Reference Manual

Catalog of Programs for IBM Data Processing Systems

KWIC Index

April 1962

No. 1
CONTENTS

Introduction .................................................. 1

Keyword-in-Context (KWIC) Index .......................... 5

Program Abstracts, Section "A"

RAMAC 305 Data Processing System ...................... 35
650 Data Processing System ................................ 37
704 Data Processing System ................................ 41
705 Data Processing System ................................ 43
709 Data Processing System ................................ 47
1401 Data Processing System ............................... 49
1410 Data Processing System ................................ 55
1620 Data Processing System ............................... 59
1710 Data Processing System ................................ 63
7070 Data Processing System ............................... 65
7080 Data Processing System ................................ 69
7090 Data Processing System ................................ 71

Program Abstracts, Section "B"

RAMAC 305 Data Processing System ...................... 73
650 Data Processing System ................................ 75
704 Data Processing System ................................ 179
705 Data Processing System ................................ 229
709 Data Processing System ................................ 245
1401 Data Processing System ............................... 253
1620 Data Processing System ............................... 265
7070 Data Processing System ............................... 287
7090 Data Processing System ................................ 307
INTRODUCTION

This catalog has been published as a service to computer users. It con­
tains a keyword-in-context index and the abstracts of the computer programs
which may be ordered from the IBM Program Information Department,
formerly known as IBM Library Services.

This department distributes four types of programs. The "A" section of
the catalog contains Type I and II programs which are written, tested,
published and maintained by IBM. The "B" section consists of Type III
and IV programs. In the case of the Type III and IV programs, the Program
Information Department acts only as a publishing and distributing agency.
Checking and testing of these programs is done by the contributors, and
questions concerning them should be directed to the author.

How to Order Programs

"A" Section
From local IBM branch office

"B" Section
Order programs directly from:
Program Information Department
IBM Corporation
2 William Street
White Plains, New York - USA

World Trade Users order programs from the WTC Program
Library in their Area if this Library services their computer.
Otherwise programs may be ordered from the United States
Program Information Department.

IBM World Trade Program Libraries:

<table>
<thead>
<tr>
<th>Area</th>
<th>Librarian</th>
<th>Computers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Europe</td>
<td>Central European Program Library</td>
<td>1401 1410</td>
</tr>
<tr>
<td></td>
<td>162 Rue de Charenton</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Paris 12, France</td>
<td></td>
</tr>
<tr>
<td></td>
<td>H. C. Koehler</td>
<td>650 1620</td>
</tr>
<tr>
<td></td>
<td>IBM Deutschland</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Postfach 66</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Sindelfingen/Wuertt, Germany</td>
<td></td>
</tr>
<tr>
<td></td>
<td>A. H. Lugtenburg</td>
<td>7070</td>
</tr>
<tr>
<td></td>
<td>IBM Deutschland</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Postfach 66</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Sindelfingen/Wuertt, Germany</td>
<td></td>
</tr>
<tr>
<td>Canada</td>
<td>K. C. Avann</td>
<td>650 1401 1410 1620 7070</td>
</tr>
<tr>
<td></td>
<td>IBM Company, Limited</td>
<td></td>
</tr>
<tr>
<td></td>
<td>844 Don Mills Road</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Don Mills, Ontario, Canada</td>
<td></td>
</tr>
</tbody>
</table>
The catalog contains three main parts:

Keyword-in-context (KWIC) Index for locating program abstracts
Program abstracts, Section "A" (by system type)
Program abstracts, Section "B" (by system type)

Keyword Index

The keyword-in-context index lists available programs arranged alphabetically by the keywords in the program titles. There are as many entries for each program as there are keywords in its title. Nonsignificant words such as "a," "the," "and," "for," "at," etc. (see complete list below) are not treated as keywords.

To prepare this KWIC index, each title was shifted to the right, one keyword at a time. After this was done, the multiple entries for each title were sorted in alphabetic order by keyword and listed on the IBM 1403 Printer to produce the master copy.

The first three entries for the program are shown below:

#CARD SYSTEMS ERROR DETECTION AIDS A 1401--AT-017
IDS #CARD SYSTEMS ERROR DETECTION A A 1401--AT-017
#CARD SYSTEMS ERROR DETECTION AIDS A 1401--AT-017

Notice that the keyword for each entry is located near the center of the column and that some or all of the title may precede or follow — that is, wrap around — the keyword. The pound sign (#) indicates the first word in each title. Each line is concluded with a reference code which relates the entry to the corresponding program abstract in the abstract section of the catalog.
Using the KWIC Index

To locate a program, begin by thinking of the significant words describing the desired program. Then look in the index for the keyword entry. The reference code adjacent to the title will then direct you to the corresponding program abstract. The reference code is set up as follows:

<table>
<thead>
<tr>
<th>Section</th>
<th>System Type</th>
<th>Reference Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>XXXX</td>
<td>XXXXXXXXXX</td>
</tr>
<tr>
<td>A or B</td>
<td>The number of the IBM system for which the program is written</td>
<td>The IBM library code for filing and ordering a program.</td>
</tr>
</tbody>
</table>

To locate the required abstract, first turn to the "A" or "B" section. Then find the corresponding system type, then the reference number. The reference numbers are in numerical sequence within system. The "A" or "B" designation and the machine type are printed on the top right-hand corner of the page to facilitate finding the abstract. The abstracts describe the programs in enough detail to help you determine whether they meet your requirements.

Words Prevented from Indexing

These words will never appear as keywords

- A
- ADD
- ADDS
- AN
- AND
- ANY
- AS
- AT
- ARITH.
- BY
- DECK
- FOR
- FROM
- GENERAL
- GENERATOR
- IBM
- IF
- IN
- INTO
- KIND
- MODIFIED
- NO
- NO.
- NUMBER
- OF
- ON
- ONLY
- OR
- OUT
- PACKAGE
- POINT
- PROBLEM
- PROG
- PROGRAM
- PROGRAMS
- PT
- PT.
- ROUTINE
- ROUTINES
- SOLUTION
DER ELF LOADING E o E 650 SED CHECKING.

LOADER. NABSOLUTE BINARY CARO AHO CORRECTION CAR WITH FL.PT .. OFL.

FLRELOCATABLE OCTAL-COLUMN I3INARY TO COLUMN BINARY CONVERTER no9 S ROW BINARY TO COLUMN BINARY CONVERSION..

FOURIER SERIES ONTO BINARY RElOCATABLE CARDS.

DIMENSIONAL ARRAY BINARY INFO ON RANGE.

NRELOCATING BINARY LOADER. TAPE FOR NBCD TO BINARY CONVERSION OF UNRESTRIC

NDAYS BETWEEN OATES B 0650-01.6.021

IBINARY ANO OCTAL LOAOER B

BE5SEl FUNCTIONS OF BESSEL

BEHAVIOR OF...

TABLE...

CENTROIDS OVER TAPE...

INTERGER...

TSF .. CARDS B 0704-10120RCBL

HOLLERITH,...

NORMALIlL B 0104-0370RS013

Vl/X

142 B

0 0704-0455BETtB

B

0709-095INA901

0104-0503ANI11

0704-0788IBRFS

0709-0935NGBSF

0104-0283MUBPU

0704-0525PKCSB

011... Utility Programs Outlined

ORTRAN.

100 MOt-'ENT OF

INOICES NPROGRAM TO CALCULATE SEASONALLY ADJUSTED B 0650-06.0.042

DISTRIBUTION AND

LOG INTERPRETATION IIPOROSITY CALCULATION FROM RADIOACTIVITY

• CALCOR •

THE

IIPUNCH ABSOLUTE (.OLUMN BINARY..

*ABSOLUTE BINARY UPPER lOADER ONE CARO 0 0709-1102SE9DU

RELAXATION LOADER

WITH

*ABSOLUTE BINARY UPPER lOADER ONE CARO 0 0709-1102SE9DU
### Nuclear-Code Laboratory

**FOR PART. OR ELL LABRS OF CODE • ARMOUR REACTOR KINETICS lARK-Ie CODE NUCLEAR-**

**0650-0S.2.019 OPEN. "SORT, ALGEBRAIC, KEY INTERPRETIVE SAVES 08 • IINE**

**ION 'FLOATING POINT OPTIMIZED THE 'L1NEAR WORD KEYS, IWHOLE WORD KEYS ONLYI ISORT, THE PROG**

**RAM TO SORT INVERSION WITH TRANSPOSE ON PROP**

**'INVERSE LAPLACE TRANSFORM, SYSTEM. IPS. * II OR**

**LISTING TAPE INDEX VARIABLE PROGRAM A**

**B 0650-04.0.00S**

**B 0650-09.6.012**

**B 0650-10.3.008**

**B 0704-0491RWAV3**

**B 0704-0914NCKSP**

**B 7090-1149AS012**

**B 0104-1144NCOI4**

**B 7070--10-904**

**B 0704-1234AAWEG**

**B 7010-03.4.002**

**B 0704-0345ELSAV'**

**B 0704-0742RWLE3**

**B 0704-05700RSRT**

**B 1401-10.3.001**

**B 1401-10.3.002**

**B 1401-10.3.003**

**B 1620-10.3.002**

**B 1620-10.3.003**

**B 008**
MULTIKEY cards: Whole word keys only. 

A B R I S: FUNCTIONS OF ORDER N

A S B: M I N I M I Z A T I O N OF U N I T - A R M E D R A N D - R A R T I C L E S.

A S B: M I N I M I Z A T I O N OF U N I T - A R M E D R A N D - R A R T I C L E S.

A S B: M I N I M I Z A T I O N OF U N I T - A R M E D R A N D - R A R T I C L E S.

A S B: M I N I M I Z A T I O N OF U N I T - A R M E D R A N D - R A R T I C L E S.

A S B: M I N I M I Z A T I O N OF U N I T - A R M E D R A N D - R A R T I C L E S.

A S B: M I N I M I Z A T I O N OF U N I T - A R M E D R A N D - R A R T I C L E S.

A S B: M I N I M I Z A T I O N OF U N I T - A R M E D R A N D - R A R T I C L E S.

A S B: M I N I M I Z A T I O N OF U N I T - A R M E D R A N D - R A R T I C L E S.

A S B: M I N I M I Z A T I O N OF U N I T - A R M E D R A N D - R A R T I C L E S.

A S B: M I N I M I Z A T I O N OF U N I T - A R M E D R A N D - R A R T I C L E S.

A S B: M I N I M I Z A T I O N OF U N I T - A R M E D R A N D - R A R T I C L E S.
CULATIONS NUMERICAL POWER SYSTEM TRANSIENT STABILITY CALCULATIONS FOR DYNAMIC VARIOUS SYSTEMS.

#744 SEQUENTIAL MONOTONE TRAPEZOID SCHEMATICS.
#4140 VELOCITY AXIOMS METHODOLOGY.

#2644 INTERVAL MONOTONE TRAPEZOID SCHEMATICS.
#4140 VELOCITY AXIOMS METHODOLOGY.

#0244 INTERVAL MONOTONE TRAPEZOID SCHEMATICS.
#4140 VELOCITY AXIOMS METHODOLOGY.

#2644 INTERVAL MONOTONE TRAPEZOID SCHEMATICS.
#4140 VELOCITY AXIOMS METHODOLOGY.

#2644 INTERVAL MONOTONE TRAPEZOID SCHEMATICS.
#4140 VELOCITY AXIOMS METHODOLOGY.

#2644 INTERVAL MONOTONE TRAPEZOID SCHEMATICS.
#4140 VELOCITY AXIOMS METHODOLOGY.

#2644 INTERVAL MONOTONE TRAPEZOID SCHEMATICS.
#4140 VELOCITY AXIOMS METHODOLOGY.

#2644 INTERVAL MONOTONE TRAPEZOID SCHEMATICS.
#4140 VELOCITY AXIOMS METHODOLOGY.

#2644 INTERVAL MONOTONE TRAPEZOID SCHEMATICS.
#4140 VELOCITY AXIOMS METHODOLOGY.

#2644 INTERVAL MONOTONE TRAPEZOID SCHEMATICS.
#4140 VELOCITY AXIOMS METHODOLOGY.

#2644 INTERVAL MONOTONE TRAPEZOID SCHEMATICS.
#4140 VELOCITY AXIOMS METHODOLOGY.

#2644 INTERVAL MONOTONE TRAPEZOID SCHEMATICS.
#4140 VELOCITY AXIOMS METHODOLOGY.

#2644 INTERVAL MONOTONE TRAPEZOID SCHEMATICS.
#4140 VELOCITY AXIOMS METHODOLOGY.

#2644 INTERVAL MONOTONE TRAPEZOID SCHEMATICS.
#4140 VELOCITY AXIOMS METHODOLOGY.

#2644 INTERVAL MONOTONE TRAPEZOID SCHEMATICS.
#4140 VELOCITY AXIOMS METHODOLOGY.

#2644 INTERVAL MONOTONE TRAPEZOID SCHEMATICS.
#4140 VELOCITY AXIOMS METHODOLOGY.

#2644 INTERVAL MONOTONE TRAPEZOID SCHEMATICS.
#4140 VELOCITY AXIOMS METHODOLOGY.

#2644 INTERVAL MONOTONE TRAPEZOID SCHEMATICS.
#4140 VELOCITY AXIOMS METHODOLOGY.

#2644 INTERVAL MONOTONE TRAPEZOID SCHEMATICS.
#4140 VELOCITY AXIOMS METHODOLOGY.

#2644 INTERVAL MONOTONE TRAPEZOID SCHEMATICS.
#4140 VELOCITY AXIOMS METHODOLOGY.

#2644 INTERVAL MONOTONE TRAPEZOID SCHEMATICS.
#4140 VELOCITY AXIOMS METHODOLOGY.

#2644 INTERVAL MONOTONE TRAPEZOID SCHEMATICS.
#4140 VELOCITY AXIOMS METHODOLOGY.

#2644 INTERVAL MONOTONE TRAPEZOID SCHEMATICS.
#4140 VELOCITY AXIOMS METHODOLOGY.

#2644 INTERVAL MONOTONE TRAPEZOID SCHEMATICS.
#4140 VELOCITY AXIOMS METHODOLOGY.

#2644 INTERVAL MONOTONE TRAPEZOID SCHEMATICS.
#4140 VELOCITY AXIOMS METHODOLOGY.

#2644 INTERVAL MONOTONE TRAPEZOID SCHEMATICS.
#4140 VELOCITY AXIOMS METHODOLOGY.

#2644 INTERVAL MONOTONE TRAPEZOID SCHEMATICS.
#4140 VELOCITY AXIOMS METHODOLOGY.

#2644 INTERVAL MONOTONE TRAPEZOID SCHEMATICS.
#4140 VELOCITY AXIOMS METHODOLOGY.

#2644 INTERVAL MONOTONE TRAPEZOID SCHEMATICS.
#4140 VELOCITY AXIOMS METHODOLOGY.

#2644 INTERVAL MONOTONE TRAPEZOID SCHEMATICS.
#4140 VELOCITY AXIOMS METHODOLOGY.

#2644 INTERVAL MONOTONE TRAPEZOID SCHEMATICS.
#4140 VELOCITY AXIOMS METHODOLOGY.

#2644 INTERVAL MONOTONE TRAPEZOID SCHEMATICS.
#4140 VELOCITY AXIOMS METHODOLOGY.

#2644 INTERVAL MONOTONE TRAPEZOID SCHEMATICS.
#4140 VELOCITY AXIOMS METHODOLOGY.

#2644 INTERVAL MONOTONE TRAPEZOID SCHEMATICS.
#4140 VELOCITY AXIOMS METHODOLOGY.

#2644 INTERVAL MONOTONE TRAPEZOID SCHEMATICS.
#4140 VELOCITY AXIOMS METHODOLOGY.

#2644 INTERVAL MONOTONE TRAPEZOID SCHEMATICS.
#4140 VELOCITY AXIOMS METHODOLOGY.

#2644 INTERVAL MONOTONE TRAPEZOID SCHEMATICS.
#4140 VELOCITY AXIOMS METHODOLOGY.

#2644 INTERVAL MONOTONE TRAPEZOID SCHEMATICS.
#4140 VELOCITY AXIOMS METHODOLOGY.

#2644 INTERVAL MONOTONE TRAPEZOID SCHEMATICS.
#4140 VELOCITY AXIOMS METHODOLOGY.

#2644 INTERVAL MONOTONE TRAPEZOID SCHEMATICS.
#4140 VELOCITY AXIOMS METHODOLOGY.

#2644 INTERVAL MONOTONE TRAPEZOID SCHEMATICS.
#4140 VELOCITY AXIOMS METHODOLOGY.

#2644 INTERVAL MONOTONE TRAPEZOID SCHEMATICS.
#4140 VELOCITY AXIOMS METHODOLOGY.

#2644 INTERVAL MONOTONE TRAPEZOID SCHEMATICS.
#4140 VELOCITY AXIOMS METHODOLOGY.

#2644 INTERVAL MONOTONE TRAPEZOID SCHEMATICS.
#4140 VELOCITY AXIOMS METHODOLOGY.

#2644 INTERVAL MONOTONE TRAPEZOID SCHEMATICS.
#4140 VELOCITY AXIOMS METHODOLOGY.

#2644 INTERVAL MONOTONE TRAPEZOID SCHEMATICS.
#4140 VELOCITY AXIOMS METHODOLOGY.
A SURVEY TRAVERSE

A pattern- recognition program, process panel, post-tracing.

A program for finding the centroids over a two-dimensional space.

A program for finding the centroids over a three-dimensional space.

A method for testing the accuracy of a data analysis program.

A program for finding the centroids over a two-dimensional space.

A program for finding the centroids over a three-dimensional space.

A program for finding the centroids over a two-dimensional space.

A program for finding the centroids over a three-dimensional space.

A program for finding the centroids over a two-dimensional space.

A program for finding the centroids over a three-dimensional space.

A program for finding the centroids over a two-dimensional space.

A program for finding the centroids over a three-dimensional space.

A program for finding the centroids over a two-dimensional space.

A program for finding the centroids over a three-dimensional space.

A program for finding the centroids over a two-dimensional space.

A program for finding the centroids over a three-dimensional space.

A program for finding the centroids over a two-dimensional space.

A program for finding the centroids over a three-dimensional space.

A program for finding the centroids over a two-dimensional space.

A program for finding the centroids over a three-dimensional space.

A program for finding the centroids over a two-dimensional space.

A program for finding the centroids over a three-dimensional space.

A program for finding the centroids over a two-dimensional space.

A program for finding the centroids over a three-dimensional space.

A program for finding the centroids over a two-dimensional space.

A program for finding the centroids over a three-dimensional space.

A program for finding the centroids over a two-dimensional space.

A program for finding the centroids over a three-dimensional space.

A program for finding the centroids over a two-dimensional space.

A program for finding the centroids over a three-dimensional space.

A program for finding the centroids over a two-dimensional space.

A program for finding the centroids over a three-dimensional space.

A program for finding the centroids over a two-dimensional space.

A program for finding the centroids over a three-dimensional space.

A program for finding the centroids over a two-dimensional space.

A program for finding the centroids over a three-dimensional space.

A program for finding the centroids over a two-dimensional space.

A program for finding the centroids over a three-dimensional space.

A program for finding the centroids over a two-dimensional space.

A program for finding the centroids over a three-dimensional space.

A program for finding the centroids over a two-dimensional space.

A program for finding the centroids over a three-dimensional space.

A program for finding the centroids over a two-dimensional space.

A program for finding the centroids over a three-dimensional space.

A program for finding the centroids over a two-dimensional space.

A program for finding the centroids over a three-dimensional space.

A program for finding the centroids over a two-dimensional space.

A program for finding the centroids over a three-dimensional space.

A program for finding the centroids over a two-dimensional space.

A program for finding the centroids over a three-dimensional space.

A program for finding the centroids over a two-dimensional space.

A program for finding the centroids over a three-dimensional space.

A program for finding the centroids over a two-dimensional space.

A program for finding the centroids over a three-dimensional space.

A program for finding the centroids over a two-dimensional space.

A program for finding the centroids over a three-dimensional space.

A program for finding the centroids over a two-dimensional space.

A program for finding the centroids over a three-dimensional space.

A program for finding the centroids over a two-dimensional space.

A program for finding the centroids over a three-dimensional space.

A program for finding the centroids over a two-dimensional space.

A program for finding the centroids over a three-dimensional space.

A program for finding the centroids over a two-dimensional space.

A program for finding the centroids over a three-dimensional space.

A program for finding the centroids over a two-dimensional space.

A program for finding the centroids over a three-dimensional space.

A program for finding the centroids over a two-dimensional space.

A program for finding the centroids over a three-dimensional space.

A program for finding the centroids over a two-dimensional space.

A program for finding the centroids over a three-dimensional space.

A program for finding the centroids over a two-dimensional space.

A program for finding the centroids over a three-dimensional space.

A program for finding the centroids over a two-dimensional space.

A program for finding the centroids over a three-dimensional space.

A program for finding the centroids over a two-dimensional space.

A program for finding the centroids over a three-dimensional space.

A program for finding the centroids over a two-dimensional space.

A program for finding the centroids over a three-dimensional space.

A program for finding the centroids over a two-dimensional space.

A program for finding the centroids over a three-dimensional space.

A program for finding the centroids over a two-dimensional space.

A program for finding the centroids over a three-dimensional space.

A program for finding the centroids over a two-dimensional space.

A program for finding the centroids over a three-dimensional space.
Bessel Functions of Order Zero

---

Bessel Functions of Order Zero

---

**FORTRAN** subroutine for calculating Bessel functions of order zero. This subroutine calculates the first-order Bessel function J_0(x) for a given argument x.

**Usage:**
```
CALL JBES10(X, J_0, IER)
```

- **X**: The argument of the Bessel function.
- **J_0**: The value of the Bessel function J_0(x).
- **IER**: Error code.

**Description:**
This subroutine is designed to compute the Bessel function of the first kind, order zero, J_0(x), for a given argument x. It uses a series expansion method to approximate the function value. The accuracy of the calculation depends on the value of x and the desired precision.

**Examples:**
- **Example 1:**
  ```
  REAL X, J_0
  INTEGER IER, I
  REAL ZERO, PI
  CALL D000 (X, J_0, IER)
  CALL DBS10(X, J_0, IER)
  WRITE (6, 100) J_0
  100 FORMAT (1X, E15.10)
  ```
- **Example 2:**
  ```
  REAL X, J_0
  INTEGER IER, I
  REAL ZERO, PI
  CALL D000 (X, J_0, IER)
  CALL DBS10(X, J_0, IER)
  WRITE (6, 100) J_0
  100 FORMAT (1X, E15.10)
  ```

**Notes:**
- **Accuracy:** The subroutine provides high accuracy for a wide range of x values. However, for very large or very small x, the accuracy may degrade.
- **Performance:** The subroutine is efficient and suitable for general use in applications requiring Bessel functions of order zero.

---

**FORTRAN** subroutine for calculating Bessel functions of order zero. This subroutine calculates the first-order Bessel function J_0(x) for a given argument x.

**Usage:**
```
CALL JBES10(X, J_0, IER)
```

- **X**: The argument of the Bessel function.
- **J_0**: The value of the Bessel function J_0(x).
- **IER**: Error code.

**Description:**
This subroutine is designed to compute the Bessel function of the first kind, order zero, J_0(x), for a given argument x. It uses a series expansion method to approximate the function value. The accuracy of the calculation depends on the value of x and the desired precision.

**Examples:**
- **Example 1:**
  ```
  REAL X, J_0
  INTEGER IER, I
  REAL ZERO, PI
  CALL D000 (X, J_0, IER)
  CALL DBS10(X, J_0, IER)
  WRITE (6, 100) J_0
  100 FORMAT (1X, E15.10)
  ```
- **Example 2:**
  ```
  REAL X, J_0
  INTEGER IER, I
  REAL ZERO, PI
  CALL D000 (X, J_0, IER)
  CALL DBS10(X, J_0, IER)
  WRITE (6, 100) J_0
  100 FORMAT (1X, E15.10)
  ```

**Notes:**
- **Accuracy:** The subroutine provides high accuracy for a wide range of x values. However, for very large or very small x, the accuracy may degrade.
- **Performance:** The subroutine is efficient and suitable for general use in applications requiring Bessel functions of order zero.

---

**FORTRAN** subroutine for calculating Bessel functions of order zero. This subroutine calculates the first-order Bessel function J_0(x) for a given argument x.

**Usage:**
```
CALL JBES10(X, J_0, IER)
```

- **X**: The argument of the Bessel function.
- **J_0**: The value of the Bessel function J_0(x).
- **IER**: Error code.

**Description:**
This subroutine is designed to compute the Bessel function of the first kind, order zero, J_0(x), for a given argument x. It uses a series expansion method to approximate the function value. The accuracy of the calculation depends on the value of x and the desired precision.

**Examples:**
- **Example 1:**
  ```
  REAL X, J_0
  INTEGER IER, I
  REAL ZERO, PI
  CALL D000 (X, J_0, IER)
  CALL DBS10(X, J_0, IER)
  WRITE (6, 100) J_0
  100 FORMAT (1X, E15.10)
  ```
- **Example 2:**
  ```
  REAL X, J_0
  INTEGER IER, I
  REAL ZERO, PI
  CALL D000 (X, J_0, IER)
  CALL DBS10(X, J_0, IER)
  WRITE (6, 100) J_0
  100 FORMAT (1X, E15.10)
  ```

**Notes:**
- **Accuracy:** The subroutine provides high accuracy for a wide range of x values. However, for very large or very small x, the accuracy may degrade.
- **Performance:** The subroutine is efficient and suitable for general use in applications requiring Bessel functions of order zero.
IBM Application & Systems Programs Library Abstract  File Number 0305-AT-007

THREE TRACE PROGRAMS, STORED PROGRAM, PROCESS PANEL, POST TRACE
Abstract:
Purpose: One program traces the store process; the second allows the control panel to be traced by the RAMAC 305 independent of the store program.

IBM Application & Systems Programs Library Abstract  File Number 0305-LM-005

PROGRAMMED DIVISION
Abstract:
Purpose: This program presents two methods of division. They are division using a tape of reciprocals, and division by iterative techniques.
Restrictions: The method of reciprocals is feasible if there are not more than 17,000 divisors.

IBM Application & Systems Programs Library Abstract  File Number 0305-LM-006

FLOATING POINT SUBROUTINES FOR THE 305 RAMAC
Abstract:
Purpose: Six floating point subroutines have been developed: Three perform the arithmetic operations of (1) floating point add or subtract; (2) floating point multiply; and (3) floating point divide. Three routines provide for comparison of floating point numbers and conversion routines between fixed and floating point numbers.
Restrictions: The range of floating point numbers may extend from +1,000,000 to 10^20 or -1,000,000 to 10^-20. Three versions of each routine are available. One utilizes the general purpose process control panel and the other requires a special wired panel.
Storage Requirements: Three drum tracks.
Remark: All operations take approximately 1/2 to 1 second. The shorter times are gained by use of the special purpose panel.

IBM Application & Systems Programs Library Abstract  File Number 0305-MI-005

LINEAR PROGRAMMING ROUTINE
Abstract:
Purpose: The program allows the solution of linear programming problems.
Method: The simplex method is used.
Restrictions: The maximum array that can be operated upon is 81 x 91.
Storage Requirements: One disk.
Machine Requirements: Automatic division.
Additional Requirements: All arithmetic computations are performed by floating point subroutines. This may be entered in fixed or floating point format.

IBM Application & Systems Programs Library Abstract  File Number 305-LM-004

305 GENERAL PURPOSE BOARD TEST DECK
Abstract:
Purpose: This card deck is utilized to insure the proper wiring of a General Purpose Process Control Panel. Proper communications with the punch, printer, and typewriter are checked. The program prints out the results of program exit tests as they are accomplished.
Method: Not applicable
Restrictions, Range: Not applicable
Storage Requirements: No disk storage area is required.
Equipment Specifications: No optional features are required.
(Continued on next column)

A - 305

Additional Remarks: User should be aware of "Record Advance Overflow" modifications which must be made to General Purpose Process Control Panel before operating test deck. Program is written for use with the 370 Printer.

IBM Application & Systems Programs Library Abstract  File Number 0305-PI-001

A COMPUTER PACKAGE FOR THE IBM 305 RAMAC
Abstract:
Purpose: The computer package is an interpretive programming system for performing scientific and engineering computations on the RAMAC 305.
Restrictions: The package will handle either fixed or floating point numbers.
Fixed point numbers are carried as 10 digits. Floating point numbers are carried in a 2 and 8 notation.
Additional Remarks: The simulated instructions are of the 3 address variety. Each address may be defined by one of 9 pseudo index registers. The following functions are included:
- Square root
- Sine
- Cosine
- Logarithm
- Exponential
- Arctangent
- Arccos
Machine Requirements: Automatic division.
Storage Requirements: 60 disk tracks.

IBM Application & Systems Programs Library Abstract  File Number 300-SP-003

SYMBOLIC PROGRAMMING AND ASSEMBLY FOR THE IBM RAMAC 305
Abstract:
Purpose: This system provides the programmer with a symbolic programming language for the IBM RAMAC 305. In addition, an assembly program is provided for translating the symbolic language into the machine language of the RAMAC 305.
The language contains operations for handling normal programs and general purpose process control panel instructions. The output of the program is a deck of self-loading, one-instruction-per-card load cards, and a listing of the symbolic program steps and their translation.
Method: Not applicable.
Restrictions, Range: Not applicable.
Storage Requirements: The General Purpose Process Control Panel is required for operating the assembly program. Any control panel may be used for operating the assembled program. The assembly program requires 300 sectors of disk storage.
Equipment Specifications: The program requires no optional features.
Additional Remarks: The 300 sectors of disk storage referred to for operating the assembly program must be contained in the file containing addresses 000000 to 000999 on a RAMAC 305 which has six character RAMAC addresses. No op code which contains a disk storage address as an operand can be utilized with a six digit disk address.

IBM Application & Systems Programs Library Abstract  File Number 305-UY-008

305 UTILITY PROGRAMS
Abstract:
Purpose: The programs contained in this package may be classified as follows:
(1) programs which transfer data from punched cards to a specific location within the RAMAC;
(2) programs which transfer data from one location within the RAMAC to another (e.g., from processing drum to disk storage, and vice versa); and
(3) programs which transfer data from specific locations in disk storage to cards or printed output.
Method: Not applicable.
Restrictions, Range: Not applicable.
Storage Requirements: All of the programs operate from track 1.
Equipment Specifications: No optional features are required.
Additional Remarks: The programs which utilize disk storage will only operate on the file containing sectors 000000 - 000999 on an IBM RAMAC 305 which utilizes six digit disk addresses.

