ADDITIONAL ABSTRACTS OF IBM 650 LIBRARY PROGRAMS

Enclosed is the second distribution of 650 Library Program Abstracts for inclusion in the loose-leaf manual of "Library Program Abstracts for the IBM 650." The file numbers of the 32 abstracts included are:

1.1.004	1.6.015	5.2.008	6.0.016
1.2.006	1.6.016	5.2.009	8.2.005
1.3.008	2.0.007	6.0.010	8.2.006
1.3.009	2.0.008R	6.0.011	8.2.007
1.4.003	2.0.009	6.0.012	8.2.008
1.4.004	2.0.010	6.0.013	9.4.001
1.6.013	2.0.011	6.0.014	9.4.002
1.6.014	5.2.007	6.0.015	12.0.001

Also enclosed is the initial distribution of errata material. These pages apply to programs as indicated below, not to the abstracts; thus they will be of interest only to those persons who have received the pertinent write-ups and listings.

1.3.003	3.1.014	5.2.004	9.2.001
1.3.004	5.1.001	5.2.005	9.2.003
2.0.001	5.1.002	5.2.006	9.6.001
2.0.002	5.2.001	7.0.001	12.0.001

IBM assumes no responsibility for editing, evaluating, or checking the accuracy of the contributed programs. All technical questions should be directed to the original contributor.

A complete write-up for any of the programs abstracted to date may be obtained from F. E. Ross, Applied Programming Publications, IBM, 590 Madison Avenue, New York 22, N. Y. Each request should specify the program file number and the name of the account for which the request is made. Inasmuch as almost all of the write-ups are printed on ozalid masters which permit easy duplication, only one copy per account will be supplied.

F. E. Ross

March 21, 1957

650 PROGRAM LIBRARY MATERIAL

Table of Contents

Page

Inserts for the Manual of "Library Program Abstracts for the IBM 650":

Table of Contents (cont'd.)

Addenda/Errata for 650 Library Program Detailed Write-ups:

LIBRARY PROGRAM ABSTRACTS

for the

IBM 650

INITIAL DISTRIBUTION PAGES

1.1.001
1.1.002
1.1.003

1.2.001
1.2.002
1.2.003
1.2.004
1.2.005

1.3.001
1.3.002
1.3.003
1.3.004
1.3.005
1.3.006
1.3.007

1.4.001
1.4.002

1.6.001
1.6.002
1.6.003
1.6.004
1.6.005
1.6.006
1.6.007
1.6.008
1.6.009
1.6.010
1.6.011
1.6.012

2.0.001
2.0.002
2.0.003
2.0.004
2.0.005
2.0.006

3.1.001
3.1.002
3.1.003
3.1.004
3.1.005
3.1.006
3.1.007
3.1.008
3.1.009
3.1.010
3.1.011
3.1.012
3.1.013
3.1.014
3.1.015
3.1.016
3.1.017
3.1.018

3.2.001
3.2.002

4.0.001
4.0.002
4.0.003
4.0.004
4.0.005

5.1.001
5.1.002
5.1.003
5.1.004
5.1.005

5.2.001
5.2.002
5.2.003
5.2.004
5.2.005
5.2.006

6.0.001
6.0.002
6.0.003
6.0.004
6.0.005
6.0.006
6.0.007
6.0.008
6.0.009

7.0.001
7.0.002

8.1.001

8.2.001
8.2.002
8.2.003
8.2.004

9.2.001
9.2.002
9.2.003

9.5.001

9.6.001

10.1.001
10.1.002
10.1.003
10.1.004
10.1.005

PREFACE

This manual of IBM 650 Library Program Abstracts is published as a reference book for the library of 650 programs maintained at the headquarters of the IBM Data Processing Division. The library was formed in the belief that present and future users of the IBM 650 can profit by exchanging programs through the medium of a central library. To this end, IBM is acting as publisher and distributor of contributed programs of general interest. Additional contributions are cordially invited and encouraged; detailed information regarding the submission of contributed programs is given in subsequent pages.

At present, the 650 Program Library includes well over one hundred programs. An abstract of the detailed write-up for each program is included in this manual. These abstracts are designed to provide sufficient information to enable the potential user to determine whether or not a particular program will meet his needs. Thus the amount of reading required of the potential user is minimized, and, at the same time, distribution of the detailed write-up is limited to those 650 users who need and intend to use the particular program.

The manual is published in loose-leaf form to simplify updating and revision. All library programs have been classified by application, and the arrangement of the abstracts in this manual is in accordance with this classification. (The various categories and their assigned index numbers are shown on a page following this preface.) From time to time, as new programs are incorporated in the library, pertinent abstracts will be distributed to holders of the manual.

IBM assumes no responsibility for editing, evaluating, or checking the accuracy of contributed programs. The diverse nature and complexity of the many programs in the library prevent the 650 Program Library staff from testing contributions and answering inquiries pertaining thereto. Any person who discovers an error in a program is requested to correspond directly with the author, with copies of the correspondence to the addressee listed in the last paragraph of this preface. If the author wishes to submit a correction, the information will be included in the errata sheets which are published periodically and distributed to holders of the manual.

Submission of Contributed Programs

The following specifications concerning the contents and format of contributions were drawn up and adopted by a group of persons representing various 650 users at the IBM 650 Scientific Computing Seminar held at Endicott, N. Y., on April 22-26, 1957.

1. Program Write-Up. The contribution should include a write-up of the program in sufficient detail as to make further communication between the author and the potential user unnecessary. The printing process used (photo-offset) requires that all items be in black ink on opaque white paper. Typing or line printing is satisfactory, provided a fairly fresh black ribbon is used. The write-up should include the following information.

   a. Purpose: A concise indication of the task which the program is designed to accomplish.

   b. General description of the method of functioning of the program.

   c. Limitations of program: Range, accuracy, floating or fixed numbers; precautions and restrictions. If the program works in conjunction with some other routine, such as an interpretive routine, note the particulars in this regard.

   d. Mathematical methods. If several methods could have been used, state why the particular method was chosen.

   e. Storage requirements; speed; relocatability.

   f. Description or layout of input and output card forms.

   g. Wiring diagrams or such layouts of input-output areas as will enable the user to wire the control panels.

   h. Equipment specifications.
   (Note: A minimum 650 is considered to be a 650 Model II with one 533.)

   i. A statement indicating the amount of testing the program has undergone, including type and number of data, prior to submission.

   j. Operating instructions in the format indicated in the following pages. (Copies of the form are available from Stationery Stores, Endicott, under Form Number 32-7966.)

k. Flow charts of a general nature and/or of intermediate detail level. It is desirable for the flow charts to contain symbolic or actual references to the program listing.

l. Program listings: One listing in executive order with comments (one instruction per card) and a second listing showing the condensed cards (several instructions per card) in location sequence.

2. Program Abstract. The contribution should include an abstract of the write-up prepared on a standard form (see sample in following pages). In addition to the author's name, organization, and address, the following information should be included:

a. Title and purpose of routine. The paragraph should explain the need for the routine and if similar routines exist, it should explain the differences and advantages of the submitted program.

b. Range, accuracy, floating or fixed.

c. Mathematical methods. If several methods could have been used, state why the particular method was chosen.

d. Storage requirements, speed, relocatability.

e. Additional remarks, precautions or restrictions. If this program works in conjunction with some other routine, such as an interpretive routine, note the particulars in this regard.

f. Equipment specifications. (Note: A minimum 650 is considered to be a 650 Model II with one 533.)

3. Program Card Decks. The contribution should include a program deck which usually will be a condensed deck, preferably seven instructions per card. Subroutines will ordinarily be supplied in SOAP II form. Decks should be adequately identified and the cards should show a serial number, either by end-printing or by interpreting.

4. Sample Case. The contribution should include a sample case, i.e., input data cards and correct output data cards obtained by using the program.

5. Permission to Publish. The contribution should be accompanied by written permission to publish and to distribute the program to interested parties.

Contributed programs should be submitted to the addressee indicated on page iv.

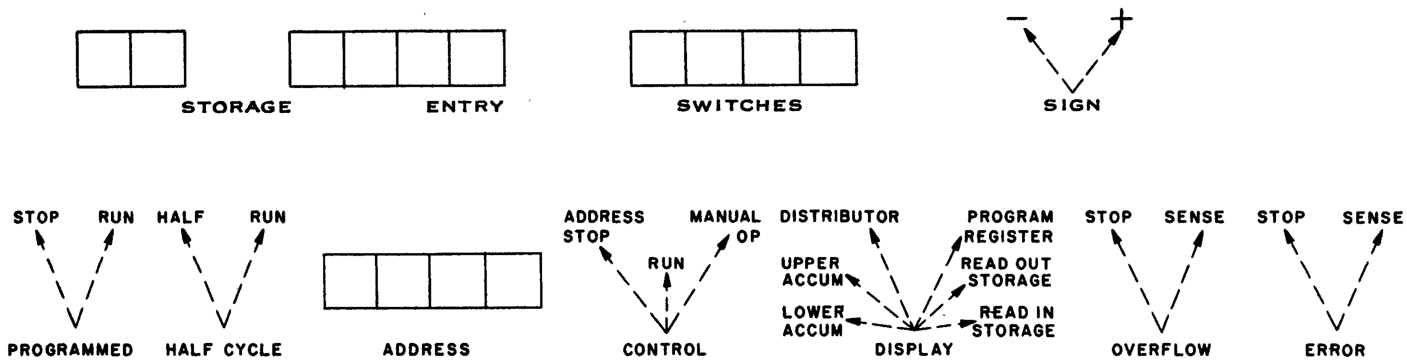
Requests for Library Programs

Requests for copies of the detailed write-up and/or program deck for any of the programs abstracted to date should be addressed to

Mr. F. E. Ross
Applied Programming Publications
IBM Corporation
590 Madison Avenue
New York 22, N.Y.

Each request should specify the program file number(s) and the name of the account for which the request is made. Inasmuch as the write-ups are printed on ozalid masters which permit easy duplication, only one copy per account will be supplied. Card decks will be furnished only to accounts where a 650 is installed or will be installed within sixty days, unless special circumstances can be shown to prevail.

650 PROGRAM OPERATING INSTRUCTIONS



I. Initial Console Setting as shown above.

A. Normal Starting Procedure: Computer Reset; Program Start.

B. Special Instructions: ______________________

II. Card Input - Output (533 or 537)

READ FEED

NO. OF CARDS	FILE DESCRIPTION

PUNCH FEED

CARD FORM

CONTROL PANELS

III. TAPE UNITS

ADDRESS	INPUT, OUTPUT OR OTHER	FILE PROTECTION RING		LABEL CHARACTERISTICS	FILE DESCRIPTION
		IN	OUT		
8010					
8011					
8012					
8013					
8014					
8015					

IV. 407 Input - Output

A. Card Input ______________________

B. Paper Forms Required ______________________

C. Carriage Control Tape ______________________

D. Control Panel Required ______________________

V. Other Instructions and Remarks:

VI. Program Stops and Required Action:

STOP ADDRESS	MESSAGE – EXPLANATION – ACTION

650 PROGRAM ABSTRACT

Author | Date

Organization and Address

For this abstract to describe the usefulness and properties of your 650 program adequately, please include comments relevant to the following items:

a. Title and purpose of routine.
b. Range, accuracy, floating or fixed.
c. Mathematical methods.
d. Storage requirements, speed, relocatability.
e. Additional remarks, precautions or restrictions.
f. Equipment specifications.

CLASSIFICATION OF 650 LIBRARY PROGRAMS

1.0.000 UTILITY PROGRAMS
- 1.1.000 Assembly and Optimizing
- 1.2.000 Loading
- 1.3.000 Punching
- 1.4.000 Tracing
- 1.5.000 Tape Handling
- 1.6.000 Miscellaneous General Purpose

2.0.000 PROGRAMMING SYSTEMS
- 2.0.000 Interpretive Programs
- 2.1.000 Compilers

3.0.000 MATHEMATICAL FUNCTIONS
- 3.1.000 Elementary
- 3.2.000 Higher

4.0.000 DIFFERENTIAL AND INTEGRAL EQUATIONS

5.0.000 MATRIX PROGRAMS
- 5.1.000 Fixed
- 5.2.000 Floating

6.0.000 STATISTICAL PROGRAMS

7.0.000 MATHEMATICAL ROUTINES

8.0.000 PHYSICAL SCIENCES
- 8.1.000 General Physics
- 8.2.000 Nuclear Physics
- 8.3.000 Geology and Geophysics
- 8.4.000 Other

9.0.000 ENGINEERING APPLICATIONS
- 9.1.000 Aeronautical
- 9.2.000 Civil
- 9.3.000 Chemical
- 9.4.000 Electrical
- 9.5.000 Mechanical
- 9.6.000 Petroleum
- 9.7.000 Other

10.0.000 MANAGEMENT SCIENCE
- 10.1.000 Linear Programming
- 10.2.000 Simulations
- 10.3.000 Other

11.0.000 DEMONSTRATION PROGRAMS

12.0.000 UNCLASSIFIED

SYMBOLIC OPTIMAL ASSEMBLY PROGRAM: SOAP

G. E. Mitchell and S. Poley
IBM, New York

a) This routine produces an optimumly coded absolute program from a set of instructions written in "free" symbolic form.

b) Does not apply.

c) Does not apply.

d) The program occupies approximately 875 storage locations exclusive of the 600 word symbol table. Output is 50 to 75 cards per minute with one word per card.

e) Programs with not more than 600 symbols are assembled and optimized in one pass; those with more than 600 symbols require additional passes. Relocatable absolute or symbolic library programs may be incorporated in the program being assembled. The form number for the write-up is 22-6285-1. A deck listing may be obtained through the 650 Program Library.

f) Alphabetic device is necessary.

OPTIMIZING PROGRAM

B. Gordon and A. Dalton
Equitable Life, New York

July 15, 1955

a) Automatically assigns optimum locations to the instructions and data of a program.

b) Does not apply.

c) Does not apply.

d) The program occupies approximately 500 storage locations in addition to 1216 locations for tables. Both input and output are one word per card.

e) Addresses may be left fixed or optimized. Addresses being optimized are 4 digit decimal numbers but are symbolic in the sense that they are assigned new optimum locations. A flow chart is included.

f) Minimum 650.

AN AUTOMATIC METHOD OF OPTIMUM PROGRAMMING

Elmer F. Shepherd
John Hancock, Boston, Mass.

April 8, 1955

a) Automatically assigns optimum locations to the instruction and data of a program.

b) Does not apply.

c) Does not apply.

d) The program occupies approximately 250 storage locations in addition to 1700 locations for tables. Both input and output are one word per card.

e) Addresses being optimized are written as a pseudo address in the 9000 series. Drum locations available to the optimizing program are indicated by manually removing the restricted addresses from a deck of 2000 cards numbered 0000 to 1999 and running those remaining through the 533 as part of the load deck. A flow chart is included.

f) Minimum 650.

FAR: FLOPS SYMBOLIC ASSEMBLY PROGRAM

Dr. John W. Young
IBM, Poughkeepsie

a) To convert FLOPS programs written with mnemonic op codes and symbolic locations and addresses, to FLOPS numeric pseudo-code and actual memory locations.

b) Data are entered in standard FLOPS floating point word format.

c) Does not apply.

d) 0001-0197 assembly program. 0200-0999 assembly table if FLOPS system is retained in memory. 0200-1999 assembly table if FLOPS system is not retained in memory. In most cases assembly proceeds at or close to maximum punching speed. Not relocatable.

e) The assembly table occupies two words of memory per symbol. Provision is made for reloading the assembly table if desired. Two passes are required for assembly.

f) Alphabetic device required.

SOAPY

Texas Highway Department
Austin, Texas

a) SOAPY is a modification of the original SOAP so that it may be used on a numeric 650.

b) Allows up to 900 symbolic addresses. Includes all the features of original SOAP.

c) Not applicable.

d) Uses most of 2,000 word drum. Can accommodate relocatable subroutines.

e) Reference should be made to original SOAP for details of program's capacity.

f) Minimum 650.

FOUR-PER-CARD LOADER

E. C. Kubie and G. R. Trimble, Jr.
IBM, New York 11/16/55

a) Loads one to four words per card into random drum locations specified by control words in the card.

b) Does not apply.

c) Does not apply.

d) Storage required is 5 words, 1995 to 1999. Locations 1951 to 1960 are used as the read band. Cards are loaded at 200 per minute.

e) Self-loading.

f) Minimum 650.

SEVEN-PER-CARD LOADER

E. C. Kubie and G. R. Trimble, Jr.
IBM, New York

a) Loads one to seven words per card into consecutive drum locations beginning at the location specified by a control word in each card.

b) Does not apply.

c) Does not apply.

d) Storage required is 23 locations, 1977 to 1999. Locations 1951 to 1960 are used as the read band. Cards are loaded at 200 per minute.

e) Self-loading.

f) Minimum 650.

FIVE-PER-CARD LOADING ROUTINE

J. M. Kibbee
IBM, Houston 1-1-56

a) Loads five words per card into random drum locations specified by control words in the card.

b) Does not apply.

c) Does not apply.

d) Storage required is 30 locations, 1970 to 1999. Locations 1951 to 1960 are used as the read band; 1950 and 1961-1969 are used to load the loading routine. Cards are loaded at 200 per minute.

e) Self-loading.

f) Minimum 650.

SIX-PER-CARD LOADING ROUTINE

J. M. Kibbee
IBM, Houston

1-1-56

a) Loads six words per card into consecutive drum locations beginning at the location specified by a control word in each card.

b) Does not apply.

c) Does not apply.

d) Storage required is 11 locations, 1950 and 1961 to 1970. Locations 1951-1960 are used as the read band. Cards are loaded at 200 per minute.

e) Self-loading.

f) Minimum 650.

H 806

1163

LOADING PROGRAM FOR MATRICES AND SIMULTANEOUS EQUATIONS

A. O. Garder
IBM, Houston

4-2-56

a) Loads fixed-point data for square matrices of order n, or for simultaneous equations with an nth order coefficient matrix and b constant vectors, converting the data from fixed to floating data.

b) Does not apply.

c) Does not apply.

d) Storage required is 236 locations, 1764 to 1999, including a self-loading routine and a read band. Data cards are read at approximately 75 cards per minute.

e) Self-loading. The elements of the matrix are entered row wise starting at location 0000 in the form ee aaaa aaaa = a. aaaaaaa $\cdot$ 10^{ee-50}. This routine may be used to load a matrix to be used by Matrix Inversion by Gausian Elimination, file number 5.2.002.

f) Minimum 650.

EIGHT PER CARD LOADING ROUTINE

D. W. Hagelbarger and E. F. Moore
Bell Telephone Laboratories, Murray Hill, New Jersey June 16, 1956

a) Loads eight words per card into consecutive drum locations beginning at the location specified by control punches ~~in each card~~.

b) Does not apply.

c) Does not apply.

d) Storage required is approximately 25 locations in the lower part of the drum in addition to the read area of the 1950 band. Cards are loaded at 200 per minute.

e) Provision is made for checking the deck being loaded for cards which are missing or out of order. This routine uses a control panel which is a modification of the one used in Bell Lab's interpretive routines.

f) Minimum 650.

SEVEN-PER-CARD PUNCH ROUTINE

D. W. Sweeney
IBM, New York

11-16-55

a) Punches, seven words to a card, the contents of consecutive drum locations between two address limits specified on a control card.

b) Does not apply.

c) Does not apply.

d) Storage required is 27 locations, 1950, 1961 to 1976, and 1985 to 1994. The read and punch areas of band 1950 are used for input - output.

e) The self-loading routine is not included in the listing. Output is in a form loadable by the seven-per-card loader, file number 1.2.002.

f) Minimum 650.

SYMBOLIC SEVEN-PER-CARD PUNCH ROUTINE

S. Poley
IBM, New York 1-4-56

a) A symbolic routine which will punch, seven words per card, the contents of consecutive drum locations between two specified address limits.

b) Does not apply.

c) Does not apply.

d) The routine uses 34 words exclusive of the punch band. Punching is at 100 cards per minute unless optimization is particularly poor.

e) Output is in a form loadable by the Seven-per-card Loader, file number 1.2.002. The routine may be used either as a subroutine by the main program or as an emergency drum punch routine under control of appropriate console settings. A SOAP symbolic deck listing with a sample absolute deck listing is included.

f) Alphabetic device if the SOAP symbolic version is used.

SYMBOLIC SEVEN-PER-CARD PUNCH ROUTINE

S. Poley
IBM, New York

1-4-56

a) A symbolic routine which will punch, seven words per card, the contents of consecutive drum locations between two specified address limits.

b) Does not apply.

c) Does not apply.

d) The routine uses 34 words exclusive of the punch band. Punching is at 100 cards per minute unless optimization is particularly poor.

e) Output is in a form loadable by the Seven-per-card Loader, file number 1.2.002. The routine may be used either as a subroutine by the main program or as an emergency drum punch routine under control of appropriate console settings. A SOAP symbolic deck listing with a sample absolute deck listing is included.

f) Alphabetic device if the SOAP symbolic version is used.

PUNCH OUT FOR THE SOLUTION OF SIMULTANEOUS EQUATIONS

A. O. Garder
IBM, Houston

4-2-56

a) Converts solutions of b systems of simultaneous linear equations with a common nxn coefficient matrix and b right hand sides (constant vectors) from floating to fixed point form and punches each solution as a row vector.

b) Does not apply.

c) Does not apply.

d) Storage required is 234 locations, 1766 to 1999 including the punch band and a self-loading routine. Solutions are punched at 100 cards a minute.

e) Each solution is given a one digit signed decimal point indicator. The first solution vector is assumed to be in (O, n+b, 2(n+b), $\cdots$, (n-1) (n+b)); the second solution in (1, n+b+1, 2(n+b)+1, $\cdots$, (n-1) (n+b)+1); etc. The routine is self-loading and has a one card restart procedure. May be used with Matrix Inversion by Gaussian Elimination, file number 5.2.002.

f) Minimum 650.

INVERSE MATRIX PUNCH OUT

A. O. Garder
IBM, Houston

a) Punches out a floating point inverse matrix into load cards containing seven matrix elements per card.

b) Does not apply.

c) Does not apply.

d) Storage required is 250 locations, 1750 to 1999, including the punch band and a self-loading routine. Cards are punched at 100 per minute.

e) The program selects and punches in each card a one digit decimal point indicator which is constant for the entire inverse matrix. The inverse matrix is assumed to be stored row-wise starting at location 0000. The routine is self-loading and has a one card restart procedure. May be used with Matrix Inversion by Gaussian Elimination, file number 5.1.002.

f) Minimum 650.

SIX-PER-CARD CONDENSING ROUTINE

G. E. Mitchell
IBM, Houston 1-1-56

a) Punches out a program which is on the drum into a six-per-card format.

b) Does not apply.

c) Does not apply.

d) Storage required is 150 locations, 0000 to 0149; locations 1950 to 1970 are used for a read band and self-loading routine. The deck being condensed must not have instructions in these locations. Cards are punched at 100 per minute.

e) Self-loading. The first location to be punched is specified in a header card. The routine will punch out all following locations, skipping those that contain a minus zero. A loading routine, file number 1.2.004, is placed ahead of the condensed deck so that it is self-loading.

f) Minimum 650.

AUTOPSY

Author Unknown
IBM, Endicott

a) Searches for those locations with any desired 4 digit number in either the data or instruction address position and punches the drum address and contents of each such location.

b) Does not apply.

c) Does not apply.

d) Storage required is 27 locations in the 1950 band; the read area of band 1900 is used for loading the routine.

e) Self-loading. The four digit number being searched for is set in console switches. Each output card has the location and contents of one word.

f) Minimum 650.

STORAGE DUMP

R. Haberman
G. E., Schenectady

January 20, 1956

a) Punches a specified block of storage, 8 words per card.

b) Does not apply.

c) Does not apply.

d) Storage required is 55 locations, 1900 to 1950, and 1961 to 1964. No speed information given.

e) The upper limit of the block being punched must be less than 1900. The block may be specified by a master card or entry may be programmed. If the number of locations being punched is not an even multiple of 8, additional storages will be punched to fill the last card with 8 words. The first card punched is a master card for use when these cards are loaded with L-2, see Technical Newsletter No. 8, pp. 50-52.

f) Minimum 650.

MEMORY DUMP AND RELOAD ROUTINE

George A. Rupprecht December 17, 1956
Office of the Chief of Naval Operations, Pentagon Building, Washington 25, D. C.

a) Punches a compact, self-reloading deck of load cards which replace 1990 words of memory.

b) Accurately replaces all except the ten card input words of any band desired.

c) Does not apply.

d) Punching time: 3 1/2 minutes. Reloading time: 1 1/2 minutes.

e) The instruction address and sign on the storage entry switches are necessary as specified despite the fact that only load cards are being read. Illegal information in the 1990 words to be replaced causes validity check stops requiring accurate console corrections for completing operation.

f) Minimum 650.

