCE instruction by

```
BEGIN
COALESCE ON  1.0  1,M  101,2  1,(M+2)
INCREMENT  1  1.0  0,0  0,0  1,0
FINISH
REPEAT 1  2  k
```

The statistical treatment of experimental data frequently includes, at some intermediate point, an analysis of variance based on a particular statistical model, often one representing an experimental design. Many texts provide convenient computational formulas for specific designs. But these formulas were developed in the days of the desk calculator. Anyone wanting to have the numerical calculation performed on an electronic computer is immediately faced with two problems.

First, should he seek or write a subroutine appropriate for his particular problem or should he seek a general purpose analysis of variance program? The first alternative is seemingly neither appealing nor efficient. The second can be very frustrating. If a general purpose code exists, it is often cumbersome to use and may not cover the particular model in question.

The second problem concerns the accuracy of the computing algorithm used. For example, many people still insist on using the so-called computational formula

$$\sum_{i=1}^{n} (x_i - \bar{x})^2 = \sum x_i^2 - [\sum x_i]^2/n ,$$

despite the fact the right-hand formula is most inaccurate.

Some of us use a technique which has several desirable features and seems worthy of recommendation until the time comes when a whole new approach to statistical computing is developed. The technique, very simply, is to treat each problem as a problem in least squares (the general linear hypothesis).

The least squares approach is well known, but its computational advantages are often overlooked. The technique is completely general; the FIT instruction which is used is simple and there is substantial evidence that the algorithm used in FIT is of high quality for statistical applications.(1)

To illustrate the technique and ideas involved, two examples are given.

Suppose we merely wish to calculate the mean and standard deviation of a set of data. This can be done, of course, by using STATISTICAL ANALYSIS. But it can also be done using FIT. Briefly, the underlying statistical model here is(2)

- - - - - - - - - - - -

(1) LONGLEY, J. W. (1967). An appraisal of least squares programs for the electronic computer from the point of view of the user. J. Amer. Statist. Assoc., 62, 819-841.

(2) HICKS, C. R. (1964). Fundamental Concepts in the Design of Experiments. Holt, Rinehart and Winston.

$$y_i = \mu + e_i \quad ,$$

where the y_i represent the measurements, μ is the ''true'' mean and the e_i represent the random component. This model written in the least squares framework is

$$y_i = \mu x_i + e_i \quad ,$$

where x_i is identically equal to one and μ is a parameter to be estimated. The following instructions will provide the answer:

```
SET  Y  IN  COL  1
[measurements]
ADD  0.0  TO  1.0  STORE IN COL  2
FIT  Y IN COL  1  WTS IN COL  2  X IN COL 2
```

The automatic printout gives the coefficient, in this case $\bar{y}$ (the estimate of μ), its standard deviation ($s_{\bar{y}} = s/\sqrt{n}$), the standard deviation (s) and the residuals (deviations).

Any advantage here is dubious as STATIST gives more information and in a form more recognizable. But in the second example the advantage is made clearer.

Suppose we have a two-way table with three rows and four columns, as illustrated, for measurements y_{ij}.

	Col 1	Col 2	Col 3	Col 4
Row 1	y_{11}	y_{12}	y_{13}	y_{14}
Row 2	y_{21}	y_{22}	y_{23}	y_{24}
Row 3	y_{31}	y_{32}	y_{33}	y_{34}

A common statistical model in this situation, stated briefly, is

$$y_{ij} = \mu + \rho_i + \gamma_j + e_{ij} ,$$

with the constraints $$\sum_{i=1}^{r=3} \rho_i = \sum_{j=1}^{c=4} \gamma_j = 0 \quad .$$

The parameter μ represents the overall mean, ρ_i the effect of the ith row and γ_j the effect of the jth column. In the least squares framework the model can be written

$$y_k = \beta_1 x_{1k} + \beta_2 x_{2k} + \beta_3 x_{3k} + \beta_4 x_{4k} + \beta_5 x_{5k} + \beta_6 x_{6k} + \epsilon_k \quad ,$$

where $k=1,2,\cdots,rc=12$

and $\beta_1 = \mu$, $\beta_2 = \rho_1$, $\beta_3 = \rho_2$, $\beta_4 = \gamma_1$, $\beta_5 = \gamma_2$ and $\beta_6 = \gamma_3$ are the parameters to be estimated and x_{1k}, x_{2k}, $\cdots$, x_{6k} are fixed, known constants. As before, x_{1k} is identically equal to one,

$x_{2k} = 1, \quad x_{3k} = 0$, if y_k is in the first row

$= 0, \quad\quad = 1$, if y_k is in the second row

$= -1, \quad\quad = -1$, if y_k is in the third row.

The constraint $\sum_{i=1}^{r=3} \rho_i = 0$ implies that $\rho_3 = -\rho_1 - \rho_2$ so in fact we have two rather than three parameters to be estimated. It explains the relation $x_{2k} = x_{3k} = -1$. The constants x_{4k}, x_{5k} and x_{6k} are defined in a similar manner.

The model can be conveniently expressed in matrix notation as $y = X\beta + \epsilon$. For clarity the matrices are written out in full.