35
GENERAL TRACING ROUTINE

Abstract:

Purpose: This program has been designed to aid programmers in debugging programs written in SOAP II language for any 650 system.

Range: Does not apply.

Accuracy: Does not apply.

Floating/Fixed: Does not apply.

Mathematical Method: Does not apply.

Storage Required: The program is available in either regional or symbolic form. The symbolic program requires \(200 + 3N + 5M + K\) drum locations, where \(N\) is the number of points within the program to begin tracing, \(M\) the number of distinct loops to be traced, and \(K\) the number of stopping points. The regional program does not require the additional \(K\) locations, and is available for output synchronizers 1 or 2. A maximum of 45 stopping points is allowed in either program.

Speed: Not given.

Relocatability: Not given.

Remarks: The program will trace all 650 system instructions. There are two conditions which will cause an automatic skip-out: if a load card is read, or if an inquiry is made while in the tracing mode. When either of these occurs tracing ceases, and the program being traced will resume at high speed. The tracing program will be re-entered at the next encountered skip-in point. If the D-address of a branch-on-inquiry instruction is chosen as a skip-in point, the inquiry subroutine may be traced. The programmer, if he so desires, may trace index registers by including a control card.

Requests for program decks should specify which type is desired, i.e., symbolic or regional for output synchronizer 1, or regional for output synchronizer 2.

IBM Application \\n& Systems Programs Library Abstract File Number 650-FO-301

FORTRANSIT I

Abstract:

Purpose: Program converts source program written in FORTRAN language into machine language instructions. Three card passes are required.

Restrictions: The program processes the following statements: Arithmetic; GO TO \(m\); GO TO \(m\), ..., \(n\); IF; PAUSE; STOP; DO; CONTINUE; DIMENSION; READ; PUNCH; END.

Machine Requirements: 533 with alphabetic device. 533 required.

Special Devices: Alphabetic device for SOAP assembly.

FORTRANSIT II

Abstract:

Purpose: Program converts source program written in FORTRAN language into machine language instructions. Three card passes are required.

Restrictions: The program processes the following statements: Arithmetic; GO TO \(m\); GO TO \(m\), ..., \(n\); IF; PAUSE; STOP; DO; CONTINUE; DIMENSION; READ; PUNCH; END.

Machine Requirements: 533 with alphabetic device.

FORTRANSIT II S

Abstract:

Purpose: Program converts source program written in FORTRAN language into machine language instructions. Three card passes are required.

Restrictions: The program processes the following statements: Arithmetic; GO TO \(m\); GO TO \(m\), ..., \(n\); IF; PAUSE; STOP; DO; CONTINUE; DIMENSION; READ; PUNCH; END.

Machine Requirements: 533 with alphabetic device; three 727 tape drives; standard 407.

FORTRANSIT III

Abstract:

Purpose: Program converts source program written in FORTRAN language into machine language instructions. Three card passes are required.

Restrictions: The program processes the following statements: Arithmetic; GO TO \(m\); GO TO \(m\), ..., \(n\); IF; PAUSE; STOP; DO; CONTINUE; DIMENSION; READ; PUNCH; END.

Machine Requirements: 533 with alphabetic device; three 727 tape drives; standard 407.
FLOATING POINT SINE A AND COSINE A

Abstract:

Purpose: This subroutine computes the sine or cosine of the angle A expressed in radians.

Range: Accepts any argument where \(|A| \leq \pi / 2\).

Accuracy:

Range of Argument | Maximum error
--- | ---
\(|A| < \pi / 2\) | 3.5 in the 8th significant digit
\(1.2 \times 10^{-6} < |A| < 2\pi\) | \(2.7 \times 10^{-7}\)
\(2 \pi \leq |A| < 2\pi\) | \(5.4 \times 10^{-7}\)
\(2\pi \leq |A| < 2\pi - 10^{-8}\) | \(3.1\) in the 8th decimal place

Floating/Fixed: Uses floating point.

Mathematical Method: The Rand Approximation is used for \(x = x / \pi \leq 1/2\). The method of reduction and the solution originated with Mr. D. W. Sweeney.

Storage Required: The routine requires 55 storage locations between 0000 and 0068 inclusive. The 14 unused locations are available to the programmer.

Speed: The routine takes 121 ms. for Sine and 128 ms. for Cosine.

Relocatability: Relocatable SOAP II cards.

Remarks: Relocate only by an even amount.

Note: As the power of 10 increases, the number of significant digits in the result decreases. This is due to the limitation of significant digits available in the original Angle A.

533 System: One 533 and automatic floating decimal arithmetic.

SQUARE ROOT

Abstract:

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

b) Range: Any floating decimal argument, \(0 \leq \text{machine exponent} \leq 99\). 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: 66 locations. Relocatable. Execution time approximately 75 milliseconds.

e) The program is in relocatable SOAP II form.

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

FLOATING POINT ARC TANGENT

Abstract:

Purpose: This subroutine computes the arctangent of floating point numbers. The result is in radians.

Range: The routine accepts all arguments \(X\) where \(3.1622777 \times 10^{-26} < |X| < 3.1622777 \times 10^{26}\)

Accuracy: The absolute error is less than \(10^{-7}\).

Floating/Fixed: The routine is written utilizing automatic floating point arithmetic.

Mathematical Method: The method is based on the work of Dr. E. C. Hultstrom, IBM, WHQ, and utilizes a continued fraction form of the expansion of \(\pi / 2\) arctan \(X\) in the interval \((0, 1)\).

Storage Required: The routine requires 49 locations.

Speed: Execution time is 127 milliseconds.

Relocatability: Routine is written in relocatable SOAP II form.

Remarks: Relocate by an even amount. One indexing register is used; the contents are not restored.

650 System: One 513, automatic floating decimal arithmetic, and one indexing register are required.

Special Devices: For SOAP assembling, an alphabetic device is required.

FLOATING POINT EXPONENTIAL

Abstract:

Purpose: This routine computes \(10^X\) and \(e^X\) for floating point arguments using automatic floating decimal arithmetic and three indexing registers.

Range: The routine accepts arguments for \(10^X\) and \(e^X\).

Accuracy: The maximum error is 1 in the 8th significant digit for positive arguments and less than 1 in the 7th significant digit for negative arguments.

Floating/Fixed: Floating decimal arithmetic.

Mathematical Method: (Adapted for floating decimal arithmetic from register from W. E. Stuart’s ‘PRATS’ library program J. 1. 026) \(e^X\) is reduced to \(10^{0.43429448X}\), which is computed in fixed point using a Hastings polynomial approximation over the range \(0 \leq X \leq 1/10\) for negative exponents, \(e^X = 10^{0.43429448X}\).

Storage Required: Requires \#8 drum within a group of 100 locations. The unused locations are available to the programmer.

Speed: 120 ms. for \(10^X\), 127 ms. for \(e^X\).

Relocatability: Relocatable SOAP II form.

Remarks: Three indexing registers are used and are not restored in the original values.

650 System: One 513, automatic floating decimal arithmetic, and three indexing registers.

Special Devices: Alphabetic device for SOAP II assembly.
IBM Application & Systems Programs Library Abstract  File Number 650-ES-009

FORTRAN SQUARE ROOT SUBROUTINE

Abstract:

Purpose: This routine computes the square root of a single precision floating point argument A.

Accuracy: The subroutine exits to the main program when two successive approximations differ by $2 \times 10^{-8}$.

Floating/Fixed: The format of the floating point number is \( \pm xxxXXXXXXXm \), with floating zeros in the form \( 00 0000 0000 \).

Mathematical Method: Iteration of Newton's Function.

Storage Required: 21 Permanent drum locations including a programmed stop for negative argument for.

Special Devices: Single and automatic floating decimal arithmetic.

Remarks: Rectangular indexing device B which is not reset.

IBM 650 System: This routine requires a 650 with floating decimal arithmetic device and one index register. An alphabetic device is needed for SOAP II assembly.

IBM Application & Systems Programs Library Abstract  File Number 650-ES-010

FORTRAN SQUARE ROOT SUBROUTINE

Abstract:

Purpose: This routine computes the square root of a single precision floating point argument A.

Accuracy: The subroutine exits to the main program when two successive approximations differ by $2 \times 10^{-8}$.

Floating/Fixed: The format of the floating point number is \( \pm xxxXXXXXXXm \), with floating zeros in the form \( 00 0000 0000 \).

Mathematical Method: Iteration of Newton's Function.

Storage Required: 21 Permanent drum locations including a programmed stop for negative argument for.

Special Devices: Single and automatic floating decimal arithmetic.

Remarks: Rectangular indexing device B which is not reset.

IBM 650 System: This routine requires a 650 with floating decimal arithmetic device and one index register. An alphabetic device is needed for SOAP II assembly.

IBM Application & Systems Programs Library Abstract  File Number 650-ES-011

SORT II

Abstract:

Purpose: Sort II is a generalized tape sorting program.

Restrictions: Program sorts unblocked fixed-length records. Maximum record is 60 words, maximum of 5 control fields. File must be within 1 or 2 reels of tape.

Methods: 2-way merge.

Equipment Specifications: 4 T77 Magnetic Tape Units

Additional Remarks: Routines for tape labeling, error corrections, restart procedures, record count, and hash totals are included.

IBM Application & Systems Programs Library Abstract  File Number 0650-SP-201

BASIC SOAP 2A

Abstract:

Purpose: This program processes programs written in symbolic language and produces one-to-one machine language instructions.

Restrictions: A maximum of 300 labels are processed per pass of card deck. It assembles instructions for a 2K machine.

Machine Requirements: 533 with alphabetic device.

IBM Application & Systems Programs Library Abstract  File Number 0650-SP-202

TAPE SOAP 2A

Abstract:

Purpose: This program processes programs written in symbolic language and produces one-to-one machine language instructions.

Restrictions: A maximum of 300 labels are processed per pass. It assembles instructions for a 2K machine.

Machine Requirements: 533 with alphabetic device; two 727 tape drives.
SOAP SI

Abstract:

Purpose: This program processes programs written in symbolic language and produces one-for-one machine language instructions. SOAP SI will process LITERALS and three other pseudo-ops, not handled by SOAP II.

Restrictions: A maximum of 300 labels are processed per pass of card deck.

Machine Requirements: 533 with alphabetic device.

SOAP II

Abstract:

Purpose: This program processes programs written in symbolic language and produces one-for-one machine language instructions. SOAP II processes LITERALS and three other pseudo-ops, not handled by SOAP II.

Restrictions: A maximum of 300 labels are processed per pass of card deck.

Machine Requirements: 533 with alphabetic device.

TAPE SOAP SI

Abstract:

Purpose: This program processes programs written in symbolic language and produces one-for-one machine language instructions. SOAP SI processes LITERALS and three other pseudo-ops, not handled by SOAP II.

Restrictions: A maximum of 300 labels are processed per pass of card deck.

Machine Requirements: 533 with alphabetic device.

SOAP II A - 4000

Abstract:

Purpose: This program processes programs written in symbolic language and produces one-for-one machine language instructions.

Restrictions: A maximum of 1200 labels are processed per pass of card deck. It assembles instructions for a 4K machine.

Machine Requirements: 533 with alphabetic device. 4K drum.

SOAP 2L

Abstract:

Purpose: This program processes programs written in symbolic language and produces one-for-one machine language instructions.

Restrictions: A maximum of 300 labels are processed per pass of card deck. It assembles instructions for a 2K machine.

Machine Requirements: 533 with alphabetic device.

CARD-TO-TAPE ROUTINE

Abstract:

Purpose: This utility routine for the 650 tape system is designed to convert card records to tape records.

Range: Numerical or alphanumerical records contained in from one to fifteen cards can be converted to tape records of from one to sixty words.

Accuracy: Does not apply.

Floating/Fixed: Does not apply.

Mathematical Method: Does not apply.

Storage Required: The program and its five-per-card loading routine use 177 card locations including the 1951 read band.

Speed: When tape writing is in the alphanumerical mode, operating speed is approximately 200 cards per minute if not more than six words are taken from each card. If writing is in the numeric mode, the same speed will be maintained regardless of the number of words taken from each card.

Relocatability: Not in relocatable form.

Remarks: None.

650 System: One 727 tape unit and any card input device.

Special Devices: None.

TAPE-TO-PRINTER/PUNCH ROUTINE

Abstract:

Purpose: This utility routine is designed to punch or print records from a reel of magnetic tape. Output is eight words per card or per line.

Range: Numerical or alphanumerical records of any length can be processed.

Accuracy: Does not apply.

Floating/Fixed: Does not apply.

Mathematical Method: Does not apply.

Storage Required: The routine requires 50 locations plus the read and punch areas of the 1950 band. (If indexing registers are not used, 56 locations are needed.)

Speed: Operates at maximum punch or print rates.

Relocatability: Written in SOAP II regionalized form.

Remarks: The program consists of two versions: one for a system with indexing registers and one for a system without that feature. Requests for card decks should specify which version is desired.

650 System: One 727 tape unit and any card input device.

Special Devices: None.
IBM Application & Systems Programs Library Abstract  File Number 0704-F0-008

4K 704 FORTRAN PROGRAMMING SYSTEM

Abstract:

Purpose: The IBM Formula Translating System, 4K 704 FORTRAN, is an automatic coding system for the IBM 704 Data Processing System. More precisely, it is a 704 program which accepts a source program written in the FORTRAN language, closely resembling the ordinary language of mathematics, and which produces a machine-language object program ready to be run on a 704.

IBM Application & Systems Programs Library Abstract  File Number 0704-F0-042

8K 704 FORTRAN PROGRAMMING SYSTEM

Abstract:

Purpose: The IBM Formula Translating System, 8K 704 FORTRAN, is an automatic coding system for the IBM 704 Data Processing System. More precisely, it is a 704 program which accepts a source program written in the FORTRAN language, closely resembling the ordinary language of mathematics, and which produces a machine-language object program ready to be run on a 704.

IBM Application & Systems Programs Library Abstract  File Number 0704-F0-049

32K 704 FORTRAN PROGRAMMING SYSTEM

Abstract:

Purpose: The IBM Formula Translating System, 32K 704 FORTRAN, is an automatic coding system for the IBM 704 Data Processing System. More precisely, it is a 704 program which accepts a source program written in the FORTRAN language, closely resembling the ordinary language of mathematics, and which produces a machine-language object program ready to be run on a 704.

IBM Application & Systems Programs Library Abstract  File Number 0704-F0-041

Simulation of the 1410 with the 704/709/704P

Abstract:

Purpose: The program enables the user to test and correct 1410 programs prior to installation of an IBM 1410 data processing system. The system will trace or dump simulated programs.

Restrictions: The program simulates standard card and tape systems. The simulated 1410 has 20,000 core storage positions. Using Basic Autocodes the simulator will assemble 1410 programs. A maximum of one disk of 1405 storage can be simulated.

Timing: The 704 takes approximately 30 times longer than if the program was running on a 1410.

Equipment Specifications:

32,876 words of core storage
4 tape units + 1 for simulated 1410 tape units + 2 for disk

Additional Remarks: This program is distributed on a systems tape.

Simulation of the 1410 with the 704/709/704P

Abstract:

Purpose: The program enables the user to test and correct 1410 programs prior to installation of an IBM 1410 data processing system. The system will trace or dump simulated programs.

Restrictions: The program simulates standard card and tape systems. The simulated 1410 has 20,000 core storage positions. Using Basic Autocodes the simulator will assemble 1410 programs. A maximum of one disk of 1405 storage can be simulated.

Timing: The 704 takes approximately 30 times longer than if the program was running on a 1410.

Equipment Specifications:

32,876 words of core storage
4 tape units + 1 for simulated 1410 tape units + 2 for disk

Additional Remarks: This program is distributed on a card deck.

Simulation of the 1410 with the 704/709/704P

Abstract:

Purpose: The program enables the user to test and correct 1410 programs prior to installation of an IBM 1410 data processing system. The system will trace or dump simulated programs.

Restrictions: The program simulates standard card and tape systems. The simulated 1410 has 20,000 core storage positions. Using Basic Autocodes the simulator will assemble 1410 programs. A maximum of one disk of 1405 storage can be simulated.

Timing: The 704 takes approximately 30 times longer than if the program was running on a 1410.

Equipment Specifications:

32,876 words of core storage
4 tape units + 1 for simulated 1410 tape units + 2 for disk

Additional Remarks: This program is distributed on a systems tape.

Simulation of the 1410 with the 704/709/704P

Abstract:

Purpose: The program enables the user to test and correct 1410 programs prior to installation of an IBM 1410 data processing system. The system will trace or dump simulated programs.

Restrictions: The program simulates standard card and tape systems. The simulated 1410 has 20,000 core storage positions. Using Basic Autocodes the simulator will assemble 1410 programs. A maximum of one disk of 1405 storage can be simulated.

Timing: The 704 takes approximately 30 times longer than if the program was running on a 1410.

Equipment Specifications:

32,876 words of core storage
4 tape units + 1 for simulated 1410 tape units + 2 for disk

Additional Remarks: This program is distributed on a systems tape.
IBM Application & Systems Programs Library Abstract  File Number 0705-AY-007

APTS 80

Abstract:

Purpose: An automatic program testing system for the IBM 705 III, consisting of a coordinated set of the "705 Better" utility programs that are used in testing, modified so that the utility programs themselves may be loaded automatically from a utility tape, and their control cards from the card reader or other input device independent of the utility tape. With APTS 80, all programs being tested may be loaded from a single tape, and test data cards and program correction cards may be read from the card reader.

IBM Application & Systems Programs Library Abstract  File Number 0705-CV-016

705-1401 A ASSEMBLY PROGRAM

Abstract:

Purpose: To assemble, on the 705, programs written in 1401 symbolic language; to produce as the end result of the assembly a listing and program cards in 1401 machine language.

Machine Requirements: The 705-1401A Assembly Program will run on a Model 1, M, II, III, IIIA, IIIIB.

Magnetic Tape Drives Required: Three (3) if card reader input. Three (3) if tape input-single assembly. Four (4) if tape input-multiple assemblies.

IBM Application & Systems Programs Library Abstract  File Number 0705-ID-001

705 III EDCS

Abstract:

Purpose: EDCS handles reading and writing, checkpoint and restart, error correction, beginning and end-of-file and beginning and end-of-file processing, tape record blocking and de-blocking, and label checking. Macro-instructions and control parameters coded by the programmer cause generation of linkages to EDCS subroutines, which in turn perform the specified functions.

An input/output memory resident system (IBM 705 BSR) operates in conjunction with EDCS to restore program status from periodically recorded checkpoints, so that in the event of program interruption, previous processing need not be repeated.

Storage Requirements: Preassambled EDCS occupies 17,074 locations.

Equipment Specification: 705 Model III

IBM Application & Systems Programs Library Abstract  File Number 0705-10-001

LST 75

Abstract:

Purpose: This program, using program cards as input, produces a sorted listing of a program's instructions by storage location, storage unit, mnemonic, operation code, and address. This output is helpful in analyzing a program for transfer points, modified instructions, instructions that set or reset switches, etc.

Equipment Specification: 705 Model I or Model II

IBM Application & Systems Programs Library Abstract  File Number 0705-MI-006

LST 97

Abstract:

Purpose: This program, using program cards as input, produces a sorted listing of a program's instructions by storage location, storage unit, mnemonic, operation code, and address. This output is helpful in analyzing a program for transfer points, modified instructions, instructions that set or reset switches,etc.

Equipment Specification: 705 Model I or Model II

IBM Application & Systems Programs Library Abstract  File Number 0705-MI-130

IBM Application & Systems Programs Library Abstract  File Number 0705-PB-364

7056 PROCESSOR

Abstract:

Purpose: The 7056 Processor accepts six programming languages: Autocoder III, |Commercial Translator|, |Commercial Translator|, |Table-Creating|, and |FORTRAN|. It will operate with any input/output device, on a 705, 705 II, or 7050 and assemble programs for any model 706 or a 7060.

7056 Processor language, described below, permits a wide variety of programming to be stated in terms of the data processing results desired, rather than the machine operations required to accomplish it. Extensive use of these languages will greatly reduce coding effort and incidence of clerical and logical errors, and will simplify problems of debugging and program maintenance. A statement in any of the languages may cause generation of an entire program that will efficiently perform the data processing defined by the statement. Within any one program, routines in the various Processor languages may be intermixed.

Autocoder III: This advanced programming language provides a vocabulary of automatics corresponding to actual machine operations, and a set of macro-instructions which, when processed, produce coding sequences that will transmit data, control program branching, perform automatic-decimal-point arithmetic, and modify addresses. The operands on Autocoder III statements may be written as symbolic representations of the information to be operated upon, and symbolic addresses, or tags, may be used to define the memory locations of data or of particular routines within the program. Data input and output fields may be defined in terms of the format of the data including the placement of decimal points, commas, dollar signs, etc.

Report/File Writing: This language consists of a vocabulary of nineteen words which, when used in a prescribed manner, cause generation of routines that will create tape files or produce printed reports. Statements in this language describe the format of print lines or tape records by specifying the contents and spacing of report headings, line headings, and detail lines. A date and page numbering may be included in the report. Provision is made also for accumulating counts or totals of any designated fields in the records being processed, and for printing those in stated formats upon the occurrence of changes in selected fields of the records. Routines in the Report/ File Writing language may be included in an appropriate point in the program and compiled by the Processor will result in error-free sequences of optimal coding that will produce reports or tape files, the contents and format of which will be precisely as specified.

Decision-Making: By use of this language, a single logical statement may be written at any point in an Autocoder III portion of a program to specify all the conditions on which a program decision is to be based, and the alternative courses the program is to follow if the conditions are satisfied or not satisfied. A single word, IF, in the vocabulary of the language, may be used with special codes that express the relationship (e.g., higher, lower, etc.) that define the individual conditions. Conditions are linked within a statement by logical connectors and are grouped in a prescripted manner to form the complete conditional statement. Decision-Making statements are translated by the Processor into instructions sequences that will perform the necessary analysis and other processing by the best possible methods.

Arithmetic: With statements similar to Decision-Making statements, mathematical operations upon any number of fields may be specified, in order to create a result. The word MATH in the operation field signals that the operand contains a free-form arithmetic expression consisting of tags and/or literals separated by operators, add, subtract, multiply or divide symbols, with possible parentheses, and special codes that express the relationships (e.g., higher, lower, etc.) that define the individual conditions. Conditions are linked within a statement by logical connectors and are grouped in a prescripted manner to form the complete conditional statement. Decision-Making statements are translated by the Processor into instruction sequences that will perform the necessary analysis and other processing by the best possible methods.

Table-Creating: This language permits automatic use of memory searching techniques by creating a string of variables with their associated data and a set of controls to accomplish the searching. Following a statement with TBLCF to the operation field and containing defining parameters, the programmer supplies the table entries or range of entries. These Processor entries are translated into a table suitable for serial or binary searching. Such a table may be utilized by macro-instructions, Report/File Writing statement and/or Decision-Making statements.

FORTRAN: This is a language for programming generalized computational problems. 705 FORTRAN programs may contain Autocoder statements at appropriate points. 705 FORTRAN permits three subscripts and constant values of range $10^{39}$. All the advantages of 705 Processor assembly are available to the user.

Equipment Specifications: 64,000 positions of storage 8 tape drives.

IBM Application & Systems Programs Library Abstract  File Number 0705-PR-131

705/COBOL and COMMERCIAL TRANSLATOR PROCESSOR

Abstract:

Purpose: The processor translates programs written either in COBOL or |Commercial Translator| into machine language programs for the 705 Models I, II, and III, and the 7060. Use of the processor in programs written for the 705 Models I and II is restricted, in that input/output routines must be written in Autocoder language. Both the 705 Model III and the 7060 it is possible to write programs completely in COBOL or |Commercial Translator|. (Continued on next page)
IBM Application & Systems Programs Library Abstract  File Number 0705-SM-051

SORT 54 /

Abstract:
Purpose: Sort 54 is a generalized three-way merge sorting program. It is capable of modifying itself according to control card specifications.

Equipment Specifications:
- IBM 705 (Model I or Model II)
- 754 Tape Control
- 727 Tape Drives
- 717 Printer

Additional Remarks: Sort 54 incorporates checkpoint, restart, and interrupt sort procedures. It accepts single or blocked fixed length records or single variable length records.

IBM Application & Systems Programs Library Abstract  File Number 0705-SM-052

SORT 54T/

Abstract:
Purpose: Sort 54T is a generalized three-way merge sorting program. It is capable of modifying itself according to control card specifications.

Equipment Specifications:
- IBM 705 (Model I or Model II)
- 777 Tape Record Coordinator
- 727 Tape Drives
- 717 Printer

Additional Remarks: Sort 54T incorporates checkpoint, restart, and interrupt sort procedures. It accepts single or blocked fixed length records or single variable length records.

IBM Application & Systems Programs Library Abstract  File Number 0705-SM-053

SORT 57

Abstract:
Purpose: Sort 57 is a generalized four-way merge sorting program. It is capable of modifying itself according to control card specifications.

Equipment Specifications:
- IBM 704 (Model I or Model II)
- 777 Tape Record Coordinator
- 727 Tape Drives
- 717 Printer

Additional Remarks: Sort 57 incorporates checkpoint, restart, and interrupt sort procedures. It accepts single or blocked fixed length records.

IBM Application & Systems Programs Library Abstract  File Number 0705-SM-054

SORT 80

Abstract:
Purpose: A generalized sorting program that will sort files of fixed- or variable-length data records, single or blocked, on a control data word as long as 100 characters and consisting of as many as five fields. To facilitate program coding, Sort 80 will use whatever tape units are specified in the control information supplied by the user.

Optional features of Sort 80 include an Extended-Edit mode for sorting particularly large files, and provisions for label processing and for the accumulation and checking of high totals. Edits are provided at logical points in the program to allow the user to include additional routines. Sort 80 also provides checkpoints, interrupt and restart procedures, and routines which facilitate the correction, or deletion and later recovery of unsalvageable records.

Equipment Specifications:
- 705 Model II or 7080
- 974 Data Synchronizer
- 8 Tape Drives
MERGE 80

Abstract:

Purpose: A generalized two- to ten-way merging program that will merge files of fixed- or variable-length data records, single or blocked, on a control data word as long as 100 characters and consisting of as many as five fields. To facilitate program scheduling, Merge 80 will use whatever tape units are specified in the control information supplied by the user.

Optional features of Merge 80 include provisions for label processing and for the accumulation and checking of hash totals. Exits are provided at logical points in the program to allow the user to include additional routines. Merge 80 also provides checkpoint, interrupt and restart procedures, and routines which facilitate the correction, or deletion and later recovery of unreadable records.

Equipment Specifications:

705 Model III or 7090

767 Data Synchronizer

4 tape drives

80 SERIES UTILITIES

Abstract:

Purpose: All "80 Series" utility programs except LOAD 80 and CLRMB80 contain routines that will check labels set up in conformance with IBM standards, if desired.

Single Card Load (LOAD80): Loads standard 705 program cards from the card reader or a 729 DS tape.

Clear Memory (CLRMB80): Sets memory positions 00160 - 25599 (or 79999) to blanks, and resets the accumulator and ASUs 01 - 11 without interrupting automatic operation.

Expanded Loads (LOAD81 and LOAD82): Load standard and/or expanded format program cards from one or a combination of two input units. Both programs feature the ability to locate a specified program on a tape.

Tape File Assembler (TPF80): Assembles tape files from cards or card images on tape. Output may be fixed- or variable-length tape records, single or blocked. Tapes must be used on 729 tape units.

Memory Print (MEPR80): Produces a printed listing of the contents of any tape mounted on a 729 tape unit, either directly on a 717, 705, or 720 printer or on a 720 I tape for later off-line printing.

Tape Duplication (TPDP80): Duplicates any 767 Data Synchronizer-controlled tape of tapes, or any selected file or files thereof.

Equipment Specifications:

705 Model III, or

7090
IBM Application & Systems Programs Library Abstract  File Number  0709-CW-005

IBM Application & Systems Programs Library Abstract  File Number  0709-FB-003

SHARE OPERATING SYSTEM - IBM MONITOR VERSION

Abstract:
Purpose: This is a collection of three programs to simulate off-line peripheral equipment. They are:
1. IBM 08 Card-to-Printer
2. IBM 08 Tape-to-Card Hollerith
3. IBM 08 Tape-to-Card Binary

Restrictions: Hollerith input may contain only those characters listed in Appendix I of the SHARE 70X System (IBM Manual, Part I, Preliminary Edition, July, 1956), including the symbol normally not used. Any other character will cause an error halt.

Column binary input must be identified by 't' in the sign positions of the 8-left and 7-left words of the card image (corresponding to the control punches in a column binary card). Absence of these bits will cause the routine to treat the image as Hollerith, or to transfer to the error returns specified by the calling sequence.

Timers:
IBM 07 36-106 ms
IBM 08 36 ms
IBM 09 118 ms
IBM 11 30-40 ms

Storage Requirements:
IBM 08 IBM 09 IBM 11
IBM 08 IBM 09 IBM 11
IBM 08 IBM 09 IBM 11
IBM 08 IBM 09 IBM 11

IBM Application & Systems Programs Library Abstract  File Number  0709-FI-001

SIMULATE PERIPHERAL EQUIPMENT

Abstract:
Purpose: This is a collection of three programs to simulate off-line peripheral equipment. They are:
1. IBM 08 Card-to-Printer
2. IBM 08 Tape-to-Card Hollerith
3. IBM 08 Tape-to-Card Binary
4. IBM 08 Tape-to-Printer

Restrictions: Hollerith input may contain only those characters listed in Appendix I of the SHARE 70X System (IBM Manual, Part I, Preliminary Edition, July, 1956), including the symbol normally not used. Any other character will cause an error halt.

Column binary input must be identified by "control punches" in the sign positions of the 8-left and 7-left words of the card. Absence of these punches will cause the program to treat the card as Hollerith, or to come to an error halt, as specified by the entry key.

Only the first 13 columns of each card are used. Tape records may be any length.

Storage Requirements:
IBM 08 IBM 09 IBM 11
IBM 08 IBM 09 IBM 11
IBM 08 IBM 09 IBM 11
IBM 08 IBM 09 IBM 11

IBM Application & Systems Programs Library Abstract  File Number  0709-CM-002

SORT 709

Abstract:
Purpose: This is a generalized sort program. This program uses 8 through 6-way merge. Input is binary or ECD from tape. The tape may consist of one or more reels of fixed-length records. Input file is sorted into ascending sequence based upon 1 through 8 control fields arbitrarily arranged within the record. The control fields may have a total of up to 258 bits.