AVAILABILITY

James D. Chappell
IBM, Washington

December 31, 1956

a) Produces a SOAP Availability Punchout from a deck of load cards that may be single-instruction, four-per-card, seven-per-card, or any mixture of these three types.

b) Does not apply.

c) Does not apply:

d) Entire drum used by program. Running time is approximately read speed when processing single instruction or four-per-card load cards and about 1/2 read speed on seven-per-card load cards.

e) Load routines 1.2.001 and 1.2.002 transfer cards, and blank cards will be processed. The d address of less than 1 % of all constants will improperly be marked as unavailable.

f) Minimum 650.

SYMBOLIC TRACING ROUTINE

W. P. Heising and S. Poley 1-4-56
IBM, New York

a) To be assembled by SOAP, along with an untested main program, in anticipation of utilizing tracing as an aid in debugging.

b) Does not apply.

c) Does not apply.

d) Storage required is 46 locations in addition to 5 successive words of any punchable area. Tracing proceeds at 100 instructions per minute.

e) For each instruction traced a card is punched with the location of the instruction, the instruction itself, and the contents of the distributor and upper and lower accumulators. The location of the first instruction to be traced is set in the storage entry switches. A SOAP symbolic deck listing with a sample absolute deck listing is included.

f) Alphabetic device if the SOAP symbolic version is used.

FLOW TRACER

S. Poley
IBM, New York

5-15-56

a) A symbolic program to be assembled by SOAP which will trace designated locations only, called "bus stops."

b) Does not apply.

c) Does not apply.

d) Storage required is 60 locations and two successive bands should be designated as an assembly area for the routine. The symbolic deck contains 52 cards.

e) A maximum of 27 bus stops are allowable. When a bus stop is reached a single card is punched giving the location of the bus stop along with the contents of the distributor and accumulator. A SOAP symbolic deck listing with a sample absolute listing is included.

f) Alphabetic device if the SOAP symbolic version is used.

TRACING ROUTINE

D. W. Hagelbarger
Bell Telephone Laboratories, Murray Hill, New Jersey

July 27, 1956

a) A tracing routine for use with machine language programs.

b) Does not apply.

c) Does not apply.

d) Storage required is 150 locations, 1800 to 1949 (OR 0800 – 094. Tracing is at 100 card per minute.

e) Traces any program that the computer can execute. For each instruction traced the following information is punched: card number, location of instruction, the instruction, and contents of upper and lower accumulator and distributor (before execution of the instruction). Entry to, exit from and tracing of branch orders only is under control of console switches. Designed for use with the general purpose control panel used by the Bell Interpretive System, Technical Newsletter No. 11.

f) Minimum 650.

SYMBOLIC BRANCH TRACING ROUTINE

S. G. Reed and E. O. Godbold
IBM, Washington

December 31, 1956

a) A symbolic program to be assembled by SOAP which will trace all branch instructions (except READ) in a program in order that the flow of the program with respect to a block diagram or a test problem may be easily followed. The tracing routine may be modified to trace all instructions.

b) Does not apply.

c) Does not apply.

d) Storage required is 53 locations including a six word punchable area. Time required is approximately 95 milliseconds per non-traced instruction and 135 milliseconds per traced instruction except that a speed of 100 traced instructions per minute cannot be exceeded.

e) For each instruction traced a card is punched giving the location of the instruction, the instruction itself, the contents of the distributor and upper and lower accumulators, and the location of the next instruction.

f) Alphabetic device if the SOAP symbolic deck is used.

SELECTIVE TRACING ROUTINE

Barry Gordon
Equitable Life Assurance Society
New York, N. Y.

a) Traces all instructions, or only those instructions with a minus sign.

b) Does not apply.

c) Does not apply.

d) Uses one band of 50 locations; is relocatable.

e) This program was previously published in IBM Principles of Operation Bulletin #135 (Form 22-7135-0) and is reprinted here to bring it within the scope of the 650 Program Library.

f) Minimum 650

TAPE DUPLICATION

J. M. Venning
IBM DPC, Chicago, Illinois

a) This program will duplicate either numerical or alphabetic tapes. Tapes may be labeled or unlabeled.

b) Does not apply.

c) Does not apply.

d) Requires any 73 drum locations.

e) The set ring, read, and write instructions must be fitted to the tape to be duplicated. It must be known whether the tape is labeled or unlabeled.

f) This routine requires a 650 with high speed storage, index registers, two 727 tape drives, and a 533 or 407 for input.

SEVEN PER CARD PUNCH FROM TAPE

J. A. Taylor
IBM 650 Data Processing Center
Endicott, New York

a) Designed to load consecutively the drum on tape, and then punch seven instructions per card. This procedure provides the following advantages:

1. Punches only those locations needed in the program. Any seven consecutive minus zero locations are bypassed.

2. Requires only the user's read-in location on the drum for writing the drum on tape. Therefore, all locations other than the user's read-in and punch areas can contain instructions.

b) Does not apply.

c) Does not apply.

d) This routine is made up of three programs:

1. Write drum on tape.
2. Punch seven instructions per card from tape.
3. Load seven instructions per card program.

Storage requirements are critical only for the write drum on tape, and the load routine. The storage requirements are as follows:

1. Write drum on tape requires locations in IAS and the user's read-in area on the drum.

2. The load routine requires locations in IAS.

e) The punch and load routines are standard routines and can be used with any write drum on tape program for punching and loading.

f) 650 Tape System.

STOP NUMBER

P. S. Herwitz
IBM, Washington 3-20-56

a) This routine stores the number 01 (n) 8000 in location n, where $0 \leqq n \leqq 1949$.

b) Does not apply.

c) Does not apply.

d) Storage required is 10 locations in the 1950 band. The 1950 memory positions are numbered in 15.2 seconds.

e) When numbering is completed, control is transferred to the read order of routine 1.3.001 or 1.3.002 depending upon which has been used to load this one.

f) Minimum 650.

SET DRUM TO MINUS ZEROS

W. S. Korwan
IBM, New York

2-20-56

a) Sets all drum locations except 0000 and 0001 to minus zeros.

b) Does not apply.

c) Does not apply.

d) There are two cards in the deck. 17 seconds are required to set the drum.

e) Self-loading. Location 0000 is set to 01 0000 8000; 0001 is set to 00 0001 0000.

f) Minimum 650.

SET DRUM TO STOP CODES

F. J. Chrinko
IBM, New York

2-17-56

a) Sets each drum location except 0001 to -01 xxxx 8000 where xxxx is the address of the location.

b) Does not apply.

c) Does not apply.

d) There are two cards in the deck. 32 seconds are required to set the drum.

e) Self-loading. Location 0001 is set to 00 0001 0000. Transfers control to the storage entry switches.

f) Minimum 650.

CLEAR DRUM TO MINUS ZEROS

J. M. Kibbee
IBM, Houston

a) Clears the drum to minus zeros except locations 0001 and 0000.

b) Does not apply.

c) Does not apply.

d) Two cards in the deck.

e) Self-loading. Location 0000 is set to 01 0000 8000; location 0001 to 00 0001 0000. Control is transferred to 0000 resulting in a programmed stop.

f) Minimum 650.

LOAD DRUM WITH ERROR STOPS

J. M. Kibbee
IBM, Houston

1-1-56

a) To set each drum location except 0001 to -01 XXXX 9999 where XXXX is the address of the location.

b) Does not apply.

c) Does not apply.

d) There are two cards in the deck.

e) Self-loading. Location 0001 is set to -00 0001 0000. Control is transferred to 0000 resulting in a programmed stop.

f) Minimum 650.

CLEAR BLOCK TO ZERO

S. Fleming
G. E., Schenectady 3-30-56

a) Clears a specified block of storage to zero.

b) Does not apply.

c) Does not apply.

d) Storage required is 8 locations, 1951 - 1958.

e) Self-loading. The block limits are punched in the one card deck.

f) Minimum 650.

FIVE-PER-CARD CONDENSING ROUTINE

G. E. Mitchell
IBM, Houston

1-1-56

a) Condenses a one-word-per-card deck to a five-word-per-card deck and places a loading routine, file number 1.2.003, ahead of the condensed deck.

b) Does not apply.

c) Does not apply.

d) The deck contains 47 cards. Output is 100 cards per minute.

e) Self-loading. A trailer card placed at the end of the condensed deck makes it self-transferring.

f) Minimum 650.

ONE TO FOUR CONVERTER

S. Poley
IBM, New York

a) Converts single word load cards to four-per-card load cards which may be used with the four-per-card loader, file number 1.2.001.

b) Does not apply.

c) Does not apply.

d) Storage required is 75 locations and with a read area, punch area, and self-loader 1900 to 1999 are used. Output is approximately 50 cards per minute.

e) Self-loading. SOAP output decks may be converted.

f) Minimum 650.

ONE TO SEVEN CONVERTER

P. S. Herwitz
IBM, Washington

3-20-1956

a) Converts single-word load cards to seven-per-card load cards which may be used with the seven-per-card loader, file number 1.2.002.

b) Does not apply.

c) Does not apply.

d) Storage required is 37 locations, 0000 to 0035 and 1950. In addition, 25 locations are used in the 1900 and 1950 bands for reading, punching, and loading. Cards read at 200 per minute and punch at approximately 28 per minute.

e) Loading routine not included in listing.

f) Minimum 650.

BLOCK TRANSFER

T. A. Gorman
IBM, Washington

March 20, 1956

a) Transfers a block of n drum locations to another block of n drum locations.

b) Does not apply.

c) Does not apply.

d) The routine requires 16 memory locations and is available in symbolic SOAP form. With best optimization, the time needed to transfer n words is .016 n seconds.

e) The transfer may be from a block whose addresses have any desired increment (either positive or negative) to another block whose addresses have any other (or the same) constant increment (positive or negative). A SOAP symbolic deck listing with a sample absolute deck listing is included.

f) Alphabetic device if the SOAP symbolic version is used.

SEVEN TO ONE CONVERTER

P. S. Herwitz
IBM, Washington

a) Converts seven-per-card load cards to single instruction load cards.

b) Does not apply.

c) Does not apply.

d) Storage required is 8 locations 1961 to 1967 and 1986. The 1950 band is used for a read area, punch area, and self-loading routine. Cards are punched at 100 per minute.

e) Self-loading.

f) Minimum 650

A PROCEDURE FOR USING SOAP WITH A NUMERIC 650

Jack N. Graham
USAF, Directorate of Intelligence
Mathematical Analysis Branch
Washington, D. C.

a) Enables SOAP to be used with a minimum 650 provided a 407 with summary punch is available.

b) Does not apply.

c) Does not apply.

d) Approximately 850 storage locations are required.

e) A SOAP deck is partially converted to 650 alphabetic code using the 407 and summary punch. This routine completes the conversion at which time the regular SOAP program performs the assembly. No special characters may be used for any part of symbolic addresses.

f) Minimum 650 and 407 with summary punch.

FIXED TO FLOATING DECIMAL CONVERTER

P. S. Herwitz and H. J. Norman
IBM, Washington

a) Converts a fixed point number into a floating point number.

b) The fixed point argument may have up to 23 digits including any leading or following zeros. The floating point equivalent consists of an 8 digit mantissa and a 2 digit exponent.

c) Does not apply.

d) Storage required is 111 locations between 0000 and 0277.

e) None.

f) Minimum 650.

SOAP TO SEVEN

James D. Chappell
IBM, Washington

December 31, 1956

a) Will convert single instruction load cards to seven-per-card load cards. SOAP output cards may be converted immediately without removing special type cards. Only those locations from the FWA to the LWA are punched with the further provision that no output card shall begin with an unused location.

b) Does not apply.

c) Does not apply.

d) Uses entire 1950 band. Running time is approximately read and punch speed.

e) The 1.2.002 loader is punched along with the 1.6.001 stop number routine prior to punching the converted program deck. A 1.2.002 transfer card is the last card punched. No single instruction load cards can be processed for loading into the area used by the 1.2.002 loader.

f) Minimum 650.

CHECKMATE

E. M. Davis and B. Lefkowitz
IBM, Portland, Oregon

October 25, 1956

a) This routine loads post-SOAP format, one-per-card load cards and tests for duplicate use of drum locations.

b) Does not apply.

c) Does not apply.

d) CHECKMATE uses 16 locations 1961-1976. The load cards are read into 1951-1960. Program cards are loaded at full read speed. The drum labeling takes 30 seconds.

e) The post-SOAP load card must contain the constant instruction 24 aaaa 8000 + in cc 21-30. The word to be loaded in cc 31-40.

f) Minimum 650.

SOAP I TO SOAP II TRANSLATOR

S. Poley
IBM, New York

December 1, 1956

a) Translates symbolic cards prepared for SOAP I into symbolic cards acceptable to SOAP II.

b) Does not apply.

c) Does not apply.

d) Storage required including tables is approximately 220 locations. Timing is approximately 100 cards per minute.

e) It is assumed that errors detectable by SOAP I have been corrected and that relocatable addresses are in the range 0000 - 1999. Only the first ten columns of the remarks field will be retained. A SOAP II symbolic deck listing and a four-per-card absolute deck listing are included.

f) Alphabetic device is necessary.

AN INTERPRETIVE OPERATION FOR THE CONVERSION OF NUMBERS FROM FIXED POINT REPRESENTATION TO FLOATING POINT REPRESENTATION AND VICE VERSA

R. W. Klopfenstein
RCA Laboratories
Princeton, New Jersey

a) Designed as an adjunct to the interpretive system developed at Bell Telephone Laboratories and described in IBM Technical Newsletter No. 11.

b) Floats a fixed point number or fixes a floating point number. Rounds in the last place in both floating and fixing.

c) Not applicable.

d) Programmed for locations 001-049. (Note: Interpretive system proper occupies locations 1000-1999).

Running Time: Approximately 60 milliseconds.

Relocatable to any 49 consecutive locations in lower memory (excepting 0000) by means of the Bell Telephone Laboratories translation routine. Preferably relocated by multiples of 50 locations.

e) Programmed stop with 8888 in the address lights occurs if an overflow would result upon fixing a given floating point number.

f) Minimum 650.

TABLE LOOK UP SUBROUTINE

J. Sider
IBM, Glendale, California

a) This subroutine will linearly interpolate from tabulated values of a function and its argument.

b) Arguments and functions are in floating point. The range of the argument table is 10^8.

c) Mathematical methods. Linear interpolation.

d) Storage requirements: 30 locations including 5 erasable. Speed is approximately 60 ms. The routine is in S. O. A. P. and may be assembled with the main program.

e) There are no programmed stops. It is assumed that the given argument is in the range of the argument table. In order to utilize the table look up command, the argument table must be in unnormalized floating point form. That is, all elements in the argument table must be entered with the same characteristic. A maximum of 25 entries in the argument table is permitted.

f) 650 equipped with alphabetic device, floating decimal, index registers.

DOUBLE TABLE LOOK UP SUBROUTINE

J. Sider
IBM, Glendale, California

a) Title and purpose: Double Table Look Up Subroutine. A function of two variables is given in tabulated form. The routine linearly interpolates with respect to both arguments to compute the desired function.

b) Arguments and functions are in floating point form. The range of each argument table is 10^8.

c) Mathematical methods: Linear interpolation.

d) Storage requirements: 79 locations including 8 erasable. The routine is in S.O.A.P. and may be assembled with the main program.

e) There are no programmed stops. It is assumed that the given argument is in the range of the argument table. The elements of the argument tables must be in unnormalized form. That is, all arguments in a given table must have the same characteristic. The ARG1 table may have as many as 48 entries. The ARG2 table may have as many as 25 entries.

f) 650 equipped with alphabetic device, floating decimal, index registers.

INTERPRETIVE FLOATING DECIMAL ROUTINE

R. R. Haefner
E. I. du Pont de Nemours & Co., Inc.
Savannah River Laboratory
Aiken, South Carolina

a) This routine is a modification of the Trimble interpretive floating decimal system described in IBM Technical Newsletter No. 8. It is designed for the 650 installation equipped with the automatic floating decimal device to provide a compromise between rewriting infrequently used programs which incorporate the Trimble routine and inefficient machine utilization while running such programs.

b) Floating arithmetic.

c) Modification of methods in Trimble routine.

d) Uses 243 storage locations in a block of 390 locations. The routine is 75% faster than the Trimble routine with no recoding required.

e) None

f) 650 with automatic floating decimal device.

SIR: SOAP INTERPRETIVE ROUTINE

B. G. Oldfield and W. Hemmerle
IBM, New York 3-21-56

a) A relocatable library program which is used with the SOAP system to handle floating decimal interpretive operations.

b) Does not apply.

c) Does not apply.

d) The program is separated into 9 sections and only those required for a particular problem need be assembled. Storage for individual sections varies from 31 to 184 locations. Timing is a function of the operation being performed.

e) Included, in addition to the arithmetic operations, are trace, float, fix, square root, sin-cos, ln, exp, and arctan. Entry and exit from the interpretive routine are at the discretion of the programmer.

f) Alphabetic device necessary.

MITILAC

R. H. Battin, R. S. O'Keefe, M. B. Petrick
MIT, Boston

September, 1955

a) A general purpose multiple address interpretive routine for floating point numbers.

b) Does not apply.

c) Does not apply.

d) The complete routine requires all but 390 locations 0010 to 0399. This amount may be increased to approximately 850 by not using all the features of MITILAC. Timing is a function of the operation being performed.

e) Included, in addition to the arithmetic operations, are sin, cos, arctan, square root, exp, ln, log as a special case, absolute value, solutions for simultaneous differential equations, 10 index registers, read, punch, and various branch operations.

f) Alphabetic device necessary.

COMPLEX ARITHMETIC INTERPRETIVE ROUTINE

Tsai H. Lee
Detroit Edison, Detroit

a) Interprets and executes multiple address complex arithmetic instructions in addition to performing the normal 650 instructions.

b) All complex numbers are assumed to be of the form .xxxxx xxxxx + j .xxxxx xxxxx.

c) Does not apply.

d) The interpretive routine occupies 284 locations, 0000 to 0283. Timing is a function of the operation being performed.

e) Twelve instructions may be interpreted: add, subtract, multiply, divide, shift left, shift round, store complex accumulator, transfer complex number from memory to memory, sum a block of complex numbers, square of absolute value, vector-vector multiplication, and unconditional transfer. Negative instructions are interpreted; positive instructions are executed normally.

f) Minimum 650.

DRUCO I

G. R. Trimble and E. C. Kubie
IBM, Washington

a) An interpretive floating decimal routine which is a modification and extension of the interpretive floating decimal system, IFDS, described in Technical Newsletter 8, pp. 17-36.

b) Does not apply.

c) Does not apply.

d) Storage required is 1000 locations, 0000 to 0999. Timing is a function of the operation being performed.

e) There are 13 new operations in addition to the 18 of IFDS. Included are the arithmetic operations, square root, sin, cos, arcsin, arctan, exp, ln, sum of products, and branch operations.

f) Minimum 650.

SPEED CODING SYSTEM

H. M. Sassenfeld
Redstone Arsenal, Huntsville, Alabama

a) A three address interpretive routine for both fixed and floating-point decimal arithmetic.

b) Does not apply.

c) Does not apply.

d) Storage required is from 600 to 855 locations depending upon how many of the function subroutines are needed.

e) There are 45 possible instructions including mathematical functions, memory, dump, restart procedure, three index registers, and optional use of normal 650 operations. Programs coded in the Speed Coding System may be simulated on the 704 by use of the 650 simulator program prepared by Redstone Arsenal.

f) Minimum 650.

NINE OPERATION SPLIT INSTRUCTION ROUTINE: NOSIR

L. M. Harvey and J. C. White
G. E., Schenectady

August 3, 1956

a) A floating-point interpretive routine using 5 digit instructions so that problems with a large number of instructions may be solved with a single program loading.

b) The interpreted operations use the built-in floating-point operations.

c) Does not apply.

d) Storage required is 94 locations 0000 to 0093.

e) Instructions consist of a one-digit operation code and a four-digit data address. Operations include the arithmetic operations, store, branch minus, branch zero, and exit. Interpreted instructions are stored two to a word and are executed in sequence; the two instructions in a word are performed before proceeding to the next word. Subroutines and normal 650 instructions may be used as needed.

f) Floating decimal device is required.

ERCO SPACE SAVER

W. G. Rouleau and E. H. Weiss
ERCO Division, ACF Industries, Inc., Riverdale, Maryland

a) This routine is designed to save programming space by executing two instructions per line. The floating decimal point instructions are add, subtract, multiply, negative multiply, divide and add absolute as well as reset add, reset subtract, store and branch minus.

b) Range: $-10^{50} < x < 10^{50}$. Accuracy: 8 places. Number system: floating arithmetic.

c) Does not apply.

d) Storage required is 150 locations.

e) This routine embellishes the 650 computer, but all ordinary 650 instructions can be used in conjunction with this system. A tracing routine has been developed and can be put into any punch band.

f) Minimum 650.

GENERAL PURPOSE SYSTEM FOR THE 650: L_2

R. W. Hamming and Miss R. A. Weiss
Bell Telephone Laboratories, Inc., Murr y Hill, N. J.

August 24, 1956

a) A general purpose three address floating point interpretive system designed to be easy to learn and use. The orders are not assigned definite locations so that program changes are very easy to make.

b) The 8 place floating point ystem of numbers with exponent range of -50 to+ 49. A fixed point addition is also included.

c) Does not apply.

d) Storage required for the interpretive system is 1100 locations, 0900 to 1999. System is not relocatable but library routines are relocatable. The main program of a problem automatically relocates itself as required.

e) In addition to the standard arithmetic operations there are: square root, e^x, $\log_e x$, 10^x, $\log_{10} x$; sin x, cos x, arctan x (both degrees and radians) all with full range of arguments and 8 place accuracy; block read in, punch out, and move; five index registers; transfers on minus, zero, and exponent; transfer to library and subroutines; and tracing orders. Conditional error stops for division by zero, square root of negative numbers, etc., for which error cards are automatically punched. Calculations can be continued after these stops by pushing the program start button.

f) Minimum 650

(2.0.008R - Replaces 2.0.008)

ERCO FLOATING DECIMAL POINT SUBROUTINES

J. K. Carl and E. H. Weiss
ERCO Division, ACF Industries, Inc., Riverdale, Maryland

a) Performs eight floating decimal point instructions, namely: add, multiply, divide, subtract, negative multiply, negative divide, add absolute and subtract absolute.

b) Range: $-10^{50} < X < 10^{50}$. Accuracy: 8 places. Number system: floating decimal point.

c) Does not apply.

d) This routine uses only memory locations 1900-1999.

e) Does not apply.

f) Minimum 650.

DOPSIR: DOUBLE PRECISION FLOATING POINT SOAP INTERPRETIVE ROUTINE

Hebron E. Adams
IBM, Washington

January 2, 1957

a) DOPSIR is both a system of coding (uses a set of mnemonic operation codes in which all arithmetic operations are performed with double precision floating decimal numbers) and a relocatable library program, which interprets the said system.

b) Range of variables: 10^{-49} to 10^{+50}. Accuracy: 18 places. Floating point.

c) Conventional floating point methods.

d) Storage required: 670 locations maximum. Speed: interpretation-execution time averages 60 milliseconds. Relocatable library program.

e) DOPSIR is, in most ways, analagous to SIR, and all SIR operations are included in DOPSIR. In addition, such features as interpretive floating decimal to fixed decimal and fixed decimal to floating decimal commands, an improved interpretive tracing system, and an addressable pseudo-accumulator have been included. Inasmuch as DOPSIR is a somewhat extensive system, the text of the report should be referred to for precautions and restrictions.

f) Alphabetic device is necessary.

FIASCO

G. V. Maverick and S. Togasaki
IBM, Los Angeles

August, 1956

a) Simulation routine which will simulate a 650 with indexing accumulators, floating decimal arithmetic, and high-speed (core) storage. This program is designed for use in checking out programs written for the augmented 650.

b) Does not apply.

c) Does not apply.

d) FIASCO occupies locations 0000-0949 of the drum. It can be used to simulate any routine which is located (or can be relocated) entirely within 0950-1999, 8000-8003, 8005-8007, 9000-9059. One of the two loading routines built into FIASCO may be used to relocate a program during loading. Timing is a function of the operation.

e) In a read instruction into core every word on the card which actually gets into core must be a legal word. In a punch from core all words of storage exit on the 533 control panel, on the augmented 650, which would have blanks will have -0303030303 with FIASCO.

f) Minimum 650.