MATRIX NOTATION

$$y = \begin{vmatrix} y_{11} \\ y_{12} \\ y_{13} \\ y_{14} \\ y_{21} \\ y_{22} \\ y_{23} \\ y_{24} \\ y_{31} \\ y_{32} \\ y_{33} \\ y_{34} \end{vmatrix} = X\beta + \epsilon = \begin{vmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & -1 & -1 & -1 \\ 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & -1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 0 & 0 \\ 1 & -1 & -1 & 0 & 1 & 0 \\ 1 & -1 & -1 & 0 & 0 & 1 \\ 1 & -1 & -1 & -1 & -1 & -1 \end{vmatrix} \begin{vmatrix} \mu \\ \rho_1 \\ \rho_2 \\ \gamma_1 \\ \gamma_2 \\ \gamma_3 \end{vmatrix} + \begin{vmatrix} \epsilon_{11} \\ \epsilon_{12} \\ \epsilon_{13} \\ \epsilon_{14} \\ \epsilon_{21} \\ \epsilon_{22} \\ \epsilon_{23} \\ \epsilon_{24} \\ \epsilon_{31} \\ \epsilon_{32} \\ \epsilon_{33} \\ \epsilon_{34} \end{vmatrix}$$

With this basic background the computing is straightforward. One can SET the measurements in any specified column, READ in the design matrix X and then FIT. Often, it is convenient to compute the design matrix X using the matrix instructions as is illustrated in the following set of instructions for this problem. The X matrix is stored in the first twelve rows of columns 10 through 15.

```
OMNITAB   8/23/67  TWOWAY ANALYSIS
RESET  4
DIAGONALIZE  1. INTO COLS 13 TO 16
ADD  1. TO  0.  STORE COL 10
ADD  0.0  TO -1. STORE  16
MTRANS  1,16  3,1  STORE 4,13
DUPLICATE  2 TIMES  1,10  4, 6  STORE  5, 10
BLOCKTRANSFER    1,16  4,1  STORE  9,11
BLOCKTRANSFER    1,16  4,1  STORE  9,12
BLOCKTRANSFER    1,10  4,1  STORE  1,11
BLOCKTRANSFER    1,10  4,1  STORE  5,12
SET Y IN COL 1
(   .....   )
FIT Y IN COL 1 WTS IN COL 10 VECTORS 10 11 12 13 14 15
```

The sums of squares for an analysis of variance can be picked off from the automatic printout. The first SQUARED FOURIER COEFFICIENT (see for example p. 134) is the so-called correction factor or sum of squares due to the mean. The sum of the second and third SQUARED FOURIER COEFFICIENTS gives the Row sum of squares and the fourth, fifth and sixth give the Column sum of squares. The last SUM OF SQUARED RESIDUALS is the Residual sum of squares. Finally, the last FOURIER COEFFICIENT is the Total sum of squares.

The above analysis is easily modified for other experimental designs by simply reading in or constructing the appropriate design matrix. For example, if there are replications, all that is required is to duplicate the X matrix for each replication. The method is simple and gives a small chance of making programming errors. The residuals (or deviations), which are vital in assessing the adequacy of the model, are automatically available for STATISTICAL ANALYSIS and/or plotting. Also, in the least squares framework, problems of non-orthogonality and missing values (or rejected outliers) are minimized. All that is required is to erase the row containing the missing observation or assign zero weight to the observation and do separate FITs for each of the analyses. For example, in a two-way classification fitting the mean, rows and columns in that order would give the unadjusted row sum of squares and the adjusted (for rows) column sum of squares for testing differences between column means. Then fitting the mean, columns and rows in that order provides the unajusted column sum of squares and the adjusted row sum of squares for testing differences between row means. Experience has shown that programs can easily be written and punched in less than one-half hour, free of errors, to analyze data for more complex designs such as a replicated partially balanced incomplete block design.

To obtain the greatest assurance of numerical accuracy it is usually best to subtract a constant from all the measurements before using FIT. Often, the mean is used, but since this introduces some round-off error it may be preferable to subtract some other number like the smallest measurement or the median.

INDEX TO OMNITAB COMMANDS

THE NATIONAL BUREAU OF STANDARDS

The National Bureau of Standards is a principal focal point in the Federal Government for assuring maximum application of the physical and engineering sciences to the advancement of technology in industry and commerce. Its responsibilities include development and maintenance of the national standards of measurement, and the provisions of means for making measurements consistent with those standards; determination of physical constants and properties of materials; development of methods for testing materials, mechanisms, and structures, and making such tests as may be necessary, particularly for government agencies; cooperation in the establishment of standard practices for incorporation in codes and specifications; advisory service to government agencies on scientific and technical problems; invention and development of devices to serve special needs of the Government; assistance to industry, business, and consumers in the development and acceptance of commercial standards and simplified trade practice recommendations; administration of programs in cooperation with United States business groups and standards organizations for the development of international standards of practice; and maintenance of a clearinghouse for the collection and dissemination of scientific, technical, and engineering information. The scope of the Bureau's activities is suggested in the following listing of its three Institutes and their organizational units.

Institute for Basic Standards. Applied Mathematics. Electricity. Metrology. Mechanics. Heat. Atomic Physics. Physical Chemistry. Laboratory Astrophysics.* Radiation Physics. Radio Standards Laboratory:* Radio Standards Physics; Radio Standards Engineering. Office of Standard Reference Data.

Institute for Materials Research. Analytical Chemistry. Polymers. Metallurgy. Inorganic Materials. Reactor Radiations. Cryogenics.* Materials Evaluation Laboratory. Office of Standard Reference Materials.

Institute for Applied Technology. Building Research. Information Technology. Performance Test Development. Electronic Instrumentation. Textile and Apparel Technology Center. Technical Analysis. Office of Weights and Measures. Office of Engineering Standards. Office of Invention and Innovation. Office of Technical Resources. Clearinghouse of Federal Scientific and Technical Information.**

---

*Located at Boulder, Colorado, 80301.
**Located at 5285 Port Royal Road, Springfield, Virginia, 22171.

☆ U. S. GOVERNMENT PRINTING OFFICE : 1968 O - 291-189