Use: Control cards specify record length, input and output blockings, control fields, memory available, merge order, and tape units. Program may be interrupted at any point and later restarted.
GENERALIZED MERGE

Abstract:

Purpose: This is a generalized merge on 2, 3, 4 or 5 BCD or binary files. The input may be one or more reels of fixed-length records. The files are merged into ascending sequences as many as 360 bits of controlled data contained in up to 5 control fields. Output is in the same format as input, but blocked as per control card. Sequenced input files may arise from partitioning a large file to stay within the capacity of Sort 709, or from batch processing.

Timing: Timing is essentially that of one-tape pass for the output file.

739 UTILITIES

Abstract:

Purpose: This is a collection of 8 utility routines:
1. RAFG generates a file of random binary or BCD digits.
2. GOAL loads instructions punched in absolute octal with their alphabetic mnemonic operation codes.
3. YM:SG prints on-line messages.
4. TEMP compares two tapes word for word.
5. SPTR checks the sequence of a file of records. Records may be blocked and have up to five control fields.
6. SPTR provides a high-speed spot trace. The information is stored in upper memory and printed upon completion of program.
7. TELD builds short tapes for testing and other special purposes.
8. TD provides an octal or BCD print of tape.

739 DATA PROCESSING PACKAGE

Abstract:

Purpose: The 739 Data Processing Package is a collection of miscellaneous programming aids to the handling of commercial data on the 709. At present it consists of generalized subroutines which permit numeric data to be converted from and to binary and to be edited for visible output, and alphanumeric data to undergo movement, validity checking, and comparison.
IBM Application & Systems Programs Library Abstract  File Number 1401-A07-007

1401 CARD SYSTEM ERROR-DETECTION AIDS

Abstract:

Purpose: To provide a simple 1401 system for checking out programs.

Method: Does not apply.

Restrictions, Usage: Does not apply.

Storage Requirements: Does not apply.

Equipment Specifications: No special features required.

Remarks: The programs provide a control card method for "patching" a 1401 program with instructions that will either:

1. Halt the program at selected times;
2. Print selected areas of storage at selected times.

Means for conveniently removing the patches are also provided.

IBM Application & Systems Programs Library Abstract  File Number 1401-A05-007

1401 AUTOCODER PROGRAM

Abstract:

Purpose: To provide more powerful tools for programmers to enable them to concentrate their efforts on the problems of program logic rather than coding. In addition, it provides an extremely fast assembly system.

Method: Does not apply.

Restrictions, Usage: Does not apply.

Storage Requirements: Does not apply.

Equipment Specifications:

- 4000 core storage positions
- For compilation of 1401 source programs (Model 4400 for 1401 FORTRAN, Model 4400 for 1401 FORTRAN, Model 4400 for 1401 FORTRAN, Model 4400 for 1401 FORTRAN)
- 1401 Card Read-Punch
- 1401 FORTRAN

Additional Remarks:

1. Some of the tools provided are:
   (1) Macro instructions
   (2) Literals
   (3) Symbolic origins
2. Compatibility with 1401 is provided.
3. Assembly is completely automatic.
4. Complete diagnostics are provided.
5. Many optional outputs are provided.
6. The user can provide his own macro-instructions and subroutines.

IBM Application & Systems Programs Library Abstract  File Number 1401-A05-007

1401 FORTRAN

Abstract:

Purpose: 1401 FORTRAN makes available to 1401 DDS installations the standardized FORTRAN programming language, the principal use of which is to describe solutions to scientific and engineering problems. The FORTRAN compiler translates such descriptions, or source programs, into machine language. Use of the FORTRAN system will produce higher program writing efficiency, i.e., more reliable programs produced more quickly. In addition, because of the machine-independence of the FORTRAN language, programs written in FORTRAN and tested on the 1401 can be applied directly and quickly to any other machine for which a FORTRAN system is available.

1401 FORTRAN features are:
1. Fast compiling speed,
2. Usability on a 1401 Card System (no tape required), and
3. "Load-and-go" system organization.

(Continued on next column)

Use of programs:
The user's FORTRAN programs, punched on cards, are entered into the 1401 DDS, followed by the FORTRAN compiler, which may be on cards or tape. The source program is translated to the compiler into the environment 1401 machine language program in core storage, ready for execution. A listing is provided during the compilation which includes the source program statements, diagnostic information relating to the intelligibility and consistency of the source program, and other useful information comprising a record of the compilation.

Machine Configuration:

For compilation of source programs:

1. 1401 Processing Unit (any model with 8000 or more core storage positions)
   Advanced Programming Feature
   High-Low-Equgal Compare Feature
   Multiply-Divide Feature
2. 1401 Printer Model 1 or 2
   Tape Unit Model 719 II, 719 IV, 719 V, 719 VI, or 7129 may be used if installed in the 1401 FORTRAN compiler
   Sense switches may be used if installed to provide a 1401 listing of the object program during various stages of the compilation.

For execution of compiled programs:

1. 1401 Processing Unit (any model with 8000 or more core storage positions)
   Advanced Programming Feature
   High-Low-Equgal Compare Feature
   Multiply-Divide Feature
2. 1401 Card Read-Punch
3. 1401 Printer Model 1 or 2

Tape Unit Model 719 II, 719 IV, 719 V, 719 VI or 7129 - only as required for input and output data.

IBM Application & Systems Programs Library Abstract  File Number 1401-K0-006

1401 TAPE READING AND WRITING SUBROUTINES

Abstract:

Purpose: To provide 1401 users with closed subroutines which are consistent with the Applied Programming Tape Standards for Tape Reading and Writing.

The Subroutines consist of a Tape Read/Write Routine, a Read Routine and a Write Routine.

Included are:

1. Error checking procedures
2. Noise record procedures
3. Dumping of unreadable records

Method: Does not apply.

Restrictions, Usage: Does not apply.

Storage Requirements: Does not apply.

Equipment Specifications: Any 4K tape system

Advanced Programming Feature

*Necessary only with 1401 Read/Write Tape Routine
IBM Application & Systems Programs Library Abstract  File Number 1401-TD-055

1401 INPUT/OUTPUT CONTROL SYSTEM

Abstract:

Purpose: The 1401 I/OCS is a set of library routines which, when called from a 1401 Autocoder source program by macro instructions, are selected and tailored and included in the object program. These routines perform I/O functions and provide linkage to the user's object program. The specific statements generated at assembly time depend completely on the particular specifications contained in the user's source program.

Use of Program: The 1401 I/OCS library routines are to be placed in the 1401 Autocoder system (Version 3 or later Versioned through a liberation run.

Machine Configuration: The 1401 I/OCS will perform the I/O functions and associated housekeeping for tape, card reader, card punch and printer. The object machine must have, in addition to those of the above types, advanced programming features and the high-low equal compare feature. The amount of core storage required varies widely from program to program and must be determined at assembly time.

IBM Application & Systems Programs Library Abstract  File Number 1401-1M-007

1401 CARD SYSTEM SUBROUTINES

Abstract:

Purpose: To provide a few frequently used arithmetic subroutines.

Method: Does not apply.

Restrictions. Range: Does not apply.

Remarks: Programs provided:

Multiply I (for storage space economy)
Multiply II (for speed economy)
Divide
Decimal-to-Decimal Conversion
Units-to-Decimal Conversion

Note: Closed subroutine linkage instructions provided.

IBM Application & Systems Programs Library Abstract  File Number 1401-RG-048

1401 REPORT PROGRAM GENERATOR

Abstract:

Purpose: 1401 RPG is a programming system which generates report writing programs which are specified by the user in the RPG language established for IBM 1400-series machines. The generated report program will accept source data contained in either a card file, magnetic tape file or disk storage file. The language facilities specifying the report writing functions of heading and detail lines, total lines controlled by field breaks, offset total printing, summary punching, cross-footing and calculations, page and serial numbering, etc. The output report can be obtained at the printer, on cards, on tape, or on any combination of the three.

Use of Program: Report specifications, entered on cards, are entered into the 1401 DPS together with the RPG system deck. The output is a punched deck containing the generated report program in symbolic (1401 RPG) language. This deck is further processed by one of the 1401 assembly systems (3901-1, 3902, or Autocoder) to obtain the machine language report writing program ready for loading.

Machine Configuration:

For report program generation:

1 1401 Processing Unit (any model with 4000 or more Core Storage positions)
1 1402 Card Reader Punch
1 1403 Printer (Model 1 or 2)

For report program execution:

1 1401 Processing Unit (any model - core storage size required depends upon complexity of report)
1 1402 Card Reader Punch
1 Tape Units (Model 729 II, 729 YV, 729 V, 729 Y, or 7293)
1 1403 Printer (Model 1 or 2, 1406 Disk Storage Unit - only as required for input data file and output report media.

Multiply-Divide Special Feature - may be used if installed.

Sense Switches Special Feature - may be used if installed.

IBM Application & Systems Programs Library Abstract  File Number 1401-SM-029

1401 SORT I

Abstract:

Purpose: To provide a generalized 3-way SORT program for 1401 users. The program internally sorts input records and merges the sorted blocks into sequenced output records. SORT I may also be used as a merge program if input tapes are already ordered.

Method: Does not apply.

Restrictions. Format: Does not apply.

Storage Requirements: Does not apply.

Equipment Specifications: 4000 positions of storage

High-Low-Simple compare

Minimum of four tape drives

Additional Remarks:

1. SORT I may handle single or blocked records.
2. The sort will be on a maximum of five 18 core fields.
3. SORT I will allow a maximum of 800 character blocking for single core field records and 720 for multiple core field records.

(Continued on next page)
Abstract:
Storage Requirements: Does not apply.

1. Merge II is a generalized merge program adapted from a control card for each specific job.
2. The program can handle both blocked and unblocked records, with or without header and/or trailer labels.
3. The header and/or trailer labels may be altered by use of additional label cards.
4. Output may be relabeled if desired by user.
5. The merge may be accomplished on a minimum of ten (10) control fields.
6. Patch area is provided for user application.
7. Unreadable record options are similar to those of 1401 Sort 1 and II.

1401 CARD SYSTEM UTILITY PROGRAMS

Abstract:

Purpose: Utility programs to load or output programs and data.

Method: Does not apply.

Restrictions, Range: Does not apply.

Storage Requirements: Does not apply.

(Continued on next page)
IBM Application & Systems Programs Library Abstract  File Number: 1401-UT-026

1401 TAPE-TO-PRINTER UTILITY PROGRAM

Abstract:

Purposes:
1. To enable the printing of various tape configurations in many print configurations without the need for specific programs.
2. To simulate the 717, 720 and 730 off-line printers for tapes prepared on 700-700 series computers.

Methods: Does not apply.

Restrictions, Range: Does not apply.

Storage Requirements: Does not apply.

Equipment Specifications:
*1401 Model C3
1403 Model 2 Printer
1 (one) 729 Model II or IV
*1403 Card Read Punch
High-Low-Equal Compare

*May run on Model D3 if system tape produced on Model C3.

Additional Remarks:

Timing: Varies according to record types (i.e., Fixed length or Variable length), and according to spacing and skipping requirements. Fixed length records which are single spaced obtain maximum speed 600 lines/min/disk.
1. Maximum block size allowable is 1296 characters with editing; 1279 without editing.
2. Multi-record files and multi-file reels may be handled.
3. Sequence checking and exception testing are provided.

IBM Application & Systems Programs Library Abstract  File Number: 1401-UT-027

1401 CARD-TO-TAPE UTILITY PROGRAM

Abstract:

Purpose: The Card-to-Tape program provides for writing information contained in punched cards onto magnetic tape.

Methods: Does not apply.

Restrictions, Range: Does not apply.

Storage Requirements: Does not apply.

Equipment Specifications:
1401 Model C3
High-Low-Equal Compare
1403 Model 1 Card Reader-Punch
1 (one) 700 Model II or IV
1403 Model 2 Printer

Additional Remarks:

1. Input record is from 1 to 98 cards.
2. Rearrangement of input prior to output is allowed.
3. Up to 16 fields may be selected for output.
4. Blocking of 1480 characters of BCD records and 1589 characters for Column Binary records.
5. Sequence checking of cards and records can be performed.
6. An exception record procedure is provided.
7. Header and trailer labels may be inserted.
8. Column Binary records and intermixed Column Binary and BCD records can be written on tape if the 1401 system being used has the Column Binary Device.

IBM Application & Systems Programs Library Abstract  File Number: 1401-UT-028

1401 TAPE-TO-CARD UTILITY PROGRAM

Abstract:

Purpose: To transfer information recorded on magnetic tape into punched cards, with a variety of output column designations.

Methods: Does not apply.

Restrictions, Range: Does not apply.

Storage Requirements: Does not apply.

Equipment Specifications:
1401 Model C3
1403 Model 2 Printer
729 Model 1 or 4 Tape Drive
1401 Card Reader-Punch
High-Low-Equal Compare

Additional Remarks:

Timing: Varies from 300 to 230 c.p.m., depending upon the number of operations desired by the user.
1. Maximum block size allowable is 1297 characters.
2. Additional information not contained within the record may be punched.
3. Field sequence checking and field selection is permitted.
4. Multiple file reels are processed according to the user's specifications.
5. Exception record processing and card sequence numbering is allowed.
6. Header and Trailer labels are optionally treated.

IBM Application & Systems Programs Library Abstract  File Number: 1401-UT-029

MULTIPLE UTILITY PROGRAM FOR IBM 1401 TAPE SYSTEM

Abstract:

Purpose: To stimulate current off-line processing by 700 series equipment, and allow any combination of Tape-to-Card, Card-to-Printer and Card-to-Tape operations to be performed at the same time.

Methods: Does not apply.

Restrictions, Range: Does not apply.

Storage Requirements: Does not apply.

Equipment Specifications:
1401 Model C3
1403 Reader-Punch
1403 Model 2 Printer
729 Model 3 or 4 Tape unit (as many as user desires for 1, 2, or 3 simultaneous operations)
1401 High-Low-Equal Compare
Advanced Programming Features
Column Binary feature (if user desires)
*Print Storage

*Print Storage is not a mandatory specification. More rapid processing of data will occur when this feature is a part of 1403 system.

Additional Remarks:

Timing: Maximum speed will be effected when any one single operation is being performed. Tape-to-Printer 600 c.p.m., Card-to-Tape 800 c.p.m., Tape-to-Card 270 c.p.m.

When more than one operation is desired simultaneously, the following time speeds are applicable: (Continued on next page)
IBM Application & Systems Programs Library Abstract  File Number 1401-UT-001

FILE UTILITIES

Abstract:
This is a set of six independent programs to perform many common tasks associated with the 1405 disk storage. The programs are: Clear Disk, Disk-to-Printer, Tape-to-Disk, Card-to-Disk, Card-to-Printer.

Restrictions: The Tape-to-Disk and Card-to-Disk are companion programs to the Clear Disk program. The Card-to-Printer program requires a single character format control card.

Equipment Specifications: 1401 Model F, 1403, 1405, 1406, tape drives as required for programs.

IBM Application & Systems Programs Library Abstract  File Number 1401-UT-005

FILE ORGANIZATION ROUTINES

Abstract:

Purpose:
The chaining method of File Organization is an efficient method of handling the problem of duplicate file addresses. When control data (item number, man number, etc.) are converted to disk storage addresses, the 1401 File Organization Program will efficiently load and maintain a chained disk file as to minimize the amount of unused storage, as well as the retrieval time for each record.

1401 File Organization features are: 1) an edit program which will edit a symbolic version of the program so as to provide the most efficient program for any size 1401; 2) ability to make additions and deletions to a chained file; 3) load and add trailer records to a file; 4) unbind a file into cards or tape for reorganization; 5) an audit list consisting of the control data of records being loaded and their addresses; 6) input data records may be on card or tape.

Use of Programs:
The Load and Additions programs are used in conjunction with the edit program. The user provides the specifications of his file and machine in a control card which is examined by the edit program to create a symbolic version of the load and additions programs which meet those specifications. The edited program and the users conversion routine (routine to convert control data to disk address) are assembled with either SPS or Autocoder. The assembled program will then load the users data file (on card or tape) with a given format onto the disk file in the desired area. The program will create the necessary chain linkages.

The remaining programs are not edited, but must be assembled with the users conversion routine. The control card is examined at object time and the users data is operated upon according to the specifications in the control card.

All of the programs provide for all I/O error checking. The programs utilize one or two access arms depending upon the number available. If there are two arms, and one fails, the program will continue to operate with one arm.

Machine Configuration:
1401 Processing Unit (1000 core storage positions are minimum)
1405 Card Read-Punch
1405 Printer (Model 1 or 2)
1405 Disk Storage Unit (Model 1 or 2)
1 or 2 Tape Units (Model 729 II, 729 III, 729 V, or 7330) may be used if data is on tape.
IBM Application & Systems Programs Library Abstract  

**1410 PAT UTILITY SYSTEM (10/503)**

**Abstract:**

**Purpose:** The 1410 PAT System facilitates the testing of newly-developed 1410 programs by reducing the amount of machine time and programmer effort required during the testing stage of program development. The PAT System also lends itself to remote testing. The PAT System provides the automatic testing facility plus a number of core card, tape, and 1405 disk utility programs.

**Use of Programs:** The 1410 PAT System comprises a series of program testing routines and utility programs for each direction of the user and under control of the PAT program, are arranged in conjunction with the program to be tested on a PAT tape.

**Machine Configuration:** The 1410 PAT System (10/503) requires an IBM 1410 system with the following minimum configuration:

- 10,000 positions of core storage
- IBM 1405 Card Reader-Punch
- IBM 1402 Printer, Model 2
- IBM 729 II, 725 IV, or 7330 Magnetic Tape Units on Channel one (1)

IBM Application & Systems Programs Library Abstract  

**1410 PAT UTILITY SYSTEM (4030)**

**Abstract:**

**Purpose:** The 1410 PAT System facilitates the testing of newly-developed 1410 programs. This automatic testing procedure reduces the amount of machine time and programmer effort required during the testing stage of program development. The PAT System also lends itself to remote testing. The PAT System provides the automatic testing facility plus a number of 1410 core, tape, and 1405 disk utility programs.

**Use of Programs:** The 1410 PAT System comprises a series of program testing routines and utility programs for each direction of the user and under control of the PAT program, are arranged in conjunction with the program to be tested on a PATA tape.

The routines and programs are arranged on tape in the order they are to be executed. Testing the program remotely requires the loading of the PAT tape and an identification card for each program to be tested. The routines and programs on tape are automatically executed in predetermined sequence.

**Machine Configuration:** The 1410 PAT System requires:

a. An IBM 1410 with 40K positions of core storage
b. An IBM 1402 Card Reader-Punch
c. An IBM 1403 Model 2 Printer
d. At least two IBM 729 or 7330 Tape Units on Channel one (1)

IBM Application & Systems Programs Library Abstract  

**1410 BASIC AUTOCODER**

**Abstract:**

**Purpose:** The 1410 Basic Autocoder relieves the user from writing his routines in machine language. He may write his routines using a well defined set of mnemonic operation codes in conjunction with useful and significant labels, which he defines, and then processes with Autocoder to produce an operating object program. He may also write macro statements and include subroutines in the library. A more detailed description of this program is contained in the Autocoder Bulletin listed in the references.

**Use of Programs:** The source symbolic program is set up in a prescribed manner and is operated on by the Autocoder to produce an operating system deck.

**Machine Configuration:** The machine configuration required by the Autocoder is:

1. Minimum of 30 K storage.
2. Four IBM 729 II, IV, or 7330 Magnetic Tape Units.
3. An IBM 1401 Card Reader-Punch.
4. An IBM 1403 Printer, model 2.

Options are available to trade 1, 2, or 3 magnetic tape units for the 1020 and 1403 unit record devices.

IBM Application & Systems Programs Library Abstract  

**1410 COBOL PROCESSOR**

**Abstract:**

**Purpose:** 1410 COBOL Processor accepts programs written in the COBOL 41 language as input and produces complete object programs to perform the functions specified in the source statements.

**Use of Program:** The process involves a COBOL run (which produces COBOL diagnostics and the source program translated into Autocoder language and format) followed by an Autocoder run (which produces the object program assembly listing and a condensed deck). The process is continuous and complete if

1. No serious diagnostic errors are discovered, and
2. If the system configuration provides tape input to the Autocoder Processor.

**Machine Configuration:** Basic requirements are:

1. Minimum of 20 K storage.
3. An IBM 1403 Printer, model 2.
4. Four IBM 729 II, IV, or 7330 Magnetic Tape Units (may be interchange).

IBM Application & Systems Programs Library Abstract  

**1410 FORTRAN II PROCESSOR**

**Abstract:**

**Purpose:** The 1410 FORTRAN II Processor is a 1410 machine-language program. This program converts a source program written in the FORTRAN II language (which closely resembles the language of mathematics) into an object program ready to run on the IBM 1410. The FORTRAN II processor thus makes it possible for personnel trained in mathematics but not in programming to prepare problems for the computer.

**Use of Program:** The processor is used in two phases: a FORTRAN phase and an Autocoder phase. During the FORTRAN phase, the processor compiles a symbolic program in Autocoder format. During the Autocoder phase, the processor converts this Autocoder program into a 1410 object program.

**Machine Configuration:** Minimum machine requirements for the use of the processor are:

- 20,000 positions of core storage
- 1 IBM 1402 Card Reader-Punch, Model 2
- 1 IBM 1403 Printer, Model 2
- 4 IBM 729 II, IV, or 7330 Magnetic Tape Units (May be interchange)
IBM Application & Systems Programs Library Abstract  File Number 1410-10-909

1410 INPUT/OUTPUT CONTROL SYSTEM (CARD/TAPE I/OCS)

Abstract:

Purpose: The 1410 Card/TAPE I/OCS relieves the user from coding input and output routines for unit record equipment and magnetic tapes. It enables the programmer to handle logical records merely by using GET, PUT, and control instructions. The blocking and deblocking of records is handled automatically by I/OCS. Also, I/OCS can be instructed to provide the coding required for the overlapping of input and output operations with processing if the 1410 is equipped with the Overlap and Priority special features.

Use of Program: For each program which is to utilize the I/OCS, the programmer must:

1. Use the I/OCS macro-instruction in his program.
2. Write one set of I/OCS statements.
3. Write one set of DTI statements for each file used by his program.
4. Write proper DA statements for each area used by the I/OCS.

The I/OCS routines are generated by the Autocoder and placed in the user's program when it is compiled.

Machine Configuration: I/OCS has no machine configuration requirements. Autocoder configurations are, of course, required during I/OCS generation.

IBM Application & Systems Programs Library Abstract  File Number 1410-10-901

1410 INPUT/OUTPUT SYSTEM FOR 1405 DISK STORAGE

Abstract:

Purpose: The 1405 Disk I/OCS provides several macro-instructions and utility routines that handle the scheduling of 1405 input and output operations for random and/or sequential processing.

Use of Program: This I/OCS is used in conjunction with 1410 Card/TAPE I/OCS. The appropriate disk I/O routines are generated by 1410 Autocoder according to the file specifications and placed in the user's program when it is compiled.

Machine Configuration: The machine configuration required by the Input/Output system for 1405 Disk Storage is:

1. Minimum of 20K storage
2. 1405 Disk storage
3. Processing Overlap and Priority special features.

IBM Application & Systems Programs Library Abstract  File Number 1410-PR-108

PROCESSOR OPERATING SYSTEM TAPE

Abstract:

Purpose: This is a system tape containing the following 7 programs:

1410-SY-907 System Supervisor
1410-AU-908 Autocoder
1410-ID-909 I/OCS Card/TAPE
1410-ID-911 I/OCS 1405 Disk
1410-RO-910 Report Program Generator
1410-CB-912 COBOL II
1410-FO-913 FORTRAN II

IBM Application & Systems Programs Library Abstract  File Number 1410-RO-109

1410 REPORT PROGRAM GENERATOR

Abstract:

Purpose: The 1410 Card RPG will handle card input and card-printer output. Machine requirements are:

- 16K storage
- 1462 card reader/punch
- 1403 printer (either 100 or 132 character positions)

The report program generated by RPG will have machine requirements dependent on the specifications provided. The column width would be:

- 16K storage
- 1462 card/reader punch

IBM Application & Systems Programs Library Abstract  File Number 1410-RO-210

1410 REPORT PROGRAM GENERATOR (CARD/TAPE/1405 + DECK RPG)

Abstract:

Purpose: The 1410 RPG accepts report specifications and produces a symbolic program deck (Autocoder format) for the desired report program. The generated report program can produce a wide range of formats, extracting its data from a card, tape or disk file (one only) and performing calculations at any point in the reporting process. RPG-generated programs utilize the 1410 I/OCS.

Use of Program: A control card and the report specifications cards are placed in proper order in the card reader. The Processor Operating System Tape, 1410-PR-108, and one work tape are used in the RPG run. An Autocoder run must follow to produce the program deck for the report program. The output of the generated program can be a printed report and/or punched cards, or tape records in the move mode, even parity.

Machine Configuration: Minimum requirements are:

1. For RPG (to generate the report program) - 1410 system... 20K storage... 1402 Card Read Punch... two magnetic tape units (1255 IL, IV, or 7330).
2. For Autocoder (to assemble the report program) - 1410 system... 20K storage... 1402 Card Read Punch... four magnetic tape units (1255 IL, IV, or 7330)... 1403 Printer, model 2. (See configuration of Autocoder for options.)
3. For the report program (to produce the report) - 1410 system... 20K storage... 1402 Card Read Punch... other I/O units appropriate to the program.

IBM Application & Systems Programs Library Abstract  File Number 1410-SL-062

Simulation of the 1410 with the 704/709/7090

Abstract:

Purpose: The program enables the user to test and correct 1410 programs prior to installation of an IBM 1410 data processing system. The system will trace or dump simulated programs.

Restrictions: The program simulates standard card and tape systems. The simulated 1410 has 50,000 core storage positions. Using Basic Autocoder the simulator will assemble 1410 programs. A maximum of one disk of 1405 storage can be simulated.

Timing: The 709 takes approximately 20 times longer than if the program was running on a 1410.

Equipment Specifications:
- 32,896 words of core storage
- 4 tape units + 1 for simulated 1410 tape units + 2 for disk

Additional Remarks: This program is distributed on a card deck.

IBM 1401 PROGRAM LIBRARY ABSTRACT  File Number 1410-IT-101

SIMULATION OF THE IBM 505 ON THE IBM 1410

(Continued on next page)
IBM Application & Systems Programs Library Abstract  File Number 1410-SM-110

1410 SORT 10

Abstract:

Purpose: Sort 10 is a generalized sorting program which employs a multibuffer tape sorting and merging program designed to permit either the sorting or the merging of data as to produce ordered output data. Input records can be fixed or variable length, single or blocked. Output can be either in ascending or descending order. Any order of merge up to 5-way may be employed.

Uses: A minimum of two control cards must be prepared by the user prior to operating Sort 10 on the 1410. These cards supply the program with information it needs to make itself specific for the machine configuration and for the functions to be performed. for the data characteristics and for the machine configuration.

Machine Configuration: Sort 10 requires an IBM 1410 Data Processing System with the following minimum configurations:
   a) 20,000 positions of core storage.
   b) 1 IBM 1405 Disk Storage Unit.
   c) The machine configuration for the individual programs on the Processor Operating System Tape.
   d) 1 IBM 292 II, 292 IV, or 7310 Magnetic Tape Units.
   e) IBM 1425 Card Read-Punch, Model 2.
   f) If storage size is 40K, Sort 10 will use the additional storage, when necessary, to increase the size of its input/output areas and work areas.

IBM Application & Systems Programs Library Abstract  File Number 1410-SM-111

SORT/MERGE 11

Abstract:

Purpose: Sort-Merge 11 is a generalized on-buffered tape sorting and merging program designed to permit either the sorting or the merging of data so as to produce ordered output data. Input records can be fixed or variable length, single or blocked. Output can be either in ascending or descending order. Any order of merge up to 5-way may be employed.

Uses: A minimum of two control cards must be prepared by the user prior to operating Sort-Merge 11 on the 1410. These cards supply the program with information it needs to make itself specific for the functions to be performed, for the data characteristics and for the machine configuration.

Machine Configuration: Sort-Merge 11 requires an IBM 1410 Data Processing System with the following minimum configurations:
   a) 20,000 positions of core storage.
   b) 4 IBM 729 II, 729 IV, or 7310 Magnetic Tape Units (may be intermixed) if Sort-Merge 11 is to function as a Sort. (To perform a 2-way Merge, only three tapes are needed.)
   c) IBM 1425 Card Read-Punch, Model 2.
   d) If storage size is 40K, 50K or 60K, Sort-Merge 11 will use the additional storage, when necessary, to increase the size of its input/output Areas and Work Areas.

IBM Application & Systems Programs Library Abstract  File Number 1410-SM-112

SORT/MERGE 12

Abstract:

Purpose: Sort-Merge 12 is a generalized tape sorting and merging program which employs the processing Overlap and Priority Special Features. It is designed to permit either the sorting or the merging of data so as to produce ordered output data. Input records can be fixed or variable length, single or blocked. Output can be either in ascending or descending order. Any order of merge up to 5-way may be employed.

Uses: A minimum of two control cards must be prepared by the user prior to operating Sort-Merge 12 on the 1410. These cards supply the program with information it needs to make itself specific for the functions to be performed, for the data characteristics and for the machine configuration.

Machine Configuration: Sort-Merge 12 requires an IBM 1410 Data Processing System with the following minimum configurations:
   a) 20,000 positions of core storage.
   b) The machine configuration for the individual programs on the Processor Operating System Tape. The System Supervisor consists of three programs contained in the system tape. These are self loading, or are called by control cards, and perform the functions listed above as directed by control information.
   c) IBM 292 II, 292 IV, or 7310 Magnetic Tape Units.
   d) IBM 1425 Card Read-Punch.

IBM Application & Systems Programs Library Abstract  File Number 1410-SM-113

1410 SYSTEM SUPERVISOR

Abstract:

Purpose: The System Supervisor has several functions in the operation of the Processor Operating System Tape.

1. In the role of a Supervisor, it picks up information from control cards and, acting upon this information, positions the Tape System, calls in the required phases of program and then turns control over to the program called.
2. The System Supervisor also accomplishes the duplication of new system tapes as well as the maintenance of the system tape.
3. Another part of the System Supervisor is the Library PRINT Program, which prints any desired section of the library that is on the Processor Operating System Tape.

Use of Programs: The System Supervisor consists of three programs contained in the system tape. They are self loading, or are called by control cards, and perform the functions listed above as directed by control information.