COMPLEX ARITHMETIC OPERATIONS IN THE BELL LABORATORIES INTERPRETIVE SYSTEM

P. M. Marcus
Carnegie Institute of Technology
Pittsburgh, Pa.

D. L. Blackhurst
Mellon Bank
Pittsburgh, Pa.

a) Complex Arithmetic Operations in the Bell Laboratories Interpretive System provides the five arithmetic operations - addition, subtraction, multiplication, division and negative multiplication - with the same code structure as for real operations. The 650 must be sent into a complex mode of operation by a special command; however, previous results and looped operations are preserved, and there is also a complex move; all other operations send the 650 back to the usual mode. Complex numbers are stored in two floating decimal parts in successive registers.

b) Floating point numbers between 10^{-50} and 10^{+49} with eight significant figures (for both real and imaginary parts).

c) Not relevant.

d) Uses 1000-1999; and 0002-0004 erasable storage, 0000-0001 for previous result. Sacrifices arctangent, but provides supplementary (slower) program to evaluate arctangent, using 950-999. Operation times much slower than for real floating decimal operations.

e) Special functions are not available for complex arguments.

The Bell Laboratories Interpretive System is described in IBM Technical Newsletter No. 11.

f) Minimum 650.

INTERNAL TRANSLATOR (IT)
A COMPILER FOR THE 650

A. J. Perlis
J. W. Smith
H. R. Van Zoeren
Carnegie Institute of Technology, Pittsburgh 13, Pa.

a) Programs written as a sequence of statements in a general algebraic language (roughly similar to that of FORTRAN) are translated into programs in symbolic, i. e., SOAP I form.

b) Programs employing both fixed and floating point constants and variables may be translated.

c) Does not apply.

d) The translator requires the entire drum. Output is approximately 50 SOAP I cards/minute.

e) The SOAP I type programs produced are assembled by a modified SOAP I deck whose output is a machine language program punched 5 words/card. These machine language programs require, during operation, an auxiliary package of subroutines which include floating point, input-output, and optional logarithm, power and exponential routines. Depending on the option, these packages require from 270 to 500 locations. The remainder of the drum is available for program and data. A general technique may be used to incorporate additional subroutines.

The system includes a programming manual, 533 wiring diagram, the translation program, the modified SOAP I program, reservation and subroutine packages, and sine, cosine, and square root floating point subroutines.

f) Alphabetic device is required.

MODIFICATIONS OF THE INTERNAL TRANSLATOR* (IT) COMPILER FOR USE OF SPECIAL CHARACTERS

J. N. Rogers
C. M. White
GE Vallecitos Atomic Laboratory
Pleasanton, California

a) These revisions are to take advantage of some of the FORTRAN symbols in writing IT statements for the compiler. The following table gives the correspondence between the revised symbols and the representation for the computer.

Symbol Name	Representation
Left Parenthesis	(
Right Parenthesis	)
Decimal Point	.
Equality (substitution sense)	=
Comma	,
Addition	+
Division	/
Negation	-

A sample statement would appear as below:

Y2 = (CI3 x Y5) - (2.85 + C (I2 + I4)) / 5.82

b) Does not apply.

c) Does not apply.

d) All other aspects of the IT system remain the same. The card deck and the listing appended to the write-up include only the change cards for the IT deck.

e) Alphabetic device and Group II special character device are required.

* 650 Library Program Abstract Number 2.1.001, Internal Translator (IT) A Compiler for the 650, A. J. Perlis, J. W. Smith, H. R. Van Zoeren, Carnegie Institute of Technology, Pittsburgh 13, Pa.

IT - 2

H. R. Van Zoeren
Computation Center
Carnegie Institute of Technology
Pittsburgh 13, Pa.

a) Programs written as a sequence of statements in IT language (see 650 Abstract 2.1.001) are translated directly into machine language represented in standard 5 instructions/card form.

b) Same as 2.1.001.

c) Does not apply.

d) The translator requires the entire drum. Output is approximately 20 cards per minute (100 instructions per minute).

e) The machine language programs produced require, during operation, an auxiliary package of subroutines which include floating point, input-output, and optional logarithm, power and exponential routines. Depending on the option, these packages require from 270 to 500 locations. The remainder of the drum is available for program and data. A general technique may be used to incorporate additional subroutines.

The system includes the translation program, relocation routine and subroutine packages, and associated function subroutines.

f) Alphabetic device is required.

SQUARE ROOT SUBROUTINE

G. E. Collins
IBM, New York

3-22-56

a) Computes the square root of a single-precision fixed-point number.

b) The argument must be such that at least one of the two highest order digits is non-zero and that the decimal point must be an even number of places from the extreme left. All 10 digits of $\sqrt{x}$ are significant.

c) The method is a table look-up operation followed by two modified Newtonian iterations.

d) LWA is 0064 in the relocatable version with 8 words open. Average execution time is approximately 72.9 ms.

e) Both absolute and SOAP relocatable deck listings are included.

f) Alphabetic device if relocatable version is used.

SQUARE ROOT SUBROUTINE

G. R. Trimble, Jr.
IBM, Houston

1-30-55

a) Computes the square root of a single-precision fixed-point number.

b) Range: $0 \leqq A \leqq .9999999989$. Maximum error is $3 \cdot 10^{-10}$.

c) Newton's method is used.

d) LWA is 0039 with 16 words open in the relocatable version. For a random argument 120 ms. are required.

e) Both absolute and SOAP relocatable deck listings are included.

f) Alphabetic device if relocatable version is used.

CUBE ROOT

W. K. Pence June 24, 1955

a) Computes the cube root of a single-precision fixed-point number.

b) Range: $0 \leqq A \leqq .999999999$. Accuracy information not given.

c) The method is to make first approximation followed by an iterative formula.

d) Storage required is 22 locations, 0000 to 0021; the routine may be translated an even number of locations. Requires approximately 14.4 + 24n ms., where n is the number of iterations.

e) None.

f) Minimum 650

EXPONENTIAL

S. Fleming
G. E., Schenectady

March 28, 1956

a) Computes e^X for a single-precision fixed-point number.

b) Range: $-16.11 < X \leqq 23.02585092$.

c) Method not given.

d) Storage required is 50 locations, 0000 to 0049; the routine may be translated by an even number of locations. Not more than 6 iterations are required.

e) None.

f) Minimum 650.

EXPONENTIAL

S. Fleming
G. E., Schenectady

March 28, 1956

a) Computes e^X for single-precision fixed-point number.

b) Range: $-20.5 < X \leqq 23.02585092$. Accuracy: error is less than one in the eighth significant digit.

c) Method not given.

d) Storage required is 49 locations, 0000 to 0048; the routine may be translated by an even number of locations.

e) None.

f) Minimum 650.

EXPONENTIAL SUBROUTINE

George E. Collins
IBM, New York

4-27-56

a) Computes e^x for a single-precision fixed-point number.

b) Range: $|x| < 10$. Accuracy: relative error varies from $13 \cdot 10^{-10}$ for no leading zeros in e^x to $6 \cdot 10^{-1}$ for 9 leading zeros in e^x.

c) Method is a rational approximation by E. Kogbetliantz.

d) LWA is 0079 with 9 words open in the relocatable version. Average execution time is 109.7 ms.

e) Both absolute and SOAP relocatable deck listings are included.

f) Alphabetic device if relocatable version is used.

EXP A

A. O. Garder
IBM, Houston

2-30-56

a) Computes e^A for a single-precision fixed-point number.

b) Range: $-59 \leqq A \leqq 20$. For A = 15 the accuracy is one significant digit; for $A < 0$ the maximum error is $2 \cdot 10^{-7}$.

c) Method is a polynomial approximation by Hastings.

d) LWA is 0105 with no words open in the relocatable version. Running time is 600 ms.

e) Both absolute and SOAP relocatable deck listings are included.

f) Alphabetic device if relocatable version is used.

SINH A, COSH A, EXP A

A. O. Garder
IBM, Houston

2-30-56

a) Calculates sinh A, cosh A, or e^A for a single-precision fixed-point number.

b) Range: $-8 \leqq A \leqq 8$. Accuracy: For e^A error varies from at most $2 \cdot 10^{-7}$ for $A < 0$ to at most .5 for $A = 8$. Sinh A and cosh A are found from e^A.

c) Method for e^A is a polynomial approximation by Hastings.

d) LWA is 0195 with 34 words open in the relocatable version. Running time is 620 ms.

e) Both absolute and SOAP relocatable deck listings are included.

f) Alphabetic device if relocatable version is used.

SINH X AND COSH X

Barbara Martin
Detroit Edison, Detroit

August 8, 1955

a) Computes sinh X or cosh X for a single-precision fixed-point number.

b) Range: $0 < X < 2$. Accuracy information not given.

c) Method is to calculate e^X from the subroutine given in Technical Newsletter No. 9, page 50, and then determine sinh or cosh from the standard formulas.

d) Storage required is 62 locations, 0000 to 0061, including the e^X subroutine. The routine may be translated an even number of locations.

e) The e^X subroutine is not included in the deck listing.

f) Minimum 650.

SIN-COS SUBROUTINE

G. R. Trimble, Jr.
IBM, Houston

1-30-55

a) Calculates sin X or cos X for a single-precision fixed-point number.

b) Range: For sin X, $-7.2 \leqq X \leqq 7.2$; for cos X, $-8.8 \leqq X \leqq 8.4$. Maximum error is $3 \cdot 10^{-9}$.

c) Method: 12th power in Taylor series. Reference: Technical Newsletter No. 9, p. 34.

d) LWA is 0099 with one word open in the relocatable version. Running time is 123 ms.

e) Both absolute and SOAP relocatable deck listings are included.

f) Alphabetic device if relocatable version is used.

SINE OR COSINE

S. G. Fleming
IBM, Schenectady

January 18, 1956

a) Calculates sin X or cos X of a single-precision fixed-point number.

b) Range: $-2\pi \cdot 10^{10} < X < 2\pi \cdot 10^{10}$ where X is in radians. Accuracy not given.

c) Method not given.

d) Storage required is 105 locations, 0000 to 0104; the routine may be translated by an even number of locations. Speed not given.

e) None

f) Minimum 650.

ARC SIN A

G. R. Trimble, Jr.
IBM, Houston

1-30-55

a) Computes arc sin A for a single-precision fixed-point number.

b) Range: $0 \leqq A \leqq 1$. Accuracy: maximum error $5 \cdot 10^{-8}$.

c) Method is polynomial approximation by Hastings.

d) LWA is 0068 with no words open in the relocatable version. For a random argument, running time is 200 ms.

e) SOAP relocatable deck listing is included.

f) Alphabetic device required.

$LOG_{10}A$, Ln_eA

E. B. West and A. O. Garder
IBM, Houston

2-30-56

a) Computes $\log_{10}A$ or $\ln_e A$ for single-precision fixed-point numbers.

b) Range $10^{-5} \leqq A < 10^5$. Accuracy: maximum error is $2 \cdot 10^{-7}$.

c) Method: polynomial approximation by Hastings.

d) LWA is 0099 with 34 words open in the relocatable version. Running time is 130 ms.

e) Both absolute and SOAP relocatable deck listings are included.

f) Alphabetic device if relocatable version is used.

NATURAL LOGARITHM

S. Fleming
G. E., Schenectady 3-28-56

a) Computes ln X for a single-precision fixed-point number.

b) Range: $10^{-9} \leqq X < 10^{10}$. Accuracy: error is less than 2 in the 7th decimal.

c) Method not given.

d) Storage required is 54 locations, 0000 to 0053.

e) None.

f) Minimum 650.

POLAR TO CARTESIAN COORDINATES

Barbara Martin
Detroit Edison, Detroit 7-27-55

a) Converts single-precision fixed-point polar coordinates to single-precision fixed-point cartesian coordinates.

b) Range: $r < 100$, $0 < \theta < 2\pi$.

c) Method is to use the sin-cos subroutine in Technical Newsletter No. 9, page 39, and then to use the standard conversion formulas.

d) Storage required is 67 locations, 0000 to 0066, including the sin-cos subroutine. The routine may be translated by an even number of locations.

e) The sin-cos subroutine is not included in the deck listing.

f) Minimum 650.

SQUARE ROOT | X |

D. W. Sweeney
IBM, Endicott

a) Calculates square root of | X | for any x in floating decimal form.

b) Range: any floating decimal argument. Accuracy information not given.

c) Method is a first approximation involving a table look-up followed by three iterations with Newton's formula.

d) Storage required is 46 locations. As written the 0000 band is used but may be relocated an even amount. Execution time is 113 ms.

e) Indexing accumulators A and B are used.

f) Floating decimal device and indexing accumulators are required.

ARCTAN X AND LN | X |

D. W. Sweeney
IBM, Endicott

a) Computes arctan x and ln | x | for floating-point numbers.

b) Range: $x > 0$ for ln x; any x for arctan x. Accuracy information not given.

c) A rational function approximation to the continued fraction expansion is used.

d) Storage required is 0000 to 0099 and may be relocated by a multiple of 50. Approximate running time is 180 ms. for either function.

e) None.

f) Floating decimal device and indexing accumulators are required.

SUM AND SUM OF SQUARES

L. G. Hunsicker
IBM, New York

October 2, 1956

a) Computes the sums and the sums of the squares of N groups of K numbers.

b) N unlimited, K $\leqq$ 100,001 but may be different for each N. Size of the numbers is limited to five decimal digits. The sum of numbers is fixed-point single precision; the sum of squares fixed-point double precision.

c) Does not apply.

d) Storage required is 147 locations, 0000 to 0146. Time information not given.

e) Self-loading.

f) Minimum 650.

FLOATING POINT LOG |A| AND LN |A|

Prepared by IBM 650 Applied Programming

G. J. Porter
IBM, New York

a) This subroutine computes $\text{Log}_{10}A$ and Ln A utilizing the floating decimal arithmetic device and indexing register A. This routine has maximum range and accuracy with running time minimized as much as possible.

b) Range: $|A| > 0$
Accuracy: Error $< 10^{-8}$
Floating Point

c) Method: $A = M \times 10^P$, where P is an integer
Multipliers A_i are found such that $m = M \prod_i^k A_i$

The A_i are chosen so that $1 < m < 1.1$

$\text{Log}_{10}m$ is computed by use of a relaxed Taylor series for

$$\text{Log}_{10}(1 + x), \quad 0 < x < .1$$

Finally, $\text{Log}_{10}M = \text{Log}_{10}m - \sum \text{Log}_{10}A_i$

Ln A is secured by multiplying Log A by Ln 10

This subroutine uses multipliers in which the sum of the digits is minimized thus taking advantage of the variable multiplication time of the 650.

d) Storage requirements: 100 locations with 15 open.
Speed: Log: 130 m.s. Ln: 140 m.s.
Relocatable SOAP II cards.

e) Indexing Registers: Indexing register A (8005) is used in this subroutine, thus the information in A before entrance into the subroutine is destroyed.

f) 650 equipped with floating point device and indexing registers. The alphabetic device is also required because of the relocatable (SOAP II) feature.

FLOATING POINT e^A, 10^A, SINH A, COSH A

Prepared by IBM 650 Applied Programming

G. J. Porter
IBM, New York

a) Subroutine for e^A, 10^A, Sinh A and Cosh A utilizing the floating decimal arithmetic device and indexing register A. Maximum accuracy and range have been secured with reasonable running time and storage requirements.

b) Range: e^A: $A < 100$; 10^A: $A \leq 43.4$; Sinh A and Cosh A: $|A| < 100$
Accuracy: Relative accuracy of 10^{-8}
Floating Point

c) Mathematical methods:

e^A: By several reductions A is reduced to the range $|A| < .054$. A relaxed Taylor series is then used.

10^A: A is multiplied by Ln 10 converting to an exponential function. The method used in e^A is then used.

Sinh A, Cosh A: These are simply extensions of the e^A method. For more detail refer to the program write-up.

d) Speed: e^A: 180 m.s.; 10^A: 185 m.s.; Sinh A and Cosh A: 240 m.s.
Storage: 150 Locations for the entire routine. If only e^A and 10^A are desired, 25 Locations can be omitted. For convenience these 25 are located at the end of the program.
Input: Relocatable SOAP II cards.

e) Indexing register A is used in the program and is not restored to its original state. If it is necessary to save the contents of this register changes can be made in the program to accomplish this. These changes are listed in the program write-up.

f) 650 equipped with floating decimal arithmetic device and indexing registers is required. The alphabetic device is also required because of the relocatable (SOAP II) feature.

FLOATING POINT SINE A AND COSINE A

Prepared by IBM 650 Applied Programming

G. J. Porter
IBM, New York

a) This subroutine computes Sine A and Cosine A for floating point arguments utilizing the floating decimal arithmetic device and indexing register A.

b) Range: $|A| < 10^8$
Accuracy: Error $< 10^{-8}$
Floating Point

c) The function is then computed by use of the multiple angle formulas and relaxed series approximations for sine and cosine.

d) Storage Requirements: 150 locations, 1 open.
Speed: 225 m. s.
Relocatable SOAP II cards.

e) Indexing register A (8005) is used in the subroutine and is not reset to its previous condition.

f) 650 equipped with floating decimal arithmetic device and indexing registers. The alphabetic device is also required because of the relocatable (SOAP II) feature.

FLOATING POINT ARCTANGENT A

Prepared by IBM 650 Applied Programming

G. J. Porter
IBM, New York

a) This subroutine utilizes the floating point device and indexing accumulator A to calculate the arctangent of a floating point number with maximum accuracy and range and minimum running time.

b) Range: Any floating decimal number.
Accuracy: Absolute Error < 10^{-7}
Floating Point

c) Mathematical Methods: The range (-00, +00) is reduced to (0, .13734) by considering the transformation

$$Z = \frac{|A| - X_m}{1+|A|\, X_m}, \quad \text{where } X_m \text{ is determined by the size of A.}$$

Then Arctan Z is computed by a relaxed polynomial approximation.
Finally, Arctan A = Arctan Z + Arctan X_m

d) Storage: 75 locations with 11 open.
Speed: 140 m. s.
Relocatability: Relocatable SOAP II cards (relocation must be by multiple of 50).

e) The contents of indexing register A are not saved.

f) 650 equipped with floating decimal arithmetic device, and indexing registers is required. The alphabetic device is also required because of the relocatable (SOAP II) feature.

SQUARE ROOT

Prepared by IBM 650 Applied Programming

E. Pulley
D. L. Mordy
IBM, New York

a) Computes the square root of X for any X $\geq$ 0 in floating decimal form.

b) Range: Any floating decimal argument,

00 $\leq$ machine exponent $\leq$ 99. The error is less than one in the eighth place.

c) Method is a linear approximation involving a table look up followed by two iterations with Newton's formula.

d) Storage required: 56 locations. Relocatable. Execution time approximately 75 milliseconds.

e) The program is in relocatable SOAP II form.

f) Alphabetic device used (for SOAP II assembly).

Nth ROOT FIXED POINT SUBROUTINE

Prepared by IBM 650 Applied Programming

George Wu
IBM, New York

a) Computes the Nth root of a single precision fixed point argument A.

b) Range: $0.0000\,00001 \leq A \leq 0.9999\,99999$, $N > 0$. The number of significant places is approximately equal to ten minus the number of preceding zeros in A. Maximum accuracy - nine digits.

c) Iteration of Bailey's function.

d) Relocatable SOAP II; occupies 78 locations. Speed is dependent upon N and the desired accuracy. The average speed is approximately 600 m. s.

e) The desired accuracy may be determined by the adjustment of a constant.

f) Minimum 650.

EXPFN

W. Guebert
IBM, Cambridge, Massachusetts

a) To calculate e^x

b) Range $-117 \leq x \leq 112$. Accuracy: 8 digits in the floating point number.

c) Method is a rational approximation by Kogbetliantz given in the IBM Journal of Research and Development, April, 1957. In his notation $m = k = 3$.

d) Requires 88 drum locations. Approximately 140 ms are required for each e^x.

e) None.

f) A 650 equipped with floating point device and indexing accumulators is required as well as alphabetic device for the SOAP assembling.

FRATS
(Fast, Relocatable, Arithmetic and Transcendental Subroutines)

W. E. Stewart
Department of Chemical Engineering
University of Wisconsin
Madison, Wisconsin

a) Provides general utility routines for floating point calculation. The operations are listed below.

b) The routines deal with floating point numbers in the form

± (xxXXXXXXXX)	=	± (X. XXXXXXX) (10^{xx-50})
Digits in the 650		Scientific notation

The range of the exponent, xx, is therefore $0 \leq xx \leq 99$. Unnormalized numbers may be used as input to any of the routines. Results are normalized, except in FIX and unnormalized ADD. Given exact, normalized input, the maximum result error is about ± 0.56 units of the last result digit, except for logarithms of numbers near unity, which are correct within $\pm 3 \times 10^{-10}$ before normalization and rounding. Unnormalized input is handled with equal precision, except when added or used as numerator in division.

c) Square root is computed by the Newton iteration method, using three iterations. The exponential function, e^x or a^x $(a \leq 10)$, is evaluated using a table of $y = 10^w$ at interval $\Delta w = 0.1$, and a fifth-degree polynomial for interpolation; the 650 table lookup operation is not used. The logarithmic function, ln Z, is evaluated using a seventh-degree expansion in odd powers of $\frac{Z - y}{Z + y}$. Values of y and $\log_{10} y$ are obtained, by table lookup, from the same table used for the exponential function.

d) The complete set of routines occupies 398 locations including temporary storage, and can be loaded in locations 0001 - 0399 or any 8 consecutive bands on the drum. The routines are relocatable by SOAP II to any higher region on the drum, except that the address increment for Natural Logarithm must be evenly divisible by 50. Any block of routines may be omitted without affecting the others, except that Multiply-Add requires Blocks 1 and 2.

Block	Operation	Drum Locations Used, Unrelocated Lowest	Highest	Total Number	Execution Time, Milli-seconds
1	ADD (normalized or unnormalized)			76	29
1	FLOAT, and set ADD to normalize			8*	20
1	FIX, and set ADD to not normalize			8*	39
2	MULTIPLY			59	31
1, 2	MULTIPLY-ADD, link and execute			6*	64
1, 2	MULTIPLY-ADD, execute only				59
3	Divide by 8002			42	37
3	Divide 8002 by (k)				32
4	Square Root			55	103
5	Exponential, e^x or a^x	0000	0099	75	108 for e^x
6	Natural Logarithm	0063	0149	90	126
1-3		0001	0199	196	
1-4		0001	0249	248	
5, 6		0000	0149	150	

* In addition to parent operations.

The above execution times do not include access time for factors and exit instructions. Access time ranges from 0 to 20 milliseconds for random access, depending on the number of new factors.

e) The invalid-address stops use addresses above 9990, and are effective for any combination of accessories now available. Programs which will utilize these subroutines may be written in symbolic form for SOAP assembly, or coded directly in machine language.

f) Minimum 650.

CIRCULAR AND HYPERBOLIC FUNCTIONS: REGULAR BESSEL FUNCTIONS

W. V. Baxter
Savannah River Laboratory, duPont, Augusta, Georgia

July, 1955

a) Computes sin x, cos x, sinh x, cosh x, $J_n(x)$, and $I_n(x)$ for n = 0, 1, 2, or 3.

b) Arguments are fixed-point in the form xx. xxxxxxxx; answers are given in both fixed and floating-point form. Range for sin x and cos x is $|x| < 100$; for sinh x and cosh x, $|x| < 5.29$; $I_0(x)$, $x < 6.32$; $I_1(x)$, $x < 6.52$; $I_2(x)$, $x < 6.77$; $I_3(x)$, $x < 7.15$; $J_0(x)$ and $J_3(x)$, $x < 7.82$, $J_1(x)$, $x < 9.62$; $J_2(x)$, $x < 8.94$. The series is summed until the new term is $< 10^{-8}$.

c) Series expansions are used.

d) Storage required is 150 locations, 0000 to 0149, and may be translated by an even amount.

e) None.

f) Minimum 650.

IRREGULAR BESSEL FUNCTIONS

Julius C. English
Savannah River Laboratory, duPont, Augusta, Georgia

May, 1956

a) Computes ln x, $Y_n(x)$, and $K_n(x)$ for n = 0, 1, 2, or 3.

b) Arguments are fixed-point in the form xx.xxxx xxxx; answers are given in both fixed and floating-point form. Range for ln x is $.0086 \leq x < 100$; $Y_0(x)$, $.021 \leqq x \leqq 6.30$; $Y_1(x)$, $.021 \leqq x \leqq 6.46$; $Y_2(x)$, $.21 \leqq x \leqq 6.64$; $Y_3(x)$, $.55 \leqq x \leqq 6.98$; $K_0(x)$, $.021 \leqq x \leqq 5.20$; $K_1(x)$, $.021 \leqq x \leqq 5.30$; $K_2(x)$, $.21 \leqq x \leqq 5.57$; $K_3(x)$, $.55 \leqq x \leqq 5.98$. The series is summed until the new term is $< 10^{-8}$.

c) Series expansions are used.

d) Storage required is 449 locations, 0000 to 0448, and may be translated by an even amount.

e) This program includes W. V. Baxter's routine for sin, cos, sinh, cosh, $J_n(x)$, and $I_n(x)$, file number 3.2.001.

f) Minimum 650.