Machine Configuration: The machine configuration required by the System Supervisor for system maintenance runs is:
   1. Minimum of 8K storage.
   2. Two IBM 292 II, 292 IV, or 7310 Magnetic Tape Units.
   3. IBM 1425 Card Read-Punch.

The machine configuration for the individual programs on the Processor Operating System Tape are specified in the Abstracts of the programs.

IBM Application & Systems Programs Library Abstract  File Number 1410-SM-114

1410 UTILITY PROGRAMS

Abstract:

Tape File Generator 1: This program prepares blocked or unblocked tape files from fixed-length card records.

Tape File Generator 2: This program prepares blocked or unblocked tape files from variable-length card records.

Tape Compare Program: This program compares the contents of two magnetic tapes, each of which can be in odd or even parity, and high or low density. They may have fixed or variable-length records and may be blocked or unblocked. Only one file can be compared on a run, and the comparisons may start at any file or record on either tape. If the records are not identical, they will be written out.

Tape Duplicate Program: This program duplicates the contents of one magnetic tape on a second tape. The duplicated tape can be written in high or low density and in odd or even parity, regardless of the density and parity of the original tape. The original tape may contain fixed or variable-length records, and may be blocked or unblocked. Up to nine files of a multi-file reel can be duplicated.

Snapshot Program: The Snapshot Program is in a program testing aid. It prints out the contents of a specified area of core storage following the execution of any specified instruction in the object program. Following the execution of the Snapshot Program, control is returned to the object program. The Snapshot Program also prints the contents of the Index Registers and the settings of the HIGH-LOW-EQUAL, ARITHMETIC-OVERFLOW, or ZERO RESULT indicators.

Storage Print Program: The Storage Print program prints out the entire contents of 1410 core storage. Subscript characters are used in place of those not available on the user's 1403 Printer. Word marks are represented by the digit "3" printed above the character with which the word mark is associated.
1410-1405 DISK UTILITY PROGRAMS

Abstract:

Clear Disk Program: The Clear Disk Storage Program erases all data in all or selected portions of disk storage by writing blanks. The user also has the option of filling these areas with any one of the other 63 valid characters, and the ability to write a six-digit address in the first six positions of each sector cleared by this program.

Disk-to-Tape Program: The Disk to Tape 'A' Program enables the user to preserve data contained in all or selected portions of a disk file before data is updated or altered.

Tape-to-Disk Program: The Tape to Disk 'A' Program enables the user to reload into disk storage all or selected portions of the tape records that have been unloaded by the Disk to Tape Program.

Disk-to-Printer Program: The Disk to Printer Program is used to print out on the IBM 1403 Printer data contained in all or portions of a disk file.

Disk File Generator: The Disk File Generator enables the user to load data from punched cards into disk storage.

Use of Programs: The 1410-1405 Disk Utility Programs are used in conjunction with a Machine Specifications Card, and with Area Control Card(s). The programs will allow the user to clear all of disk storage or selected areas of it to blanks or any other allowable character, generate data in all or selected areas of disk storage, write the contents of all or selected areas of disk storage on tape or on the printer, and reload areas of disk storage that were previously written on tape. The smallest area that may be acted upon, however, is a single track of ten sectors.

Machine Configuration

Basic Requirements for all programs.

Each program requires a minimum of:

- 10,000 positions of core storage
- 1 IBM 1405 Disk Storage Unit, Model 1 or 2
- 1 IBM 1402 Card Reader-Punch

Additional requirements:

- 1410-1405 Disk-to-Printer Program
  - 1 IBM 1403 Printer, Model 1 or 2
- 1410-1405 Disk-to-Tape Program
  - 1 IBM 729 II, 729 IV, or 7330 Magnetic Tape Unit
- 1410-1405 Tape-to-Disk Program
  - 1 IBM 729 II, 729 IV, or 7330 Magnetic Tape Unit

1410-1405 DISK FILE PROTECTION PROGRAMS

Abstract:

Disk-to-Tape with Overlap: The Disk-to-Tape File Protection Program enables the user to preserve data contained in all or specified portions of a disk file before data is updated or altered. Because of the utilization of the Overlap special feature this program is considerably faster (approximately 35%) than the DISK-TO-TAPE utility program. This program is primarily written to be used in conjunction with the users production programs.

Tape-to-Disk with Overlap: The Tape-to-Disk File Protection Program enables the user to reload into disk storage all or specified portions of the tape records that have been unloaded by the TAPE-TO-DISK File Protection Program. Because of the utilization of the Overlap special feature this program is considerably faster (approximately 20%) than the DISK-TO-TAPE utility program. This program is primarily written to be used in conjunction with the users production programs.

Use of Programs

These File Protection Programs can only be used on a machine that has the Processing Overlap special feature, and only full tracks are written and loaded. The programs are used in conjunction with a Machine Specifications Card, and with Area Control Card(s). The user can unload onto tape or reload from tape either a complete disk file or selected areas of the file. Either the Move mode or the Load mode may be used.
1620 FLOW TRACE PROGRAM

Purpose: To enable the programmer to check that the path (flow) of his program is correct. Should the program deviate from the expected, the trace helps localize the trouble.

Method: The trace program detects every branch that actually occurs in the object program, types the address of the branch, controller, and the address to which it branched.

Restrictions, Range: Cannot discontinue the trace in the middle of the subroutine linked to the main program by a BT or BTM and a BB instruction.

Storage Requirements: 514 positions of core storage. Program is relocatable.

Equipment Specifications: IBM 1620 with paper-tape reader. No restriction on 1620 core storage (16K, 32K, 64K). Trace output is in typewriter. Cannot be used on machines with indirect addressing feature.

1620 SELECTIVE TRACE PROGRAM

Purpose: To provide more detailed checking than the FLOW TRACE PROGRAM. To help pinpoint the exact location of the trouble. To enable the programmer to check each instruction as it appears in memory and see the data fields as they are manipulated.

Method: Not applicable.

Restriction, Range: If instruction contains a record mark, only that part of the instruction up to, but not including the record mark, will be typed. Cannot terminate the trace during the execution of a subroutine linked to the program by a BT or BTM and a BB instruction.

Storage Requirements: Program requires 2443 core locations. The program is completely relocatable.

1620 FORTRAN (Tape)

Purpose: Program converts source program written in FORTRAN language into machine language instructions.

Method: Not given.

Restrictions, Range: Permissible FORTRAN language is a subset of 704/709/7000 FORTRAN language. Number of symbols is limited to 200.

Storage Requirements: Requires 50,000 storage positions 1620.

Equipment Specifications: 1620 CPU
1621 Paper Tape Reader
961 Tape Punch
1623 Core Storage Unit may be added, at the user's option.

1620 FORTRAN (Card)

Purpose: Program converts source program written in FORTRAN language into machine language instructions.

Method: Not given.

Restrictions, Range: Permissible FORTRAN language is a subset of 704/709/7000 FORTRAN language. Number of symbols is limited to 200.

Storage Requirements: Requires 50,000 storage positions 1620.

Equipment Specifications: 1620 CPU
1621 Card Reader-Punch Unit
1623 Core Storage Unit may be added, at the user's option.

1620 FLOW TRACE PROGRAM

Purpose: To enable the programmer to check that the path (flow) of his program is correct. Should the program deviate from the expected, the trace helps localize the trouble.

Method: The trace program detects every branch that actually occurs in the object program, types the address of the branch, controller, and the address to which it branched.

Restrictions, Range: Cannot discontinue the trace in the middle of the subroutine linked to the main program by a BT or BTM and a BB instruction.

Storage Requirements: 514 positions of core storage. Program is relocatable.

Equipment Specifications: IBM 1620 with paper-tape reader. No restriction on 1620 core storage (16K, 32K, 64K). Trace output is in typewriter. Cannot be used on machines with indirect addressing feature.

1620 SELECTIVE TRACE PROGRAM

Purpose: To provide more detailed checking than the FLOW TRACE PROGRAM. To help pinpoint the exact location of the trouble. To enable the programmer to check each instruction as it appears in memory and see the data fields as they are manipulated.

Method: Not applicable.

Restriction, Range: If instruction contains a record mark, only that part of the instruction up to, but not including the record mark, will be typed. Cannot terminate the trace during the execution of a subroutine linked to the program by a BT or BTM and a BB instruction.

Storage Requirements: Program requires 2443 core locations. The program is completely relocatable.

1620 FORTRAN (Tape)

Purpose: Program converts source program written in FORTRAN language into machine language instructions.

Method: Not given.

Restrictions, Range: Permissible FORTRAN language is a subset of 704/709/7000 FORTRAN language. Number of symbols is limited to 200.

Storage Requirements: Requires 50,000 storage positions 1620.

Equipment Specifications: 1620 CPU
1621 Paper Tape Reader
961 Tape Punch
1623 Core Storage Unit may be added, at the user's option.

1620 FORTRAN (Card)

Purpose: Program converts source program written in FORTRAN language into machine language instructions.

Method: Not given.

Restrictions, Range: Permissible FORTRAN language is a subset of 704/709/7000 FORTRAN language. Number of symbols is limited to 200.

Storage Requirements: Requires 50,000 storage positions 1620.

Equipment Specifications: 1620 CPU
1621 Card Reader-Punch Unit
1623 Core Storage Unit may be added, at the user's option.
TRANSPORTATION PROBLEM

Abstract:
Purpose: This program solves the transportation problem. That is, it minimizes the total cost of shipping from M warehouses to N retailers.
Method: A iterative search technique applied to the stepping-stone method.
Restrictions: Problem sizes are indicated by the formula:
\[ M \times N \times (M + N) \times (M + N) \times (M + N + 1) \]
where \( M \) = number of warehouses, \( N \) = number of retailers, \( M \) = total of 99, \( M + N \) = maximum number of digits used to specify units, \( M + N + 1 \) = number of positions of core memory.
Typical sizes are 40 x 40 with both MDCC and MODC equal to 6 digits fields, 40 x 80 with MDCC and MODC reduced to 5 digits fields, or if 65K additional memory is available, a 60 x 300 problem may be solved using 3 digit fields.
Equipment Specifications: Card or tape I/O, indirect addressing.
Additional Remarks: Results of a 40 x 50 Problem
Calculation time for a 40 x 50 test problem varied from 3 min. using 3 digit cost and unit fields to 3 1/4 min. using 8 digit fields. The variation of core storage used was from about 15,000 to over 26,000. The total card input required approximately 8 1/6 additional min. while the output added another 1 1/6 min. for a total running time of less than 7 minutes.
Other 40 x 50 test problems have required as much as 8 minutes of calculation time, using 8 digit fields and occupying over 26,000 core locations.

IBM Application & Systems Programs Library Abstract  File Number  1620-LM-018

Production Line Balancing

Abstract:
Purpose: This routine assigns operators to jobs on an assembly line. The assembly line is divided into zones and the assignment is done in a manner which tends to balance the work load in each zone.
Method: A hit approximation method.
Restrictions: There can be up to 99 zones. The maximum number of jobs per zone is 97 to 99 depending on the average number of precedence jobs per job. The maximum number of can do jobs is 98.
Timing: A problem with 320 input cards and 367 can do jobs took about 3 minutes inclusive of I/O.
Equipment Specifications: Paper tape reader or card reader.

IBM Application & Systems Programs Library Abstract  File Number  1620-LM-028

1620 FORTRAN with FORMAT - AUTOMATIC FLOATING POINT SUBROUTINES, CARD SYSTEM

Abstract:
Purpose: This subroutine package may be used with 1620 FORTRAN with FORMAT, Card System (Program #1620-FO-003) to realize the advantages of the Automatic Floating Point feature. Storage requirements for the subroutines are reduced and execution time of object programs decreased.
Use of the Program: The subroutines may be incorporated into the object program tape at compilation or may be loaded separately prior to the execution of the object program. Messages are automatically types during compilation and loading, indicating appropriate action by the user. This subroutine tape is fully compatible with the two distributed with the 1620 FORTRAN with FORMAT processor.
Machine Configuration: The subroutine package operates on a 1620 with the card re Spart and the Automatic Floating Point feature.

IBM Application & Systems Programs Library Abstract  File Number  1620-LM-029

1620 FORTRAN with FORMAT - AUTOMATIC FLOATING POINT SUBROUTINES, TAPE SYSTEM

Abstract:
Purpose: This subroutine package can be used with the 1620 FORTRAN with FORMAT, Tape System (Program #1620-FS-003) to realize the advantages of the Automatic Floating Point feature. Storage requirements for the subroutines are reduced and execution time of object programs decreased.
Use of the Program: The subroutines may be incorporated into the object program tape at compilation or may be loaded separately prior to the execution of the object program. Messages are automatically types during compilation and loading, indicating appropriate action by the user. This subroutine tape is fully compatible with the two distributed with the 1620 FORTRAN with FORMAT processor.
Machine Configuration: The subroutine package operates on a 1620 with punched tape input-output and the Automatic Floating Point feature.

IBM Application & Systems Programs Library Abstract  File Number  1620-MI-015

1620 RANK TOTAL PROGRAM

Abstract:
Purpose: The purpose of this program is to determine quickly and to a high probability whether a duplicated tape is an exact character for character copy of its original. This is accomplished by taking an arithmetic “hash” total of all the characters on any given tape.
Restrictions: Range: Does not apply.
Method: After each record in read in, it is split into fields of twenty digits and then each of these fields, in turn, is extracted from an area called the accumulator. At the conclusion of the routine the accumulator is compared with a previously entered check total and a message indicating the result is typed.
Storage Requirements: The program occupies core locations 402 to 1116 and 19650 to 19999. The remainder is available for input records.
Equipment Specifications: This program may be used on a basic IBM 1620 paper tape machine with no optional features.

IBM Application & Systems Programs Library Abstract  File Number  1620-MI-026

1620 NUMERIC TAPE DUPLICATOR/CORRECTOR

Abstract:
Purpose: To duplicate or correct 1620 tapes consisting only of numeric records, separated by end-of-line characters.
Method: Punching a tape which is an exact copy of the original or punching a second tape incorporating the desired changes.
Restrictions: Range: Maximum permissible record length is 8,60. Also, corrections may not increase or decrease the length of any record.
Storage Requirements: Program is loaded into memory from 00400 to 00800. Each record to be duplicated is loaded from 00000. The program also uses an area of core storage, ending in 99999 and equal to the length of the record, as a dump area.
Equipment Specifications: 1620 with paper tape and 30K memory.

IBM Application & Systems Programs Library Abstract  File Number  1620-MI-004

1620 GOTOAN (Tape)

Abstract:
Purpose: A relatively fast compiler for programs which will generally be executed only once.
Method: GOTOAN stores the compiled program in memory during compilation. The object program is then executed in an interpretive mode. No object tape or deck is produced. After execution of an object program, compilation of a new object program is possible without loading the processor.
Restrictions: Range: The language used in GOTOAN is a modified subset of FORTRAN, including the functional subroutines. Arithmetic statements are restricted to one arithmetic operation per statement. (Continued on next page)
IBM Application & Systems Programs Library Abstract  File Number 1620-SP-011

IBM 1620 SYMBOUC PROGRAMMING SYSTEM - CARD I/O

Abstract:

Purpose: This programming system assembles symbolic instructions into absolute machine language. The source program, consisting of the symbolic instructions, is read only once.

Restrictions, Range: The system can accommodate 511 labels.

Method: Does not apply.

Storage Requirements: The system occupies memory from position 100 to 19999.

Equipment Specifications: The system is designed to operate on a basic 1620 with card I/O and can be modified for the additional storage unit 1623.

IBM Application & Systems Programs Library Abstract  File Number 1620-SP-010

IBM 1620/1710 SPS, CARD SYSTEM

Abstract:

Purpose: 1620/1710 SPS, a symbolic programming system in use since late 1960. It provides many additional features in the assembly of source programs, and includes five sets of floating point subroutines for use on 1620 or 1710 systems of a variety of configurations. These are:

a) Fixed length floating point numbers not using the Automatic Divide feature.

b) Fixed length floating point numbers using the Automatic Divide feature.

c) Variable length floating point numbers not using the Automatic Divide feature.

d) Variable length floating point numbers using the Automatic Divide feature.

e) Variable length floating point numbers using the Automatic Floating Point feature.

The range of floating point numbers is:

\[ 2 \times 10^{-308} \ldots 2 \times 10^{308} \]

For variable length subroutines the fractional part of the floating point number may vary from 2 to 45 digits.

Use of Programs: A 1620 or 1710 system with 20,000 digits of storage. The processor can be modified for 45,000 or 60,000 digits of storage to allow an extension of the label table.

For assembly of source programs:

Basic Card 1620 or 1710 with 20,000 digits of storage. The processor can be modified for 45,000 or 60,000 digits of storage to allow an extension of the label table.

For execution of assembled programs:

A 1620 or 1710 system with any optional features.

IBM Application & Systems Programs Library Abstract  File Number 1620-SP-009

IBM 1620 SYMBOUC PROGRAMMING SYSTEM - TAPE I/O

Abstract:

Purpose: This programming system assembles symbolic instructions into absolute machine language. The source program, consisting of the symbolic instructions, is read twice.

Restrictions, Range: The system occupies memory from position 100 to 19999.

Equipment Specifications: The system is designed to operate on a basic 1620 with tape I/O, and can be modified for the additional storage unit 1623.
SPS is an extension of 1620 BPS, a symbolic programming system in use since late 1959. It provides many additional features in the assembly of source programs, and includes five sets of floating point subroutines for use on 1620 or 1710 systems of a variety of configurations. These are:

a) Fixed length floating point numbers not using the Automatic Divide feature.
b) Fixed length floating point numbers using the Automatic Divide feature.
c) Variable length floating point numbers not using the Automatic Divide feature.
d) Variable length floating point numbers using the Automatic Divide feature.
e) Variable length floating point numbers using the Automatic Floating Point feature.

The range of floating point numbers is:

$1.00000 \ldots 0 \times 10^{-99}$ to $1.99999 \ldots 9 \times 10^{99}$

For variable length subroutines, the fractional part of the floating point number may vary from 2 to 45 digits.

Use of Program

With the SPS processor loaded in the storage, the source statements may be entered on the typewriter or through the tape reader. In the first pass, the statements are scanned, certain errors detected, and label table constructed. In the second pass the source statements are again scanned; additional errors are indicated, and the program assembled in machine language. A condensed self-loading tape may be punched. Listing on the typewriter is also possible. A map of storage assignments may be typed. If subroutines are required, the proper subroutine tape will be processed and subroutines selected for inclusion in the object program.

Machine Configuration

For assembly of source programs:

Basic tape 1620 or 1710 with 20,000 digits of storage. The processor can be modified for 40,000 or 80,000 digits of storage to allow an extension of the label table.

For execution of assembled programs:

A 1620 or 1710 system with any optional features.
IBM Application & Systems Programs Library Abstract

1710 Simulator/7090

Abstract

Purpose: The 7090 Simulator of the 1710 Control System provides the ability to perform program checkout:

1. Prior to the installation of a 1710 System.

2. Subsequent to the installation but without requiring that the 1710 be removed from its normal task of Data Acquisition, Operator Guide or Closed Loop Control.

3. Without requiring modification of a physical system to conform to the program requirements, i.e., modifying a 1710 System to have the proper function, and filter and matching cards, at a given 1712 Multiplexer and Terminal Unit Address.

Machine Configuration:

For simulation of the 1710:

- 7090 with two tape channels (A & B)
- 4 tapes on channel A
- 2 tapes on channel B
- 32,000 words of core storage
- On line printer (SHARE II Board)

The simulator will simulate the following 1710 features:

1. Random & Sequential Addressing
2. Interrupt (AOCITAS Complete: Indicator #40)
3. Contact Sense (200 pt/sec)
4. Contact Operate
5. Analog Input (20 pt/sec)
6. Analog Output (set point positioner)
7. 300 M.T.U. Addresses
8. 1711 Manual Entry Switches
9. Process Branch Indicators
10. Indirect Addressing
11. Additional Instruction (TNF-TNS-MF)
12. Divide
13. 1623 Additional Core Storage 1 or 2 20,000 Digit Modules

MACHINE CONFIGURATION

For assembly of Source Programs:

- 709/7090 with two tape channels (A & B)
- 4 tapes on channel A
- 2 tapes on channel B
- 32,000 words of core storage
- On line printer

For execution of assembled programs:

A 1620 or 1710 System with either paper tape or card I/O and those optional features required by the Source Language Program, such as the 1630 additional instructions or 1710 Random Addressing Features.

IBM Application & Systems Programs Library Abstract

1710 SP/709-7090 PROCESSOR

Abstract

Purpose

The 709/7090 Processor provides the 1710 user with the ability to assemble programs for a 1710 installation without removing the 1710's capability to perform its normal task of Data Acquisition, Operator Guide or Closed Loop Control. The processor provides the user with all of the features of the 1401/1710 SPS while increasing the assembly speed and the size of the programs that may be assembled.

Use of Programs

With the SPS Processor loaded into storage under control of the IBM 7090 Monitor, the source statements are read from Tape A3. In the first pass, the statements are scanned, certain errors detected, and the label table is constructed locally in SPS labels. The processor writes the scanned statement on an intermediate tape (B3) along with certain control information to be used during the second pass.

Prior to the second pass of the source language (from tape B3), the label table is examined to determine the number of entries. If there are more than 35 entries, binary search indices are built up by the processor and a binary search is made when looking up labels during the second pass.

In the second pass, the statements are read from the intermediate tape (B3) assembled, written on the punch output tape (A3) in the format specified in the control card for this assembly, i.e., condensed card or paper tape format, and written on the print tape (A2).

At the end of Pass II if any subroutines were used, the processor selects the subroutine set specified from the subroutine tape (B7) and assembles and writes the output for listing (A2) and punching (A3). At the end of Pass II, the processor writes the resultant map of 1710 storage on the printer tape. The processor will repeat the assembly process until all source language programs have been assembled.

A - 1710
**AUTO-TAP TEST GENERATOR SYSTEM**

Abstract:

The AUTO-TEST Generator System provides a highly flexible and efficient method of creating tapes for automatic testing. The test tape is created by the ATG System in a one pass generation run.

The minimum system configuration required for a Generation Run is a 7070 capacity of 5K, one tape channel, and three tape drives. If available, a capacity over 5K, 4 tape channels, 40 tape drives, the 7500 Card Reader, the 7501 Console Card Reader, the 7550 Card Punch, and the 7400 On-Line Printer may also be used in generating the test tape. One control card (the ATG Control Card) and the settings of the Console Alteration Switches specify the machine configuration to be used for the generation run.

Testing may be performed with the generated tape on a system even more basic than the minimum needed for generating the test tape or may be done on any combination of the units mentioned above. One control card for each program object (or Tape Card) specifies the machine configuration to be used for testing that object program.

The configuration of the system which generates the test tape does not have to be the same as the configuration of the system which performs the testing.

**AUTO-CODER 74**

Abstract:

Purpose: Autocoder 74 is a symbolic programming system designed to simplify the preparation of programs for the 7070 Data Processing System. With the increased capacity and versatility of data processing systems, machine-language instructions have increased correspondingly in both number and complexity. Coding in machine language today is an extremely tedious and time-consuming task. The 7070 Autocoder 74 is a symbolic programming system designed to permit the programmer to code more easily and with greater meaning than is possible with numerical machine language. Symbolic programming systems also perform automatically many burdensome tasks such as assigning and keeping track of storage locations and checking for errors. Use of these systems will save the programmer a significant amount of valuable programming time and effort.

Autocoder 74 allows the use of 3008 macro-instructions.

Machine Requirements: 4 tape units.

**AUTO-CODER 710**

Abstract:

Purpose: To translate a program written in the Autocoder language including macro statements and/or one-for-one instructions, into an operative machine language program.

Machine Requirements: (Include machine components, special features, storage requirements, control panel-standard or special)

Minimum

1. 5,000 words of core storage
2. 6 IBM 729 model II, IV, VI, or 7310 tape units.
3. Channel 1 or Channels 1 and 2.

Optional

1. IBM 7500 Card Reader (Utility Panel)
2. IBM 7500 Card Punch (Utility Panel)
3. IBM 7400 Printer (Utility Panel)
4. Up to four additional IBM 729 model II, IV, VI, or 7310 tape units
5. 16,000 words of core storage

Capabilities and Limitations:

Autocoder can process any program written for Basic Autocoder or 4-Tape Autocoder. If additional tape units are available, it can process stacked input and/or output. Additional macro generators can be added to the system to allow new input statements. There is great flexibility in entering new loads, patching existing loads, and dropping unneeded loads. Only one macro generator can be added or dropped in a single run.

**COBOL PROCESSOR**

Abstract:

Purpose: The COBOL processor translates a source program written in accordance with the rules specified in the IBM COBOL General Information Manual, form P28-8053-1 into a 7070 or 7074 machine - language program which, when read into the computer, will execute the instructions specified in the source program.

Use of Program: The program is to be used as described in the reference material listed in the accompanying letter with the exception of the following items whose implementation will be deferred:

Procedures Division

1. The CORRESPONDING option of the MOVE verb.
2. The EXAMINE verb (including the TALLY register).
3. Class conditions in conditional statements.
4. Numeric literals as operands of DISPLAY statements.
5. The use of the figurative constant ALL.
6. The ability to optimally round or truncate the results of arithmetic computations. The MOUNT OPTION is standard; truncation is deferred.

(Continued on next page)
IBM Application & Systems Programs Library Abstract  File Number 7070-CX-003

7070 COMMERCIAL TRANSLATOR

Abstract:

Purpose: 7070 Commercial Translator makes available to users of the 7070 a problem oriented-language for the formulation of commercial problems.

Use of Program: The program is to be used as described in the Commercial Translator material listed in the accompanying letter.

Machine Configuration: The 7070 Commercial Translator processor is designed to operate on a 7070 or 7074 of the following configurations:

1. 10,000 words of Core Storage.
2. Input/Output requirements - Seven tapes are required by the system. The input medium for the source program may be one of these seven tapes, an eight tape or a card reader.

IBM Application & Systems Programs Library Abstract  File Number 7070-FG-073

BASIC FORTRAN

Abstract:

Purpose: The IBM BASIC TRANSLATING system, FORTRAN, is an automatic coding system which consists of a source-language (closely resembling the ordinary language of mathematics) and a processor which converts source programs written in the FORTRAN language into machine-language object programs.

Use of Program: FORTRAN is essentially a problem-oriented language designed to facilitate the writing of programs which will perform scientific and engineering type computations. It can also be adopted in the solution of many business problems which can be expressed in a mathematical formula.

Machine Configuration: IBM 7090 Card Reader (Utility Panel) IBM 7090 Card Punch (Utility Panel)

Capabilities and Limitations: Programs may be compiled for any configuration of 7070 equipment. Basic FORTRAN accepts FORTRAN I features in a source program.
Restrictions: Operates in conjunction with 7070 EBCDIC.
Storage Requirements: 400 words + EBCDIC requirements.
Equipment Specifications: 7200 Card Reader and necessary I/O.

IBM Application & Systems Programs Library Abstract File Number 7070-SC-204

**INPUT/OUTPUT CONTROL SYSTEM 7090**

**Abstract:**

**Purpose:** To provide users of the IBM 7090/2's Data Processing Systems with routines for reading and writing card and tape records.

**Use of Program:** The Input/Output Control System is used in conjunction with other programs to provide standardized routines which perform the input and output functions.

**Machine Configuration:**

1. Machines requirements at compile time are dictated by the specifications of the program being used in conjunction with the Input/Output Control System. Reference should be made to the manual or abstract describing these programs.
2. The storage requirements of the Input/Output Control System vary from 750 to 2000 words depending upon the number of files specified and the parameters in the DIOCS statement.

Capabilities and Limitations:

1. The reading and writing of tape records is controlled by the Input/Output Control System and will occur simultaneously with processing.
2. Macro-instructions are provided for processing which will, when required, block and unblock data records that are to be written on, or read from, tape.
3. A program which uses the Input/Output Control System may be interrupted at any time and continued from that point at another time by the use of these macro-instructions.
4. Macro-instructions are provided for processing unit records.
5. Error routines for both tape and unit records are provided.
6. The Input/Output Control System has been designed to allow the using of SPOOL programs with programs using the Input/Output Control System.

**IBM Application & Systems Programs Library Abstract File Number 7070-10-084**

**TAPE FILE GENERATOR FOR TESTING**

**Abstract:**

**Purpose:** The tape files needed to test programs which read input records from tape can be generated from cards using this utility program. Practically any form of tape file can be created with this program.

**Equipment Specifications:** 7200 Card Reader 1 729 Tape Drive

**IBM Application & Systems Programs Library Abstract File Number 7070-2G-026**

**REPORT PROGRAM GENERATOR 7070**

**Abstract:**

**Purpose:** Programs for writing reports from data on magnetic tapes can be created by the programming system through the use of the Report Program Generator.

**Use of Program:** The Report Program Generator acts as a pre-processor to 7070/2's Autocoder. Input consists of the layout of the data tapes, the format of the desired report, and the conditions for inclusion of items of the data.

**Machine Configuration:**

Minimum
1. 5,000 words of core storage
2. 6 IBM 729 Model II, IV, V, VI or 7330 tape units.
3. Channel 1 or Channels 1 and 2.

Optional
1. IBM 7530 Card Reader (Utility Panel)
2. IBM 7530 Card Punch (Utility Panel)
3. IBM 7450 Printer (Utility Panel)
4. Up to four additional IBM 729 Model II, IV, V, VI or 7330 tape units.
5. 10,000 words of core storage

Capabilities and Limitations:

The data file may consist of form 1, 2 or 3 records. The data file records may include no more than 99 fields to be used for the report. A given variable field to be edited may be no more than 10 characters.

**IBM Application & Systems Programs Library Abstract File Number 7070-2G-079**

**SIMULATE 660 ON 7070**

**Abstract:**

**Purpose:** Programs written for the 660 (except 500 Model IV) may be run on an IBM 7070 using this program. The machine configuration of the 7070 system must be the same as a 660 system for the program to be simulated. The simulation program was written for standard 660 systems.

**IBM Application & Systems Programs Library Abstract File Number 7070-3M-077**

**SORT 30**

**Abstract:**

**Purpose:** Tape files containing records from 1 through 999 words in length can be sorted according to a control word that may have from 1 through 100 characters located in from 1 through 12 fields. The tape records may be fixed- or variable-length in single or blocked form. The maximum number of tape records that may be sorted is equal to the number of records which can be contained on 6 full reels of tape.
Equipment Specifications: 4 through 16 magnetic tape units.

Additional Comments: The order of merge of the program depends on the number of tape units available; the order of the merge may be either 2, 3, 4 or 5.