AN INTERPRETIVE SUBROUTINE FOR THE ERROR FUNCTION AND THE COMPLEMENTARY ERROR FUNCTION

R. W. Klopfenstein
RCA Laboratories,
Princeton, N. J.

a) This subroutine computes the error function, or, alternately its complement. It is designed for use with the interpretive system developed at Bell Telephone Laboratories and described in IBM Technical Newsletter No. 11.

b) Floating point input and output. Accepts any argument (positive and negative) accepted by the interpretive system, viz.,

$$10^{-50} \leqq |x| < 10^{+50}, \text{ and } x = 0.$$

Maximum error of 3 units in the eighth significant figure for Erf (x) and 3 units in the seventh significant figure for Erfc (x).

c) Power series for small values of argument. Laplace continued fraction for large values of argument.

d) Programmed for locations 900-999 (Note: Interpretive system occupies locations 1000-1999.) Addition of 5 cards to Erf (x) deck converts it to Erfc (x) deck preserving constant significant figure accuracy but not changing storage requirements. Maximum running time: 2.58 seconds.

Relocatable to any 100 consecutive storage locations in lower memory (excepting location 0000) by means of Bell Telephone Laboratories translation subroutine. Preferably relocated by multiples of 50 locations, however, in order to preserve optimization in basic language portion of the program.

e) See write-up for explanation of programmed CONDITIONAL STOP and means for eliminating it if it is not desired.

f) Minimum 650.

(August 1957, Bulletin 10 - 17)

AN INTERPRETIVE SUBROUTINE FOR THE SINE INTEGRAL AND COSINE INTEGRAL FUNCTIONS

R. W. Klopfenstein
RCA Laboratories,
Princeton, N. J.

a) This subroutine computes the sine integral and cosine integral functions. It is designed for use with the interpretive system developed at Bell Telephone Laboratories and described in IBM Technical Newsletter No. 11.

b) Floating point input and output. Accepts any argument (positive and negative) accepted by the interpretive system, viz.,

$$10^{-50} \leqq |x| < 10^{+50}, \text{ and } x = 0.$$

Maximum error of 1 unit in the eighth significant figure for Si (x) and 5 units in the eighth decimal for Ci (x).

c) Power series for small values of argument. Legendre continued fraction for large values of argument.

d) Programmed for locations 800-999. (Note: Interpretive system occupies locations 1000-1999.)

Running time: Average running time - 3.0 seconds.
Maximum running time - 4.18 seconds.

Relocatable to any 200 consecutive storage locations in lower memory (excepting location 0000) by means of the Bell Telephone Laboratories translation subroutine. Preferably relocated by multiples of 50 locations.

e) Ci (x) has singularity at $x = 0$. Subroutine stores - 99999999 99 (-10^{50}) in the Ci (x) output for $|x| < 10^{-49}$ as an approximation to minus infinity.

f) Minimum 650.

(August 1957, Bulletin 10 - 19)

BESSEL FUNCTIONS SUBROUTINE

R. R. Haefner
E. I. du Pont de Nemours & Co., Inc.
Savannah River Laboratory
Aiken, South Carolina

a) Computes e^x, ln x, $\sqrt{x}$; $I_n(x)$, $K_n(x)$, $J_n(x)$, and $Y_n(x)$ for n = 0, 1, 2, and 3.

b) Automatic floating decimal; range and accuracy are discussed in the write-up.

c) Various mathematical methods are used; they are described in the write-up.

d) 490 storage locations are required - SOAP II relocatable or fixed in locations 0500-0989.

e) None.

f) 650 with automatic floating decimal device and indexing registers.

MATHIEU AND MODIFIED MATHIEU FUNCTIONS SUBROUTINE

E. T. Kirkpatrick
Mechanical Engineering Department
Carnegie Institute of Technology
Pittsburgh 13, Pa.

a) Computes Mathieu and modified Mathieu Functions

using canonical forms $y'' + (a-2q\cos 2u)\, y = 0$

$$y'' - (a-2q\cosh 2u)\, y = 0$$

and solutions of the form $y = \sum_{r=0}^{\infty} A_{2r}^{2n} \cos 2r\,u$

$$y = \sum_{r=0}^{\infty} A_{2r}^{2n} \cosh 2r\,u$$

b) Range: $n = 0(1)\,3$ $\quad 0 \le q \le 25 \quad 0 \le u < 1.0$
Accuracy: 5 significant figures.
Floating point interpretive system of Dr. V. M. Wolontis of Bell Laboratories is used (IBM Technical Newsletter No. 11).

c) The characteristic numbers and Fourier coefficients are found by evaluating the continued fraction and recurrence relations which are found as a consequence of assuming a solution in the form of an infinite trigonometric or hyperbolic series.

d) The Mathieu Function subroutine requires locations 50 to 549, not relocatable. Since the program is written in the Bell Laboratories interpretive mode, locations 1000 to 1999 are also unavailable. Given n, q, u and an approximation to a, the time required to compute y varies from 30 to 90 seconds.

e) The normalization used is that of Goldstein-Ince.

f) Minimum 650.

SIMULTANEOUS FIRST ORDER DIFFERENTIAL EQUATIONS

R. W. De Sio
IBM, Schenectady

a) A fixed-point routine that solves a system of equations of the form:

$$y_i' = f_i(t_1, y_1, y_2, \ldots, y_n) \quad (i = 1, 2, \ldots, n)$$

with the given initial values

$$(y_i)_{t=0} = K_i \quad (i = 1, 2, \ldots, n).$$

b) The decimal locations of each y_i' and y_i are up to the programmer.

c) A predictor-corrector method is used with iterations until the desired convergence is obtained.

d) Storage required for the program is 380 locations, 0000 to 0377, 0803, and 0804.

e) The user of this routine must program the y_i variable routines and must provide a punch routine.

f) Minimum 650.

MULTIPLE NUMERICAL INTEGRATION

F. Edelman
RCA, David Sarnoff Research Center, Princeton

a) This subroutine uses the floating-point interpretive system developed by Dr. V. M. Wolontis, Technical Newsletter No. 11, and performs up to a triple integration.

b) The upper limits of integration may be finite or infinite.

c) Methods used are the Trapezoidal Rule, Simpson's Rule, or Newton's 3, 4, or 5 point formulas.

d) Storage required is practically the entire drum. Machine time is measured for the integration of a basic block of five points, excluding computation time of the integrands. The time is 5 seconds, 28 seconds, or 168 seconds for a single, double, or triple integration respectively.

e) Only programming of the integrands and specification of the integration limits are required. The integration increment can be varied to a certain extent during any one integration. Program decks are available upon request from the author.

f) Minimum 650.

FOURIER SYNTHESIS

H. T. Evans, Jr. and Paul S. Herwitz
IBM, Washington

a) Computes double sums of the form:

$$S(x, y) = \sum_k \left[\sum_h A(h,k) \begin{Bmatrix}\sin\\ \cos\end{Bmatrix} (2\pi hx)\right] \begin{Bmatrix}\sin\\ \cos\end{Bmatrix} (2\pi ky),$$

where either trigonometric function may be the sine or the cosine.

b) A (h, k) is an integer $<10^4$. The range for both x and y is 0 to .25 with increments of .01. The quantities h and k are integers <100, and for a given summation either one takes on only odd values or only even values.

c) Does not apply.

d) Storage required is approximately 390 locations. Time required is 30 + (550+.5n)m seconds where n is the average number of values of A(h, k) for the m series to be summed.

e) Self-loading. There may be up to 300 non-zero values of A(h, k) and up to 300 inner sums. A theoretical discussion of the routine is given in Technical Newsletter No. 10, pp. 229 - 237.

f) Minimum 650.

LAPLACE TRANSFORMATION

J. A. Painter
IBM, Endicott

a) Solves linear differential equations by evaluating the Laplace Transform of the equation. Input is $X(S) = A(S)/B(S)$ which is obtained by taking the transform and solving for $X(S)$. $A(S) = \sum_{i=0}^{n} A_i S^i$, $B(S) = \sum_{i=0}^{m} b_i S^i$.

b) Floating-point arithmetic is used. $1 \leqq m \leqq 6$.

c) $B(S)$ is factored using Lin's method and $X(S)$ split into partial fractions. The inverse transformations are evaluated using a RAND polynomial for e^x.

d) The entire drum is used. Timing information is not given.

e) Final output is in complex form. This routine may also be used to solve algebraic equations.

f) Minimum 650

AN INTERPRETIVE SUBROUTINE FOR THE SOLUTION OF SYSTEMS OF FIRST ORDER ORDINARY DIFFERENTIAL EQUATIONS

Franz Edelman
RCA, David Sarnoff Research Center, Princeton

a) Solves systems of first order ordinary differential equations.

b) Systems of up to 30 equations may be solved. Floating decimal arithmetic is used. Precision is specified by the programmer.

c) The programmer has a choice between the Runge-Kutta-Gill and the Milne methods.

d) The interpretive routine occupies locations 0600 to 0999. Execution time per point is about 6 + 3N seconds for the RKG method and about 2.5 + 1.5N seconds for the Milne method where N is the number of equations to be solved.

e) The programmer need specify only initial conditions, the equations to be solved and their number, and the precision. The program is written for the Wolontis Interpretive Routine described in Technical Newsletter No. 11. Program decks are available upon request from the author.

f) Minimum 650.

ELLIPTIC INTEGRALS

R. Pexton
R. Carpenter
University of California Radiation Laboratory
Livermore, California

a) Computes complete and incomplete elliptic integrals of the first and second kinds.

b) The elliptic integrals contain two parameters whose ranges are: $0 \leq k \leq 1 \cdot 0$; $0 \leq Q \leq \pi/2$. k is defined as the modulus and Q is defined as the amplitude of the elliptic integrals.

Magnitudes of parameters are expressed in floating point notation. The two high order digits determine the location of the decimal point: XXYYYYYYYY.
i.e. 5010000000 = 1·0 Q is measured in radians.

The results are accurate to seven decimal digits when the parameters are in the following ranges: $0 \leq k \leq \cdot 8$ and $0 \leq Q \leq 1 \cdot 4$ ($\sim 80^{\circ}$). Outside this range, the accuracy decreases, particularly when both parameters are close to their upper bounds.

c) Repeated application of Landen's transformation permits one to replace a numerical integration process with an algebraic expression whose members are easily produced. The magnitudes of the algebraic members rapidly converge to a constant value (0 <u>or</u> 1.0) and hence only a few terms are required for the desired accuracy.

d) The total program occupies cells 0000 through 1045. The IBM Basic Floating Point Routine plus the transcendental subroutines sin, cos, ln, and arctan are located in cells 0000 through 0772.

The following commands in the IBM Basic Floating Point Routine are <u>not</u> used: 04, 11, 12, 13, 15, 17, 18.

Four values are computed for a specified set of parameters in 15 seconds, on the average.

The program may be relocated by a multiple of 50.

MATRIX INVERSION

A. O. Garder and J. M. Kibbee
IBM, Houston

2-28-56

a) Inverts matrices of 25th order or less.

b) Matrix elements are ten-digit fixed-point numbers.

c) The inverting part of the routine is that of Mr. Dura Sweeney's, and performs Gaussian Elimination using eight-digit floating-point arithmetic.

d) The program with storage space for the matrix utilizes essentially the complete drum. For a matrix of order n $.00004\,n^2(n+5)$ hours are required.

e) The output consists of the inverse in fixed-point form and two figures of merit which represent the accuracy with which the product of the matrix and its inverse approximate the unit matrix.

f) Minimum 650.

SOLUTION OF SIMULTANEOUS LINEAR EQUATIONS

A. O. Garder
IBM, Houston

April 1, 1956

a) Solves b systems of n simultaneous linear equations with b righthand sides and a common coefficient matrix.

b) Arithmetic is fixed-point form.

c) Method not given.

d) Storage required is 450 locations, 1200 to 1649. Speed not given.

e) It is required that $(n+1)(n+b) < 1200$. The routine is self-loading and self-restoring.

f) Minimum 650.

COMPLEX ARITHMETIC MATRIX INVERSION

Tsai H. Lee
Detroit Edison, Detroit

a) Computes the inverse of a complex matrix up to size 27 x 27 or the solutions to b systems of linear equations with a common coefficient matrix.

b) Matrix elements are fixed-point of the form xx. xxxx xxxx.

c) Standard elimination method is used.

d) Storage required for the program is 135 locations, 0300 to 0434. Storage for the complex matrix requires $2n^2$ locations; working storage 2n locations. Approximate running time is $n^2(.27n + .22)$ sec.

e) None.

f) Minimum 650.

MATRIX-VECTOR MULTIPLICATION

J. D. Brown
IBM, New York

July 9, 1956

a) Multiplies a fixed-point, single-precision, square matrix M of order $n \leqq 42$ by a vector X.

b) Each partial product is half-adjusted to reduce truncation error.

c) Does not apply.

d) LWA is 0075 in the relocatable version with no words open. Maximum time required is $(89.1+37.2n+43.0n^2)$ ms.

e) All elements are treated as fractions and only the high-order half of the products are accumulated. Overflow may occur if $\sum_j m_{ij} \cdot x_j > 20$ digits. Absolute and SOAP relocatable deck listings are included.

f) Alphabetic device if relocatable version is used.

MATRIX ADDITION

J. D. Brown
IBM, New York

July 9, 1956

a) Adds two fixed-point, single-precision, $m \times n$ matrices A and B.

b) The elements a_{ij} and b_{ij} must be such that $a_{ij} + b_{ij} \leqq 10$ digits. Also $m \times n \leqq 987$.

c) Does not apply.

d) LWA is 0025 in the relocatable version with no words open. Maximum time required is $[28.4+53.0(m \cdot n)]$ ms.

e) Both absolute and SOAP relocatable deck listings are included.

f) Alphabetic device if relocatable version is used.

MATRIX INVERSION

D. W. Sweeney
IBM, New York

October 6, 1955

a) Inverts matrices of order $\leqq 42$ or solves b sets of simultaneous equations for $n^2 + nb \leqq 1764$.

b) Matrix elements are in floating-point form.

c) Method not given.

d) Storage required is 236 locations, 1764 to 1999. The matrix inversion, exclusive of input and output time, is executed in approximately $.072n^3$ seconds.

e) Locations 0000 to n^2-1 are occupied by the elements of the input matrix. The inversion program is destroyed after use and must be reloaded for each new inversion.

f) Minimum 650.

MATRIX INVERSION BY GAUSSIAN ELIMINATION

A. O. Garder
IBM, Houston

April 2, 1956

a) Inverts a floating-point matrix of order n or solves b systems of simultaneous linear equations with b constant vectors and a common coefficient matrix of order n.

b) All numbers are of the form ee aaaaaaa = a.aaaaaaa $\cdot$ 10^{ee-50}.

c) Method is Gaussian Elimination. Pivotal elements are selected in order without regard to size.

d) Storage required is approximately 350 locations 1650-1999. Time required for one inversion, or solution, is $.00002(n+b)^2 n$ hours.

e) Storage limitations require that $n^2+(n+1)(b+1) \leqq 1650$. The inverse of the coefficient matrix is obtained with solution of a system of simultaneous linear equations. This is a modified version of a program originally written by Dura Sweeney which is now self-restoring on the drum.

f) Minimum 650.

LARGE SCALE MATIRX MULTIPLICATION

L. W. Ayres
IBM, Washington

a) Multiplies an mxn matrix A by an nxk matrix B to produce an mxk matrix C.

b) Matrix elements are floating-point of the form x.xxxx xxxxx ee where ee represents an exponent modulo 50.

c) Does not apply.

d) Storage required is 400 locations, 1600 to 1999. Speed not given.

e) The postmultiplier is stored in locations 0000 to 1199; the premultiplier in 1200 to 1599. Only non-zero elements of either matrix are loaded into storage. More than one pass will be required if there are more than 576 non-zero elements in the postmultiplier. Also $n \leq 200$. A SOAP symbolic deck listing is included in addition to an absolute listing of the program assembled in locations 1600 to 1999.

f) Minimum 650.

DOUBLE PRECISION MATRIX INVERSION

W. J. Hemmerle
IBM, Washington

a) Inverts a matrix and solves systems of simultaneous linear equations in double precision arithmetic,

b) Matrices up to 25 x 25 may be inverted and b systems of n equations may be solved where $2(n + 1)(n + b) \leqq 1300$.

c) Method is that of D. W. Sweeney's matrix routine, file number 5.2.001.

d) LWA is 0649 in the relocatable version. For an n x n matrix running time is approximately $.36n^3$ seconds.

e) The program is self-restoring. A SOAP relocatable deck listing is included.

f) Alphabetic device is required.

COMPLEX AND REAL EIGENVALUES

R. W. De Sio
IBM, Schenectady

a) Determines real and complex eigenvalues for an n x n matrix A.

b) Matrix elements are in floating-point form. For large n (> 6) coefficients of small powers in the characteristic equation lose significance.

c) Method consists of three phases: (1) matrix-vector multiplication, (2) solution to a system of equations by Dura Sweeney's Gaussian Elimination routine, file number 5.2.001, and (3) calculation of roots of a polynomial equation by De Sio's program Real and Complex Roots of Algebraic Equations, file number 7.0.001.

d) With respect to c) above (1) requires approximately 380 storage locations, (2) 236 locations, and (3) 336 locations. A fifth-order matrix requires about 3 minutes.

e) Only one of the three phases is on the drum at a time. The deck listing with this write-up includes only phase (1), the matrix-vector multiplication.

f) Minimum 650.

SMALL SCALE MATRIX MULTIPLICATION

L. W. Ayres
IBM, Washington

a) Multiplies an m x n matrix by an n x k matrix to produce an m x k matrix.

b) Data must be in floating decimal form; $n \leqq 41$; $nk \leqq 1681$; no restriction on the size of m.

c) Does not apply.

d) Storage required for the program is approximately 220 locations; data storage requires $n(k + 1)$ locations. Time estimate for calculation is .061 mnk seconds and for input-output $\frac{6nk + 26}{140}$ seconds.

e) None.

f) Minimum 650.

LARGE SCALE MATRIX INVERSION

a) Computes the inverse of large order matrices.

b) Matrix elements are floating-point of the form x. xxxx xxxxx ee, where ee represents an exponent modulo 50. A matrix of order $n \leqq 500$ may be handled.

c) The Jordan method is used.

d) Approximately 330 storage locations are used for the program. Time required is $\frac{n^2(n+1)}{100}$ minutes.

e) Both absolute and SOAP symbolic deck listings are included. Each step in the elimination process requires a separate pass through the 650. The output from the kth elimination step is supplied as input for the k+1st step. A total of n passes is necessary.

f) Alphabetic device if SOAP symbolic version is used.

MATRIX INVERSION

H. L. Norman
IBM, Washington

December 31, 1956

a) This program has modified 5.2.002 to include load and punch routines so that any number of matrices may be loaded, inverted and punched out without reloading the program. This program will invert a matrix of order N or will solve b systems of simultaneous linear equations with b constant column vectors on the righthand side of a common coefficient matrix of order N, where $N^2 + (N+1)(b+1) \leqq 1600$.

b) Input data and solution are in floating point form.

c) The inversion is performed by the method of Gaussian Elimination.

d) The program, including the load and punch routines, utilizes storage locations 1600 - 1999. Locations 0000 - (N+1) (N+b) are used for storage of matrix elements and temporary storage. Loading and punching are at full speed; the calculation requires approximately $.0012N(N+b)^2$ minutes. The program is not in relocatable form.

e) A non-load starting card is required for each matrix inverted.

f) Minimum 650.

DOUBLE PRECISION MATRIX INVERSION

James D. Chappell
IBM, Washington

December 31, 1956

a) Inverts a matrix and solves systems of simultaneous linear equations in double precision floating point arithmetic, a revision of 5.2.004 to provide greater flexibility of input and output and increased speed.

b) Matrices up to 25 x 25 may be inverted and V systems of N equations may be solved where $2\ (N+1)\ (N+V) \leq 1300$.

c) Method is Gaussian elimination, pivotal elements are selected in order without regard to size.

d) Not relocatable, running time is approximately $.30N^3$ seconds.

e) The program contains its own load and punch routines and is self-restoring.

f) Minimum 650.

SYMMETRIC SIMULTANEOUS LINEAR EQUATIONS

H. L. Norman
Service Bureau Corporation
Washington, D. C.

a) This program will solve "b" systems of "n" simultaneous linear equations consisting of "b" constant right-hand column vectors with a common symmetric nxn coefficient matrix and/or solve the determinant of the symmetric coefficient matrix. Both load and punch routines are incorporated in such a way that any number of systems can be solved with one program setup. By taking advantage of symmetry, this program is twice as fast as the corresponding non-symmetric general solution. Many desirable options are incorporated to increase the flexibility of the input and output.

b) Both input data and the solutions are in floating decimal point form. The size of the system to be solved is limited such that $(n+b)^2 - b \leqq 1450$.

c) The simultaneous equations are solved by the Doolittle method, the b column vectors of constants considered to be on the right-hand side of the equation. The determinant is obtained by the product of the diagonal elements of the diagonalized matrix.

d) The program uses locations 1451 to 1999 with the exception of 46 scattered locations. The input matrix occupies locations 0000 to $n(n+b) - 1$ and the solution uses locations 0000 to $(n+b)^2 - b$. Calculation time is roughly $.03\,n(n+b)^2$ seconds. Loading and punching are at full speed. The program is not in relocatable form.

e) The coefficient matrix must be symmetric.

f) Minimum 650.

MATRIX INVERSION AND SOLUTION OF SIMULTANEOUS LINEAR EQUATIONS

Prepared by 650 Applied Programming, IBM, New York

B. N. Carr
IBM Corporation

a) Inverts matrices and solves simultaneous linear equations. This routine is more than three times as fast as programs which do not use index registers and the floating decimal device.

b) Square matrices, (nxn), can be inverted where $n(n+1) \leq 1999$. Rectangular arrays, $n \times (n+m)$, can be solved where $(n+1)(n+m) \leq 1999$. As with any similar procedure, error due to accumulated roundings may be large.

c) A progressive elimination technique is used to perform the inversion.

d) The entire drum, except 0000, can be used for matrix element storage. For any matrix, $(n+1)(n+m)$ consecutive locations are used starting with 0001. Immediate access storage is used for the load routine, the inversion program, and the output routine. The program is not relocatable. The time for inversion is approximately $.02n^3$ seconds. The program contains 32 instructions and 2 constants.

e) The inversion program fails if $a_{1,1}$ or any element which takes its place during the calculation is zero. The program is written in machine language.

f) This routine requires a 650 equipped with the floating decimal device, index registers, and immediate access storage.

MATRIX INVERSION ROUTINE 1 (MIR 1)

K. B. Williams
University of California Radiation Laboratory
Livermore, California

a) MIR 1 inverts a matrix of order n or solves b sets of linear equations with a common coefficient matrix.

b) Matrix elements are floating point numbers of the form .XXXXXXXX YY where Y is the exponent (excess 50) base 10.

c) The method is by Gaussian Elimination.
The programming technique is a modification of one devised by R. W. DeSio.

d) MIR 1 occupies 79 locations from 0000 to 0078. It can be translated to any desired block of locations by an even amount (using a translating routine supplied with MIR 1). Approximately $10n^3$ milliseconds are required to invert a matrix assuming average times for floating point operations.

e) Location of the matrix on the drum is arbitrary.
Also, $(n + 1)(n + b) \leq 1921$. MIR 1 must be loaded with a loading routine, SLR 2, which is supplied with the program.

f) 650 equipped with indexing accumulators and floating decimal device.

SYMMETRICAL MATRIX INVERSION

J. Giblin
Detroit Edison Company
Detroit, Michigan

a) Computes the inverse of a symmetrical matrix up to size 54 or inverts and solves a rectangular system satisfying the inequality $n^2 + n(1 + 2b) \leq 3298$, where b is number of b vectors, with 1900 band open for punch routine.

b) All operations are in floating point arithmetic. Accuracy is that obtained by conventional elimination techniques.

c) The method is based upon standard elimination methods modified to require knowledge of only the elements on and above the main diagonal.

d) Speed is that of fastest standard method to size 12 x 12, but from this point the necessarily complex address modification increases running time as n, and hence the number of iterations, increases.

e) Since the product of a matrix and its transpose is a symmetrical matrix, the routine can be extended to non-symmetrical matrices to size 54 x 54.

f) Minimum 650.

VECTOR BY SYMMETRICAL MATRIX MULTIPLICATION

S. Young
Detroit Edison Company
Detroit, Michigan

a) Performs and punches the results of a vector by symmetrical matrix multiplication.

b) Multiplies an n-dimensional vector by an n x n symmetrical matrix, where $n \leq 45$. All operations are in floating point arithmetic.

c) Conventional vector by matrix multiplication methods are used, with modifications such that only those elements of the matrix which lie on or above the diagonal and the elements of the vector need to be loaded into the machine.

d) Speed and storage requirements are dependent on the size of the matrix. In the case of an n x n matrix, $n\left[\frac{(n+1)}{2}\right]$ storage locations are needed to put the matrix in memory.

e) None

f) Minimum 650.

MATRIX INVERSION

J. C. English
F. K. Townsend
E. I. du Pont de Nemours & Co., Inc.
Savannah River Laboratory
Aiken, South Carolina

a) Provides a matrix inversion routine with load and punch routines.

b) The routine will invert up to a 40th order matrix. The automatic floating decimal arithmetic of the 650 is utilized.

c) Gaussian Elimination.

d) Approximately 350 storage locations are used. The code is given in SOAP II format. Computation time for n^{th} order matrix is about $0.029\ n^3$ seconds.

e) If a matrix system has b constant vectors, then n + b working storage locations are required beyond the matrix and vector storage locations. Location 1936 contains zero to prevent optional punch out.

f) 650 with automatic floating decimal device and indexing registers. The alphabetic device is desirable.

LATENT ROOTS AND VECTORS OF A MATRIX

W. Granet
Boston University
Boston, Massachusetts

a) Calculates all the latent roots and vectors of a real but otherwise arbitrary matrix. All the latent roots and vectors are assumed real.

b) Matrix input is assumed to be in floating decimal form. The SIR routine is used for floating arithmetic operations.

c) The method used is described by Bodewig in "Matrix Calculus," pages 309-310.

d) As a guide to time estimation, one iteration for an 8 x 8 matrix requires approximately 15 seconds. Iterations dominate latent vector computations.

e) Three programs are included:

1. Program I can calculate all the latent roots and vectors of a matrix up to a maximum size of 20 x 20 (unless round-off errors interfere).

2. Program II can handle a maximum size of 25 x 25, but will calculate, at most, seven latent roots and vectors for this maximum size.

3. Program III involves more card handling than the other programs, but will handle a maximum size of 34 x 34 and obtain all 34 latent roots and vectors (unless round-off errors interfere).

f) Minimum 650.

SOLUTION OF SIMULTANEOUS LINEAR EQUATIONS
Prepared by IBM 650 Applied Programming

G. A. Black
IBM Endicott

a) The purpose of this program is to make available a routine to solve sets of simultaneous linear equations quickly, without inverting the associated matrix.

b) Range: For n equations in m unknowns, (n + 1) (n + m) must be less than or equal to 1999.

Accuracy varies with many factors.

c) Floating point computations and output. Input may be in fixed point.

d) The program is fabricated in immediate access storage and subsequently the entire drum is available for storage of elements. Additional routines may be added at the end of the drum, resulting in a smaller maximum number of elements. Solution time is approximately 1 1/2 seconds for a 3 by 6 matrix, 5 seconds for an 8 by 9 matrix, 33 seconds for a 10 by 26 matrix (16 identical solution vectors), and 74 seconds for a 20 by 20 matrix with one solution vector.

e) The user is cautioned that dependent equations and other factors may impair the value of the results. Checking of the solution is considered necessary.

f) 650 with IAS, indexing registers, and automatic floating decimal device.

MULTIPLE REGRESSION ANALYSIS

Arthur Cohen
IBM, Washington

September, 1955

a) Computes all components necessary for a complete regression and correlation analysis. There are four phases: (I) a logarithmic transformation of the initial data, V_i, to the form $x_i = \log V_i - C_i$ where C_i is an arbitrary constant or formation of new variables of the form $x_k = x_i x_j$; (II) Calculates means, standard deviations, and simple correlation coefficients; (III) part 1 computes the inverse of the matrix of simple correlation coefficients and part 2 computes partial correlation coefficients and multiple regression coefficients; (IV) computes the predicted values based on the regression equation or the residual between observed and computed dependent variable values.

b) For (I) initial variables ≤ 14, observations $< 10,000$; (II) variables ≤ 33, observations per variable $< 10,000$. Phases I and II are fixed-point, III and IV are floating.

c) Standard formulas are used.

d) The entire drum is used. Timing for phase (I) is at most $(45 + \frac{38}{3} N)$ sec.; (II) $(420 + N [\frac{n}{10} \{\frac{n(n-1)}{2}\} \frac{1}{2} + \{\frac{n(n-1)}{2}\} \frac{1}{4}])$ sec.; (III) part 1 $.072\, n^3$ sec., part 2 5 minutes; (IV) $(60 + \frac{nN}{5})$ sec. where n is the number of variables and N the number of observations.

e) Each phase may be used separately or in conjunction with the others. The program was designed for a specific application and some modification may be necessary in its general utilization.

f) Minimum 650.

SIMPLE CORRELATION COEFFICIENTS

R. Rind and K. Brokate
IBM, New York

February 29, 1956

a) Computes the means, standard deviations, and all simple correlation coefficients of n variables, each with k observations.

b) The maximum number of variables is $n = 31$ with $k \leqq 2002$. Input data are five-digit decimal numbers, either integers or fractions. Means and standard deviations are computed in fixed-point, with accuracy, $\bar{x} \pm 1 \cdot 10^{-10}$ and $s \pm 1 \cdot 10^{-9}$. The correlation coefficients are computed in both fixed and floating-point with respective accuracies $r \pm 1 \cdot 10^{-9}$ and $r \pm 5 \cdot 10^{-9}$. Intermediate results Σx, Σx^2, $k\Sigma x^2 - (\Sigma x)^2$, and $k\Sigma xy - \Sigma x \Sigma y$ are computed exactly.

c) The standard formulas are used.

d) Storage required is 856 locations 0000 to 0855. Data is stored in locations 0856 to 0855 + 8p where p is the number of input data cards per variable, each card containing 14 observations. The time required for $n \leqq 17$ is $\frac{n(n+3)(p+1)}{210} + .585$ minutes; for $17 < n \leqq 31$ it is $\frac{n(n+3)(p+1)}{180} + .585$ minutes.

e) No observations may be missing.

f) Minimum 650.

CORRELATION COEFFICIENT ROUTINE

J. W. Robinson, III
IBM, Houston

July 9, 1956

a) Computes the means, standard deviations, and product moment correlation coefficients of $n \leqq 50$ variables.

b) The number of observations per variable is unlimited. Input data are ten-digit fixed-point pure decimal numbers. Output is fixed-point, and computations are single-precision.

c) The standard formulas are used.

d) All locations except $\frac{n(n+1)}{2}$ to 1274 are used; for n = 50 the entire drum is used. Approximate time for 100 observations is 8 min. for n = 10; 29 min. for n = 20; 71 min. for n = 30; 125 min. for n = 40; 195 min. for n = 50. For other cases assume that the time varies linearly as the number of observations and as the square of the number of variables.

e) Self-loading and self-restoring.

f) Minimum 650.

ANALYSIS OF VARIANCE PROGRAM

W. Andrus
IBM, Endicott

a) Computes the sums of squares, with the exception of the high-order interaction term, necessary in an analysis of variance.

b) Fixed-point positive integers are used. These can be at most seven factors and eight levels per factor, one observation per cell, and a total of $\leq$ 16,500 individual digits in all data cells.

c) Does not apply.

d) Storage required is approximately 341 locations, 0000 to 0340. Timing information not given.

e) Fractions and negative numbers may usually be avoided by multiplication or addition of a constant without affecting the validity of the analysis. It is necessary that the data be punched and stored systematically by level from the innermost to the outermost factor.

f) Minimum 650.

AUTO-CORRELATION PROGRAM

W. E. Andrus, Jr.
IBM, Endicott

May 31, 1956

a) Computes the values of the auto-correlation function for up to 1500 data elements, or the values of the cross-correlation function for up to 750 data elements in each time sequence.

b) Arithmetic is fixed-point in the form x.xxxx xxxxx.

c) The standard formulas are used.

d) Storage required for the program and load routine is 301 locations 0000 to 0300; data locations are 0500 to 1999. Timing is $\frac{n^2}{2}$(.09) seconds where n is the total number of data elements.

e) The program is not optimized.

f) Minimum 650.

POLYNOMIAL OF BEST FIT BY LEAST SQUARES METHOD

M. A. Kelly and M. S. Dyrkacz
GE, Schenectady

April 2, 1956

a) Finds four polynomials, 1st through 4th degree, that give the best fit a given set of points.

b) The maximum number of points is 100. Floating-point arithmetic is used.

c) The method is least squares.

d) Storage required is 998 locations, 0000 to 0997. Time estimate not given.

e) Output includes the coefficients of the four polynomials, the original points, the values of the polynomials at the original abscissae, and the RMS of the error for each polynomial.

f) Minimum 650.

MULTIPLE CORRELATION FOR 50 VARIABLES

J. D. Hall
University of Indiana, Bloomington

a) Obtains all possible correlations (1225) of 50 variables of 3 digits each.

b) The maximum number of observations for each variable is 10,000. Arithmetic is fixed-point.

c) The standard formulas are used.

d) Storage required is approximately 350 locations. Timing information not given.

e) The output includes the sum, sum of squares, mean, sum of cross products, standard deviation, and the number of observations for each variable along with all possible correlations.

f) Minimum 650.

UNBIASED STANDARD ERROR OF THE REGRESSION COEFFICIENTS

W. T. Rooker, Jr.
IBM, Washington

a) Computes the unbiased standard error of the regression coefficients using as input the partial regression and correlation coefficients.

b) Floating decimal arithmetic is used. Maximum number of variables is 33 with a maximum of 10,000 observations per variable.

c) The standard formula is used.

d) Storage required is approximately 400 locations. Computing time for 33 variables is 25 seconds.

e) This program may use as input, without modification, the results computed in Multiple Regression Analysis, file number 6.0.001. SIR interpretive routine, not included in the listing, is used for the floating decimal arithmetic. A SOAP symbolic deck listing with a sample absolute deck listing is included.

f) Alphabetic device if the SOAP symbolic version is used.

WEIGHTED LEAST SQUARE POLYNOMIAL APPROXIMATION

R. E. von Holdt and J. R. Brousseau
University of California Radiation Laboratory, Livermore, California

May 22, 1956

a) Fits a weighted least square polynomial of order n to a set of m observation points, or obtains the solution of a system of n equations in n unknowns.

b) Limits for the least squares fit: $1 \leqq n \leqq 33$, $3 \leqq m \leqq 312$. Also $m(n + 3) \leqq 1250$ and $m \geqq n + 1$. Limits for a system of equations: $3 \leqq n \leqq 33$. Calculations are in floating-point.

c) An iterative method is used.

d) Storage required for the program is 750 locations 0000 to 0749; the rest of the drum is used to store data. Speed estimates not given.

e) The program includes an interpretive routine to perform the floating decimal arithmetic. In producing the nth order approximation, all other approximating polynomials from order one to n-1, and their respective residuals, are produced.

f) Minimum 650.

POLLY: POLYNOMIAL FIT BY LEAST SQUARES

Richard R. Haefner
Savannah River Laboratory, du Pont, Augusta, Georgia

September, 1956

a) Obtains a least squares fit of a polynomial $\sum_{i=0}^{N} a_i x^i$.

b) A maximum of n=100 experimental points is allowed. Maximum order of polynomial is N = 15. Input data are in fixed decimal mode, and output coefficients are in floating decimal.

c) Least squares method.

d) Approximately $0.0016n\ (N^2+10N+20) + 0.002(3N^3+10N^2)$ minutes are required for an Nth order polynomial with n data points. Storage required is approximately 2000 locations.

e) Four types of weighting factors are allowed: (1) uniform weighting, (2) weighting by inverse first power of the dependent variable, (3) weighting by the inverse second power of the dependent variable, and (4) arbitrary weight factors at each point.

f) Minimum 650.

STANDARD ERROR OF REGRESSION COEFFICIENTS

W. T. Rooker, Jr.
IBM, Washington

a) Computes the standard error of regression coefficients using as input the partial regression and correlation coefficients.

b) Floating decimal arithmetic is used. Maximum number of variables is 33 with a maximum of 10,000 observations per variable.

c) The standard formula is used.

d) Storage required is approximately 480 locations between 0000 and 0671, which includes the necessary SIR subroutines. Total time for 33 variables is 55 seconds.

e) This program may use as input, without modification, the results computed in phase III, b of Multiple Regression Analysis, file number 6.0.001. An absolute listing of the complete program is included in addition to a SOAP symbolic listing exclusive of the SIR subroutines.

f) Alphabetic device if the SOAP symbolic version is used.

SINH FIT

.\. R. Haefner April, 1955
Savannah River Laboratory, du Pont, Augusta, Georgia

a) Obtains a least squares fit to data obtained in a subcritical reactor. The relative activities of foils corrected or uncorrected for epithermal neutron background may be obtained.

b) Fixed point arithmetic is used.

c) Least squares.

d) Storage required is approximately 1550 locations. An average speed for a sinh fit to 20 experimental points is 3 minutes. Relative activities of foils are obtained at a speed of 20 points per minute.

e) The routine can obtain (1) a hyperbolic sine fit when the absolute experimental uncertainty of the data is of the same magnitude at each point, (2) a hyperbolic sine fit when the relative uncertainty is the same at each point, and (3) a J_0 (γ r) fit when the relative uncertainty is the same at each point. A general description of the routine is give in DP-143, January 1956, available from the Department of Commerce. Pages 29 through 34 of this report are included.

f) Minimum 650.

AUTOCORRELATION AND POWER SPECTRUM

Essor Maso and William J. Drenick
Hughes Aircraft Company, Culver City, California

January 14, 1957

a) Autocorrelation and power spectrum.

b) Fixed. Approximately 3 to 4 significant figures.

c) Numerical integration by addition of discrete input points.

d) 2,000 words. Non-relocatable.

e) Not to exceed 999 input points or 99 lags in autocorrelation.

f) Minimum 650.

CORRELATION ANALYSIS WITH ANNOTATED OUTPUT

Staff, Scientific Computing Center
IBM, Washington

December 31, 1956

a) Computes the means, standard deviations, and simple correlation coefficients for as many as 25 variables and 9999 observations providing both fixed and floating decimal output. However, with three exceptions, this routine may be substituted for phase II and output of this routine may be used as input to later phases of the "Multiple Regression Analysis on the 650." file no. 6.0.001. The exceptions are: (1) Program 6.0.014 will not handle more than 25 variables. (2) Observation numbers appear in different columns on the data cards so that 6.0.014 data cards cannot be directly used as input to phase IV. (3) 6.0.014 does not produce the means in a suitable card form for direct applications as input to phase IV.

b) Input data can be a maxium of 8 digits for each variable. Summations are accumulated in double precision fixed point.

c) The standard formulas are used.

d) The entire drum is used by the program. No accurate timing formula is available, but this routine will run at least twice as fast as phase II of "Multiple Regression Analysis" by A. Cohen.

e) Fixed point means and standard deviations are scaled. Header cards identify output.

f) Alphabetic 650.

CHI SQUARE FOR UP TO 10x10 CONTINGENCY TABLE

Albert Newhouse January 16, 1957
Computing and Data Processing Center, University Of Houston

a) This routine computes Chi square for systems up to 100 observations and up to 70 one-digit variables.

b) Chi square is computed in fixed point arithmetic for every variable versus every other variable.

c) Standard formulas are used with option for correction.

d) 1950 locations are needed. Available in SOAP and/or absolute.

e) Self-restoring, available in self-loading 5/c.

f) Minimum 650, alphabetic device if SOAP version is used.

CHI SQUARE AND PHI FOR 2x2 CONTINGENCY TABLE

Albert Newhouse
Computing and Data Processing Center, University of Houston

January 16, 1957

a) This routine computes Chi square and Phi for systems up to 100 observations and up to 70 one-digit variables.

b) Chi square and Phi are computed in fixed point arithmetic for every variable versus every other variable.

c) Standard formulas with option for correction.

d) 1286 locations are needed. Available in SOAP and/or absolute.

e) Self-restoring, available in self-loading 5/c.

f) Minimum 650, alphabetic device if SOAP version is used.

A STATISTICAL INTERPRETIVE SYSTEM FOR THE IBM 650 MAGNETIC DRUM CALCULATOR

G. E. Haynam
Case Institute of Technology
Cleveland, Ohio

a) A three address floating point statistical interpretive routine which is a modification of the interpretive routine by V. M. Wolontis described in IBM Technical Newsletter No. 11.

b) Some fixed point operations are included in order to preserve the accuracy in some statistical calculations.

c) Does not apply.

d) Storage required for the interpretive system is 1500 locations, 0500 to 1999. The time depends upon the operation being performed.

e) The trigonometric functions and negative multiply have been removed and the following operations added: float, mean, covariance, α_3^2, α_4, random number, negative, gamma function, normal probability, Poisson probability, binomial probability, cumulative Poisson, cumulative binomial, χ^2 test, t test, F test, clear, store loop box, restore loop box, general exponentiation, and two statistical read commands.

f) Minimum 650.

RAP - A REGRESSION ANALYSIS PROGRAM

C. E. Cates
T. H. Green
R. Y. Seaber
R. A. Stewart
Shell Oil Company
Houston Research Laboratory
Houston, Texas

a) A program written in SOAP and SIR to compute the constants and regression coefficients of polynomial equations which may contain up to 26 variables, of which up to 8 may be dependent. The equations may contain up to 26 terms, each of which may contain up to 5 independent variables. The variables can be independently changed by a number of different transformations as the data are entered.

b) Data are entered as positive, four digit floating decimal numbers. Internal operation is in the SIR mode.

c) Normal least squares techniques.

d) Program is in 2 parts, each of which uses the entire drum. Output from Part I is the input to Part II. Speed is a function of equation size, number of observations, and type of transformations.

e) Output includes variance of dependent variable error and value of student t for each coefficient.

f) Minimum 650. Alphabetic device permits printing header cards, but is not essential to obtain correct results.

SURFACE FITTING

W. S. Pickrell
IBM, Los Angeles

a) Computes the least squares coefficients and the sums of the squares of the residuals for a power series in either one or two independent variables.

b) Floating point arithmetic is used with the input and output in fixed point form with a decimal point indicator. Coefficients for up to a tenth degree power series in the independent variable may be computed. When used for surface fitting, a maximum size for the matrix of the observed values of the dependent variable is fifty by fifty.

c) The method of least squares with orthogonal polynomials is used.

d) The program is stored in drum locations 1000 through 1899. The first 1000 drum locations are used as working storage for the matrices. A time estimate cannot be made as it is dependent on the degree of the polynomial and the number of data points.

e) None.

f) Minimum 650.

REAL AND COMPLEX ROOTS OF ALGEBRAIC EQUATIONS

R. W. De Sio
IBM, Schenectady

a) Computes real and complex roots of equations of the form $f(z) = A_o z^n + A_1 z^{n-1} + \ldots + A_n = 0$.

b) The floating-point interpretive routine by George Trimble in Technical Newsletter No. 8 is used.

c) Method is an iterative scheme of Milne.

d) Storage required is 336 locations, 500 to 733 and 850 to 951. Timing varies with the number of iterations needed.

e) None.

f) Minimum 650.

ROOTS OF A FUNCTION OF A REAL VARIABLE

F. Edelman
RCA, David Sarnoff Research Center, Princeton

July 7, 1956

a) Locates the roots of an arbitrary function lying in a given interval and computes them to a specified precision.

b) The floating-point interpretive routine by Dr. V. M. Wolontis of Bell Laboratories in Technical Newsletter No. 11 is used.

c) Method is to detect a sign change in f(x) in an interval and then to successively halve this internal until the desired accuracy is obtained.

d) Storage required is 1200 locations, 0800 to 1999, which includes the interpretive routine.

e) The programmer specifies the function, interval, precision, and the initial increment of the independent variable. Multiple roots of an even order may not be detected. Program decks are available upon request from the author.

f) Minimum 650.

OPTICAL RAY TRACING

Dale J. Raar
IBM, Detroit

November 29, 1955

a) Determines the path of a beam of light as it passes through an optical system consisting of a number of different media with spherical boundaries.

b) Arithmetic is fixed-point in the form xx.xxxx xxxx. Any size system may be traced.

c) The standard formulas for refraction are used.

d) Approximately 300 locations are used for the program. Time required is less than one second per surface.

e) All rays are considered to be skew.

f) Minimum 650.

MOONSHINE

R. Stuart,
University of California Radiation Laboratory, Livermore, California

a) Solves the one-dimensional neutron diffusion equation. The multi-group diffusion equasion is solved for the case of a sphere, a cylinder, and a slab.

b) A maximum of three different material regions and eighteen groups can be handled. Fixed decimal arithmetic is used.

c) The method is an iterative process.

d) The entire drum is required. Total running time, using all eighteen energy groups, is about thirteen minutes.

e) Two or three iterations are usually needed for a solution.

f) Minimum 650.

PARACANTOR

S. P. Stone
University of California Radiation Laboratory, Livermore, California

a) Paracantor I is a two energy group, two region, time independent reactor code, which obtains a closed solution for a critical reactor assembly for cylindrical reactors of finite length and with a radical reflector of finite thickness. Paracantor II computes the fluxes, including the adjoint fluxes, from the output of Paracantor I.

b) Floating-point arithmetic is used.

c) The method, in general, follows the two energy group theory found in The Elements of Nuclear Reactor Theory by Glasstone and Edlund.

d) The entire drum is required. The average running time for Paracantor I is 5 to 8 minutes; for Paracantor II 5 minutes.

e) The program contains all of the load, punch, and interpretive routines, tables, and miscellaneous constants necessary for running.

f) Minimum 650.

ONE-SPACE-DIMENSIONAL MULTIGROUP

G. J. Habetler and V. A. Walbran
GE, Knolls Atomic Power Lab, Schenectady

December 1, 1956

a) Solves the one-space-dimension multigroup formulas.

b) Input is in fixed decimal form. Approximately 50 groups, each of a 50 point mesh, may be handled. The exact range of the many variables is given in the write-up.

c) The method is described in a 43 page paper which is supplied with the write-up and listing.

d) The entire drum is used. Timing is from 20 seconds to one minute per group for a 40-point mesh, depending on the choice of input data.

e) The program is divided into two parts, the Multigroup Calculation and the Power Calculation. Allowance has been made for variations in geometry, boundary conditions, and handling of scattering cross sections.

f) Minimum 650.

LOST, A CROSS-SECTION AVERAGING PROGRAM

C. J. Hibbert
G. E., Knolls Atomic Power Laboratory, Schenectady

a) Computes cross-section integrals over specified lethargy groups.

b) Input is in floating-point form. The maximum number of lethargy points is 200.

c) Integrations are performed using the trapezoidal rule.

d) Storage required for the program is 424 locations, 1571 to 1994. The rest of the drum is used for data storage. Time required for a typical compostion with six materials and self-shielding for 170 point and 15 point files is 12.5 minutes and 1.24 minutes respectively.

e) The program distinquishes between the absorption of moderator or non-fissionable materials and those of fissionable or associated fission product materials.

f) Minimum 650.

DONATE

Harvey Amster and
Roland Suarez
Westinghouse Bettis Plant, Pittsburgh Pa.

May 1956

a) Distribution of neutrons at thermal energies - a solution for the energy distribution of neutrons in equilibrium with an infinite homogeneous medium of pure monatomic hydrogen undergoing thermal motion. Allowing varying cross sections, elements other than hydrogen and a buckling turn for leakage from a finite volume.

b) Floating point.

c) Milne's Predictor-corrector formulas,
3 point Lagrangian interpolation,
5 and 8 point integration formulas.

d) 3 runs.

e) None.

f) Minimum 650.

MUFT III

R. L. Hellens
R. W. Long, and
B. H. Mount
Westinghouse Electric Corp., Pittsburgh, Pa.

July 1956

a) Computes the energy distribution of neutrons having a given Faurier mode in an infinite medium.

b) Four approximations are provided with the inclusion of isotropic inelastic scattering, resonance capture, and fast fission. Fixed point arithmetic is used.

c) The output includes flux, current, and slowing density spectra and computes the fast constants for a variety of few group schemes.

d) Solution requires two runs through the computer. The entire drum is used.

e) Twenty is the maximum number of elements that can be used as input for any one problem.

f) Minimum 650.