MERGE 91

Abstract:

Purpose: Up to 8 tape files may be merged into one file through the use of this program. The record and control word specifications are the same as for Sort 90. There is no limit on the number of reels that may be required for a file.

Equipment Specifications: From 3 through 26 magnetic tape units are required by Merge 91.

IBM Application & Systems Programs Library Abstract File Number 7070-ML-078

Utilities Programs

Abstract:

Purpose: These programs provide frequently needed routines to assist in the use of the 7000 disk files attached to the 7070. The programs are (1) Clear Disk, (2) Disk-to-Tape, (3) Tape-to-Disk.

Storage Requirements: 1500 positions per program.

Equipment Specifications: 7290 Tape Units

IBM Application & Systems Programs Library Abstract File Number 7070-ML-269

Utilities Programs

Abstract:

Purpose: These utility programs provide frequently needed routines to assist in the testing and operation of the user's 7070 programs. The following are included:

Condensed Card Load Program
Lead Program Relocator
Zero Storage Program
Tape Mark Program
Tape Erase Program
Tape File Generator Program
SNAPSHOT Program
Storage Print Program
Tape Print Program
Branch Trace Program
Tape Duplication Program
Tape Compare Program
Unload Storage Program

Storage Requirements: 7290 Card Reader, 7300 Printer, 7550 Card Punch, Tape drives as needed.

Equipment Specifications: "IBM Application & Systems Programs Library Abstract File Number 7072-UT-081"

Utility Programs for Additional Storage

Abstract:

Purpose: This is a collection of 8 commonly used programs. They are:

Condensed Card Load Program for Additional Storage: This program is designed to load a program which has been punched into cards in condensed form. It will load condensed cards with a maximum of five words in each card into specified locations. Execute cards, i.e., cards containing instructions which are to be executed as soon as they are read, may be included among the condensed cards.

Lead Program Relocator for Additional Storage: This program will allow the user to move the IBM 7072/7074 Condensed Card Load Program for Additional Storage from its current location to any twenty-five consecutive locations below location 9999. It is not necessary to know the current location of the lead program when it is to be relocated.

Zero Storage Program for Additional Storage: This general zeroing program may be used to set core storage to plus zeros regardless of the location of the load program. The Zero Storage Program may be used even though the user does not know the location of the load program.

Tape Mark Program for Additional Storage: This program is used to write a tape mark on a maximum of six tape units connected to any one channel. A separate program, which consists of one card, is required for each channel.

Tape Erase Program for Additional Storage: This program is designed to erase the tape on a maximum of six tape units connected to any one channel. A separate program, which consists of one card, is required for each channel.

Equipment Specifications: 7072/74 with Additional Storage feature.
IBM Application & Systems Programs Library Abstract File Number 7080-CV-000

INT580

Abstract:

Purpose: INT580 enables a program coded for an IBM 705 L, II or III with serial input/output equipment to operate on the IBM 7080, utilizing communication channels and 120 tape units. The 774, 745 L and II, 727, 757, 758, 759 and 724 are simulated in memory. 727, 720A, 730A, 713, 723 and 714 units are simulated on 320 tape units. Restrictions to full simulation are covered in the detailed description of interpretation of each unit, starting at page 10 of the enclosed preliminary manual (see enclosures the addenda, also enclosed) and on page 19 of the manual. These restrictions should not affect most object programs.

Use: INT580 may be loaded into memory once, and left there until that memory is needed for another application. Loading of an object program is initiated after INT580 has been entered and control cards, if necessary, have been processed for that program. The object program is entered in the normal manner and proceeds until an input/output instruction is encountered. The I/O interpret feature of the 7080, working with the Reset step switch causes an automatic interrupt to INT580. where the desired operation is initiated or fully accomplished. Control returns to the object program until the next interrupt. For a detailed description of the various ways to use INT580, see the addenda for version 3 referred to above.

Machine Configuration: The minimum 7080 configuration of 800 memory units and two communication channels is required. The program as written requires the card reader for one control card per object program, but this is easily modified. Drum simulation will require at least 900 memory units of memory. If the second file is used. Four communication channels are required for efficient simulation of simultaneous FW-VR operations on two TRC's.

IBM Application & Systems Programs Library Abstract File Number 7080-CV-006

7080 IOCS

Abstract:

Purpose: To provide the user a complete 7080 Input/Output control system in the form of obtaining two channel and minimal versions of this system.

Use: To use the 7080 IOCS, the first file of the distribution tape should be punched out and a Processor librarian run should be made using these cards. All programs using 7080 IOCS should be assembled from the new system tape.

To obtain the two channel and minimal versions, the third file of the distribution tape should be punched and separated into four decks using the Identify columns 19 to 60 of the cards.

Using the second file of the distribution tape as the reassembly master and the check deck desired as input, a reassembly should be made to obtain a program deck and listing of the desired version.

The deck with Ident IOCSSZ will produce a complete system for two channels.

The deck with Ident IOCSSR will produce a minimal system for four channels. The checkpoint routine may be included by removing the change cards which have a "C" in columns 74.

The deck with Ident IOCSSG will produce a minimal system for two channels. The checkpoint routine may be included by removing the change cards which have a "C" in columns 74.

The deck with Ident IOCSSH and with a "C" in columns 74 will produce an IOCS to run with 770V and 7V tapes. This deck may also be collated by Ident numbers in columns 1 to 5 with any of the three above decks.

The preassembled 7080 IOCS deck may be obtained by punching the fourth file of the distribution tape.

The 7080 IOCS must be in memory at the time of the running of the object program. This may be loaded in one of three ways.

1. The IOCS program deck may be placed in front of the object program deck and loaded as one block.
2. The IOCS program deck may be loaded first and then the object program loaded.
3. The IOCS program deck may be loaded and left in memory during the running of several programs.

If the program decks for the minimal or two channel systems are used, the 09 card produced by the processor should be discarded.

Machine Configuration: The 7080 IOCS complete version for four channels will occupy memory locations 590 to 20,000 with eraseable housekeeping occupying memory locations 20,000 to 25,000. The minimal system for two channels will occupy memory locations 590 to approximately 11,160 for the nonerasable portion. The size of the other versions will fall between these two.

The basic program material accompanying this memorandum includes one reel of tape.

1. The first file of this tape is the complete 7080 IOCS Library.
2. The second file is the reassembly master for IOCS00.
3. The third file consists of 4 change decks.
4. The fourth file is the preassembled IOCS00 deck.
5. The fifth file is the IOCSIO Listing.

Each file is preceded by a standard header and a toepark.

IBM Application & Systems Programs Library Abstract File Number 7080-CV-121

CSMRS

Abstract:

Purpose: CSMRS is a restart program to be used in conjunction with 7080 IOCS. It will restore the machine and tapes to the status at the time of a checkpoint taken during the running of an object program with 7080 IOCS.

Use: The CSMRS program tape must be placed on a program tape, designated to the 7080 IOCS at the time of the running of the object program. This tape will be rewound and unloaded by the checkpoint load control record, any provision should be made to locate and load the restart program from the first record on this tape. CSMRS will be put in the utility section of the SC500 program tape cards and will be loaded automatically if SC500 is indicated to 7080 IOCS.

Machine Configuration: All tapes which were being used by the object program at the time of the taking of the checkpoint must be mounted on the proper units. Also a restart program tape must be on-line. CSMRS will use approximately 80,000 memory positions. If the machine is 1400, the memory positions used will be 0 to 65,000 and 125,000 to 166,000.

IBM Application & Systems Programs Library Abstract File Number 7080-CV-114

IBM SORT 80 FOR 7080 UNDER SUPERVISORY CONTROL: SNOUSC

Abstract:

SORT 80 program specifications and features, operating instructions, etc., are detailed in the reference manual IBM 705 I/7080 Generalized Sorting Program, Sort 80 Version 0 Form C28-6125. All of the operating and modification features of the basic Sort 80 system can be utilized to full advantage with one exception: Memory positions 75000 through 99999 must be reserved for use by SC500 and SNOUSC executive routines.

In accordance with your request, the following Basic Program Material is being forwarded:

1. Two tape files on one reel of Tape at 200 cpi density. The external label reads, "IBM SORT 80 for 7080 Under Supervisory Control: SNOUSC. Program Number 7080-OM-114, Version 1. Modification Level 0." The first file, preceded by a standard IBM header label, contains the SNOUSC program deck, including INCL command and SORTUS cards. This tape can be used as input (Change Tape) to the SC500 Library. The second file is a listing of the SNOUSC executive routines - to be used as a supplement to the basic Sort 80 listings.
2. 7080 Data Processing System Bulletin "IBM SORT 80 for the 7080 Under Supervisory Control: SNOUSC" form 228-6111.
3. INCL command card to be used on a master program tape for unmodified sort applications.
4. INCL 01 command card and dummy 00 TCD cards to be used on the master program tape for modified sort applications.
5. EXEC command card enabling loading of SNOUSC from the common program tape.

Machine Configuration: The 7080 IOCS complete version for four channels will occupy memory locations 590 to 20,000 with eraseable housekeeping occupying memory locations 20,000 to 26,000. The minimal system for two channels will occupy memory locations 590 to approximately 11,160 for the nonerasable portion. The size of the other versions will fall between these two.
IBM Application & Systems Programs Library Abstract  File Number 7080-EL-037

NOSTP

Abstract:

Purpose: The NOSTP macro-instruction and a set of associated subroutines enable 7080 and 7080 programs, running on the 7080, to utilize the non-stop operation feature of that machine. The use of these routines, in conjunction with the non-stop operation feature, will permit continuous operation of the 7080 in automatic status.

Additional Remarks: When the 7080 is running in non-stop mode (i.e., interrupt mode with the non-stop switch set and in set status), the non-stop operation feature permits continuous operation of the 7080 (with the non-stop switch on) and is not tied to the interrupted program. Any condition which would normally cause the 7080 to enter manual status will result in an automatic interrupt to a location specified by interrupt word 510. The conditions which result in this automatic interrupt are:

1. Any halt operation
2. Any condition which turns on one or more of the 0000-0005 check indicators, provided the corresponding switch for these indicators is set to automatic.
3. Any condition which turns on the automatic restart indicator.

When using the NOSTP routines, the location specified by interrupt word 510 would be the entry to these routines, and the automatic interrupt would transfer program control to them.

Equipment Specifications:

7080

IBM Application & Systems Programs Library Abstract  File Number 7080-SV-115

7080 SUPERVISORY CONTROL SYSTEM (7080)

Abstract:

Purpose: To provide the time and effort required to perform the operations for "production" 7080 runs. SC80 will, upon command, locate a program on the program tape, load it into memory, verify the console set-up, and transfer control to the object program.

The program tape (a) used at object time will contain a copy of Memory Print Program (MP7080) at the beginning of each reel. This program has been placed at this location at 7080 user's request to allow them when a production 7080 job encounters trouble.

SC80 will also serve as the 7080 user in holding program file maintenance to a minimum. This is accomplished through the powerful ability to "call in" common programs and/or routines in order to "complete" object programs. Naturally, the common programs and routines need maintenance only on the "source" copy.

Use of Program: SC80 provides:

1. a program library maintenance facility,
2. ability to select "current" programs,
3. an Object Time Routine.

The data to be handled by SC80 is normally supplied by the user and constitutes his programs, interspersed with SC80 command cards. Initially, however, data is being supplied as input to the first run, input to the maintenance program is converted to a memory image program tape for use by the other two phases of the system.

This system will replace the 7080 Basic Supervisory Control System, Program Number 7080-SV-008. That program is obsolete and will not be distributed or maintained in the future. The Preliminary Reference Manual, IBM 7080 Supervisory Control System SC80, dated September 1961, is also obsolete.

Machine Configuration:

A. The Library Maintenance Program

Memory Size - 60K (minimum)
6 IBM 129 Magnetic Tape Units (minimum)
Console Card Reader

B. The Production of a Current Tape

Memory Size - 60K (minimum)
5 IBM 129 Magnetic Tape Units (minimum)
Console Card Reader

C. SC80 Object Time Routine

Memory 80 to 0159
Plus 2019 characters beginning at a 0 or 5 locations above 0000
1 IBM 129 Magnetic Tape Unit (minimum)
Console Card Reader

IBM Application & Systems Programs Library Abstract  File Number 7025-IT-039

7080 UTILITIES

Abstract:

Purpose: This is a collection of eight commonly used utility programs.

Data Assembler (DA7080): The Data Assembler is capable of creating data files from card image records on tape. There is provision for searching the input tape for the correct data set and then processing through to an "END" card. The files created by DA7080 may be of fixed or variable length, blocked or unblocked, multiple or single file and labeled or unlabeled.

Expanded Load Program (EL7080): The expanded load program for the 7080 will be capable of loading a program deck on a primary program tape, loading the program, locating a deck of patch cards on a secondary unit, and loading the patch cards. The expanded load program will occupy the upper 2000 positions of memory and the lower 300 positions. If the input is from tape, the processing will be overlapped by the reading of the next program card.

Expanded Load Program (UL7080): The expanded load program for the 7080 will provide for loading information between memory positions 000140 and 109799 on a 10K 7080 or between 000140 and 079799 on an 8K 7080. Otherwise, this program is the same as EL7080.

Load Program (LP7080): The Load Program for the 7080 will provide for the following functions:

1. Clear Memory from 0140 to the end of memory.
2. Clear the contents of Banks 1, 2, 3, and 4.
3. Set up interrupt words 200, 210, 220, 230, 240, 251, 252, and 253 so as to prevent the machine from hanging following the loading operation due to an unacknowledged interrupt signal.
4. Modify itself to load an object program from any card reader or channel tape.
5. Load an object program into an 8K or a 16K 7080.

Memory Print Program (MP7080): The memory print program for the 7080 will be capable of printing the contents of banks 0 through 5, the settings of the allocation switches, and memory from positions 500 through 156799. Memory areas may be defined as constant, instruction, and/or bit switch areas. The constant and instruction areas will be sorted sequentially so that memory will be printed sequentially by memory position and not by the order of the parameters on the control cards.

Data Print (DP7080): The Data Print program for the 7080 provides for writing records in four output formats. The two options that effect the format are:

1. Indexing - The Indexing option provides for breaking each data record into one hundred or fewer character segments and then printing each set of characters to the line.
2. Referencing - The referencing option provides for two functions.
   a. Additional output information - When the referencing option is used, a line of print will be printed before each tape record is processed. This line of information indicates the tape record number, the actual length of the tape record, and other information which was indicated by the external modification card and/or indicated by certain fields in the tape record.
   b. Record Length Checking - Provides for a length check of each data record and each tape record.

The four formats are:

1. A combination of indexing and referencing.
2. Indexing, but no referencing.
3. Referencing, but no indexing.
4. Neither indexing nor referencing.

Patch Conversion (PC7080): The patch conversion program provides for the use of certain mnemonic operations when an expanded patch card is being punched.

Data Conversion (DC7080): The Data Conversion program will allow the user to take records of any format and convert them to any other format. There is provision for labeling unlabeled files, blocking unlabeled records, reblocking blocked records, unblocking blocked records and providing IBM standards for variable length records on files containing variable-length records. Multiple and/or multi-reel tapes may be created and tapes may be duplicated by DC 7080.
IBM Application & Systems Programs Library Abstract  File Number 7090-CT-911

709/7090 COMMERCIAL TRANSLATOR

Abstract:

Purpose: To facilitate the reduction of time and effort required to program commercial problems by permitting a user to compile programs written in the Commercial Translator language, and to load and execute those programs.

Machine Configuration: The 709/7090 Commercial Translator may be used on a 7094, or on a 7090 equipped with the Data Channel Trap.

The following minimum configuration is required:
1. 32768 words of core storage.
2. One on-line printer.
3. A minimum of 5 tapes:
   a) One system tape.
   b) One listing output tape.
   c) Three utility tapes.
4. One additional tape, or a card reader for tapes.
5. One additional tape, or a punch for punch output.

IBM Application & Systems Programs Library Abstract  File Number 7090-IO-919

7090 I/OCS

Abstract:

Purpose: The I/OCS Version C is designed to relieve programmers of the necessity of writing input and output routines. A programmer can, if he so chooses, think of each file as a continuous string of words. I/OCS will automatically assign tape drives to files giving them the ability to start and stop at any point. Assignment will be on available or reserved tapes as recorded by IBSYS. During processing, I/OCS automatically handles label checking and preparation, blocking and delisting of data words, and overwriting of input and output. Provision is also made for error detection and correction, checkpointing and restart procedures, and tape switching at execution time.

Note that any program which uses I/OCS to control input/output functions must use the system for all its I/O functions, and must not use any input/output routines other than those of I/OCS.

Use: I/OCS Version C is used under the Basic Monitor Operating System. For an example, reference should be made to the 7090 I/OCS Reference Manual, 0C28-4106-2.

Machine Configuration: I/OCS Version C requires at least one tape unit (or the system tape), an on-line printer, and the Data Channel Trap.

IBM Application & Systems Programs Library Abstract  File Number 7090-PR-110

7090/7094 IBFSAP

Abstract:

Purpose: To facilitate assembly, including macro-operation compilation, and symbolic tape maintenance under the Basic Monitor (IBSYS). IBFSAP can be called with the Basic Monitor control card (EXECUTE IBFSAP). This being done, IBFSAP will recognize all cards which are in the format of FAP cards. The exception to this rule is that all IBFSAP control cards must have an asterisk (*) in column seven (7). A special feature of IBFSAP is the pseudo-operation, SIST (Save Symbol Table), which provides the symbolic definition entries most commonly needed by IBMIC and IODE.

Use: IBFSAP is used under the Basic Monitor Operating System. For an example, reference should be made to the Fap Supplement A248-A186.

Machine Configuration: 7090/7094 IBFSAP may be used on a 709 equipped with the Data Channel Trap feature. If the 709 is to be used, the request for the system must state it is going to be used on the 709 and the appropriate system will be sent.

The following minimum configuration is required:
1. 32,768 words of core storage.
2. One on-line printer.
3. One system tape.
4. One tape or a card reader for input.
5. One tape or a card punch for punched output.
6. One tape for printed output.
7. Two tapes for work tapes.

IBFSAP works under IBSYS and thus obtains its tape units from IBSYS.

IBM Application & Systems Programs Library Abstract  File Number 7090-SM-111

700 BASIC MONITOR, IBSYS

Abstract:

Purpose: To facilitate the reduction of time and effort required to perform the inter-system communication thus allowing continuous processing with a minimum of operator intervention. The Basic Monitor can be equipped with just those programming systems desired at a particular installation. The Basic Monitor can coordinate unit assignments and communicate intermediate information between the desired system facilitating continuous operation and reducing set-up time. This will affect a substantial time saving in computer operation, and will allow greater flexibility in programming.

Use of Program: Basic Monitor, IBSYS, provides:

1. An Edit routine to modify, add, and/or delete programming systems to satisfy the requirements of any users.
2. Machine installation assembly parameters need only be specified for the Basic Monitor. This information will be transmitted to each system as required.

3. A Dump routine to record core when the termination of a system's operation becomes necessary because of an error which makes recovery impossible. IBSTBS makes it possible to have system maintenance, assemblies, and selection of next system each passing information as needed to the next system to be executed. IBSTBS control cards are used to obtain the desired results with the minimum of computer time.

A complete set of instructions on the usage of IBSTBS is in the IBM 7090 Basic Monitor Manual A180-A080.

Machine Configuration: The 7090 Basic Monitor may be used on a 7090, or on a 709 equipped with the Data Channel Trap. If the 709 is used, the request for the system must state it is going to be used on the 709 and the appropriate system will be sent.

The following minimum configuration is required:
1. 32,768 words of core storage.
2. One on-line printer.
3. One tape unit.
4. One tape or a card reader for input.
5. One tape or a punch for punched output.
6. Any other requirements are determined by the system which is being monitored by Basic Monitor.

The Basic Monitor has been assembled for the following machine configuration:
1. Channel A has ten tape units, a card reader, a punch, and a printer.
2. Channel B has ten tape units.
3. Channel C has five tape units.
4. Channel D has five tape units.

IBSTBS is initialized with four tapes, a card reader, a punch, and a printer on Channel A, and four tapes on Channel B. Other units may be attached for use by IBSTBS control cards as needed.
305 RAMACODER

Henry L. Coon

Direct Inquiries to: Henry L. Coon
IBM Corporation
300 Church Street
New York 15, New York

Purpose/Description: The RAMACODER system is comprised of three elements:
1. A general purpose process control panel
2. A symbolic language for preparing 305 programs
3. The assembly program which converts symbolic programs into machine language programs.

Method: N/A

Restrictions/Range: N/A

Storage Requirements: N/A

Equipment Specifications: IBM 305 System - The assembly programs require a basic 305 with no special features but can be used to assemble programs for a broad range of 305 configurations.

305 CUT & FILL

Author Unknown

Direct Inquiries to: Author Unknown

Purpose: To perform the calculations involved in the cut and fill problem of highway construction. It may be used to compute either design volumes based on terrain cross sections or payload volumes based on final field slope staking.

Method: Average end areas

Restrictions/Range: Distances - 999.99 feet
Cut and fill volumes - 9,999,999.9 cubic yards

Storage Requirements: Total accumulated cut and fill - 999,999,999

Equipment Specifications: 10 tracks of disk file uses general purpose control panel

Additional Remarks: Timing - 45-70 seconds per station
IBM 650 Library Program Abstracts

**STANOLINK II**
C. E. Stevens
Standard Oil Company (Indiana)
Detroit, Michigan

- **Purpose:** This is a symbolic optimal assembly system comparable to SOAP II which uses numeric symbols. There are two 650 programs included in the system. One edits the symbolic coding and punches error cards for invalid conditions. The other assembles the symbolic coding into an optimally coded absolute program.

- **Range:** Does not apply.
  - **Accuracy:** Does not apply.
  - **Floating/Fixed:** Does not apply.

- **Mathematical Method:** Does not apply.

- **Storage Required:** Both programs occupy most of the drum.
  - **Speed:** The edit program reads at the rate of 180 to 230 cards per minute; punching is intermittent. The assembly program produces single instruction load cards at the rate of 75 to 80 cards per minute at the start and will slow down slightly as assembly proceeds.
  - **Relocatability:** Not relocatable.
  - **Remarks:** This system will accommodate 60 regimes and 600 symbolic addresses. Relocatable absolute or symbolic library programs may be incorporated in the program being assembled. The edit program is used to demonstrate all features of STANOLINK II. Block diagrams and listings of the edit program are included to implement the demonstration. This system will work on any 650 installation. On a 650 with one 533, it will assemble programs for the most elaborate installation.

- **IBM 650 System:** One 533 required.
  - **Special Devices:** None required.

**SOAP-TYPE OPTIMAL ASSEMBLY PROGRAM: STRAP**
L. S. Kessel
Universal Oil Products Company
Des Plaines, Illinois

- **Purpose:** This program is a modification of SOAP II which permits use of 500 general symbols throughout the program, plus an unlimited number of sets of 100 symbols used only in a particular section.

- **Range:** Does not apply.
  - **Accuracy:** Does not apply.
  - **Floating/Fixed:** Does not apply.
  - **Mathematical Method:** Does not apply.

- **Storage Required:** Entire drum and immediate access storage.
  - **Speed:** Not given.
  - **Relocatability:** Not given.

- **Remarks:** None.

- **IBM 650 System:** One 533, 1AS, and indexing registers.
  - **Special Devices:** Group II special character devices are required.
IBM 650 Library Program Abstracts

NO SOAP-G, M. Clemence IBM Washington, D. C.

1. Purpose: NO SOAP is a Numerically-Operated Symbolic-Ortbo-Assembly Program which permits the user of a machine without alphabetic device to do essentially the same things that are done by SOAP II when the alphabetic device is available.

2. Range: Does not apply.


4. Remarks: Has all the features of Basic STANOSPYCE.

Studies: One 533 equipped with a total of 12 coselectors.

IBM 650 Library Program Abstracts

A MODIFIED SOAP II FLOATING POINT PACKAGE FOR THE IBM 650

E. Vernon Griffin

IBM Applied Science

Madison, Wisconsin

1. Purpose: To enable programmers to write programs in SOAP II language as if they had a floating decimal device available, and then assemble them so that they will run on a 650 without the floating decimal device.

2. Range: Does not apply.


4. Remarks: STANOSPYCE is similar to SOAP II in its design and operation; however, only numerical symbolic addresses are used.

5. IBM 650 System: One 533 required.

IBM 650 Library Program Abstracts

STANOSPYCE

Currie E. Serviss

Standard Oil Company (Indiana)

Regional Accounting Office

Detroit, Michigan

1. Purpose: Using the 650 without the alphabetic device, this routine translates English addresses into a symbolic program language.

2. Remarks: This program occupies approximately 1800 drum locations except for worst possible order is 5.57 minutes for 1000 numbers as a check.

Storage Requirements: To any other 50 word block.

IBM 650 System: One 533 required.

IBM 650 Library Program Abstracts

SORTING SUBROUTURE

K. Reid

Navy Cyclotron Laboratory

Irvington, New York

1. Purpose: To sort a block of N numbers in descending order.

2. Restrictions, Range: Any fixed point or floating point numbers.

3. Speed: From 2200 minutes for worst possible order to 1.47 minutes for 1000 numbers as a check.

4. Remarks: Has all the features of Basic SOAP II except that on reading a floating decimal instruction it passes out instructions which automatically create linkages to appropriate subroutines. There are subroutines for each of the seven floating point operation codes. These are relocatable and are automatically assembled into the object program. Note that this is an assembly package and not an interpretive one.

5. IBM 650 System: One 533 equipped with a total of 22 selectors.

IBM 650 Library Program Abstracts

SOAP-TYPE OPTIMAL ASSEMBLY PROGRAM: STRAP 400

Louis S. Rubel

Universal Oil Products Company

Des Plaines, Illinois

1. Purpose: This is a 6000-word modification of SOAP II which permits 100 general symbols used throughout the program, plus an unlimited number of sets of 55 symbols used only in a particular section, and which is substantially faster than SOAP II.

2. Restrictions, Range: Does not apply.


4. Storage Requirements: Entire drum and IAS.

5. Remarks: This output is coded in STANOPIKE II numeric symbols. Using STANOPIKE II, the output may be assembled into an object program.

IBM 650 System: One 533.

(Continued on next column)
IBM 650 Library Program Abstracts

1401 ASSEMBLY ON THE 650 TAPE SYSTEM
Henry La Badie
U.S. Army Ordnance
Frankford Arsenal

a. Purpose: 1401 S. P. S. Assembly on the 650 Tape System
b. Range: None
c. Mathematical Method: None

d. Storage Required: 2000 Words; 150 CPM Input; 10 CPM Output

e. Remarks:
1. Only mnemonic op codes.
2. ComITlenta, DC and DCW cards must have ll-X punch in Col. 75.
3. Above cards must have no invalid 650 punches in Cob. 23.
4. Sign in Col. 23 may not be used with a constant. The units position of the constant may be signed.
5. All other 1401 S.P.S. Rules must be followed for this program.

IBM 650 System:
1. T.E.
2. Set Format
3. 1 Tape Unit
4. Index Registers
5. Both Alpha Devices
6. 12 Pilot Selectors
7. 6 Calculators
8. Rd Side - 2 Digit Selectors (or 1 digit and 1-digit title emitter, if extra pilot sel. available)
9. Pch Side - 1 Digit Selector; 1-1/2 Time Emitter

650 Library Program Abstract File Number 1.2.003

FIVE-PER-CARD LOADING ROUTINE
J. M. Kibbee
IBM, Houston

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 5 words, 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.

650 Library Program Abstract File Number 1.2.004

SIX-PER-CARD LOADING ROUTINE
J. M. Kibbee
IBM, Houston

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 to 1960 are used as the read band. Cards are loaded at 200 per minute.
e) Self-loading.
f) Minimum 650.

650 Library Program Abstract File Number 1.2.006

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.
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.
IBM 650 Library Program Abstracts

LDJ, LOADING ROUTINE

B. T. Wade
Numerical Computation Laboratory
Ohio State University
Columbus, Ohio

a. Purpose: This routine is designed to load either seven words per card or five words per card instruction card format and is used in the Ohio Department of Highways engineering programs. (See classification 9.1.000.)
b. Range: Does not apply.
Accuracy: Does not apply.
Floating/Fixed: Does not apply.
c. Mathematical Method: Does not apply.
Speed: Not given.
Relocatability: Not given.
e. Remarks: The format is the same as most 7/card loaders. This program will load the output of "7/Card Punch," File Number 1.1.018.
f. IBM 650 System: One 533, IAS, and indexing registers are required.

IBM 650 Library Program Abstracts

LAB AND L0B

T. S. Gemmell
Ohio Department of Highways
Columbus, Ohio

a. Purpose: These two routines load the seven words per card instruction card format using any band other than the 1900-1950 band as the location of the loading routine. Key cards indicate the locations of the links. This makes for flexibility in arranging subroutines, replacing subroutines, or adding to the lengths of modular sections of programming.
b. Range: Does not apply.
Accuracy: Does not apply.
Floating/Fixed: Does not apply.
c. Mathematical Method: Does not apply.
d. Storage Required: Requires 36 locations including the read area.
Speed: Cards are loaded at maximum speed.
Relocatability: LAB is relocatable by multiples of fifty.
e. Remarks: The routine’s main feature is its ability to read in and stack modular programming and subroutines. "Links" are set between routines by the loading routine. Key cards indicate the locations of the links. This makes for flexibility in arranging subroutines, replacing subroutines, or adding to the lengths of modular sections of programming.
f. 650 System: One 533 required.
Special Devices: None.

IBM 650 Library Program Abstracts

7/CARD LOADER

L. Zirkle
Computing Center
Oklahoma State University
Stillwater, Oklahoma

a. Purpose: This is a two-card routine which will load into consecutive drum locations up to seven words of data from a standard seven-word load card. Loading begins at the location specified by the control word.
b. Range: Does not apply.
Accuracy: Does not apply.
Floating/Fixed: Does not apply.
c. Mathematical Method: Does not apply.
Speed: Not given.
Relocatability: Not given.
e. Remarks: Will clear one read band for unnecessary blanks. Therefore, if two or more read bands are used, they must be free of blanks.
f. IBM 650 System: 650 with Index Registers.
INDEPENDENT TABLE LOADER

7/16, J. D. Fey
Department of Statistical Services
Elgin Air Force Base, Florida

a. Purpose: Independent Table Loader - loads tables, permits reversion of tables, additions and deletions, expansion and contraction without object program assembly or reassembly.
b. Restrictions: Range: Does not apply.
c. Method: Does not apply.
e. Remarks: Requires specially punched table cards, will sequence check tables as loaded or will not sequence check at discretion of the user.
f. IBM 650 System: Minimum 650.

650 LIBRARY PROGRAM ABSTRACT FILE NUMBER 1.3.002

SEVEN-PER-CARD PUNCH ROUTINE

D. W. Sweezy
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, 1951 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.

650 LIBRARY PROGRAM ABSTRACT FILE NUMBER 1.3.007

STORAGE DUMP

R. Riderman
G. E., Schenectady

January 20, 1956

a) Punches a specified block of storage, 6 words per card.
b) Does not apply.
c) Does not apply.
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 storage may 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.

650 LIBRARY PROGRAM ABSTRACT FILE NUMBER 1.3.008

MEMORY DUMP AND RELOAD ROUTINE

George A. Ruppert
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.

650 LIBRARY PROGRAM ABSTRACT FILE NUMBER 1.3.009

AVAILABILITY

James B. Chappell
IBM, Washington

December 31, 1956

a) Produces a SOAP availability punched 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 1/2% of all constants will improperly be marked as unavailable.
f) Minimum 650.

IBM 650 Library Program Abstracts

FILE NUMBER 1.3.009

7/CARD PUNCH

L. Zirkle
Computing Center
Oklahoma State University
Stillwater, Oklahoma

a. Purpose: This is a flexible, relocatable, 7/card punch routine which uses additional features.
b. Range: Does not apply.
c. Accuracy: Does not apply.
d. Floating/Fixed: Does not apply.
e. Mathematical Method: Does not apply.
f. Storage Required: The program uses storage locations 0000-0051, and punch region 0052-9909.
Speed: Not given.
Relocatability: Relocatable using SOAP II. (Continued on next page)
### SELECTIVE TRACING ROUTINE

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

**Purpose:** This routine traces all instructions, or only those instructions with a minus sign.

- **Range:** Does not apply.
- **Accuracy:** Does not apply.
- **Floating/Fixed:** Does not apply.
- **Relocatability:** Not given.
- **Remarks:** For each instruction traced a card is punched with the location of the instruction, the instruction itself, the contents of the distributor and accumulator, the location of the first instruction to be traced, and in the storage entry switches. A SOAP II symbolic deck listing with a sample absolute deck listing is included in the write-up.

**System:** One 533 and indexing register.

**Special Devices:** Alphabetic device if the SOAP II symbolic version is used.

---

### TRACING ROUTINE

D. W. Hagelbarger  
Bell Telephone Laboratories, Murray Hill, New Jersey  
July 27, 1956

**Purpose:** This routine traces all instructions, or only those instructions with a minus sign.

- **Range:** Does not apply.
- **Accuracy:** Does not apply.
- **Floating/Fixed:** Does not apply.
- **Relocatability:** Not given.
- **Remarks:** For each instruction traced a card is punched with the location of the instruction, the instruction itself, the contents of the distributor and accumulator, the location of the first instruction to be traced, and in the storage entry switches. A SOAP II symbolic deck listing with a sample absolute deck listing is included in the write-up.

**System:** One 533 and indexing register.

**Special Devices:** Alphabetic device if SOAP II symbolic version is used.
IBM Library Program Abstracts

MODIFIED SYMBOLIC TRACING ROUTINE
J. May
Hudson Laboratories
Columbia University
Dobbs Ferry, New York

Purpose: This program is to be assembled by SOAP II, along with an untested program, for use in tracing as a method of debugging. This routine is a modification of "Symbolic Tracing Routine," File Number 1.4.021.

Range: Does not apply.
Accuracy: Does not apply.
Floating/Fixed: Does not apply.
Mathematical Method: Does not apply.
Storage Required: This program requires 57 storage locations, including eight successive words of any punch band.

Speed: Tracing proceeds at the rate of 100 instructions per minute.
Relocatability: Not given.
Remarks: For each instruction traced, a card is punched with the location of the instruction in the instruction itself, the contents of the distributor and accumulators, and the contents of the indexing registers. The location of the first instruction to be traced is set in the Storage Entry switches.

IBM 650 System: One 533 and indexing registers.
Special Devices: Alphabetic device required.

IBM 650 Library Program Abstracts

AUTOSET
M. F. Raw
Federal Bureau of Investigation
Washington 25, D. C.

Purpose: This program will set tapes (either "read" or "write") to a predetermined position. Can be used to set tapes to the position where a partially completed job was halted on a previous run.

Range: Will preset one to six tapes.
Accuracy: Does not apply.
Floating/Fixed: Does not apply.
Mathematical Method: Does not apply.

Speed: Approximately that of tape reading.
Relocatability: May be relocated to any band.
Remarks: Identification of predetermined position on tape may be a tape mark, record number, or any word in a record which is peculiar to that specific record.

IBM 650 System: One 533, tape units, and indexing registers required.
Special Devices: None.

IBM 650 Library Program Abstracts

MULTIPLE PROGRAM DUMP AND LOADER
G. M. Stace
Office Methods & Procedures
Owens-Illinois Glass Co.
Toledo 1, Ohio

Purpose: These routines write any number of programs on a single tape. Any required program can be loaded onto the drum by means of a single load card. A program may be added to the program tape without specifying the last program number on the tape.

Range: Does not apply.
Accuracy: Does not apply.
Floating/Fixed: All routines are fixed.
Mathematical Method: Does not apply.
Storage Required: The maximum storage requirement for any routine is 1000-005 plus the first ten locations of 1AS and a read band.

Speed: Not given.
Relocatability: Not given.
Remarks: These routines will destroy instructions located in 1AS and indexing registers.

IBM 650 System: One 533, tape units and indexing registers are required.
Special Devices: None.

IBM 650 Library Program Abstracts

CROWN LIFE INSURANCE COMPANY SORTING PROGRAM
J. Ballantine
Crown Life Insurance Company
Toronto, Ontario

Purpose: Program to sort ungrouped 650 tape records. Record size and position of the index in the record are located symbolically so that the SOAP program may be assembled to sort any size record from one to fifty words in length. The program retains the sequence of equal indices from the input to the sorted output.

Range: Sorts on a single word index only. Program has two phases. Phase I sorts sorts thirty records and Phase II merges these blocks in multiple passes to complete the sort.
Accuracy: Does not apply.
Floating/Fixed: Does not apply.
Mathematical Method: Does not apply.
Storage Required: Requires bands 0150 to 1950 for the internal block sorting in Phase I, and there are seventy-seven free locations between 0000 and 0049.

Speed: Not given.
Relocatability: Not given.
Remarks: None.

IBM 650 System: One 533, six 727 Magnetic Tape Units, and indexing registers are required.
Special Devices: None.

IBM 650 Library Program Abstracts

SORT II, descending
C. E. Perkins
J. R. Casalasp
National Biscuit Company
New York, New York

Purpose: This routine sorts records in descending order rather than ascending order.
(Continued on next page)
b. Range: Does not apply.
   Accuracy: Does not apply.
   Floating/Fixed: Does not apply.
   Mathematical Method: Does not apply.
   Storage Required: Not given.
   Speed: Not quite as well optimized as SORT II.
   Relocatability: Not given.
   Remarks: The methods are covered in the SORT II Reference Manual (form 328-0415). The "High" and "Low" exits of the original comparison blocks have been interchanged.

f. IBM 650 System: An IBM 650 system with two tape units.
   Special Devices: None.

IBM 650 Library Program Abstracts

TAPE PROGRAM FINDER, WRITER, AND SALVAGE

Mr. Charles Sampson
Kentucky Department of Highways
State Office Building
Frankfort, Kentucky

a. Purpose: These programs are for the purpose of writing any programs (that is in single or 7-per card) on tape, finding the program after it is written on tape and loading it on to the 650, and then transferring the program from one tape to another.

b. Restrictions, Range: Does not apply.
c. Method: Does not apply.
d. Storage Requirements: One band used for Finder Program, four bands used for each of the others. These bands are used momentarily and there is no need for relocation.
e. Remarks: Follow instructions submitted in write-up.
f. IBM 650 System: With 245 and tape.

650 LIBRARY PROGRAM ABSTRACT FILE NUMBER 1.6.006

CLEAR BLOCK TO ZERO

S. Fleming
G. E., Schenectady

a. Clears a specified block of storage to zero.
b. Does not apply.
c. Does not apply.
e. Self-loading. The block limits are punched in the one card deck.
f. Minimum 650.

650 LIBRARY PROGRAM ABSTRACT FILE NUMBER 1.6.007

FIVE-PER-CARD CONDENSING ROUTINE

G. E. Mitchell
IBM, Houston

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. (Continued on next column)
A PROCEDURE FOR USING SOAP WITH A NUMERIC 650

Jack N. Graham
CSAF, 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) SOAP deck is partially converted to 460 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.

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 1956 hand. Running time is approximately read and punch speed.
e) The 1.2.001 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.

IBM 650 Library Program

File no. 1.6.014
ERRATA

"SOAP to Seven," by J. D. Chappell

Under INPUT on page 1 of the write-up, the statement should read as follows:
"... the location in columns 23-26, and the word to be loaded in columns 31-40."

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

R. W. Kleinfeldt
RCA Laboratories
Princeton, New Jersey

a) Designed as an add-on 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 (excluding 0000) by means of the Bell Telephone Laboratories translation routines. 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.

INTERPRETIVE FLOATING DECIMAL ROUTINE

R. B. Haedner
E. L. 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. 5. 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 759, faster than the Trimble routine with no recoding required.
e) None
f) 650 with automatic floating decimal device.

April 1958, Bulletin 18 - 11
IBM 650 Library Program Abstracts  File no. 1.6.021 Utility Programs

DAYS BETWEEN DATES
R. Strauss
IBM, Jacksonville, Florida

a. Purpose: Subroutine to determine the number of days between two dates.
b. Range: Up to the limit of the upper accumulator.
c. Mathematical Method: Does not apply.
d. Storage Required: 40 words plus 10 words for each time the subroutine is used in the program.
e. Remarks: The earliest date must be used as the first date and the most current date as the last date. The date must be six digits and read into the 650 in year, month, and day order. To compute the days between dates in different centuries, the dates must be eight digits and read in the 650 in century, year, month, and day order.
f. 650 System: One 533 required.

IBM 650 Library Program Abstracts  File no. 1.6.022 Utility Programs

FIVE-PER-CARD CONDENSING ROUTINE
J. H. Cooper
E. P. Fraser
T. H. Green
Shell Oil Company
P. O. Box 2527
Houston I, Texas

a. Purpose: Condenses one-per-card instructions of either SOAP I or SOAP II form.
b. Range: Does not apply.
c. Mathematical Method: Does not apply.
d. Storage Required: About 450 storage locations are required for program and storage.
e. Remarks: The entire drum is available to object program since object program instructions, which overlay locations used by the 5/card loader, are automatically saved until last and punched in self-loading 2/card form. The condensed cards are counted when punched and this count is punched in the last card; thus, each time the condensed deck is loaded the count is compared with the original count.
f. 650 System: One 533 required.

IBM 650 Library Program Abstracts  File no. 1.6.023 Utility Programs

MISCELLANEOUS UTILITY ROUTINES
a. Purpose: Six of the seven short utility routines originally published in IBM 650 Bulletin 12 and three contributed routines of a similar nature have been assembled to provide a convenient "package" for installations with an expanded IBM 650 system. The routines included are:

Clear Drum to Zeros between Limits
Clear IAS to Zero between Limits

Clear Drum and IAS to None Zeros
Dump IAS and Drum onto Tape
Load IAS and Drum from Tape
Print IAS and/or Drum
Universal Tape Print
Determine Footage of a Reel of Tape

"SNIP" - Measure Off Predetermined Footage of Tape
b. Range: Does not apply.
c. Mathematical Method: Does not apply.
d. Storage Required: Varies from eight locations to twenty-four depending upon routine used.
e. Remarks: None.
f. 650 System: Most of these routines require one 533 and indexing registers in addition to the equipment specified in the title.

IBM 650 Library Program Abstracts  File no. 1.6.024 Utility Programs

RELOC
E. D. Mounts
National Homes Acceptance Corp.
Lafayette, Indiana

a. Purpose: This program converts single-instruction load cards to four-per-card load cards where other than the 1950 band is used for read-in and relocates the "Four-Per-Card Loader." (File Number 1.2.001) automatically. It will also convert the 1950 band.
b. Range: Does not apply.
c. Mathematical Method: Does not apply.
d. Storage Required: The program uses 170 storage locations from location 1890 to location 1999, excluding the read-in locations 1951 to 1960, punch locations 1977 to 1986, the self-loader locations 1995 to 1999, and the trailer load card location.
e. Remarks: All routines to be converted must reserve locations 45, 46, 47, 48, and 49 (or their equivalents) in the desired read-in band, for self-loader instructions. The routine can be easily altered for other locations. Output is complete and ready for subsequent loading. It is assumed that any program being converted has been used and proved in single instruction load card form. SOAP output decks may be used without disturbing their sequence. The relocated self-loader is punched out in front of the output deck.
f. 650 System: One 533 required.

IBM 650 Library Program Abstracts  File no. 1.6.025 Utility Programs

LOAD DECK GENERATOR
C. L. Stevens
Standard Oil Company (Indiana)
Detroit, Michigan

a. Purpose: This program produces a seven-per-card load deck preceded by a zero clearing routine and a seven-per-card loading routine, for any band of the drum. The program to be punched must first be loaded on the drum. The Load Deck Generator generates the necessary variable instructions so that the zero clearing routine and the seven-per-card loading routine will read into any band specified by the programmer. Many zero locations are not punched, thus reducing the multiple-instruction-per-card deck to minimum size.
**IBM 650 Library Program Abstracts**

**STOP NUMBER DRUM AND IAS**

J. B. Reid
Trans-Canada Air Lines
Montreal Airport
Quebec, Canada

a. **Purpose:** This program loads all drum locations (except 1951-1960) and IAS locations with: 01 aaaa 8880, where aaaa is the address of the location.

b. **Range:** Does not apply.

c. **Accuracy:** Does not apply.

d. **Mathematical Method:** Does not apply.

e. **Storage Required:** Storage locations 1951-1960 and IAS locations 9000-9007.

f. **Speed:** Total of 5.7 seconds for drum and IAS loading with stop codes.

g. **Remarks:** This program is self-zero clearing with self-loading output.

h. **Relocatability:** Not given.

i. **IBM 650 System:** One 533 required.

---

**IBM 650 Library Program Abstracts**

**ON-LINE STORAGE DUMP**

H. R. Vandenburg
Princeton University
Princeton, New Jersey

a. **Purpose:** This program causes a print-out of the contents of the indexing registers, distributor, accumulators, and drum storage.

b. **Range:** Does not apply.

c. **Accuracy:** Does not apply.

d. **Mathematical Method:** Does not apply.

e. **Storage Required:** Locations 1951-1960, 8001-8003, and 8005-8007.

f. **Speed:** Not given.

---

(Continued on next column)
IBM 650 Library Program Abstracts

**Library Program Abstracts**

**IBM 650 Library Program Abstracts**

**FILE NO. 1.6.004**

**Utility Programs**

**RELOCATABLE TO REGIONAL SOAP II**

G. J. Porter
Project Matterhorn
Princeton, New Jersey

a. Purpose: This program converts subroutines written in relocatable SOAP II into standard SOAP II by making the relocatable addresses into regional addresses.

b. Range: Does not apply.

c. Mathematical Method: Does not apply.

d. Storage Required: The program includes the loader occupying locations 1980-1999.

- Speed: Not given.
- Relocatability: Not given.

Special Devices: Requires minor modifications to SOAP II board.

f. IBM 650 System: One 533 required.

Special Devices: Alphabetic device required.

---

**IBM 650 Library Program Abstracts**

**FILE NO. 1.6.035**

**Utility Programs**

**ERL GENERAL UTILITY PROGRAM**

Judy Pogoda
Electronic Research Laboratories
New York, New York

a. Purpose: This program was designed to facilitate the comparison and assimilation of sets of data output from mathematical programs. It is useful for the interpretation of output data and the preparation of data for plotting by hand or machine.

b. Range: Not given.


d. Storage Required: The entire drum is used.

- Speed: Part 1, the rewriter, runs 4 seconds per data card input, when all 8 words of the data card are processed. Part 2 runs 3.25 seconds per data card input, for processing of 9 words.
- Relocatability: Not relocatable.

e. Remarks: All auxiliary routines used are included in the seven-per-card listings and program decks.

f. IBM 650 System: One 533 required.
**IBM 650 Library Program Abstracts**

**650 FORTRAN SYMBOL EQUIVALENCE TABLE**

W. M. Compton
Arabian American Oil Company
New York, N. Y.

a. Purpose: This program automatically prepares SOAP II "EQQ" cards defining the storage locations of each one-subscripted variable and the location of the first 650 instruction compiled for each statement in a 650 FORTRAN source program. This symbol table aids in program error-detection operations.

b. Range: Does not apply.

Accuracy: Does not apply.

Floating/Fixed: Does not apply.

c. Mathematical Method: Does not apply.

d. Storage Required: Does not apply.

Speed: Symbol table punched at the rate of 100 symbols per minute.

Relocatability: None.

e. Remarks: None.

f. IBM 650 System: One 680 and indexing registers.

Special Devices: Group II special character device required.

**IBM 650 Library Program Abstracts**

**FOR TRANSIT SUBROUTINE PACKAGE**

C. W. Zahler
United Steel (Steel Corporation
Pittsburgh, Pennsylvania

W. J. Lee
IBM Corporation
Pittsburgh, Pennsylvania

(Continued on next page)
IBM 650 Library Program Abstracts

AUTOMATIC PERSONAL IDENTIFICATION CODE (AUTO PIC)

Jack Melnick
IBM - Trenton
22 West State Street
Trenton, New Jersey

a. Purpose: To numerically code alphabetic names of individuals and assign unique identifying data to each individual.
b. Range: Not applicable.
d. Storage Required: 1727 words for tables.
e. Remarks: Limits of tables: 768 first names; 959 last names broken into 10 phases of 959 words each.
f. IBM 650 System: One 533 required.

IBM 650 Library Program Abstracts

SWCF SUBROUTINE FOR 650 FORTRAN

David L. Grobstein
Concepts and Applications Laboratory
Piscataway Annex
New Brunswick, New Jersey

a. Purpose: This subroutine makes available to 650 FORTRAN a statement resembling the IF GENEUS SWCHF (IF P_1, X, Y) instruction available in 704-795 FORTRAN.
b. Range: Does not apply.
c. Mathematical Method: Does not apply.
d. Storage Required: 28 drum locations.
e. Remarks: Does not apply.

IBM 650 Library Program Abstracts

UTILITY SUBROUTINES

George Radin
Dental School
New York University College of Engineering
New York, N. Y.

a. Purpose: This package includes subroutines for ABSF, COSF, SINF, ATANF, SQRTF, EXPF, LOGF, ATANF, CLOFF, ELRF, CLOFF.
b. Range: Maximum.
c. Accuracy: Maximum.
d. Floating/Fixed: Floating decimal arithmetic is used.
e. Mathematical Method: Standard iterative techniques are employed.
f. Storage Required: Not given.
g. Remarks: Limits of tables: 768 first names; 959 last names broken into 10 phases of 959 words each.
h. IBM 650 System: One 533 required.

IBM 650 Library Program Abstracts

AUTOMATIC SOAP CONVERSION UTILITY PROGRAM (AUSCP)

TJNC: Robert R. Devery
5750 Hickman Drive
Dayton, Ohio

a. Purpose: The subroutine was the rightmost three Storage Entry Switches on the 650 console to eliminate sense switches, and control program branching.

IBM 650 System: Same as needed for 650 FORTRAN.

IBM 650 Library Program Abstracts
IBM 650 Library Program Abstracts

**BLOCK CORRELATION - COB**

Numerical Computation Laboratory
Ohio State University Research Center
Columbus 15, Ohio

a. Purpose: COB will produce all the correlations for a block of variables which are to be overlaid with themselves or with another block of variables. Results include sums, sums of squares, sums of crossproducts, means, standard deviations, variances, covariances, correlation coefficients, and the square.

b. Range: Not given.

c. Mathematical Method: COB uses the following formula in the computations.

\[ \sum_{i=1}^{n} \left( x_{i} - \bar{x} \right) \left( y_{i} - \bar{y} \right) \]

\[ \sum_{i=1}^{n} x_{i}^2 \]

\[ \sum_{i=1}^{n} y_{i}^2 \]


e. Speed: Time required for accumulation of sum is approximately (in minutes)

\[ \frac{1}{2} \times \frac{1}{2} \times \frac{1}{2} \]

f. Remarks: COB has attached to the front of the fixed desk the loading routine used by the program.

g. IBM 650 System: Basic 650; no special equipment necessary.

**IBM 650 Library Program Abstracts**

**SHIFF**

Richard E. Chandler
Research Computing Center
Florida State University
Tallahassee, Florida

a. Purpose: SHIFF is a FORTRAN I I subroutine designed to shift a fixed point number a desired number of places right or left for both.

b. Restrictions, Range: Fixed point.

c. Method: Does not apply.


e. Remarks: SHIFF operates with the argument (number to be shifted) in the lower. Since the first shift performed is to the right, all digits shifted "off" will be lost.

f. IBM 650 System: Minimum 650 with alphabetic and special character devices.

**IBM 650 Library Program Abstracts**

**TRANSLATOR - OTHER FORMATS TO SOAP - RELOCATABLE (TYPE 2)**

W. H. Lawton
Dr. L. Weimer
Ohio Department of Highways
Columbus 15, Ohio

(Continued on next column)
IBM 650 Library Program Abstracts

SIMULATOR GENERATOR

Q. J. Mahony
North American Life Assurance Co.
Toronto, Ontario, Canada

a. Purpose: Generates so SOAP II input card format a subroutine for use within a program. The subroutine generated, after assembly within a program will simulate in the 533 the operation of a 537 input, output unit to the extent of parsing the input on the input cards.

b. Range: Does not apply.

c. Mathematical Method: Does not apply.

d. Storage Required: (to the generated subroutine) the result storage area used by the subroutine is defined by the input prepared for the generator. (This area should be as large as possible for easy card handling). The subroutine programs is combined with 100 consecutive locations (with a few spaces in the middle).

-e. Remarks: The input to the generator must specify the number of "answer" words needed and the punch words from which they will be available for output. Thus there is considerable flexibility in programs design but the generator analyses the variables and puts out a complete subroutine which is ready to use.

f. IBM 650 System: One 533 required.

Special Devices: Alphabetic device required.

IBM 650 Library Program Abstracts

FORTRAN SCANNING ROUTINE

George Brook
Applied Science Representative
IBM - Tulsa, 1197 E. Boulder Avenue
Tulsa 20, Oklahoma

a. Purpose: This routine will scan a program write in the "650 FORTRAN" language and will examine the program for forty-seven types of errors. These errors fall into three major categories: (a) transcribing and keypunching, (b) violations of system restrictions, (c) logical flow errors.

b. Range: Does not apply.

c. Mathematical Method: Does not apply.

d. Storage Required: 549 locations.

-e. Remarks: Since the "650 FORTRAN" system contains virtually no diagnostic features, the use of FORSCAN should greatly reduce the number of unsuccessful compilations. Machine editing with FORSCAN is considerably faster than the 650 FORTRAN to SOAP phase of the compiling process.

f. IBM 650 System: One 650 required.

Special Devices: Indexing accumulators, special character device, and alphabetic device.

IBM 650 Library Program Abstracts

FORTRAN EDITOR

Jon Pegg
S. Yegowski
IBM Advanced Systems Development
Monterey & Castle Roads
San Jose, California


b. Range: Does not catch all errors.

c. Mathematical Method: Does not apply.

d. Storage Required: Used approximately 100 cards per minute.

e. Remarks: Notes.

f. IBM 650 System: 1AS, 497, Indexing registers, alphabetic device.
**IBM 650 Library Program Abstracts**

**SIR: SOAP INTERPRETIVE ROUTINE**

J. G. Oldfield
W. Hemmeler
IBM, New York

**Purpose:** A relocatable library program which is used with the SOAP system to handle floating decimal interpretive operations.

**Range:** Does not apply.

**Mathematical Method:** Does not apply.

**Accuracy:** Does not apply.

**Storage:** Does not apply.

**Remarks:** Standard 607 accounting machines cannot be programmed to print FORTRAN statement cards or to bring information out of storage on the first line of the first form.

**Equipment specifications:**
1) Standard 607 accounting machines (60 co-selectors, 15 pilot selectors, and 2 digit selectors) allow printing of all card formats mentioned above except FORTRAN statement cards.
2) 607 accounting machines equipped with 16 additional co-selectors, 6 additional pilot selectors, and 1 additional digit selector allow printing of all card formats mentioned above including FORTRAN statement cards.

**IBM 650 Library Program Abstracts**

**SIR: SOAP Interpretive Routine**

J. G. Oldfield and W. Hemmeler

The original SIR write-up has been rewritten by Dr. J. A. Kearns and Mrs. Helga Sharestan, IBM Education Center, New York, to conform to SOAP II. The new report, known as "SIR II" is written as a textbook rather than as a reference manual and is being added to the original write-up as an addendum.

(Continued on next column)
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.

650 LIBRARY PROGRAM ABSTRACT

FILE NUMBER 2.0.065

SPEED CODING SYSTEM

H. M. Sassenfeld
Redstone Arsenal, Huntsville, Alabama

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

b) Does not apply.

c) Does not apply.

d) Storage required is from 600 to 650 locations depending upon how many of the function subroutines are needed.

e) There are 45 possible instructions including mathematical functions, memory, branch, output procedure, three index registers, and optimal 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.

650 LIBRARY PROGRAM ABSTRACT

FILE NUMBER 2.0.096

NINE OPERATION SPLIT INSTRUCTION ROUTINE: NOSIR

L. M. Harvey and J. C. White
August 5, 1956

G. E., Schenectady

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.

650 LIBRARY PROGRAM ABSTRACT

FILE NUMBER 2.0.007

ERCO SPACE SAVER

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

(Continued on next column)
**ERCO FLOATING DECIMAL POINT SUBROUTINES**

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

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

Reuben E. Adams
IBM, Washington

January 4, 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: \(-10^{59} \text{ to } 10^{59}\). 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, analogous to SHR, and all SHR 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.

**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 set 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.

(Continued on next column)
a. Purpose: "Revised Bell Lab Interpretive System": This program is a revision of the Bell Lab Interpretive System (see Technical Newsletter No. 11) to extend its principle to include the use of indexing registers, IAS, and automatic floating decimal arithmetic feature.

"Revised Bell Lab Tape System": This program is a supplement to "Revised Bell Lab Interpretive System." Both systems were assembled separately; thus the program decks are not the same in similar parts.

The tape commands were added to permit the user of the Bell Lab System to have access to tape storage.

b. Range: Will vary depending upon the function being executed.

Accuracy: Will vary depending upon the function being executed.

Floating/Fixed: Floating decimal.

c. Mathematical Method: See the program write-up.

d. Storage Required: "Revised Bell Lab Interpretive System": 819 drum storage locations and 60 IAS locations are required. "Revised Bell Lab Tape System": 986 drum storage locations and 60 IAS locations are required.

Speed: Will vary, depending upon the function being executed.

Relocatability: Not given.

e. Remark: The unused drum storage locations could be used to add more codes to the revised systems.

f. IBM 650 System: "Revised Bell Lab Interpretive System": One 533 required. "Revised Bell Lab Tape System": Same as above plus at least two 727 tape units.

Special Devices: Alphabetic device required if reassembly is desired.

IBM 650 Library Program Abstracts

SIMULATION OF AN INDEXING REGISTER IN SIR

D. Leavenworth
American Machine & Foundry Company
Greenevich, Connecticut

a. Purpose: This program is a modification in SIR ("SOAP Interpretive Routine," File Number 1, 0, 001) to simulate an indexing register.

b. Range: Does not apply.

Accuracy: Does not apply.

Floating/Fixed: Does not apply.

c. Mathematical Method: Does not apply.

d. Storage Required: Requires the modification of 14 SIR instructions. If the function subroutines (Sin-Cos, Log, Exp, Arctan) are not used, this program requires the reservation of only seven storage locations in addition to MAIN SIR.

Speed: Not given.

Relocatability: See File Number 1, 0, 001.

e. Remark: The simulation of an indexing register in SIR is accomplished by providing for two new pseudo-operation and tagging instructions with a negative sign for address modification. The only sacrifice made is the increase of about five to one.

f. IBM 650 System: One 533 required.

Special Devices: Alphabetic device required.

IBM 650 Library Program Abstracts

UNIVERSITY OF HOUSTON ASSEMBLER FOR THE PROCESS ENGINEERING INTERPRETIVE CODING SYSTEM

V. Schrire
E. I. DuPont
University of Houston
Houston, Texas

a. Purpose: This program combines the functions of symbolic assembly with those of the executive routine. For many applications this system possesses greater advantages than either function utilized separately.

(Continued on next column)
IBM 650 Library Program Abstracts

**FLICOR: FLOATING INTERPRETIVE COMPATIBLE OPERATION ROUTINE**

E. L. Schenitzer
L. Rubin
Aeronautical Systems Incorporated

- Purpose: This routine was designed to simulate floating decimal arithmetic and indexing register operations using the IBM 650 basic card machine.

- Programs written for use with this interpretive routine are compatible with programs intended for use with the IBM 650 equipped with floating decimal device and indexing registers, and may be run on such machines by changing only two instructions. In addition to the main routine, a tracing routine for debugging is included, as are a set of certain basic arithmetic subroutines.

- Usage: Does not apply to the main routine. See the program listing for the range of the subroutines.

- Accuracy: Does not apply to the main routine. See the program listing for the accuracy of the subroutines.

- Floating/Fixed: Does not apply.

- Mathematical Method: Does not apply to the main routine. See the methods used in the subroutines.

- Storage Requirements: 675 storage locations. The listing subroutines require the number of storage locations indicated:

  - LOG X - 80
  - SIN X - 80
  - COS X - 80
  - X - 70
  - \(10 X\) - 80
  - ARCTAN X - 80

- Speed: For the main routine, the following approximate speeds are given:

  - Arithmetic operations 45 to 52 ms.
  - LOG, EXP, index operation 10 to 15 ms.

  - For the following subroutines, the approximate speeds are as follows:

    - LOG X - 105 ms.; SIN X - 200 ms.; COS X - 200 ms.
    - X - 105 ms.; \(10 X\) - 219 ms.; ARCTAN X - 240 ms.

- Relocatability: The main routine is relocatable, with some restrictions.

- Remarks: Tagging for address modification is interpreted for the data address portion of the instruction word. The subroutines (arithmetic) mentioned are independent of the main routine, and may be assembled separately.