LIL ABNER: A FEW-GROUP ONE-DIMENSIONAL CODE

H. Bohl
G. Gelbard
R. Suarez
Westinghouse Electric Corp., Pittsburgh, Pa.

a) Lil Abner is a one-to-eight group code designed, primarily, to treat one-dimensional reactor and cell problems.

b) This code will handle a maximum of ten regions and one hundred mesh points. Floating point arithmetic is used.

c) The method is an iterative process.

d) None.

e) All physical parameters in the Few-Group equations as well as the mesh width must be constant within each region. In the fast groups these parameters may be obtained directly from MUFT III (8. 2. 006) calculations or from microscopic cross sections fitted to match MUFT III results. Sample problem is enclosed.

f) Minimum 650.

K-CODE

W. V. Baxter
Savannah River Laboratory, du Pont, Augusta, Georgia

December, 1955

a) Obtains the transients of neutron flux in response to a change in the reactivity of a reactor.

b) Eleven delayed groups of neutrons and two power coefficients of different relaxation times are allowed. Floating decimal arithmetic is used.

c) Theoretical treatment is given in a paper by H. D. Brown, submitted for the journal "Nuclear Science and Engineering" under the title, "A General Treatment of Flux Transients."

d) Storage required is approximately 1800 locations. One time increment requires 30 seconds.

e) A very general change in reactivity as a function of time can be made by proper input parameters. The set of differential equations is solved by integration of the associated difference equations.

f) Minimum 650.

BEEHIVE AND HORNET
REACTOR CODES FOR SPHERICAL GEOMETRY

S. P. Stone (Beehive)
S. P. Stone and R. Shaffer (Hornet)
University of California Radiation Laboratory
Livermore, California

a) "Beehive" is a five energy group, two region, time independent, spherical reactor code. It considers the problem of a reactor system in which the core material is assumed to be at a higher energy (temperature) than the reflector material. The companion code, "Hornet," computes the neutron fluxes for the critical assembly determined by the Beehive calculations.

b) The majority of arithmetic is performed in interpretive floating point.

c) The code obtains a closed solution for the critical reactor assembly by a procedure which is a logical extension of normal two group theory. The solution is obtained by an iterative process.

d) Storage: 2,000 words. Speed: "Beehive" requires 2-1/2 minutes per iteration, and 5 or 6 iterations. "Hornet" requires 7 minutes.

e) Only a preliminary investigation has been made for cases where the G/2 2-5 spacing is "close," a situation in which the critical 10 x 10 determinant evaluation might be subject to error.

f) Minimum 650.

UNCLE I
THE DIFFUSION EQUATION IN CYLINDRICAL GEOMETRY

R. R. Haefner
E. I. du Pont de Nemours & Co., Inc.
Savannah River Laboratory,
Aiken, S. C.

a) UNCLE I - Solution of the Neutron Diffusion Equation in Cylindrical Geometry.

b) Uses network of 9 points in the r-direction and 16 in the z-direction. Fixed decimal.

c) Extrapolated Liebmann Method.

d) 20 seconds per iteration.

e) One group only.

f) Minimum 650.

UNCLE II
THE DIFFUSION EQUATION IN (x, y) SPACE

R. R. Haefner
E. I. du Pont de Nemours & Co., Inc.
Savannah River Laboratory,
Aiken, S. C.

a) UNCLE II - Solution of the Neutron Diffusion Equation in (x, y) Space.

b) Uses network of 9 points in the x-direction and 16 in the y-direction. Fixed decimal.

c) Extrapolated Liebmann Method.

d) 20 seconds per iteration.

e) One group only. $\frac{\partial \phi}{\partial x} = 0$ at $x = 0$ is a restriction on the types of problems that can be solved.

As the program for UNCLE II is the same as that for UNCLE I with a few exceptions, the write-up for UNCLE II does not include a complete listing of the program instructions, but only the exceptions. A complete listing is included in the UNCLE I write-up.

f) Minimum 650.

UNCLE III
THE DIFFUSION EQUATION IN ONE DIMENSION

R. R. Haefner
E. I. du Pont de Nemours & Co., Inc.
Savannah River Laboratory,
Aiken, S. C.

a) UNCLE III - Solution of the Neutron Diffusion Equation in One Dimension.

b) Uses network of K + 1 points, $K \leqq 36$. Fixed decimal.

c) Extrapolated Liebmann Method.

d) Time required: 0.16 K seconds/iteration.

e) One group only.

f) Minimum 650.

VALPROD

C. M. White
GE, Vallecitos Atomic Laboratory
Pleasanton, California

a) One dimensional reactor flux calculation for slab, cylinder, and sphere.

b) Fixed point. Range is discussed in the report; it is too complex for this abstract.

c) This is PROD II in a form more convenient for use. PROD II is described in abstract 8.2.003. References are KAPL-1415, KAPL-1531, and GEAP-0952.

d) Full 2000 words of drum. Non-relocatable.

e) None.

f) Minimum 650.

P-3 FLUX DISTRIBUTION

J. W. Weil
P. Cabral
GE Atomic Power Equipment Dept.
San Jose, California

a) This code computes the one-velocity neutron flux distribution in concentric cylindrical geometry using a P_3 spherical harmonics approximation to the neutron transport equation. Anisotropic scattering is included and each region may have different properties and may or may not have a neutron source. The properties of any one region and a source in that region must remain constant throughout the region.

b) There is no limit to the number of concentric cylindrical regions which can be handled. The code operates in floating point interpretive mode.

c) The P-3 Flux Code is an analytic solution of the P_3 flux problem. Details of the code have been published through the American Nuclear Society. Further information may be obtained from KAPL 1173 (Secret).

d) The program occupies virtually the entire 2000 word drum and is thus not relocatable.

e) The following difficulties have been observed but do not limit the normal utilization of the code.

i) Regions of high cross section at large radii will cause a machine stop because the calculated Bessel functions become too large for the floating point representation.

ii) Regions of small cross sections which do not include the origin will cause difficulty. This is most easily recognized by irregularities in the resulting fluxes.

iii) The code will not handle regions with zero absorption. The insertion of a small absorption cross section will, however, not affect the flux distribution and will permit the code to operate.

The P-3 Flux Code will automatically compute the neutron flux distributions throughout the regions in the problem (the number of points computed is controllable) and will also provide average fluxes in each region.

f) Minimum 650.

DMM (DIFFUSION MULTIGROUP MULTIREGION)

E. J. Leshan
American - Standard

R. J. Levit
IBM Corporation

J. Franklin
Burroughs Corporation

a) To calculate the spatial and energy distribution of the neutron flux in a spherical, cylindrical, or slab reactor as well as the reactivity of the system.

b) Single precision floating point arithmetic.

c) The method is that developed by Leshan and Franklin, which permits scattering from any energy level to all lower levels. Solution is obtained in a single pass.

d) The following times have been observed:

I	K	N	Time (in seconds per iteration)
2	3	60	32
10	1	20	66
10	3	40	131

e) Restriction. $3N + 2K + I\left[N + 4 + 1/2\ K\ (I-5)\right] \leq 1601$, where I is the number of energy groups, K is the number of regions, and N + 1 is the number of mesh points.

f) 650 equipped with indexing registers and floating decimal device.

BALL
A REACTOR CODE FOR SPHERICAL GEOMETRY

S. P. Stone
T. B. Kerr
University of California
Radiation Laboratory
Livermore, California

a) Ball is a two-energy-group, two-region, time-independent reactor code. It obtains a closed solution for a critical reactor assembly of spherical geometry, and also computes the normal and adjoint fluxes.

b) Floating point. Accuracy is dependent on input data.

c) Iterative solution.

d) Approximately 1,700 storage locations are used. A typical problem requires eight to ten iterations and takes approximately 2 1/2 minutes.

e) None

f) Minimum 650.

N E D

D. B. MacMillan
GE Knolls Atomic Power Laboratory
Schenectady, New York

a) NED is a 650 program for computing the Wigner-Wilkins kernel (reference: AECD 2275).

b) The value of the kernel is computed in fixed point arithmetic at the points of an N by N mesh, where N may not exceed 34. Accuracy of 5 to 7 decimal places is obtained; see the write-up for a more specific statement.

c) The numbers are computed in parallel, or parameter study, style.

d) The program uses the whole drum and is not relocatable. For H moderator, sample calculations required $\frac{N^2}{7}$ minutes. For Be moderator, sample calculations required $\frac{N^2}{20}$ minutes.

e) None.

f) Minimum 650.

SURVEY TRAVERSE

J. T. Ahlin and G. E. Mitchell
IBM, Houston

May 1, 1956

a) Computes the departures and latitudes for each traverse line, the x and y coordinates for each station, and the length, bearing, departure and latitude of the closure.

b) Angle data are to either the nearest second or the nearest hundreth of minute; distance data in the form xxxxx.xx feet. Sines and cosines are computed to six decimal places.

c) Does not apply.

d) Storage required is about 500 locations between 0000 and 0999. Speed is 100 stations per minute.

e) Self-restoring.

f) Minimum 650.

CUT AND FILL PROGRAM

R. W. Blaylock and J. M. Kibbee
IBM, Houston

a) Computes the amount of cut and fill volume between survey stations on a highway using the data from the original survey and from either a final survey (for billing) or design specification.

b) Fixed-point arithmetic is used with a maximum of 100 points per station with no limit to the number of stations. Volumes are punched to the nearest cubic yard, areas to the nearest hundredth square foot, horizontal distances to the nearest tenth of a foot, vertical distances to the nearest hundredth of a foot.

c) The average end-area is used for computing volumes.

d) Storage required is about 975 locations assembled between 0800 and 1950. Input data and computed tables occupy locations 0000 to 0799. Timing is a function of the number of stations and readings at each station. For 25 readings per station and 100 stations per mile computations require about 15 minutes per mile.

e) For design purposes the program also computes the slope stake points (intersections of proposed road with terrain). A SOAP symbolic deck listing in addition to an absolute deck listing of the program assembled between 0800 and 1950 is included.

f) Alphabetic device if the SOAP symbolic version is used.

SURVEY TRAVERSE WITH BALANCING

J. D. Erdwinn
IBM, Houston

July 25, 1956

a) Computes the original and balanced values of length, bearing, components of each closure, and the coordinates of each station from the field data of a closed traverse.

b) The maximum number of stations is 300; latitudes and departures may not exceed 99,999.99. A description of the error of closure is computed.

c) Balancing is achieved by means of the compass rule.

d) The program requires about 700 storage locations and has been assembled in 0000 to 0999. Locations 1000 to 1899 are used for tables. Speed is approximately 2500 courses per hour.

e) A SOAP symbolic deck listing in addition to an absolute deck listing of the program assembled between 0000 and 0999 is included.

f) Alphabetic device if the SOAP symbolic version is used.

CUT AND FILL

J. M. Kibbee and J. W. Robinson
IBM Houston

a) Computes slope stake intercepts, cut, fill, and net volumes, adjusted, and accumulated volumes.

b) Fixed decimal.

c) Average end-area method.

d) Uses entire memory: approximately 1200 program steps
approximately 800 table locations.
Speed varies with type of problem run.

e) Road is described in terms of crown height and width, and slope depth and width.

f) Minimum 650.

MOMENT DISTRIBUTION

J. D. Hutchinson
University of Houston
Computing and Data Processing Center
Houston, Texas

a) Computes the bending moments in structural members of a rigid frame, given fixed end moments.

b) Meets all engineering requirements. The program is written in fixed point.

c) The "Moment Distribution" method of Hardy Cross is used. (See Paper 1793, Trans, A. S. C. E., 1932.)

d) Program requires 540 memory locations; data require 10 words per member in the frame. Speed: 1/8 to 1/10 seconds per member per joint per iteration. Relocatability: Program is written in SOAP, but all data locations are in absolute.

e) Handles frames with up to 100 members. Not more than 8 members can meet at any given joint.

f) Minimum 650.

TRUSS ANALYSIS

A. A. Aucoin and J. D. Hutchinson
University of Houston
Computing and Data Processing Center
Houston, Texas

a) Computes axial forces in statistically loaded, simple, determinate, pinned trusses.

b) Range: Loads varying from 1 to 99999 (units arbitrary). Accuracy: Depends on number of significant figures in data; 1 part in 500 accuracy can be obtained on large trusses. Program is written in fixed point.

c) The "Method of Joints" is used. (See any standard text on truss analysis.)

d) Program requires 1200 memory locations; data require six locations per member. Speed: Approximately jj seconds where jj is the number of joints in the truss. Relocatability: Since the program and data occupy most of the drum, it is not convenient to relocate. The program is written in SOAP, however.

e) Self restoring. Will process either many loading configurations for the same truss or many trusses, or any combination, in sequence, automatically. For indeterminate trusses, see abstract entitled, "Connector and Redundancy Programs for Indeterminate Truss Analysis."

f) Minimum 650.

TRUSS ANALYSIS

A. A. Aucoin
J. D. Hutchinson
University of Houston
Computing and Data Processing Center
Houston, Texas

a) Computes axial forces in statically loaded, simple, determinate, pinned trusses.

b) Range: Loads varying from 1 to 99999 (units arbitrary). Accuracy: Depends on number of significant figures in data; 1 part in 500 accuracy can be obtained on large trusses. Program is written in fixed point.

c) The "Method of Joints" is used. (See any standard text on truss analysis.)

d) The program requires 1200 memory locations; data require six locations per member. Speed: Approximately jj seconds where jj is the number of joints in the truss. Relocatability: Since the program and data occupy most of the drum, it is not convenient to relocate. The program is written in SOAP, however.

e) The program is self restoring and will process either many loading configurations for the same truss or many trusses, or any combination, in sequence, automatically. For indeterminate trusses, see Abstract 9.2.007, "Connector and Redundancy Programs for Indeterminate Truss Analysis."

f) Minimum 650.

CONNECTOR AND REDUNDANCY PROGRAMS FOR INDETERMINATE TRUSS ANALYSIS

Irene Tung
University of Houston
Computing and Data Processing Center
Houston, Texas

a) Designed to compute true axial forces in all members of indeterminate trusses from output of "Truss Analysis" program.

b) Fixed point except the Sweeney Matrix Inversion routine which is incorporated.

c) Castigliano's Theorem of Least Work is applied. (See any standard text on indeterminate structures.)

d) The Connector requires 750 locations for program and data. The Redundancy Program requires 1725 locations for program and data. The programs are written in SOAP in fixed point except the Sweeney Matrix Inversion program which is incorporated.

e) Up to 24 redundants in a truss can be handled.

f) Minimum 650.

GEORGIA SKEWED BRIDGE PROGRAM

C. P. Reed
Rich Electronic Computer Center
J. M. Nieves-Olmo
State Highway Department of Georgia
Atlanta, Georgia

a) This program determines the placement of bents, the intersection of radial lines with concentric circles, the chord distances between bents, and other related data for substructure of a curved bridge.

b) Accuracy to tenths of a second for angles. Most calculations are performed in floating decimal with part of input being submitted in floating decimal.

c) Makes use of plane geometry and trigonometry which pertain to chords of concentric circles and radial triangles.

d) Uses entire drum. Speed: 4 seconds per radius per bent.

e) Can handle any number of bents and up to 17 concentric circles at each pass. Can handle either left, right, or partially skewed bridge.

f) Minimum 650.

GEORGIA SKEWED BRIDGE PROGRAM

C. P. Reed
State Highway Department of Georgia
Atlanta, Georgia

a) This program determines the placement of bents, the intersection of radial lines with concentric circles, the chord distances between bents, and other related data for substructure of a curved bridge.

b) Accuracy to tenths of a second for angles. Most calculations are performed in floating decimal with part of input being submitted in floating decimal.

c) Makes use of plane geometry and trigonometry which pertain to chords of concentric circles and radial triangles.

d) Uses entire drum. Speed: 4 seconds per radius per bent.

e) Can handle any number of bents and up to 17 concentric circles at each pass. Can handle either left, right, or partially skewed bridge.

f) Minimum 650.

MOMENT DISTRIBUTION

P. Yeager
L. C. McReynolds
Computer Section
Washington Department of Highways
Olympia, Washington

a) Computes final end moments in beams and in column tops of continuous beams built integrally with columns when distribution coefficients, carry-over factors and fixed-end moments are given.

b) Will solve any single story continuous frame bridge structure with up to 15 spans. All data is in fixed point.

c) Uses Hardy Cross method of moment distribution.

d) Program occupies 1158 positions of memory storage and is not relocatable. Speed is 3 seconds per joint.

e) None.

f) Minimum 650.

TEXAS ENGINEERING SUBROUTINES

Texas State Highway Department
Austin, Texas

a) To convert degrees to radians, radians to degrees, and bearing to slope, and to perform 20 digit divisions.

b) Range: 0.00000000 to 9.99999999 radians.
Accuracy: XXX° XX' XX.X"
Fixed point arithmetic.

c) Normal conversion formulas.

d) Locations: 1801-1899. Non-relocatable.

e) None.

f) Minimum 650.

FORECASTING ZONAL TRAFFIC VOLUMES

J. Petersen
Computer Section
Washington Department of Highways
Olympia, Washington

a) Computes future zone-to-zone traffic movements given the present zone-to-zone movement and the estimated growth factors for each zone, using a method of successive approximations.

b) Will solve any system with up to 192 zones. All data is in fixed point.

c) Uses the method of Howard W. Bevis presented in "Traffic Quarterly" Volume X, No. 2, April, 1956, pages 207-222, entitled "Forecasting Zonal Traffic Volumes."

d) Program occupies 930 positions of memory storage and is not relocatable. Speed is punch speed (100 per minute).

e) None.

f) Minimum 650.

MAXIMUM DENSITY OF GRANULAR MATERIALS

R. V. LeClerc
H. E. Sandahl
Materials Laboratory
Washington Department of Highways
Olympia, Washington

a) Computes points on a curve for determination of the maximum densities of coarse granular materials.

b) Input and output are in fixed point.

c) Used with laboratory method for determining maximum density developed by H. W. Humphres.

d) Program occupies 363 positions on drum and is not relocatable. Speed is 2 seconds.

e) None.

f) Minimum 650. Alphabetic device is required if alphabetic identification is used.

ANALYSIS OF LATERALLY LOADED PILES

C. B. Rader, Sr.
C. R. Hobby
E. I. Organick
University of Houston
Computing and Data Processing Center
Houston, Texas

a) Computes lateral deflection, bending moment, shear, fiber stress due to vertical as well as horizontal loading, and soil pressure for t + 1 positions along a pile divided into t sections ($t \leq 49$). Piles are assumed to be made of pipe or to have a circular cross section.

b) The program is written in fixed point machine language; range and accuracy are discussed in program write-up.

c) Focht and McClelland method (see Texas Engineer, Vol. 25, nos. 9, 10, 11, Sept., Oct., Nov., 1955).

d) The program is not relocatable and uses approximately 1000 storage locations. Time required, for each wall thickness, is (t + 3) seconds plus punch-out time, where t is the number of divisions of the pile; punch-out occurs at maximum rate.

e) Does not apply.

f) Minimum 650.

GRADE PROFILE DESIGN

D. C. Bardell
IBM, South Bend, Indiana

a) Computes all data pertinent to any grade profile design of any highway. The position of the profile may be decided by the engineer. Input consists only of the coordinates of the PI stations, the length of the vertical curves, and stationing for which calculations are desired. Output consists of station, tangent grade elevations, length of vertical curve, vertical curve corrections, per cent of tangent grade, and finish grade elevations.

b) Fixed point arithmetic is used. Any stationing $\leq 999 + 99.99$ may be accommodated. All calculations are rounded to hundredths of a foot, and percentage is rounded to thousandths.

c) Standard formulas are used.

d) The program consists of approximately 220 instructions and constants and is contained in locations 0000 - 0299. It is relocatable. Timing is dependent on the number of readings for which calculations are to be made. Assuming calculations are to be made every 25 feet, it takes approximately two minutes and ten seconds to compute the grade profile for one mile.

e) The program is written in machine language and, if it is to be relocated, one constant must be changed. The program also applies to any project where fitting of parabolic curves is desired.

f) Minimum 650.

REVISED TRAVERSE AND TRAVERSE ADJUSTMENT COMPUTATION

J. A. Haller
California Division of Highways
Sacramento, California

a) This routine calculates traverse data for the typical highway survey, right of way, or design problem. Input is in the form of one card per course. Any two unknowns within a traverse may be accepted. Results are punched one course to a card and show identification, distance, bearing, sine, cosine, latitude, departure, and coordinates for regular courses. Areas are obtained for closed figures and segment areas are also computed. The factors developed in one traverse may be stored for use in a later traverse. Where two mathematically correct solutions are possible, both solutions are presented from a single set of input data, and the engineer must choose the proper solution.

b) Ninety-eight regular courses may be submitted for each traverse. Cards need not be sorted by course number, but all cards for a given traverse must be together. Distances are given to thousandths of feet and bearings to seconds. Functions are computed to nine decimal places.

c) Library subroutines used are from Technical Newsletter #9 for sine, and cosine, arctangent, and arcsine.

d) Ninety-eight locations each are required for storage of sine, cosine, distance, and bearing. Other program and temporary storage requirements use the remainder of the two thousand drum locations, with the exception of seventy-nine locations. Speed is about two thousand courses per hour. The program is considered optimum and is not in relocatable form.

e) Some coded stops may be reached because of incorrect input data.

f) A 650 with twenty pilot selectors, half-time emitters, and alphabetic device is used.

CONTOUR CHART OF TRIP DESIRES

J. A. Haller
California Division of Highways
Sacramento, California

a) This program computes the desire line trip values for each coordinate point within a traffic survey area. The output from the program may be listed with proper spacing to post contour values. The listing may then be used to draw a contour chart of trip desires.

b) Up to approximately 1750 contour points may be posted in one pass of the trip cards. Coordinate boundaries for each pass must be set up.

c) The X and Y coordinates of each point along a straight line from origin to destination are computed. The number of points computed for any one trip will be one more than the number of ordinates crossed by the longer axis of the trip.

d) The entire program requires about 300 locations, but this number may be reduced if the punching phase is separated from the reading phase. The program should not be relocated except to separate punching from reading phases. Speed varies with the concentration of trips within the particular swath being processed.

e) Reading of trip cards may be suspended and the trip values for each coordinate point may be punched out at any time so that the 650 does not need to be reserved for the entire time necessary to compute a given swath.

f) Minimum 650.

FREEWAY ASSIGNMENT PROGRAM

California Division of Highways
Sacramento, California

a) Determines best alternate route for a proposed freeway based on time-rate-distance studies of existing traffic.

b) Fixed decimal.

c) Formula as outlined by the Traffic Section, California Division of Highways.

d) Uses all locations except 1000 and 1999.

e) Will handle one alternate freeway at a time and up to 3 speeds on city streets.

f) Minimum 650.

CURVED BRIDGE PROGRAM

Texas Highway Department
Austin, Texas

a) This program relieves the detailer of much of the laborious computation involved in the plan preparation of a curved bridge.

b) Fixed point. Accuracy varies for different variables in program.

c) Mathematical formulas as now used by bridge designers.

d) Optimized through most of memory. About 500 program steps.

e) Only 20 bents may be computed at one time. The values of radii are limited to less than 10, 000. Other limitations given in write-up.

f) Minimum 650.

COMPOSITE BEAM

J. A. Haller
R. E. Shields
California Division of Highways
Sacramento, California

a) This program will compute steel girder size and all other factors needed to complete the design of a concrete-steel composite girder.

b) Input is in fixed point, but the SIR routine in floating point is entered for the solution to several equations. Output is in fixed point.

c) The routine picks a trial size of top and bottom flange, computes the stresses on such a beam, and then modifies top and bottom flange sizes separately as a result of the test of the stresses. When both top and bottom flanges are within the proper stress band, the program computes reductions in flange sizes, end reactions, or shear stress, and punches results. A single card input produces a single card output for each beam to be designed.

d) Approximately 1700 locations of table, instruction, and temporary storage are used. Speed varies, but the average beam will be designed in 35 seconds.

e) Provision is made to compute initial factors which are not specified by the engineer. The minimum data include span length, spacing between girders, structure depth, and steel stress. If other data are given, these data will be used in place of values computed from the minimum.

f) Minimum 650.

THREE CENTER CURVES FOR SHORT RADIUS TURNS

California Division of Highways
Sacramento, California

a) This program performs the computations of short radius turns as set forth in the Planning Manual of the California State Highway Department.

b) The value of the angle Δ cannot fall within the ranges between 179°55' and 180°05', and between 359°55' and 0°5'.

c) Uses IBM sine-cosine, square root, and arc-sine subroutines.

d) Uses approximately 650 locations. Can be relocated anywhere on drum.

e) The program was written for the ranges prescribed in the Planning Manual, so not all possible variations have been tested.

f) Minimum 650.