- IBM 650 System: One 513 required.

**IBM 650 Library Program Abstracts**

**A COMPLETE FLOATING-DECIMAL INTERPRETIVE SYSTEM FOR THE IBM 650 MAGNETIC DRUM CALCULATOR AND IBM IMMEDIATE ACCESS STORAGE UNIT (BELL III)**

Robert L. Farrow, Ph.D.
Biophysics Division
Department of Physiology
Ohio State University
Columbus 10, Ohio

- Purpose: This program is a general purpose scientific and engineering interpretive program. It is designed to replace the original Bell Interpretive System program by running Bell language programs on the IBM 650 equipped with an auxiliary 653 unit.

- Restrictions, Usage: The range of this program is identical to the original Bell Interpretive System program by running Bell language programs on the IBM 650 equipped with an auxiliary 653 unit.

- Method: Subroutines for the transcendental functions are based upon the high speed approximations for digital computers, and in fact are the same as those listed in Bell I except for the calculations of the floating-decimal characteristics.

- Storage Requirements: The system programs uses core addresses 9000 to 9007 and addresses 9008 to 9015 for variable storage as well as drum locations 1500 to 1799. (Note: A separate subroutine is provided to locate some 200 plus unused 7044.)

- Remarks: The Bell II program consists of approximately 50000 instructions with a total of approximately 250000 characters. The Bell III will operate, for a given problem, at least 35 percent faster than Bell I while even greater operating speeds are attainable with extensive programs of use of the previous numerical results. It consists of a Systems Load Program (6 cards), a Systems Dec (673 cards) and Drum Clear (13 cards) in that order.

- Remarks: Precautions:

  1. There is no error stop for zero before floating divide operations. A new interpretive command TR (ZERO) (transfer on zero in 805) has been provided.

  2. For greatest advantage the systems programs have the automatic floating-decimal arithmetic feature of the auxiliary 653 unit. Consequently, the FO-

**IBM 650 Library Program Abstracts**

**ID-3 INTERPRETIVE SYSTEM**

Bonner and Moore Engineering Associates
Shreveport, Texas

- Purpose: This routine is a special interpretive system designed for use in the process industry.

- Restrictions, Range: Does not apply.

- Accuracy: Does not apply.

- Floating/Fixed: Fixed point.

- Method: Does not apply.

- Storage Requirements: 1150 drum locations are available for interpretive instructions.

- Remarks: The ID-3 system is used to provide the executive program for the Bell Operations Simulator. Operations code of ID-3 are of the type that greatly reduces the programming time for the Process Engineer.

- IBM 650 System: Basic 650 is required.

**650 LIBRARY PROGRAM ABSTRACT**

**FILE NUMBER 2.1.001**

**INTERNAL TRANSLATOR (II)**

A COMPILER FOR THE 650

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

- 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.

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

- Does not apply.

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

- 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 600 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, 532 wiring diagrams, the translation program, the modified SOAP I program, reservation and subroutine packages, and sine, cosine, and square root floating point subroutines.

- Alphanumeric device is required.
MODIFICATIONS OF THE INTERNAL TRANSLATOR (IT)
COMPILER FOR USE OF SPECIAL CHARACTERS

J. N. Rogers
C. M. White
GE Vallecros 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.

<table>
<thead>
<tr>
<th>Symbol Name</th>
<th>Representation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Left Parenthesis</td>
<td>(</td>
</tr>
<tr>
<td>Right Parenthesis</td>
<td>)</td>
</tr>
<tr>
<td>Decimal Point</td>
<td>.</td>
</tr>
<tr>
<td>Equality (substitution sense)</td>
<td>=</td>
</tr>
<tr>
<td>Orms</td>
<td>+</td>
</tr>
<tr>
<td>Addition</td>
<td>+</td>
</tr>
<tr>
<td>Division</td>
<td>/</td>
</tr>
<tr>
<td>Negation</td>
<td></td>
</tr>
</tbody>
</table>

A sample statement would appear as below:

\[ Y_2 \equiv ((C13 \times Y_3) - (2.85 + C (12 + 14))) / 5.82 \]

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.


April 1958, Bulletin 18 - 13

IBM 650 Library Program Abstracts

FILE NUMBER 2.1.004

IBM 650 Library Program Abstracts

FILE NUMBER 2.1.004

SPYCE

J. M. McKeever
IBM, Los Angeles, California

a. Purpose: This routine translates English sentences into symbolic program language. The output of this routine may then be assembled using an assembler program of the user's choice.

b. Range: Does not apply.

Accuracy: Does not apply.

Floating/Fixed: Does not apply.

c. Mathematical Method: Does not apply.

d. Storage Required: This routine requires all of drum storage except six locations.

Speed: This routine compiles at punch speed.

Relocatability: Not relocatable.

e. Remarks: By using SPYCE, programming time is greatly reduced and much of the detail effort is eliminated. At any time the programmer may switch from sentence to SOAP mode. SPYCE is applicable to both those commercial and engineering problems which require large volumes of input/output data.

f. IBM 650 System: One 533 required.

Special Devices: Alphabetic device and read half-time emitter are required.

IBM 650 Library Program Abstracts

FILE NUMBER 2.1.006

BUMP, BOSTON UNIVERSITY MATRIX PROGRAM

J. E. Beilby
Boston University
Boston, Massachusetts

a. Purpose: This is an interpretive program which will perform matrix-vector operations automatically, including: add, subtract, multiply, invert, transpose, trace, scale, scalar multiply, as well as internal operations: read, punch, move, store, etc.

b. Range: Minimum size matrix is 10 X 10, without partitioning.

Accuracy: Not given.

Floating/Fixed: Floating decimal arithmetic is used.

c. Mathematical Method: Does not apply.

d. Storage Required: Entire drum is used. 750 locations allocated for instructions, data.

Speed: Not given.

Relocatability: Not relocatable.


f. IBM 650 System: One 533 required.

IBM 650 Library Program Abstracts

FILE NUMBER 2.1.007

GENERALIZED ALGEBRAIC TRANSLATOR (GAT)

B. Acton
S. Graham
University of Michigan
Ann Arbor, Michigan

a. Purpose: This routine translates programs written in conventionally parenthesized algebraic statements into optimized IBM 650 instructions.

b. Range: Does not apply.

(Continued on next page)
IBM 650 Library Program Abstracts

**IBM 650 Library Program Abstracts**

**TLATRAN**

**Frank Doe Wichers**

University of Florida

Tallahassee, Florida

**a. Purpose:** An automatic routing system using a FORTRAN - like language and a modified FORTRAN I control panel.

**b. Restrictions:** Repetitive floating point routines with 8 digit mantissa and 2 digit exponent.

**c. Method:** Does not apply.

**d. Storage Requirement:** One or two passes, depending on optimization desired.

**e. Remarks:** The source program must be correct in every detail.

**f. IBM 650 Program:** 1000 or 4096 word 650 with or without immediate store storage.

**IBM 650 Library Program Abstracts**

**TLATRAN**

**Frank Doe Wichers**

University of Florida

Tallahassee, Florida

**a. Purpose:** An automatic routing system using a FORTRAN - like language and a modified FORTRAN I control panel.

**b. Restrictions:** Repetitive floating point routines with 8 digit mantissa and 2 digit exponent.

**c. Method:** Does not apply.

**d. Storage Requirement:** One or two passes, depending on optimization desired.

**e. Remarks:** The source program must be correct in every detail.

**f. IBM 650 Program:** 1000 or 4096 word 650 with or without immediate store storage.
SQUARE ROOT SUBROUTINE

G. E. Collins
IBM, New York

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 at an even number of places from the extreme left. All 10 digits of 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.

650 LIBRARY PROGRAM ABSTRACT
FILE NUMBER 3.1.601

EXPONENTIAL

S. Fleming
G. E., Schenectady

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

b) Range: -$16.11 < X < 33.0255032$

c) Method not given.

d) Storage required is 10 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) Maximum 650.
LIBRARY PROGRAM ABSTRACT  FILE NUMBER  3.1.005

EXPONENTIAL
S. Fleming  
G. E., Schenectady  
March 28, 1956
a) Computes $e^x$ for single-precision fixed-point number.
b) Range: $-20.5 < x < 20.0$ . Accuracy: error is less than one in the eighth significant digit.
c) Method not given.
d) Storage required is 45 locations, 0000 to 0048; the routine may be translated by an even number of locations.
e) None.
f) Minimum 650.

LIBRARY PROGRAM ABSTRACT  FILE NUMBER  3.1.006

LOG$_{10} A$, IN$_{10} A$
E. B. West and A. O. Garder  
IBM, Houston  
2-30-56
a) Computes $\log_{10} A$ or $\ln_{10} A$ for single-precision fixed-point numbers.
b) Range $10^{-5} < A < 10^{10}$. Accuracy: maximum error is $2 \times 10^{-7}$.
c) Method: polynomial approximation by Hastings.
d) LWA is 0009 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.

LIBRARY PROGRAM ABSTRACT  FILE NUMBER  3.1.007

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: $0 < X < 20$. 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 by an even number of locations.
e) The $e^X$ subroutine is not included in the deck listing.
f) Minimum 650.

LIBRARY PROGRAM ABSTRACT  FILE NUMBER  3.1.008

NATURAL LOGARITHM
S. Fleming  
G. E., Schenectady  
2-30-56
a) Computes $\ln X$ for a single-precision fixed-point number.
b) Range: $10^{-9} < X < 10^9$. 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.

LIBRARY PROGRAM ABSTRACT  FILE NUMBER  3.1.009

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 $X$, $-7.2 < X < 7.2$; for cos $X$, $-8.8 < X < 8.4$. Maximum error is $2 \times 10^{-9}$.
c) Method: 12th power in Taylor series. Reference: Technical Newsletter No. 9, p. 34.
d) LWA is 0039 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.

LIBRARY PROGRAM ABSTRACT  FILE NUMBER  3.1.010

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: $r < 100, \theta < 3\pi$.
c) Method is to use the sin-cos subroutine in Technical Newsletter No. 9, page 29 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.
FLOATING POINT LOG |AI| AND LN |AI|
Prepared by IBM 650 Applied Programming

G. J. Porter
IBM, New York

a) This subroutine computes Log |AI| and Ln |AI| 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: |AI| > 0
Floating Point

Accuracy: Error ≤ 10^-5

Storage: 100 locations

c) Method: A = M × 10^n, where P is an integer
Multiplicities A_i are found so that J - 1 < i < J + 1

Log |AI| is computed by use of a relaxed Taylor series for

\[ \log |AI| = A \times \sum_{i=1}^{J} A_i \]

Finally, Log |AI| = \log |AI| ≤ \frac{1}{2} \sum_{i=1}^{J} A_i

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: 170 m.s.
Ln: 140 m.s.
Relocatable SOAP II cards.

3) Indexing Registers: Indexing register A (8000) 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, 10^A ≤ 100; A ≤ 43; Sinh A and Cosh A: |AI| < 100
Accuracy: Relative accuracy of 10^-5

Floating Point

c) Mathematical methods:
e^A: By several reductions A is reduced to the range 1A < 0.054.
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: 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. 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.
IBM 650 Library Program Abstracts

Arcsin X, Arccos X, Square Root X

V. E. Kohman
Curtiss-Wright Corporation
Propeller Division
Caldwell, New Jersey

a. Purpose: Computes arc sin X, arc cos X, square root X for a single-
precision floating point number.
b. Range: Arcsin / Arccos: X ≤ 1.5 x 10^7
   Square root: Any positive floating point argument.
   Accuracy: Maximum error < 1.5 x 10^-7
   Floating/Fixed: Floating.
c. Mathematical Method: Arcsin
   Square Root: First approximation involving a table look-up followed by
   three iterations with Newton's formula.
d. Storage Required: 140 locations are required.
   Speed: Average running time is 110 ms. for arc sin or arc cos, or
   155 ms. for square root.
   Relocatability: As written, the 0000, 0050 and 0100 bands are used but
   may be relocated an even amount.
e. Remarks: SOAP II symbolic and relocatable decks are included. Error
   stops are provided for a negative argument for square root routine or an
   argument greater than 1 for arc sin / arc cos routine.
   650 System: One 533, automatic floating decimal arithmetic, and indexing
   registers are required.
   Special Devices: Alphabetic device for SOAP II assembly.

IBM 650 Library Program Abstracts

Parabolic Interpolation

A. R. Barton, Jr.
Curtiss-Wright Corporation
Propeller Division
Caldwell, New Jersey

a. Purpose: Interpolate the f(x) value corresponding to a given x value by
   fitting 3 parabolas through 3 given points which define the curve on which
   f(x) lies. All values must be in normalized floating point form.
b. Range: The routine will use any set of numbers supplied.
   Accuracy: The region of the curve under consideration must be parabolic,
   and the axis of symmetry of the assumed parabola must be perpendicular
   to the x-axis for most accurate results.
   Floating/Fixed: Floating.
c. Mathematical Method: The three given points are used to set up 3 simulta-
   neous linear equations. Solution of these equations yields the equation of
   the parabola from which f(x) is calculated.
   Storage Required: 18 locations in 2 adjacent bands plus a previously
   defined region K of 6 words are required.
   Speed: Not given.
   Relocatability: Not given.
   Remarks: There are no error stops. It is left to the programmer to
   determine if a curve of the form f(x) = a x^2 + b x + c is applicable and
   if the unknown f(x) will lie on the curve defined by the 3 given points
   before using this routine.
   650 System: One 533, automatic floating decimal arithmetic, and indexing
   registers are required.
   Special Devices: None.

IBM 650 Library Program Abstracts

WISCONSIN FUNDAMENTAL FLOATING - DECIMAL FUNCTION SUBROUTINES

G. W. Struble
Department of Mathematics
Numerical Analysis Laboratory
University of Wisconsin
Madison, Wisconsin

a. Purpose: This program consists of five subroutines designed to evaluate
   the following functions: e^x, ln x, arc tan x, sin x or cos x and √x, where
   x is expressed in normalized floating decimal form.
b. Range: For subroutine given in (a) above, respectively:
   |x| < 1.1 x 10^7, x ≠ 0, no restriction, |x| (2 x 10^7), a = 0.
   Accuracy: Variable, but in general the result has seven significant
   figures.
   Floating/Fixed: Floating decimal.
   Mathematical Method: The square root subroutine uses a Newton-Raphson
   iteration. All others use relaxed polynomial approximations. The
   methods were chosen primarily to yield subroutines taking little space
   and yet maintaining suitable accuracy and speed.
   Storage Required: For the subroutines given in (a) above, the number of
   storage locations required is, respectively: 41, 57, 48, 56 and 21.
   Speed: For the subroutines given in (a) above, the average computation
   times are, respectively: 95, 147, 175, 156, 130 and 188 milliseconds.
   Relocatability: The program decks are in relocatable SOAP II form, and
   should be relocated an even number of locations to preserve optimization.
   Remarks: Indexing register A is used for e^x and arctan x only, but is
   reset by the subroutine to its contents upon entry.
   IBM 650 System: One 533, indexing registers, and automatic floating
   decimal arithmetic feature are required.
   Special Devices: Alphabetic device is required.

B — 650
IBM 650 Library Program Abstracts

PRIME NUMBER GENERATOR

J. J. Di Giorgio
New York Test Center
New York City

a) Purpose: To generate prime numbers within a given range.

b) Range: 1-2,048,000,000.

Accuracy: Does not apply.

Floating/Fixed: Not given.

c) Storage Required: The program is stored in the first 200 drum locations. A table is created from 0030 upwards, depending on the range of numbers desired.

d) Speed: A function of the range. For example, program execution time for the range 20,000 to 31,000 is ten minutes.

Relocatability: Not given.

e) Remarks: None.

f) IBM 650 System: One 650 required.

IBM 650 Library Program Abstracts

STANDARDIZED UTILITY DECK OF SUBROUTINES (SUDS)

T. A. Wall
Bayard Company
Wayland, Mass.

a) Purpose: Computes sine, cosine, tangent, arctangent, square root, log, natural log, anti-log, anti-natural-log, hyperbolic sine, hyperbolic cosine, and sinh, cosh, tanh.

b) Restrictions: Range: Floating point throughout, angles in radians.

Accuracy generally 7 significant digits or better.

Range: Sine, Cosine, Tangent

<table>
<thead>
<tr>
<th>Argument range</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 ≤ x ≤ 2π</td>
<td>±10^-7</td>
</tr>
</tbody>
</table>

Arctangent, Square Root

<table>
<thead>
<tr>
<th>Argument range</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>x ≤ 10^3</td>
<td>±10^-7</td>
</tr>
</tbody>
</table>

Log, Natural Log, Anti-log, Anti-Natural-log

<table>
<thead>
<tr>
<th>Argument range</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 ≤ x ≤ 50</td>
<td>±10^-7</td>
</tr>
</tbody>
</table>

Hyperbolic Sine, Hyperbolic Cosine

<table>
<thead>
<tr>
<th>Argument range</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>x ≤ 10^3</td>
<td>±10^-7</td>
</tr>
</tbody>
</table>

Series expansions are used.

d) Storage required is 160 locations, 0000 to 0169, and may be translated by an even amount.

e) None.

f) Minimum 650.

650 LIBRARY PROGRAM ABSTRACT

FILE NUMBER 3.2.002

IRREGULAR BESSEL FUNCTIONS

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

a) Computes y_n(x), for n = 0, 1, 2, or 3.

b) Arguments are fixed-point in the form x, x ≤ 1000, and y_n(x) answers are given in both fixed and floating-point form. Range for sin x and cos x is 0 ≤ x < 100; for such x and cos x, n ≤ 5.39; y_n(x) ≤ 6.32; y_n(x) ≤ 6.92; y_n(x) ≤ 2.27; y_n(x) ≤ 1.25. The series is summed until the new term is < 10^-6.

c) Series expansions are used.

d) Storage required is 160 locations, 0000 to 0169, and may be translated by an even amount.

e) None.

f) Minimum 650.

650 LIBRARY PROGRAM ABSTRACT

FILE NUMBER 3.2.003

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, alternatively its complement. It is destined 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 ≤ |x| ≤ 10^50, and x = 0.

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

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

d) Programmed for locations 900-999 (Note: Interpretive system occupies locations 1000-1999.) Additions 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.

Reallocable to any 10 consecutive storage locations in lower memory (excluding location 0000) by means of Bell Telephone Laboratories translation subroutine. Preferably relocated by means of 50 locations, however, in order to preserve optimization in basic language portion of the program.

(Continued on next page)
An Interpretive subroutine for the sine integral and cosine integral functions

R. W. Kingenstein
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. 51.

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

-10^5 ≤ |x| ≤ 10^5, and x = 0.

Maximum error of 1 unit in the eighth significant figure for Si (x) and 3 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 600-699. (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 (excluding location 2000) 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 - 9999999999 (-1)^6 in the Ci (x) output for |x| < 10^-18 as an approximation to minus infinity.

f) Minimum 650.

Bessel Functions Subroutine

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

a) Computes e^x, ln x, √x; I_0(x), K_0(x), J_0(x), and Y_0(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 0500-0999.

e) None.

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

IBM 650 Library Program Abstract

Mathematical Functions

A set of interpretive subroutines for cylindrical and spherical Bessel functions of the first and second kinds and their derivatives

H. E. Kulsrud
RCA Laboratories
Princeton, New Jersey

a) Purpose: Subroutines compute any or all of the Bessel functions J_n(x), Y_n(x), J_n(x), and Y_n(x). This subroutine uses a similar set of subroutines for Bessel functions of the third and fourth kinds.

b) Accuracy: 5 significant figures.

April 1958, Bulletin 18 - 19
IBM 650 Library Program Abstracts

**Mathematical Functions**

**RACA**

Mike Marjory Simmons
University of California
Radiation Laboratory
Berkeley, California

a. **Purpose:** This is a subroutine to compute Clebsch-Gordan coefficients, \( A, B, C \).

b. **Range:** \( 1 \leq A + B + C \leq 125 \). Always the first kind are desired, but must be positive if Clebsch-Gordan coefficients of the second kind are called for.

c. **Mathematical Method:** Based on a recursion method suggested by Stegun and Abramowitz.

d. **Storage Required:** Programs are stored beginning at 0001 and occupy 250 to 300 locations. (Note: The Bell system occupies locations 0000 and 1000-1999)

Accuracy: @8 significant figures.

Floating/Fixed: Floating decimal.

Storage Speed: Not given.

Relocatability: Not applicable. (Continued on next column)

IBM 650 Library Program Abstracts

**FORSCOLEIT**

Arthur Wachowski
Automatic Electric Laboratories, Inc.
600 North Wall Road
Northlake, Illinois

a. **Purpose:** FORSCOLEIT is a modification of Fortran II-3 at the PDP-7 program level, which evaluates Boolean Expressions for constructing truth tables or evaluation of Boolean functions into canonical form. This is accomplished by reinterpreting \( \oplus \) and \( \ominus \) as the Boolean binary operations of "inclusive or" and "and".

b. **Range, Accuracy, Floating/Fixed:** Same as Fortran II-3

c. **Mathematical Method:** Same as Fortran II-3 or as described in program write-up.

d. **Storage Required:** 8 locations.

Accuracy: Not applicable.

(Continued on next column)

**LAPLACE TRANSFORMATION**

J. A. Painter
IBM, Endicott

a) Solve linear differential equations by evaluating the Laplace Transform of the equation. Input is \( \text{X}300 = A300/500 \) which is obtained by taking the transform and solving for \( \text{X}300 \). \( A32 = A \), \( 3 = 3 \), \( B300 = B300 \), \( B = B) \).

b) Floating-point arithmetic is used. \( \pm 2 \leq 6 \).

c) \( B300 \) is factored using Lin's method and \( X300 \) 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
In addition, this program is capable of extracting roots of equations of the degree M, where 6 < M < 25, when the degree and coefficients are properly loaded. To accomplish this, punch 0000002200 where 22 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, 1904, 1905. ... The transfer card should be replaced by these single "instruction" load cards with a new transfer to IBM following.

In either event, the roots are stored at 1811, 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 1904, i.e., the first word of the final card of the load deck should be punched 00001000 instead of 0000104B.

April 1956, Bulletin 18 - 43

IBM 560 Library Program Abstracts

Relaxation Program: Laplace’s Equation in Rectangular Coordinates

D. Dorfman

LYNDA DIVISION OF AVCO MFG. CORP.

Gas Turbine Department

Brantford, Connecticut

a. Purpose: Solves problems for systems that can be represented by the Laplace partial differential equation in rectangular coordinates.

b. Range: An effective field of up to 1500 points can be represented with a limitation of 960 interior points distributed as follows:

1. Up to 50 vertical distances, including boundaries.
2. Up to 30 horizontal distances excluding boundaries.
3. Up to 30 interior points along any of the vertical coordinate strips (including the boundaries).

Accuracy: Can be controlled up to 3 significant digits.

Floating/Fixed: Floating.

Mathematical Method: Finite difference method for unequal spacing, allowing both over-relaxation and under-relaxation.

Storage Required: Full drum storage required.

Speed: Speed is approximately .3 seconds per interior point per iteration.

Relocatability: Not relocatable.

Remarks: Program must be reloaded for each new case.

i. IBM System: One 533, indexing registers, and automatic floating decimal arithmetic are required.

Special Devices: None.

IBM 560 Library Program Abstracts

Elliptic Integrals

R. Pickus

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 < a < 1; 0 < q < 71. k is defined as the modulus and Q is defined as the amplitude of the elliptic integrals.

Magnitude of the parameters are expressed in floating point notation. The two high-order digits determine the location of the decimal point: XXXXXX.YYYYYYYYY, i.e., 0.00000000000001 < x < 0.00000000000002. The modulus q is measured in radians.

The results are accurate to seven decimal digits when the parameters are in the following ranges: 0.0 < k < 1 and 0.0 < q < 14.0067. Outside this range, the accuracy decreases, particularly when both parameters are close to their upper bounds.

c) Degenerate application of Landau’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 (off 1.4) and hence only a few terms are required for the desired accuracy.

d) The total program occupies cells 0000 through 1043. The IBM Basic Floating Point Routine plus the transcendental subroutine min, max, ln, and arcsin are located in cells 0000 through 0777.

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

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

The program may be relocated by a multiple of 50.

e) Locate k in cell 0772. Q in cell 0773. Incomplete elliptic integral of the first kind will be stored in 0774. Complete elliptic integral of the first kind will be stored in 0775. Incomplete elliptic integral of the second kind will be stored in 0776. Complete elliptic integral of the second kind will be stored in 0777.

Final instruction is in 0778.

Input exit command in 0779.

Load and punch routines are included.

g) Minimum 416.
There is a further restriction on a parabolic point near the upper boundary: if a parabolic point occurs near the upper boundary, the point following the parabolic point cannot have as neighbors any points, either to the right or left, that fall on the boundary.

For Example:

- Not allowed as a parabolic point because this point has all interior points as neighbors.
- This is allowed as a parabolic point because the following point has all interior points as neighbors.

IBM 650 Library Program Abstracts

**Differential and Integral Equations**

**RELAXATION PROGRAM: LAPLACE’S EQUATION IN THE CYLINDRICAL COORDINATE SYSTEM**

D. Dorfman

Lycoming Division of AVCO Mfg. Corp.

Gas Turbine Department

Stratford, Connecticut

a. **Purpose:** Solves axisymmetric incompressible flow problems with variables \( r \) (radial distances), and \( z \) (axial distances) only.

b. **Range:** An effective field of up to 1500 points can be represented with a limitation of 900 interior points distributed as follows:

1. Up to 50 radial distances, including boundaries.
2. Up to 30 axial distances excluding boundaries.
3. Up to 10 interior points along any radial coordinate strip (32 including the boundaries).

Accuracy: Can be controlled up to 8 significant digits.

Floating/Fixed: Floating.

Mathematical Method: Finite difference method for unequal spacing, allowing both over-relaxation and under-relaxation.

Storage Required: Full drum storage required.

Speed: Speed is .45 seconds per interior point per iteration.

Relocatability: Not relocatable.

Remarks: Program must be reloaded for each new case.

f. 650 System: One 533, indexing registers, and automatic floating decimal arithmetic are required.

Special Devices: None.

**IBM 650 Library Program Abstracts**

ERRATA/ADDENDA

"Relaxation Program: Laplace’s Equation in the Cylindrical Coordinate System," by D. Dorfman.

The following changes in the deck and listings should be made:

<table>
<thead>
<tr>
<th>Location</th>
<th>In</th>
<th>Should Be</th>
</tr>
</thead>
<tbody>
<tr>
<td>1290</td>
<td>24 1958 1340</td>
<td>24 1958 0154</td>
</tr>
<tr>
<td>1853</td>
<td>24 1955 1803</td>
<td>24 1955 1807</td>
</tr>
<tr>
<td>1903</td>
<td>24 1955 1803</td>
<td>24 1955 1808</td>
</tr>
</tbody>
</table>

The following additions should be made to the program write-up:

Restrictions on types of parabolic points:

Experience in using the relaxation programs dictates that parabolic points should be avoided wherever possible, because account is not taken about points in the neighboring strips, or the proximity of the boundary.

If parabolic points cannot be avoided:

There is a further restriction on a parabolic point near the upper boundary: if a parabolic point occurs near the upper boundary, the point following the parabolic point cannot have as neighbors any points, either to the right or left, that fall on the boundary.

For Example:

- Not allowed as a parabolic point because this point has all interior points as neighbors.
- This is allowed as a parabolic point because the following point has all interior points as neighbors.

IBM 650 Library Program Abstracts

ERRATA/ADDENDA

"Relaxation Program: Poisson’s Equation in Rectangular Coordinates," by D. Dorfman.

The following changes in the deck and listings should be made:

<table>
<thead>
<tr>
<th>Location</th>
<th>In</th>
<th>Should Be</th>
</tr>
</thead>
<tbody>
<tr>
<td>2540</td>
<td>24 1958 0154</td>
<td>24 1958 0154</td>
</tr>
<tr>
<td>1853</td>
<td>24 1955 1803</td>
<td>24 1955 1807</td>
</tr>
<tr>
<td>1903</td>
<td>24 1955 1803</td>
<td>24 1955 1808</td>
</tr>
</tbody>
</table>

(Continued on next column)
The following additions should be made to the program write-up:

Restrictions on types of parabolic points:

Experience in using the relaxation programs dictates that parabolic points should be avoided wherever possible, because account is not taken of points in the neighboring strips, or the proximity of the boundary.

If parabolic points cannot be avoided:

There is a further restriction on a parabolic point near the upper boundary: If a parabolic point occurs near the upper boundary, the point following the parabolic point cannot have any neighbors to the right or left, that fall on the boundary.

For Example:

Not allowed as a parabolic point. This can be eliminated by adding the dotted vertical grid lines or by removing the horizontal grid on which this point lies.

This is allowed as a parabolic point because the following point has all interior points as neighbors.

IBM 650 Library Program Abstracts

NUMERICAL SOLUTION OF LAPLACE, POISSON, AND HEAT FLOW EQUATIONS

J. B. Annable

Jack & Heine, Incorporated

Cleveland, Ohio

a. Purpose: This program will solve partial differential equations such as the Laplace or Poisson which apply to any given two-dimensional region for a field T, where T is known for the boundaries. The field is represented by a grid approximation. T is found for each intersection by a finite difference approximation to the partial differential equation, yields equations of the form:

\[ A T_i + B T_j + C T_k + D T_{i+1} + E T_{i-1} = F + R \]

where \( A \), \( B \), \( C \), \( D \), \( E \), \( F \), and \( R \) are coefficients determined by an analysis of the properties of the matrix \( A, B, C, D, E, F, R \) and using an iterative process. Convergence is controlled by:

\[ \sum_{i=1}^{n} \left| T_{i+1} - T_{i-1} \right| < 10^{-6} \]

where \( m = \) iteration number, \( i = \) point number, \( n = \) number of points and \( 0 < x < 5 \).

d. Storage Required: The entire drum is used, however, locations may be used with a corresponding decrease in the maximum values of \( T \) and \( E \).

Speed: Running time is approximately 4 seconds per point per iteration.

Relocatability: Not given.

c. Remarks: Convergence is not trivial and should be analyzed by a careful study of the problem to be solved. The equation for these parabolic equations is determined by an analysis of the right hand side. Consequently, the error analysis is extremely difficult.

e. Recommendations: None.

f. 440 System: One 513 required.

Special Devices: None.