TRAVERSE AND COORDINATE PROGRAM

K. F. Kohler
R. R. DeClark
Bureau of Public Roads
Portland, Oregon

a) Using either Stations and Deflection Angles right or left, Length of Courses and Deflection Angles right or left, or Stations and Azimuths as input, the Bearings, Stations, Length of Courses, Course Lats. and Deps. and Coordinates of angle points are computed. Using P. I. Numbers and Coordinates as input, the Bearings, Delta Angles, and Length of Courses are computed. In all, fourteen different problem types are computed.

b) Coordinates CC, CCC, CCC. CC, Bearings N. or S. DDMMSS E. or W., Stations SSSS+SS. SS, Deflection Angles DDDMMSS R. or L., Delta Angles DDDMMSS, P. I. Numbers PP, PPP, PPP, and Course Lengths LLL, LLL. LL, (L, LLL. LL when using coordinates as input). The subroutines used are SR-3 (Square Root), SC-1 (Sine - Cosine) and AS-1 (Arcsine). Program is in fixed point.

c) Does not apply.

d) Storage required is about 1000 locations between 0000 and 1836. Speed is 40 courses per minute.

e) Program is written in SOAP.

f) 650 with alphabetic device.

DETERMINATION OF COEFFICIENTS FOR THE BENEDICT EQUATION OF STATE

C. R. Hobby
University of Houston
Computing and Data Processing Center
Houston, Texas

a) Determination of Coefficients for the Benedict Equation of State.

b) Floating point (SOAP - SIR)

c) Special least square fitting originally developed by Brough, H. W., Schlinger, W. G., and Sage, B. H., Industrial and Engineering Chemistry, 43, p. 2442, November, 1951.

d) Entire drum is used.
Speed: (7N + 140) seconds for first set of coefficients, (1. 5N + 140) for succeeding sets,
2N seconds for statistical summary.
N = the number of data points.

e) Does not apply.

f) Minimum 650.

ELECTRICAL POWER SYSTEM TRANSIENT STABILITY CALCULATIONS

J. E. Rowe and J. L. Gabbard, Jr.
Union Carbide Nuclear Co., Oakridge, Tenn. November 1, 1956

a) It is possible to make the transient stability calculations for any system that can be represented by 19 equivalent machines or less. However, if the number of equivalent admittances required to represent the network does not exceed 200, a program limit of approximately 50 machines is possible (a 30 machine system has been studied). Induction machines as well as synchronous machines can be handled.

b) Uses fixed decimal arithmetic.

c) Uses transient stability theory, symmetrical component theory, and network theory. Makes use of Starr's equivalent circuit for the n - terminal network expressed in matrix form and as admittances rather than impedances. Calculations are made in the per unit system and care must be exercised in selecting the system base in order to avoid field excessions with the fixed decimal program. The transient stability differential equations are solved using the method of 1st order forward differences.

d) Uses 718 words plus data and output. Time approximately 1 1/2 - 2 1/2 hours depending on variables.

e) Contains an excellent flow chart.

f) Minimum 650.

NETWORK REDUCTION

P. E. Scott and E. M. Kidd
Union Carbide Nuclear Co., Oak Ridge, Tenn.

October 19, 1956

a) A network reduction program - discribes an automatic method of reducing an electrical power network to a smaller equivalent network.

b) Limitations as to size of matrix to be handled are $n \leq 20$, $n^2 + nb \leq 800$

$n =$ order of M $b =$ order of K

Uses floating point arithmetic. The matrix of coefficients for the entire system is partitioned into M and K which represent those junctions to be eliminated and those to remain respectively.

c) Matrix theory and network theory.

d) Approximate time $(.576n^3 + 1.273nb + .726)$ seconds storage required 460 words plus data and output.

e) Number of output cards $= 1 + b(b+1)/2$
Has an excellent flow chart. Applicable to linear, bilateral, passive networks.

f) Minimum 650.

CALCULATION OF PIPING SYSTEM EXPANSION STRESSES

M. Alfieri, B. Whipple, P. O'Neill
General Dynamics Corp., Electric Boat Division, Groton, Conn.

a) Calculates piping systems with three anchors and no intermediate constraints or the equivalent case of two anchors with one constraint.

b) Input-output is in fixed decimal form.

c) The Kellog method is used.

d) The program is divided into three parts with a total of 2500 instructions. The three parts are processed as one complete operation and the entire drum is used.

e) A write-up of this program is in Technical Newsletter No. 10, pp. 195-213. Operator's notes, deck listing and description, and 533 wiring instructions are available from the 650 Program Library.

f) Minimum 650.

PIPE STRESS ANALYSIS

W. S. Pickrell
J. H. Rogers
L. S. Woo
IBM, Los Angeles

a) Computes the bending moment, torsional moment, bending stress, torsional stress, and the resulting combined stress at each end and the midpoint of every bend or elbow in a piping system. Also, the three moments and three forces acting at each anchor are computed..

b) Either two or three anchor problems with no intermediate restraints may be analyzed. The piping system may include any number of members in any arrangement in space. There may be any changes in section or material within the system and the branches may be at different operating temperatures. All computations are performed in floating point while both the input and output are in fixed point form.

c) The Kellogg Method is used for the calculations, while the stresses and the anchor reactions are computed according to the ASA Pressure Piping Code.

d) The program consists of two parts, each of which uses the entire drum. An average two anchor problem is completed in approximately six minutes, while the average three anchor problem uses approximately twelve minutes of machine time.

e) Part I of the program is loaded on the drum and intermediate results for all problems to be analyzed are punched. These are used with Part 2 of the program and the final answers for all problems are punched. Two test problems and detailed instructions as to how to prepare the input data are included in the write-up.

f) Minimum 650.

WELL BORE DEVIATION RECORD

J. T. Ahlin and G. E. Mitchell
IBM, Houston May 1, 1956

a) Given the distances, bearings, and inclinations at various stations in a well bore, this routine computes the well bore deviation record, the depth and horizontal components of the bottom hole, and the x, y, and z components and coordinates for each station.

b) Angle data are to either the nearest second or the nearest hundreth of minute; distance data in the form xxxxx.xx feet.

c) Does not apply.

d) Storage required is about 500 locations between 0000 and 0999. Speed is about 60 stations per minute.

e) None.

f) Minimum 650.

P-V-T DATA CALCULATIONS

A. Cohen
IBM, NY DPC

a) Program uses the Benedict equation to compute the density roots, entropies, enthalpies and other quantities for methane, ethane, propane, butane and pentane at pre-selected temperatures and pressures given in either English or c. g. s. units.

b) Fixed point arithmetic with different scaling for English and c. g. s. units. Accuracy depends on quantity considered.

c) Uses Benedict equation. Exponential and logarithmic routines are employed.

d) Program scattered optimumly over the whole drum. A temperature-pressure combination takes 3-4 seconds, depending on number of iterations required.

e) None.

f) Minimum 650.

EQUILIBRIUM FLASH CALCULATION

M. E. Klecka
R. Y. Seaber
Shell Oil Company
Houston Research Laboratory
Houston, Texas

a) Calculates isothermal equilibrium flash vaporizations where the feed composition and K values are specified.

b) A maximum of 30 components can be used. Floating point arithmetic is employed, and closure accuracy is ± 0.0001 mole fraction, based on the liquid product from the flash stage.

c) Conventional isothermal equilibrium flash calculation equations are used.

d) 1400 locations are used for program and data. The time per calculation depends upon number of components and the system but is generally 3-6 minutes per completed calculation.

e) Three check features are incorporated into the program:
1. The system must be above the bubble point.
2. The system must be below the dew point.
3. The sum of the mole fractions of the feed must equal 1.

A violation of any one of the above conditions will cause rejection of the particular problem by the machine. The name card identifying the problem will be punched followed by another card which gives the reason for rejection.

f) 650 equipped with alphabetic device.

LINEAR PROGRAMMING

H. F. Smith
IBM, Chicago

a) Solves a linear programming problem.

b) All numbers are of the form xxxx.xxxxxx. An M by N system may be solved where $M \leqq 30$, $N \leqq 59$ and $M(N+1) < 1400$ (these values pertain to the system after the slack vectors and artificial vectors have been adjoined).

c) Method not given.

d) The entire drum is used. Time required is approximately .09 MN seconds for one iteration.

e) Input consists of matrix elements, cost coefficients, indices of basis, and constants. At the end of each iteration the program punches out the number identifying the variables in the basis, the values of these variables, the value of the functional, and an iteration count.

f) Minimum 650.

LINEAR PROGRAMMING

L. S. Woo
IBM, Los Angeles

March 23, 1956

a) Solves a linear programming problem.

b) A maximum of 97 equations, not including the objective functions, is possible. The number of variables is unlimited. Input is 10 digit fixed-point numbers which are converted to double precision floating-point numbers for the calculations.

c) Method is Recursive Generation of Vectors for the Modified Simplex Method as described by Kurt Eisemann.

d) The entire drum is used. Timing varies from 4 minutes per iteration for the first 10 up to 13 minutes per iteration for the 31st through 40th.

e) A SOAP symbolic deck listing is included in addition to an absolute deck listing of the assembled program.

f) Alphabetic device if the SOAP symbolic version is used.

TRANSPORTATION PROBLEM

S. Poley
IBM, New York

May 17, 1956

a) Solves the transportation problem, i.e., given the requirements at m destinations, and amounts available at n origins, and the cost of shipment from any origin to any destination the program will determine the minimal mode of transportation of a homogeneous product.

b) All input data are restricted to a maximum size of five digits and all operations are in fixed-point. An approximation to the maximum number of destinations, m, and origins, n, is $5m+6n<2300$ with $n<100$.

c) Method is essentially the same as the iterative method proposed by A. Charnes and W. W. Cooper in "Management Science," October, 1954.

d) The entire drum is used. Time estimates not given.

e) Provision is made for alternate solutions which yield the same minimum total cost. A SOAP symbolic deck listing with a sample absolute deck listing is included.

f) Alphabetic device if the SOAP symbolic version is used.

LINEAR PROGRAMMING

J. W. Davis and D. H. Brown
Esso Standard Oil, Baton Rouge, Louisiana

March 29, 1956

a) Solves a linear programming problem.

b) Fixed decimal arithmetic of the form xxxxx.xxxxx is used. Up to 40 equations and any number of variables may be handled.

c) The modified simplex method is used.

d) The program is divided into four parts. Storage required is approximately 211, 37, 44, and 114 locations respectively. The parts occupy the same area of the drum and are read in only when needed. Timing information not given.

e) Information on alternate optima or near optima is supplied by the program.

f) Minimum 650.

LINEAR PROGRAMMING

R. L. Graves
Standard Oil, Indiana

a) Solves a minimizing linear programming problem.

b) A maximum of 33 equations in 1000 variables can be accommodated. All numbers are in floating-point form.

c) The dual and direct forms of the revised simplex method are used.

d) The entire drum is required. About 26 minutes are required for a 22 x 46 system.

e) A modified Trimble-Kubie interpretive system is used for the floating-point arithmetic, see Technical Newsletter No. 8.

f) Minimum 650.

LINEAR DECISION RULE FOR PRODUCTION AND EMPLOYMENT SCHEDULING

W. Folsom
C. C. Holt
Industrial Administration
Carnegie Institute of Technology
Pittsburgh, Pa.

a) Calculates optimal linear rules for making decisions on aggregate production and employment utilizing quadratic cost functions.

b) Floating decimal point.

c) The mathematical methods are described in papers appearing in "Management Science" Volume 2, No. 1 and 2, October 1955, January 1956.

d) The program requires the following decks:

(1) The Wolontis System* deck
(2) Complex Operations deck
(3) Arctan Relocated deck (decks 2 and 3 developed by Dr. P. Marcus, C. I. T.)
(4) The Linear Decision Rule Program deck

These programs are not relocatable.
All four decks are supplied in a single package.

e) Standard Wolontis* 533 and 402 boards are used.

f) Minimum 650.

* Bell Laboratories Interpretive System described in IBM Technical Newsletter No. 11.

NIM DEMONSTRATION

J. W. Robinson III
IBM Houston

a) NIM is a demonstration in which a player competes against the stored program in a version of an old game played with counters or markers.

b) The game is played with five "piles," each containing not more than nine counters.

c) The mathematical method is based on the standard binary solution for a nim game in which the object of play is to avoid taking the last counter.

d) Approximately one second or less is required for each pair of "moves" after manual entry. The program is in the form of a self-loading, self-transferring, condensed deck.

e) An attached card-input device need only be used for loading the program and may be turned off at the conclusion of loading.

f) Minimum 650.

THREE DIMENSIONAL TICK-TACK-TOE

H. F. Smith, Jr.
Watson Laboratory
New York 25, N. Y.

a) This program is a demonstration routine for the IBM 650; it permits a human opponent to compete with the 650 in a three-dimensional version of the children's game of tick-tack-toe, or crisscross. Plays are made by entering in the storage entry switches the coordinates of a cell in a cube of order 4 and depressing the program start key; the machine will reply and stop, awaiting the opponent's next play.

b) Does not apply.

c) Does not apply.

d) The program uses approximately 1700 storage locations.

e) None.

f) Minimum 650.

DEBUGGING PROGRAMS

A. M. Pietrasanta
IBM, New York

October, 1956

This paper describes a complete, automatic debugging procedure designed to provide the maximum amount of information about a malfunctioning program in the minimum amount of programmer and machine time. The following routines are used in the debugging procedure and complete information about them is given: Flow Tracer, Snapshot Tracer, Symbolic Seven-Per-Card Punch, all by S. Poley; Symbolic Tracing Routine by W. P. Heising and S. Poley; and Stop Codes by F. J. Chrinko.

The above routines, except the last one, are written in SOAP symbolic form, and are designed to be used by the SOAP programmer most effectively. The routines, however, can be used by the non-SOAP, or absolute, programmer, but a rudimentary knowledge of the SOAP system is necessary.

402 CONTROL PANEL FOR SOAP II

D. P. Baird
IBM, Schenectady

This paper describes the wiring, function, and application of a control panel for the IBM 402 Accounting Machine designed for listing the input and output of SOAP II. This 402 board will perform essentially the same functions as the 407 control panel described in the "SOAP II Programmers' Reference Manual" (Form No. 32-7646). The 402 printing format differs slightly from that of the 407.

FLOW DIAGRAMMING FOR THE IBM 650

B. Dimsdale
A. K. Charnow
I. M. Sobul
Service Bureau Corporation
Los Angeles, California

This paper describes a flow diagramming technique for the IBM 650. The method is an adaptation of the von Neumann-Goldstine system, and is designed primarily for mathematical and scientific problems.

ERRATA

for

THE LIBRARY PROGRAM FOR THE IBM 650

The following corrections, appendices and amendments have been received for the indicated programs. Revisions are made or have been made in material distributed but in some cases the incorrect version may have been sent out. This is the initial distribution of the series.

February 1957

1.3.003 PUNCH OUT FOR THE SOLUTION OF SIMULTANEOUS EQUATIONS

In the listing:

Card No. (c.c. 7-10)	c.c.	loc.	should read
0013	51-60	1816	45 1823 1871
0014	61-70	1823	16 1832 1824
0014	71-80	1824	46 1830 1849
0015	71-80	1830	15 8001 1820
0018	71-80	1849	21 1915 1871

1.3.004 INVERSE MATRIX PUNCH OUT

In the listing:

Card No. (c.c. 7-10)	c.c.	loc.	should read
0032	71-80	1918	45 1826 1873
0017	51-60	1826	16 1834 1846
0020	71-80	1846	46 1870 1876
0024	71-80	1870	15 8001 1872
0025	71-80	1876	21 1816 1873

"One to Seven Converter," by P. S. Herwitz

In the one-page listing appended to the detailed write-up for 1.6.009, instruction number 29 (location 0029) should read:

	65	0028	0030
instead of	65	0028	0039

This is a typographical error in the preparation of the listing; the program deck is not affected.

2.0.001 SIR, SOAP INTERPRETIVE ROUTINE

Since there are no stops in Main SIR for the eventuality that a computed number has three digit characteristic, Dr. Albert Newhouse of the University of Houston suggests the following changes in Main SIR:

Cards reading:

8	0063	SUP	0119	0025	MAIN SIR	064
8	0164	AUP	0119	0025	MAIN SIR	065
8	0025	SRT	F0002	0037	MAIN SIR	066

Should be replaced by:

8	0063	SUP	0119	0292	MAIN SIR	064
8	0064	AUP	0119	0782	MAIN SIR	065
8	0025	SRT	F0002	0682	MAIN SIR	066

And then add:

8	0682	NZU	0450	0037	MAIN SIR	195
8	0450	RAL	0122	0732	MAIN SIR	196
8	0732	ALO	0788	0242	MAIN SIR	197
8	0242	LDD	0119	F8002	MAIN SIR	198
8	0788	36	F0000	F1720	MAIN SIR	199
8	0292	BMI	0025	0450	MAIN SIR	200
8	0782	BMI	0450	0025	MAIN SIR	201

The write-up should then include under restrictions:

01 XXXX 2000 + : Characteristic out of range. Displaying the distributor will show the characteristic in the low order positions. XXXX indicates where in the program this stop occurred, viz. it is the I-address of the store command following the series of arithmetic steps which led to this characteristic.

"SIR: SOAP Interpretive Routine," By B. G. Oldfield and W. Hemmerle

The following error in the detailed write-up for SIR has been noted:

In the table shown on page 13, the "number of cards" for ARC TAN should read 65 instead of 66.

2.0.002 MITILAC

In the machine requirements: only minimum 650 required, alphabetic device not necessary.

3.1.014 LN-1 NATURAL LOGARITHM

Entry to subroutine:

1- Should read -- Place |x| in the accumulator with 9 positions to the right of the decimal point and <u>a maximum of 10 digits</u>.

5.1.001 MATRIX INVERSION

In the listing:

Card No. (c.c. 7-10)	c.c.	loc.	should read
0078	31-40	1717	45 1611 1524
0077	31-40	1711	15 8001 1722
0077	41-50	1712	21 1341 1524
0060	61-70	1611	16 1436 1612
0060	71-80	1612	46 1711 1712

5.1.002 SOLUTION OF SIMULTANEOUS LINEAR EQUATIONS

In the listing:

Card No. (c.c. 7-10)	c.c.	loc.	should read
0047	61-70	1501	16 1582 1601
0062	61-70	1601	46 1612 1610
0064	31-40	1610	21 1713 1569
0064	51-60	1612	15 8001 1568
0065	21-30	1615	45 1501 1569

"General Purpose System for the 650: L_2," by R. W. Hamming and Miss R. A. Weiss

An error has been discovered in certain copies of the L_2 program deck furnished to 650 users. In the main deck, column 18 of card 30 should contain a zero punch; in the incorrect copies, this column is blank.

It is recommended that all copies of this deck be examined and, if necessary, corrected. L_2 decks furnished by the 650 Program Library on or after March 3, 1958, have been corrected.

"FIASCO," By G. V. Maverick and S. Togasaki

FIASCO, which simulates the 650 with automatic floating decimal arithmetic, indexing registers, and immediate access storage, was programmed to conform with the specifications for these additional features as outlined in the publication, "IBM 650 MDDPM Additional Features: Indexing Accumulators, Floating-Decimal Arithmetic: Advanced Write-up" (Form No. 22-6258-0). Consequently, the discrepancies which exist between the "Advanced Write-up" and the augmented 650 as built, also exist between FIASCO and the augmented 650. Briefly, these discrepancies are:

1. In FIASCO, the lower accumulator is set to zero at the beginning of the execution of floating decimal operations. The 650 equipped with the floating decimal device does <u>not</u> automatically clear the lower at the beginning of floating decimal operations. (The lower should be set to zero by appropriate program steps before the floating decimal operations; failure to do so may cause erroneous results.)

2. On the 650 equipped with indexing registers, when a 50, 51, 52, 53, 58, 59, 80, 81, 82, 83, 88, or 89 operation has a data address of 8001, 8002, 8003, or 9000-9059, the contents of the location specified by the data address will be in the distributor following the operation. With FIASCO they will <u>not</u> be in the distributor.

"Internal Translator (IT), A Compiler for the 650," By A. J. Perlis, J. W. Smith, and H. R. Van Zoeren

In the SOAP listing of the Compiler,

Card No.	Should read:							
793		LDD	ML1A	TKNZ1	0940	69	1802	1094
799	MY1	LDD	ML1A	TKNZ2	1178	69	1802	1194
A793	ML1A	SLO	8002		1802	16	8002	1852
B793		STL	ABVAL	ML1	1852	20	0366	1291

The above changes are corrections to the compiler and do not represent misprints in the listing. The compiler as distributed would construct an incorrect translation of Relational statements where the right hand members were of the form:

$$\ldots \begin{matrix} U \\ V \\ W \end{matrix} \qquad A \quad V_1 \Omega \; V_2$$

where V_1 and V_2 are operands and Ω an operation.

The above four changes should be made in the seven per card deck which is in standard seven words per card form.

In the Description,

1) Page 1.41 : 2nd line after subroutine error listing should read:

"than 10^{-50} the computer......"

2) Page 1.44 : Program and remarks under b), (i) should read:

1: I2 ← I1

2: 6, I5, 0, 1, I1-1,

3: 6, I3, 0, 1, I1-1,

4: CN (I5, I3) ← 0

5: 6, I4, 0, 1, I1-1,

6: CN (I5, I3) ← CN (I5, I3) + YN (I2, I4) x YN (I4, I3)

7: H

The matrix B is represented as the matrix CN, whose (row) dimension must be specified. Statement 1 sets the correct dimension.

3) Page 1.45 : Program (II)

4th statement : I11 should read "II1"

5th statement : I12 should read "II2"

8th statement : I12 should read "II2"

4) Page 1.46 : Program (II)

1st statement should read:

"0 : C1 ← YI2"

"Internal Translator (IT), A Compiler for the 650," By A. J. Perlis, J. W. Smith, and H. R. Van Zoeren

In the SOAP listing of the Compiler:

Card No.	Should read:						
1. Correction:							
804	LDD	1971	GENN	1139	69	1971	1681
2. Correction:							
1442	STU		OPSGN	1334	21	0524	1902
3. Insertion:							
1442A	STD		V1	1902	24	0488	1384

The above changes are corrections to the compiler and do not represent misprints in the listing. Change 1 is necessary since the compiler, as distributed, would construct an incorrect translation of Relational statements where the right-hand members were of the form:

$$\ldots \begin{matrix} U \\ V \\ W \end{matrix} \quad V_1 \, \Omega \, V_2$$

where V_1 is a numerical constant. Changes 2 and 3 are necessary since the compiler, as distributed, would construct an incorrect translation of multiple parameter subroutines, at least two of whose parameters are expressions.

The above changes should be made in the seven per card deck which is in standard seven words per card form.

"Sin-Cos Subroutine," By G. R. Trimble

Page 4, Line 7, which reads:

703 87 8 0086 00 F0000 F0000 REC 12 FACTO 0186 00 0000 0000

Should read:

703 87 8 0086 00 F0000 F0002 REC 12 FACTO 0186 00 0000 0002

Existing card decks should be checked and these corrections made, if necessary.

"Square Root | x | ," By D. W. Sweeney

The entry point for this routine as indicated in the detailed write-up is incorrect. On page 1 of the write-up, the last paragraph should read:

Basic Linkage: Put x in upper accumulator, then

aaaa : RAA (80) bbbb 0020
bbbb : Next Instruction

The result is in the upper accumulator.

"Floating Point Sine A and Cosine A," by G. A. Porter

The following error has been noted in the program:

The instruction in location 0133 should read:

	RAA	F8003	0141
instead of	RAA	F8002	0141

This instruction appears on line 19 of the program listing on page 4 of the detailed write-up and is contained in card 19 of the relocatable deck. Copies of the write-up and deck furnished by the 650 Program Library on or after February 11, 1958, have been corrected.

"Laplace Transformation," by J. A. Painter

The following supplement to the program write-up has been submitted:

This program solves the algebraic equation entered on data card #1 prior to returning control to the console to read the second data card. Therefore, it has been found useful at times to replace the second data card with a self-loading program to read out or operate upon the coefficients without performing the transformation.

In addition, this program is capable of extracting roots of equations of the degree M, where $6 < M \leq 25$, when the degree and coefficients are properly loaded. To accomplish this, punch 0000XX0000 where XX is the degree of the equation, into a standard one-per-card load format to load at 1901. The coefficients are then punched one-per-card to load at 1902, 1903.... The transfer card is replaced by these single "instruction" load cards with a new transfer to 1048 following.

In either event, the roots are stored at 1851, 1852,... as complex numbers.

Restriction: This program will not solve an equation with a numerator of 1.

NOTE: Unless the special procedure for extracting roots of equations (described above) is being used, the last card of the load deck should transfer to 1000 rather than to 1048, i.e., the first word of the final card of the load deck should be punched $\overset{+}{0}00000100\overset{+}{0}$ instead of $\overset{+}{0}00000104\overset{+}{8}$.

"Small Scale Matrix Multiplication," By L. W. Ayres

PAGE 2: Under A. of CONTROL CARD FORM, columns 1 - 10 which read:

65 1952 1921

should read:

65 1953 1921.

PAGE 9: Under CONSTANTS USED, locations 1906 and 1916 which read:

1906	00	0000	0048
1916	00	0000	0049

should read:

1906	00	0000	0049
1916	00	0000	0050

PAGE 10: Locations 1711 and 1733 which read:

1711	22	1868	1943
1733	44	1937	1949

should read:

1711	22	1868	1729
1733	44	1743	1744

Existing card decks should be checked and these corrections made, if necessary.

5.2.001 MATRIX INVERSION

In the listing:

Page 1,	location 1770	should read	15 1959 1804
2,	1962 remarks n+b		n+b+1
3,	1972		24 1990 1946
4,	1808		1807
4,	1816		35 0004 1877

5.2.004 DOUBLE PRECISION MATRIX INVERSION

In the listing

Page 7,	location 0068	should read	44 0121 0172

5.2.005 COMPLEX AND REAL EIGENVALUES ON THE IBM 650

In the listing:

Page 1,	location 1513	should read	44 1514 1518+
1,	1528		24 1692 1529+
4,	1591		45 1592 1598+

5.2.006 SMALL SCALE MATRIX MULTIPLICATION

In the listing:

Page 5,	location 1699	should read	19 1953 1922

"Matrix Inversion," By H. L. Norman

The following error has been noted in 650 Library Program 5.2.008.

The instruction in location 1868 should read:

	35 0002 1626
instead of	35 0002 1616.

This instruction appears on page 5 of the listing included with the detailed write-up of the program. In the condensed (seven-word per card) load deck, it is the seventh word in card #37.

"Matrix Inversion," By H. L. Norman

PAGE 1: Location 1646 which reads:

15 1641 1695

should read:

15 1641 1702.

"Double Precision Matrix Inversion," by J. D. Chappell

The following correction should be made in the detailed write-up:

On page 3, in the paragraph headed "Deck Description," the last sentence should read: "The deck consists of 106 cards serially numbered from 001 to 106."

The program deck is correct as distributed.

"Multiple Regression Analysis," By A. Cohen

PAGE 15: CARD PREPARATION - Omit cards. After the sentence, "The row indices are punched on standard L1 load cards (12-punch in column 1) for storing in locations 1700-1715," the following should be inserted:

"The 'omit cards' should be punched as follows:

1. 'Y' punches in columns 1, 10, 20, 30, 40, 50, 60, 70, and 80.

2.	Col. 1-10	00 1700 0007	(First omit card)
		00 1707 0007	(Second omit card)
		00 1714 0002	(Third omit card)
3.	Col. 11-20	00 0000 00XX	(XX is the number of the first omitted variable.)
	Col. 21-30	00 0000 00XX	(XX is the number of the second omitted variable.)
	.	.	
	.	.	
	.	.	
	Col. 71-80	00 0000 00XX	(XX is the number of the seventh omitted variable.)

As many as three omit cards may be needed to omit 16 variables. The number in column ten is the number of variables omitted on that particular omit card. In the case of the second omit card, XX in columns 19 and 20 represent the eighth variable omitted, XX in columns 29 and 30 represent the ninth variable omitted, etc."

"Correlation Coefficient Routine," By J. W. Robinson III

Listed below are the corrections necessary to make the program conform to the specifications given in the write-up. The program deck consists of 125 cards numbered sequentially in columns 8-10; the card numbers below refer to the identification in those columns.

			Y		Y
Card #29	Col. 21-30	should be changed from	69 1430 1383	to	69 1358 1383
	41-50		00 0000 1358		24 1486 1742
62	21-30		19 8001 1329		19 8001 1484
66	11-20		64 1708 1758		30 0001 1758
	21-30		30 0001 1490		64 1708 1490
67	51-60		69 1528 1831		69 1456 1831
68	21-30		00 0000 1456		69 1704 1430
107	41-50		69 1776 1484		69 1776 1318
	51-60		24 1421 1374		44 1329 1536
109	51-60		69 1552 1546		69 1552 1318
110	11-20		24 1421 1374		45 1742 1486
112	21-30		69 1541 1846		69 1347 1846
	31-40		24 1486 1591		24 1877 1591
	41-50		69 1347 1536		69 1541 1748
	51-60		24 1877 1586		20 1708 1747
116	51-60		24 1400 1742		24 1400 1648
117	11-20		69 1747 1692		69 1850 1692
	31-40		00 0000 1850		65 1844 1712
119	51-60		69 1704 1598		69 1698 1598
120	11-20		24 1486 1648		24 1421 1586
	21-30		69 1698 1748		69 1528 1430
	31-40		24 1421 1586		24 1486 1586
121	31-40		15 1596 1742		11 1796 1546

Mr. J. W. Hamblen, Director of the Oklahoma A and M College Computing Center, has pointed out that the formula used to compute s_k is

$$\sqrt{\left[n \sum x_k^2 - (\sum x_k)^2\right]/n^2}$$

rather than as stated in the write-up. (The two formulas are equivalent; the one given by Mr. Hamblen indicates the method which is used in computing s_k.)

"Polynomial of Best Fit by Least Squares Method," by M. A. Kelly and M. S. Dyrkacz

The following error has been noted in the program deck:

In part 1 of the deck, card 001 should have a 12-punch in column 1 in addition to the 7-punch.

Copies of the program deck furnished by the 650 Program Library on or after March 3, 1958, have been corrected.

"Unbiased Standard Error of the Regression Coefficients," by W. T. Rooker, Jr.

The following errors have been noted in the program deck:

1. Card 106, which contains only a 12-punch in column 1, should be removed entirely.

2. In card 105, columns 41-50 should be punched

$$24\ 0412\ 199\overset{+}{9}$$

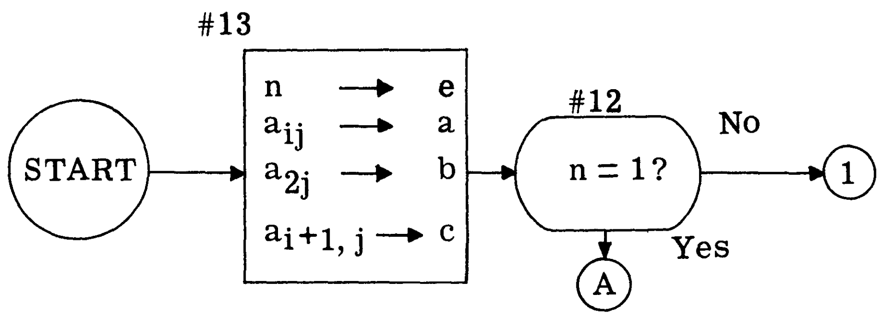
instead of $$24\ 0412\ 800\overset{+}{0}$$

Copies of the program deck furnished by the 650 Program Library on or after February 17, 1958, have been corrected.

"Weighted Least-Square Polynomial Approximation to a Continuous Function of a Single Variable," By R. E. von Holdt and R. J. Brousseau.

The following revisions are to be made:

PAGE 29: Box #12 of the flow diagram should be located following Box #13.



PAGE 34: The following sentence is omitted from the top of the page:

Multiply row (2) by B_2^2 and subtract from row (3).

New row (3) = $\left[0.02 \quad -0.01 \quad -0.02 \quad -0.01 \quad 0.02\right]$

PAGES 41, 42:

The following instructions should read:

Instr. No.	Loc. Instr.	Operation		Data Addr.	Instr. Addr.
7.23	(0208)	SRT	30	0006	0270
11.01	0380	BRNZ	45	0285	(0364)
12.02	0411	BRNZ	45	(0479)	0436
13.24	0473	STIA	23	0320	(0186)

PAGE 42: The following instructions are omitted from bottom of the page.

25.03	0483	RAL	65	0441	0445
25.04	0445	AL	15	0431	0485
25.05	0485	LD	69	0317	0439

"Polly: Polynomial Fit by Least Squares," By R. R. Haefner

The following errors have been noted in 650 Library Program 6.0.010 "Polynomial Fit by Least Squares."

The instruction in location 1532, page 13, should be 12 1315 0000-
1505, page 19, should be 24 1378 1508+
1435, page 20, should be 03 0000 0000+
0609, page 24, should be 07 0000 0000-
0616, page 24, should be 07 0000 0000-

The printed format from output cards can be improved by loading 00 1570 0001+ into location 1427 in the initial read-in.

The program in the Appendix of 6.0.010 was given in the logical sequence of computer interpretation. Since the subroutine that transfers the matrix elements to the inversion locations is used differently for elements below the largest diagonal than for elements above that diagonal, both sequences of the subroutine are given in the Appendix. Thus, if one prepared one-field load cards from the program given in 6.0.010 those cards which apply to the second use of the subroutine would be loaded last and would be executed for the first use of the subroutine, giving erroneous results. To prevent this error, one should insure that the following orders are read-in:

Location	Instruction
1344	24 0710 1342 +
1367	44 1347 1385 +
1374	21 1341 1376 +
1377	10 1183 1378 +
1378	21 1343 1340 +

The routine described in 6.0.010 permits the punching of the root-mean-square error, σ, only when $\sigma < K$. The routine was recently revised to punch σ for each order of the polynomial. The following orders are necessary for this revision:

Location	Instruction
1479	11 0057 1567 −
1523	24 1400 1521 +
1525	24 1427 0026 +
1529	21 1461 1487 +
1566	71 1427 1519 +
1567	69 1568 1569 +
1568	71 1427 7000 +
1569	24 1566 1487 +

"Chi Square for up to 10 x 10 Contingency Table," By Albert Newhouse

The following errors have been noted in Library Program 6.0.015:

On page 2 of the detailed write-up, under OUTPUT, Word 2 should read,

"χ^2_{ij} having 10 digits with decimal point after the sixth digit."

In the SOAP deck, card #504, columns 31-32 should be punched 20 instead of 16, and columns 48-50 should be punched STL instead of SLO.

In the condensed (five per card) deck, card #108, columns 21-22 should be punched 20 instead of 16.

7.0.001 REAL AND COMPLEX ROOTS OF ALGEBRAIC EQUATIONS ON THE IBM 650

In the listing:

Page 5,	location 0598	should read	66 0944 0599+
7,	0883		09 0057 0943−

$E = A_i X^i$ - If "i" is even, the leading coefficient must be equal to one.

"ValPROD," by C. M. White

The program write-up for ValPROD has been amended by the inclusion of two memoranda supplied by the original contributors. The first of these, dated June 18, 1957, deals with a revision of the program designated ValPROD II; the other, dated January 15, 1958, discusses in detail several coding errors contained in ValPROD I and ValPROD II. Program decks for the revised programs are designated ValPROD IB and ValPROD IIB.

AEC contractors and other 650 users concerned with nuclear reactor problems may obtain the amended program material in the usual manner.

9.2.001 APPENDIX H847 - SURVEY TRAVERSE - 533 PANEL

Wire from first read:

Col. 1 to load hub;
Col. 10 to XPU Pilot Selector #1;
Col. 14 to Position 3 Alphabetic First Read W3;
Col. 21 to Position 1 Alphabetic First Read W3.

CAI to ALPHA IN W3
RSU
PSU to PSU and ALPHA OUT W3

Read Card C to Storage Entry C.

Cols.	2-4	High order positions word 1 Zeros emitted in low order word 1
	5	Low order position word 2
	14	3rd lowest position word 3
	21	1st lowest position word 3
	6-13	Low order positions word 4
	22-28	Low order positions word 5
	15-20	High order positions word 6 Zeros emitted in low order word 6
	29-34	High order positions word 7 with column 34 through read column split, the X from 34 into 1st position of word 7 with an emitted zero. Zeros emitted into positions 4-2 word 7.

Pilot Selector #1
Emit 9 - N
Emit 8 - T 1 position of selector
Word 8 Entry C 1st position - C

Read Hold to Pilot Selector / Hold

Word Size Emitter

Size one	-	W2, W8
Size seven	-	W5
Size eight	-	W4
Size ten	-	W1, W3, W6, W7

Storage exits C to Punch Card C:

Word 1:	Positions 10-5	Cols. 15-20
	4-2	Cols. 6-9
	1	Common Punch Col. Split #4
		0-9 Col. 10
		11-12 Punch Col. Split #5
Word 2:	Positions 10-6	Cols. 29-33
	5	0-9 hub of punch col. split #1
	4-2	Cols. 10-12
	1	Common hub of punch column split #2
Word 3:	Positions 5	Column 5
	3	Column 14
	1	Column 21
Word 4:	Positions 10-8	Column 2-4
	7-1	Columns 22-28
Word 5:	Positions 10	Column 1
	7-1	Columns 35-41
Word 6:	Positions 7-1	Columns 42-48
Word 7:	Positions 7-1	Columns 49-55
Word 8:	Positions 7-1	Columns 56-62
Word 9:	Positions 7-1	Columns 63-69
Word 10:	Positions 10-8	Columns 78-80
	7-1	Columns 70-76

Punch Col. Split:

11-12 of Position 1	11-12 of position 2
C of Position 1	Column 34
0-9 of Position 2	0-9 of Position 5
C of Position 5	Column 13

"Cut and Fill," By Blaylock and Kibbee

In the description of the program the following changes should be made:

PAGE 3: Under Programmed Stops:

01 0004 0000	should read	01 0003 0000
01 0005 0000		01 0004 0000
01 0006 8000		01 0005 8000

PAGE 4: The Restart Procedure should read:

"Use the last header card (if more than one is needed for the problem). Start at 1100 and set address stop to 0915. Manually enter into location 0827 the station number of the station following the header cards, i.e., the first station to be processed in the RESTART. Manually enter into location 0824 columns 30 - 37 of the previous station output card. However, this quantity must be shifted two positions before entering. Manually enter into location 0880 columns 38 - 45 of the previous station output card after shifting the quantity left two positions. Start program from 0915."

In Appendix I the following change should be made under Header Card:

". . . , and the compaction factor in columns 28 - 30."

should read

". . . , and the compaction factor in columns 18 - 20 and 28 - 30."

In Appendix II the following changes should be made:

under Storage Entry C:

4 low order positions word 9.

should read

4 low order positions Word 9. Omit if no benches are included in the road plan.

under Word Size Emitters:

.
.
.

" " " 4 to W9.

should read:

" " " 4 to W9 if benches are included in the road plan.

" " " 0 to W9 if no benches are included in the road plan.

" " " 0 to W10.

The following change need be made only if actual elevations are used.

Instruction 918 which reads:

0872 00 0000 0007

should read:

0872 00 0000 0005

9.2.003 SURVEY TRAVERSE WITH BALANCING

In the write-up: **Page 1, Line 6**

"each closure" should read "each course"

"Cut and Fill on the IBM Type 650 MDDPM," By J. M. Kibbee and J. W. Robinson, III.

Attention is called to program step number 79, (0355 45 0158 0371). The program can never go to cell 0371, (C$\bar{O}$MPØ), on Header Card Read In, if the header cards are numbered consecutively in card column number 8 as per key punch instructions at the bottom of Page 3 Part VIII. The program will read in and store the first header card and STOP after reading the second header card.

Solution Number 1

More than one header card may be used if the card number (card column 8) is the same on all header cards. Storage is available for 7 header cards or a total of 49 compaction factors. The header cards must be maintained in their proper order, even though they all carry the same card number in column number 8.

Solution Number 2

The following program change allows the Header Cards to be numbered consecutively in card column number 8:

	Location	OP	D-Address	I-Address
Change Inst. No. 79	0355	46	0371	0158

"Connector and Redundancy Programs for Indeterminate Truss Analysis," By I. Tung.

It has been determined that the original C103003 and Sweeney 9980 Programs malfunctioned on problems involving only one redundant member. The following corrections and inserts should be made in order to enable the programs to handle all cases:

Deck No.	Card No. (Col. 8-10)	Columns to be Changed	Present Contents	Corrected Contents
C103003	101	47-50	$142\overset{+}{3}$	$187\overset{+}{3}$
	105	17-20	$162\overset{+}{9}$	$192\overset{+}{9}$
	107	21-30	$000000000\overset{+}{0}$	$441423097\overset{+}{6}$
		31-40	$000000000\overset{+}{0}$	$030000192\overset{+}{6}$
		41-50	$000000000\overset{+}{0}$	$441629192\overset{+}{7}$
		51-60	$000000000\overset{+}{0}$	$651851128\overset{+}{0}$
		61-70	$092619601\overset{+}{9}$	$092618731\overset{+}{8}$
		71-80	$601960196\overset{+}{0}$	$581929192\overset{+}{7}$
	108*	41-50	$000000000\overset{+}{0}$	$200001185\overset{+}{8}$
		51-60	$000000000\overset{+}{0}$	$640001185\overset{+}{7}$
		73-80	$1960196\overset{+}{0}$	$1857185\overset{+}{6}$

* It is possible that duplicate card numbers 108 exist. If so, change the card which contains an x-punch in column 10 to read $11\bar{0}$ in columns 8-10. This should be the last card in the program deck, which is the transfer card.

Deck No.	Card No. (Col. 8-10	Columns to be Changed	Present Contents	Corrected Contents
	109 (Insert)	1-10		$\overset{+}{3}10300310\overset{+}{9}$
		11-20		$600001185\overset{+}{2}$
		21-30		$640000185\overset{+}{3}$
		31-40		$240001185\overset{+}{4}$
		41-50		$200000185\overset{+}{5}$
		51-60		$600001185\overset{+}{6}$
		61-70		$185118521\overset{+}{8}$
		71-80		$531854185\overset{+}{5}$
9980 (Sweeney)	012	67-70	$180\overset{+}{5}$	$176\overset{+}{3}$
	041	6-10	$4004\bar{1}$	$4104\bar{2}$
	041 (Insert)	1-10		$\overset{+}{9}98004004\overset{+}{1}$
		11-20		$241758199\overset{+}{8}$
		21-30		$000000000\overset{+}{0}$
		31-40		$000000000\overset{+}{0}$
		41-50		$158001180\overset{+}{5}$
		51-60		$010000000\overset{+}{2}$
		61-70		$451760800\overset{+}{0}$
		71-80		$161761176\overset{+}{2}$

9.6.001 AMENDMENT TO H846 WRITEUP

The Well Bore Deviation Program has been changed to allow the following:

(1) Starting of the survey with an initial value for the depth, N-S coordinate and E-W coordinate.

(2) Input of angle of inclination in degrees and hundredths with bearing in degrees.

(3) Input of accumulated measured depth instead of course length.

(4) Computation of Pipeline Profile surveys with the same program.

In order to start the survey with initial coordinates, these coordinates should be punched on a load card (Y in column 1), in the following manner: initial E-W coordinate in columns 1-10, initial N-S coordinate in columns 11-20, initial depth in columns 21-30. Initial W and S coordinates should be punched with X-punch in columns 20 and 30 respectively. E and N coordinates are indicated by a Y-punch in columns 20 and 30.

If the angle of inclination is in degrees and hundredths, column 5 of each card should have a 2 punch. The inclination is then punched in columns 31-34, zeros in columns 29-30. Minus angles are indicated by an X-punch in column 34.

Henceforth field 4, columns 22-28 must contain the accumulated depth to the station at which the inclination is measured.

In order to compute 2 Pipeline Profile survey, the first card of the survey should have an X-punch in column 10. Whis punch causes the machine to change the necessary instructions. A Well Bore Survey cannot be computed following a Pipeline Profile Survey without reloading the program deck.

The only change in the output is in word 10 and columns 78-80 of the output card. The three high order positions of word 10 contain the course length. These are punched in columns 78-80.

The last card punched for each survey contains the closure information. It has an X-punch in column 13.

407 Output Form

The 407 panel is wired to print the heading on each sheet of the output form.

The X-punch in column 13 of the last card of the output deck from the Type 650 causes the 407 to print closure and head the appropriate column of the form with closure distance and bearing. The heading on each sheet is an follows:

COLUMN	HEADING	NUMBER SIZE
1	CO #	3 digits
2	Well #	5 digits
3	STA	3 digits
4	MEA D (Measured Depth)	5 digits (feet)
5	CRS (Course Length)	3 digits (feet)
6	VER DEPTH (Computed Vertical Depth)	7 digits and decimal (feet to nearest hundredth)
7	CRS DEPTH (Computed Course Depth)	7 digits and decimal (feet to nearest hundredth)
8	INCL	6 digits (see above for form)
9	BEARING	Two alphabetical characters and up to 6 digits. (See above for form)
10	STA N-S	An alphabetic character and 7 digits plus decimal. (Non 5 deviations to nearest hundredth)
11	STA E-W	Alphabetic character and 7 digits plus decimal, (or a deviation to nearest hundredth of a foot)
12	NORTH (Total North Coordinate)	7 digits plus dicimal (hundredth of a foot)
13	SOUTH (Total South Coordinate)	7 digits plus decimal (hundredth of a foot)
14	EAST (Total East Coordinate)	7 digits plus decimal (hundredth of a foot)
15	WEST (Total West Coordinate)	7 digits plus decimal (hundredth of a foot)

APPENDIX H846 - WELL BORE

Wire from first read:

Col. 1 to load hub;
Col. 10 to XPU Pilot Selector #1;
Col. 14 to Position 3 Alphabetic First Read W3;
Col. 21 to Position 1 Alphabetic First Read W3.

CAI to ALPHA in W3
RSU
PSU to PSU and ALPHA OUT W3

Read Card C to Storage Entry C.

Cols.	2-4	High order positions word 1 Zeros emitted in low order word 1
	5	Low order position word 2
	14	3rd lowest position word 3
	21	1st lowest position word 3
	6-13	Low order positions word 4
	22-28	Low order positions word 5
	15-20	High order positions word 6 Zeros emitted in low order word 6
	29-34	High order positions word 7 with column 34 through read column split, the X from 34 into 1st position of word 7 with an emitted zero. Zeros emitted into positions 4-2 word 7.

Pilot Selector #1	Emit 9 - N Emit 8 - T Word 8 Entry C 1st position - C	1 position of selector

Read Hold to Pilot Selector/Hold.

Word Size Emitter

Size one	- W2, W8
Size Seven	- W5
Size Eight	- W4
Size Ten	- W1, W3, W6, W7.

Storage exits C to Punch Card C:

Word 1:	Positions 10-5	Columns 15-20
	4-2	Columns 6-9
	1	Common punch column split #4
		0-9 Column 10
		11-12 Punch Column Split #5
Word 2:	Positions 10-6	Columns 29-33
	5	0-9 hub of punch column split #1
	4-2	Columns 10-12
	1	Common hub of punch column split #2
Word 3:	Position 5	Column 5
	3	Column 14
	1	Column 21
Word 4:	Positions 10-8	Column 2-4
	7-1	Columns 22-28
Word 5:	Position 10	Column 1
	7-1	Column 35-41
Word 6:	Position 7-1	Column 42-48
Word 7:	Position 7-1	Column 49-55
Word 8:	Position 7-1	Column 56-62
Word 9:	Positions 7-1	Columns 63-69
Word 10:	Positions 10-8	Columns 78-80
	7-1	Columns 70-76

Punch Column Split

11-12 of Position 1	11-12 of Position 2
C of Position 1	Column 34
0-9 of Postion 2	0-9 of Position 5
C of Position 5	Column 13

"Transportation Problem," by S. Poley

It has been discovered that the copies of the program deck for Program III (Alternate Optima) of the Transportation Problem furnished by the 650 Program Library prior to February 28, 1958, contain several erroneous cards. The corrections are too numerous to list here; 650 users who expect to run this part of the program may obtain corrected copies of the deck from the library in the usual manner.

The program listing contained in the detailed write-up is correct as issued.

"Debugging Programs," By A. M. Pietrasanta

The following information pertains to programs contained in the appendices of the detailed write-up for 12.0.001.

With reference to Appendix B, "Snapshot Tracer," it should be noted that if the symbolic location S0001= XX50, the constant, 40 9999 S0024, which is loaded into S0020 (see card #87), must be changed to 90 9999 S0024.

In Appendix E, "Set Drum to Stop Codes," an error exists in the listing on page 2: Word 8 of the first card should read "16 0001 8003" instead of "16 0001 0003."

12.0.001 DEBUGGING PROGRAMS ON THE IBM 650

In Appendix E:

Page 2, word 8 should read 16 0001 8003