IBM 650 Library Program Abstracts

SOLUTION OF N SIMULTANEOUS DIFFERENTIAL EQUATIONS

B. B. Hiebert

Savannah River Laboratory

E. I. du Pont de Nemours & Co.

Aiken, South Carolina

a. Purpose: This routine is designed to obtain the solution of a set of ordinary differential equations \( \frac{df}{dx} = A y \), where \( A \) is an \( N \times N \) matrix whose elements can depend upon the time or upon the components of the vector \( y \).

b. Range: \( N \leq 10 \).

c. Remarks: None.

f. 650 System: One 513, automatic floating decimal arithmetic, and indexing registers.

IBM 650 Library Program Abstracts

NUMERICAL SOLUTION OF ORDINARY DIFFERENTIAL EQUATIONS WITH AUTOMATIC ERROR ANALYSIS

N. J. Saber

Computation and Data Processing Center

University of Pittsburgh

Pittsburgh 13, Pennsylvania

d. Storage Required: The BSG routine requires 288 storage locations including printout subroutines. The BSGM routine requires 755 storage locations including printout subroutines.

Speed: Not given.

Relocatability: Not given.

e. Remarks: The changing of mesh size is done automatically under control of the program. The changing of mesh size is done automatically under control of the program.
The printed subroutine used is Compiler Extension 3 and may be used by any other part of the program by making the usual reference.

d. IBM 650 System: One 533, automatic floating decimal arithmetic feature, and indexing registers.
special Devices: Alphanumeric device required.

IBM 650 Library Program Abstracts

NUMERICAL SOLUTION OF DIFFERENTIAL EQUATIONS OF ORDER N

Dennis M. Sweeney
University of Michigan
Willow Run Laboratories
Computational Department
Ann Arbor, Michigan

a. Purpose: This subroutine solves differential equations of order N.
b. Restrictions: Range N = 0.

c. Method: Combined Runge-Kutta Milne method, with an option for
       Range-Kutta solution only.

d. Storage Requirements: 60 locations (N+1)/2, with 100 or less
       storage locations (N+1)N/2, depending on the order of the equation.

e. Remarks: The user specifies the function to be integrated, its order,
       and the initial conditions.

f. IBM 650 System: Uses index registers and floating decimal arithmetic.

650 LIBRARY PROGRAM ABSTRACT

FILE NUMBER 5.1.001

MATRIX INVERSION

A. O. Gardner and J. M. Kloke
IBM, Houston

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 dynam. For a matrix of order n, 0.00004*(n+10) 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.

650 LIBRARY PROGRAM ABSTRACT

FILE NUMBER 5.1.002

SOLUTION OF SIMULTANEOUS LINEAR EQUATIONS

A. O. Gardner
IBM, Houston

a. Solves b systems of simultaneous linear equations with b right-hand sides
   and a common coefficient matrix.
b. Arithmetic is fixed-point form.
c. Method not given.

(Continued on next column)

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

e. It is required that (m+1)/b <1200. The routine is self-loading and self-
   restoring.
f. Minimum 650.

650 LIBRARY PROGRAM ABSTRACT

FILE NUMBER 5.1.003

COMPLEX ARITHMETIC MATRIX INVERSION

Tsun 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.xxx xxxx xxxx.
c. Standard elimination method is used.
d. Storage required for the program is 135 locations, 0000 to 0434. Storage
   for the complex matrix requires 2n^2 locations; working storage in
   locations. Approximate running time is n^9.3 (n + 20) sec.
e. None.

f. Minimum 650.

650 LIBRARY PROGRAM ABSTRACT

FILE NUMBER 5.1.004

MATRIX-VECTOR MULTIPLICATION

J. O. Brown
IBM, New York

July 9, 1956

a. Multiplies a fixed-point, single-precision, square matrix M of order
   n x n 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 + 7.8n+43, 0.75 ms.
e. All elements are treated as fractions and only the high-order half of the
   products are accumulated. Overflow may occur if E = x^2 >900 digits. Absolus
   and SOAP relocatable deck listings are included.

f. Alphanumeric device is relocatable version is used.

IBM 650 Library Program Abstracts

FILE NUMBER 5.1.005

EIGENVALUES OF REAL SYMMETRIC MATRICES BY THE JACOBI METHOD

K. M. Howell
D. J. Hall
Research Computing Center
Indiana University
Bloomington, Indiana

a. Purpose: This program will find the roots and vectors of real symmetric
   matrices.
b. Range: The program consists of three parts:
   Part I which finds all roots and vectors of matrices up to 32 x 32.

(Continued on next page)
Part II which finds all roots only of matrices up to 56 x 56; and

Part III, the eigenvector reassembly of matrices up to 56 x 56. Part III uses rotation output of Part II.

Accuracy: Not given.

Floating/Fixed: Computation is in fixed decimal arithmetic.

c. Mathematical Method: The Jacobi Method is used in these routines.

d. Storage Required: Part I and Part II require all 2040 locations for a maximum size matrix.

Speed: The time requirement for a well conditioned matrix may be computed as follows:

Part I: (2.5 x 10^{-6} n^4 + 4 x 10^{-1} n^2) minutes, where n is the size of the matrix.

Part II: (0.06n^2) minutes, plus punch-out time.

Part III: (0.00n^2) minutes to reassemble vectors from rotation punch-out of Part II.

Relocatability: The program is not relocatable.

e. Remarks: None.

f. IBM 650 System: One 533 required.

Special Devices: None.

---

IBM 650 Library Program Abstracts

** File no. 5.1.207 **

** Matrix Programs **

** PATTERN QUARTIMAX ROTATION OF A FACTOR MATRIX **

Miss Ruth W. Broden
C. E. Holm
Educational Testing Service
Princeton, New Jersey

a. Purpose: This program employs a modification of the quartimax computation for factor rotation. In this modification a hyparachued factor pattern is given to the machine as well as the factor matrix. The machine uses the pattern to select the subset of variables to which it will attend when creating a given phase, in order to find an orthogonal solution which closely fits the hyparachued pattern. The program also provides a measure of the goodness of this fit.

b. Range: The program will handle a matrix up to 999 elements.

Accuracy: Elements are rounded to 6 decimal places.

Floating/Fixed: Fixed decimal arithmetic is used.

c. Mathematical Method: The quartimax method is used for rotation.

d. Storage Required: Locations 0000 to 0999 are used for the program, locations 1000 to 1099 for the factor matrix, and 1099 to 1999 by loading and punching routines.

Speed: Depends on the pattern used. A 6 factor, 35 variable factor matrix with a single rotation pattern required approximately 3-4 minutes per cycle.

Relocatability: Not relocatable.

e. Remarks: None.

f. IBM 650 System: One 533 required.

---

** IBM 650 Library Program Abstracts **

** File no. 5.1.210 **

** Matrix Programs **

** FACTOR ANALYSIS BY THE CENTROID METHOD **

S. G. Navarro
University of Kentucky
Lexington, Kentucky

a. Purpose: This program computes the factors of a symmetric matrix with unknown centroids by assigning each row sum equally to the largest element in each column.

b. Range: Not given.

Accuracy: Not given.

Floating/Fixed: Fixed decimal arithmetic is used.

c. Mathematical Method: The Centroid Method is used. Columns and rows are automatically reflected until all row sums are positive.

d. Storage Required: The entire drum is used.

Speed: The speed of computation depends on the number of reflections needed in the factor, and it is difficult to determine exactly. A good estimate is 4·10^6 minutes/factor.

Relocatability: Not relocatable.

e. Remarks: The program makes use of symmetry to allow factorization of matrices up to 1000.

f. IBM 650 System: One 533 required.

---

** IBM 650 Library Program Abstracts **

** File no. 5.1.208 **

** Matrix Programs **

** MATRIX - VECTOR PRODUCT **

Reverdy Wright
Agricultural Experiment Station
University of Florida
Gainsville, Florida

a. Purpose: To compute the product of the total sum of squares of deviations of observations from their means, appropriate to the n x n individual contributions to the total. To accomplish this, the program of each row, after the first, of an n x n matrix and the n x n single column observations vector is computed and summed. In the development of this method, this sum has been called the Matrix-Vector Product or M·VP.

A square matrix, better called a primary matrix, is provided for each independent variable. From these primary matrices the computer develops the expanded n x n matrix by forming the direct product of these matrices.

b. Restrictions, Range: All computations are done in either single or double precision fixed-point arithmetic.

Method: Sums of squares are obtained to 4 places of decimals in single precision.

c. Storage Requirements: The program is non relocatable, consists of approximately 760 instructions and is reasonably fast in execution.

d. Remarks: Over 100 problems have been successfully run to date, the largest involving a product matrix of order 940.

e. IBM 650 System: The basic IBM 650 computer is required.

---

** IBM 650 Library Program Abstracts **

** File no. 5.1.209 **

** Matrix Programs **

** MAXF **

Richard E. Chandler
Research Computing Center
Florida State University
Tallahassee, Florida

a. Purpose: MAXF is a FORTRANSIT subroutine designed to search a matrix of floating point numbers and to record the location of the numerically largest element. Since MAXF requires this in what is essentially a normalized floating point matrix, any routine doing the same task would have to be faster than any program accomplishing this which operates in floating point.

b. Restrictions, Range: Fixed point.

c. Method: Does not apply.

d. Storage Requirements: 80 locations plus 1455 (entry point) and 1950-1953.

Speed: Dependent on type of matrix. For an M x N matrix, operating time does not exceed 0.02 M, seconds.

e. Remarks: When using matrices in FORTRANSIT, the programmer must reserve locations for the matrix elements with a DIMENSION statement, let A be a matrix of M rows and N columns. Let A* be a submatrix of A of M rows and N columns. Let the first element of A* (A* (1,1)) be in from location L (determined from the DIMENSION statement).

The FORTRANSIT command: MM : MAXF (M, N, N, L)

causes the subroutine to search the submatrix A* for its numerically largest element. It then stores in locations MM a word of the form CC wX wY where wX is L and wY is L + M·N + 1. L, the numerically largest element of A*. MM can be split into 2 more words and can store wY by multiplying and dividing by a proper power of 10 or by using a shift subroutine such as SHIFT (IBM 1.1, 1.3).

(Continued on next column)
650 LIBRARY PROGRAM ABSTRACT

FILE NUMBER 5.2.001

MATRIX INVERSION

D. W. Sweeney
IBM, New York

a) Inverts matrices of order 42 or solves b sets of simultaneous equations for \( n^2 + b \leq 1764 \).
b) Matrix elements are in floating-point form.
c) Method not given.
d) Storage required is 236 locations. 1764 to 1959. The matrix inversion, exclusive of input and output time, is executed approximately \( 67.2 \) 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.

650 LIBRARY PROGRAM ABSTRACT

FILE NUMBER 5.2.002

MATRIX INVERSION BY GAUSSIAN ELIMINATION

A. O. Corder
IBM, Houston

a) Inverts a floating-point matrix of order \( a \) or solves \( b \) systems of simultaneous linear equations with \( b \) constant vectors and a common coefficient matrix of order \( a \).
b) All numbers are of the form \( x + 0000 x \), where \( x \) represents an exponent modulo 50. A matrix of order \( a \leq 500 \) may be handled.
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 \( .0000 b_0 \) hours.
e) Storage limitations require that \( n^2 + (b-1)b \) \( \leq 1850 \). 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 Duru Sweeney which is now self-centering on the drum.
f) Minimum 650.

650 LIBRARY PROGRAM ABSTRACT

FILE NUMBER 5.2.005

COMPLEX AND REAL EIGENVALUES

R. W. De Sio
IBM, Schenectady

a) Determines real and complex eigenvalues for an \( n \times n \) matrix \( A \).
b) Matrix elements are in floating-point form. For large \( n \) (>8) coefficients of small powers in the characteristic equation lose significance.

(Continued on next column)

650 LIBRARY PROGRAM ABSTRACT

FILE NUMBER 5.2.006

D. W. Sweeney

a) Determines real and complex eigenvalues for an \( n \times n \) matrix \( A \).
b) Matrix elements are in floating-point form. For large \( n \) (>8) coefficients of small powers in the characteristic equation lose significance.

(Continued on next column)

650 LIBRARY PROGRAM ABSTRACT

FILE NUMBER 5.2.007

LARGE SCALE MATRIX INVERSION

H. L. Norman
IBM, Washington

a) Computes the inverse of large order matrices.
b) Matrix elements are floating-point of the form \( x + 0000 x \), where \( x \) represents an exponent modulo 50. A matrix of order \( n \leq 500 \) may be handled.
c) The Jordan method is used.
d) Approximately 330 storage locations are used for the program. Time required is \( .0012 (n^+1) \) 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 last elimination step is supplied as input for the \( k+1 \) step. A total of \( n \) passes is necessary.
f) Alphabetic device if SOAP symbolic version is used.

650 LIBRARY PROGRAM ABSTRACT

FILE NUMBER 5.2.008

MATRIX INVERSION

H. L. Norman
IBM, Washington

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 right hand side of a common coefficient matrix of order \( N \), where \( N^2 + (N-1)(N-2) \leq 1600 \).
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 1660-1999. Locations 0000 - \( (N+1)(N+2) \) are used for storage of matrix elements and temporary storage. Loading and punching are at full speed; the calculation requires approximately \( .0012 N(n+6) \) minutes. The program is in relocatable form.
e) A non-load starting card is required for each matrix inverted.
f) Minimum 650.

650 LIBRARY PROGRAM ABSTRACT

FILE NUMBER 5.2.009

DOUBLE PRECISION MATRIX INVERSION

James D. Chappell
IBM, Washington

(Continued on next page)
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 $25 \times 25$ may be inverted and $V$ systems of $N$ equations may be solved where $2 (N+V)(N+V) \leq 1200$.

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

d) Not relocatable, running time is approximately $0.09^2$ seconds.

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

f) Minimum 650.

ERRATA 650 Program Library - File No. 5.2.000

"Double Precision Matrix Inversion," by J. D. Chappell

The following correction should be made in the detailed write-up:

On page 2, in the paragraph headed "Deck Description," the last sentence should read: "The deck consists of 106 cards serially numbered from 091 to 106." The program deck is correct as distributed.

April 1958, Bulletin 18 - 45

650 LIBRARY PROGRAM ABSTRACT FILE NUMBER 5.2.010

SYMMETRIC SIMULTANEOUS LINEAR EQUATIONS

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

a) This program will solve $n$ systems of $n$ simultaneous linear equations consisting of $n$ constant right-hand column vectors with a common symmetric anti-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 $(n+1)(n+1)$ - 1 and the solution uses locations 0000 to $(n+1)(n+1) - 1$. Calculation time is roughly $0.09 \times n$ seconds. Loading and punching are at full speed. The program is not in relocatable form.

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 $(n+1)(n+1) - 1$ and the solution uses locations 0000 to $(n+1)(n+1) - 1$. Calculation time is roughly $0.09 \times n$ 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.

650 LIBRARY PROGRAM ABSTRACT FILE NUMBER 5.2.011

MATRIX INVERSION AND SOLUTION OF SIMULTANEOUS LINEAR EQUATIONS

Prepared by 650 Applied Programming, IBM, New York

D. N. Curt
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) Similar matrices, (non), can be inverted where $n (n + 1) \leq 199, 999$. Rectangular arrays, $(n + m) (n + m)$, can be solved where $(n + 1) (n + m) \leq 199, 999$. 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 program uses locations 1451 to 1999 with the exception of 46 scattered locations. 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 $0.09^2$ seconds. The program contains 32 instructions and 2 constants.

e) The inversion program fails if $n$, $y$ 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.

650 LIBRARY PROGRAM ABSTRACT FILE NUMBER 5.2.012

MATRIX INVERSION ROUTINE 1 (MIR 1)

W. R. 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 $\pm 0.0000 \times Y$ base 10. The matrix is symmetric.

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

d) MIR 1 occupies 70 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 20 $\mu$s milliseconds are required to invert a matrix assuming average times for floating point operations.

e) Location of the matrix on the drum is arbitrary.

f) MIR 1 must be loaded with a loading routine, SLR 2, which is supplied with the program.

650 LIBRARY PROGRAM ABSTRACT FILE NUMBER 5.2.013

SYMMETRICAL MATRIX INVERSION

J. Giblin
Detroit Edison Company
Detroit, Michigan

a) Computes the inverse of a symmetrical matrix up to size $34$ or inverts and solves a rectangular system satisfying the inequality $n + 1 < 286$ where $b$ is number of $b$ vectors, with $1900$ load 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.

(Continued on next page)
d) Speed is in 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.

c) 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.

650 LIBRARY PROGRAM ABSTRACT

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 ≤ 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, \( n^2 + \frac{n(n+1)}{2} \) storage locations are needed to put the matrix in memory.

e) None

f) Minimum 650.

April 1958, Bulletin 18 - 23

650 LIBRARY PROGRAM ABSTRACT

MATRIX INVERSION

J. C. English
F. K. Townsend
E. L. 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 a \( n^2 \) order matrix is about 0.020 sec.

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.

April 1958, Bulletin 18 - 25

650 LIBRARY PROGRAM ABSTRACT

LATENT ROOTS AND VECTORS OF A MATRIX

W. Granet
Boston University
Boston, Massachusetts

(Continued on next column)
GENERAL SIMULTANEOUS EQUATIONS SOLUTION
J. H. Schenck
Curta~ Wright Corporation
Propeller Division
Caldwell, New Jersey

a. Purpose: This program solves a series of inhomogeneous simultaneous equations in floating-point single-precision arithmetic.

b. Range: A maximum of 40 equations may be solved.

Accuracy: Accuracy of solution is indicated by residuals calculated from the check row of the equation matrix according to Crout's method.
Floating/Fixed: Floating.

Mathematical Method: Crout's method.

Storage Required: Requires all of drum, but about 200 locations may be used to develop equations before solution instructions are entered, or most of drum may be used to operate on solution after obtained.
Speed: Speed varies from approximately 30 minutes for 40 equations to about 2 minutes for 4 equations.

Relocatability: Program is relocatable.

Remarks: None.

IBM 650 Library Program Abstracts
Filen. 5.2.019 Matrix Programs

IBM 650 Library Program Abstracts
Filen. 5.2.020 Matrix Programs

LINEAR EQUATION SOLVER
G. Pullay
J. Gillespie
J. W. Hamblen
Computing Center
Oklahoma State University
Stillwater, Oklahoma

a. Purpose: To obtain the solutions for many small systems of linear equations. Also, to evaluate the determinants of the coefficient matrices.

b. Range: The program handles systems in 2, 3, 4 or 5 unknowns.

Accuracy: Not given.
Floating/Fixed: Floating decimal.

Mathematical Method: Cholesky's scheme is used.

Storage Required: The program uses storage locations 1900-1700; the data uses IAS locations 9011-9059.

Speed: Approximately 0.6 seconds where n = the number of unknowns.

Relocatability: Not given.

Remarks: None.

f. IBM 650 System: One 533, automatic floating decimal arithmetic, and indexing registers are required.

Special Devices: None.

IBM 650 Library Program Abstracts
Filen. 5.2.021 Matrix Programs

SOLUTION OF SYSTEMS OF SIMULTANEOUS LINEAR EQUATIONS
T. R. Jackson
Ford Motor Company
21500 Oakwood Boulevard
Dearborn, Michigan

a. Purpose: This program solves systems of simultaneous linear equations of 10th order or less using the largest pivot elements. The inverse is computed and may be punched out.

b. Range: Up to 39 equations in 39 unknowns.

Accuracy: Matrix elements are ten-digit floating decimal numbers.
Floating/Fixed: Floating decimal.

Mathematical Method: The Gauss-Jordan elimination method is used.

Storage Required: Requires all of drum, but about 200 locations may be used to develop equations before solution instructions are entered, or most of drum may be used to operate on solution after obtained.

Speed: The time required for the inversion process is approximately 0.0067 seconds, where n is the order of the system.

Relocatability: Not relocatable.

Remarks: A matrix check program is included.

f. IBM 650 System: One 533, indexing registers, and automatic floating decimal arithmetic feature.

IBM 650 Library Program Abstracts
Filen. 5.2.012 Matrix Programs

MATRIX INVERSION WITH ITERATIVE IMPROVEMENT OF ACCURACY
L. G. Dean
R. H. Higgins
Development Department
Union Carbide Chemical Company
South Charleston, West Virginia

a. Purpose: This program performs matrix inversion by modified Gaussian elimination, considers the inverse as a first approximation and then multiplies the round-off errors inherent in the initial inverse by means of an iterative technique.

b. Range: This routine will handle square arrays up to the 22nd order.

Accuracy: Iterations continue until the sum of squares of the elements in the approximate "zero" matrix (the identity matrix with unity subtracted from each diagonal element) ceases to decrease.
Floating/Fixed: The matrix elements are entered in fixed point form. The calculation is in floating decimal arithmetic. The output is punched in either floating or fixed decimal form, according to the setting of the Storage Entry Sign switch.

Mathematical Method: The following method is used for the iterative improvement of the inverse:

\[ A_{k+1} = I + \frac{1}{A_k} - \frac{A_k}{1 + A_k} \]

where \( A \) is the original matrix; \( A_k \) is the kth approximation of the inverse; \( I \) is the unit or identity matrix.

d. Storage Required: Not given.

Speed: The inversion time, excluding input, is approximately 0.0067 seconds. The calculation time for the improvement iterations is approximately 0.0039 seconds per iteration.

Relocatability: Not relocatable.

Remarks: The program is loaded in two decks - the inversion routine and the iterative improvement routine. The latter deck loads the original matrix and the iterative improvement routine. The former deck loads the original matrix for each iteration. The latter deck is reloaded for each iteration. The iterative improvement routine requires that the original matrix be reread for each iteration. Iterations continue as given under Accuracy above. At this point the sum of the squares of the "zero" elements, the approximate identity matrix, and the final inverse matrix are punched.

f. IBM 650 System: One 533, IAS, and automatic floating decimal arithmetic feature are required.

IBM 650 Library Program Abstracts
Filen. 5.2.013 Matrix Programs

MOLECULAR SPECTROSCOPY
MULTIPLICATION OF MATRICES
George J. Just
Yuki Mitaka
Department of Chemistry
Rensselaer Polytechnic Institute
Troy, N. Y.

(Multiplication of matrices)
IBM 650 Library Program Abstracts

MOLECULAR SPECTROSCOPY
LATENT ROOTS AND VECTORS OF A MATRIX

George J. Zassen
Burbank, California

a) Purpose: Computes the latent roots and vectors of a symmetric matrix.
b) Restrictions: The matrix must be symmetric.
c) Method: Gaussian elimination is used to find the eigenvalues.
d) Storage Requirements: A storage of 650 to 654 is required.
e) Remarks: The program can handle matrices up to order 30.

f) IBM 650 System: IBM 650.

IBM 650 Library Program Abstracts

TO OBTAIN THE EIGENVALUES AND EIGENVECTORS OF A MATRIX

William Grant
Computing Center
Oklahoma State University
Stillwater, Oklahoma

a) Purpose: Computes the eigenvalues and eigenvectors of a general matrix.
b) Restrictions: The matrix must not be singular.
c) Method: Uses the QR algorithm.
d) Storage Requirements: Requires 650 to 654 storage units.
e) Remarks: The program is efficient for large matrices.

f) IBM 650 System: IBM 650.

650 LIBRARY PROGRAM ABSTRACT

FILE NUMBER 6.0.001

SIMPLE CORRELATION COEFFICIENTS

R. Bird and R. Brokate
IBM, New York

a) Purpose: Computes the Pearson correlation coefficient.
b) Restrictions: The data must be normally distributed.
c) Method: Uses a formula to calculate the correlation coefficient.
d) Storage Requirements: Requires 650 to 654 storage units.
e) Remarks: The program is useful for analyzing data sets.

f) IBM 650 System: IBM 650.
CORRELATION COEFFICIENT ROUTINE

J. W. Robinson, III
IBM, Endicott

a) Computes the means, standard deviations, and product moment correlation coefficients of n x 20 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 8012 to 1274 are used; for n = 50, the entire drum is used. Approximate time for 1000 observations is 8 min. for n = 10; 23 min. for n = 20; 71 min. for n = 30; 125 min. for n = 40; 193 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, Houston

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 16,500 individual digits in all data cells.
c) Does not apply.
d) Storage required is approximately 341 locations, and eight levels per factor, one observation per cell, and a total of 350 observations for each variable along with the values of the cross-correlation function for up to 20 variables. 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

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 xxxx.
c) The standard formulas are used.
d) Storage required for the program and load routine is 301 locations 0009 to 0019; data locations are 0050 to 1999. Timing is $t = 0.006$ seconds where n is the total number of data elements.
(Continued on next column)

MULTIPLE CORRELATION FOR 50 VARIABLES

J. D. Hall
University of Indiana, Bloomington

a) Determines all possible correlations (1225) of 50 variables of 2 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.
WEIGHTED LEAST SQUARE POLYNOMIAL APPROXIMATION

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

1) 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.

2) Limits for the least squares fit: \( 1 \leq n \leq 22 \), \( 2 \leq m \leq 512 \). Also \( m = 2n \) \( \leq 128 \) and \( m \geq n + 1 \). Limits for a system of equations: \( 3 \leq n \leq 22 \). Calculations are in floating-point.

3) An iterative method is used.

4) Storage required for the program is 750 locations 0000 to 0740; the rest of the drum is used to store data. Speed estimates not given.

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

6) Minimum 650.

IBM 650 Library Program


The following revised errata sheet, which replaces that published in IBM 650 Bulletin 15, has been received from one of the original contributors. The following revisions are to be made:

- Page 24: Change the following to read
- Page 25:
- Page 26: Change the following to read
- Page 27: Box 12 of the flow diagram should be located following box 13.

ERRATA

- Page 41:
- Page 42: The following instructions are missing at the bottom of the page.
- Page 43: The following sentence has been omitted from the top of the page:

Multiply row (2) by \( \frac{1}{N} \) and subtract from row (3).

New Row (3) = \[ 0.02 -0.01 -0.02 -0.01 0.00 \].

- Page 46: Decks supplied on or after May 1, 1958 include the appropriate changes shown above.

IBM 650 Library Program

FILE NUMBER 6.0.009

ABSTRACT

The memory required to store the matrix being solved must be less than or equal to the number of locations.

- Page 54:

ERRATA

- Page 58:

ERRATA

IBM 650 Library Program

FILE NUMBER 6.0.010

POLY; POLYNOMIAL FIT BY LEAST SQUARES

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

- Page 61:

ERRATA

- Page 66:

ERRATA

IBM 650 Library Program

FILE NUMBER 6.0.011

SINE FIT

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

- Page 71:

ERRATA

- Page 76:

ERRATA

IBM 650 Library Program

FILE NUMBER 6.0.012

ERRATA
AUTOCORRELATION AND POWER SPECTRUM

Ezor Muso 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.

CHI SQUARE FOR UP TO 10X10 CONTINGENCY TABLE

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

a) This routine computes Chi square 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) 126 locations are needed. Available in SOAP and/or absolute.
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
December 31, 1956

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, 6500 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: limit, mean, variance, e.g., e.g., random number, negative, gamma function, normal probability, Poisson probability, binomial probability, cumulative Poisson, cumulative binomial, X 2 test, t test, F test, clear, store loop box, restore loop box, general exponentiation, and two statistical read commands.
f) Minimum 650.
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.

IBM 650 Library Program Abstracts

FACTOR ANALYSIS

C. W. Harris, Dep. of Education
W. H. Peirce, Numerical Analysis Laboratory
University of Wisconsin
Madison, Wisconsin

a. Purpose: Using an n x n (symmetric) correlation matrix with 1's in the main diagonal the program produces a maximum likelihood solution under the assumption of random sampling from a multivariate normal population. It provides a method of converging by iteration the initial estimates of the unique variances and provides a test of significance for the residuals after the extraction of any given number of common factors.

b. Range: Maximum matrix size, 10 x 10.

Accuracy: Not given.

Floating/Fixed: Computation is in fixed point.


d. Storage Required: Practically the entire drum is required.

Speed: Exact timing information is not available, since it depends on the number of iterations necessary for convergence. One 10 x 10 matrix which was processed took 14 hours to meet the conditions of the Lawley test.

Relocatability: Not given.

c. Remarks: The number of iterations and hence the total time required may be reduced considerably by applying a less stringent significance test.

f. 650 System: One 533 required.

Special Devices: None.

IBM 650 Library Program Abstracts

CURVE AND SURFACE FITTING ON EQUALLY OR UNEQUALLY SPACED POINTS

C. Hobby
A. Newhouse
L. Giesel
Computing and Data Processing Center
University of Houston
Houston, Texas

a. Purpose: Fits a polynomial to the given data. By repeated use it will fit a polynomial to a function in several variables.

b. Range: The number of points allowed varies with the degree n 5 10 of the polynomial, e.g., for n = 2 or 3, m 5 10; n = 10, m 5 25.

Accuracy: Not given.

Floating/Fixed: Calculations are in floating point.


d. Storage Required: The entire drum is used.

Speed: Not given.

Relocatability: Not relocatable.

e. Remarks: The program consists of three decks:

Deck 1: Determines a set of polynomials orthogonal on the given set of (equally or unequally spaced) points.

Deck 2: Uses these polynomials to fit the data in the least square sense.

Deck 3: Will compute the accuracy of fit and/or compute the values of the function for intermediate points.

The program is written in SOAP I and SIR.

f. 650 System: One 533 required.

Special Devices: None.

ERRATA

IBM 650 Library Program Abstracts

"An Integrated Set of Programs for Curve and Surface Fitting on Equally or Unequally Spaced Points," by C. Hobby, A. Newhouse, and L. Giesel.

The following corrections and additions have been submitted:

1. For the write-up:

On page 7, line 8, the equation should read:

A = \sum_{i=1}^{n} X_i.

In the original the right side of the equation was positive, in error.

On page 21, for Word 5 the line should read:

Z - B

Word 5 Col 41-50 Number B for Z = B, in floating point form if this option is selected.

The underscored phrase has been added.

On page 21, Note 3 should be corrected to read: If ----, then option 4 in Deck 2 cannot ----, etc., etc.

On page 22, correct the Col numbers as follows:

Word 7 Col 61-62 Decimals point ----, etc.

Word 8 Col 63-64 Zeros

2. In the program and listings, page 60:

Correct card number 432 to read:

432 STR4A LDD CON26

Insert the following between card numbers 432 and 433:

<table>
<thead>
<tr>
<th>01540</th>
<th>24</th>
<th>1457</th>
<th>1560</th>
</tr>
</thead>
<tbody>
<tr>
<td>01550</td>
<td>24</td>
<td>1563</td>
<td>1566</td>
</tr>
<tr>
<td>01560</td>
<td>24</td>
<td>1471</td>
<td></td>
</tr>
<tr>
<td>01570</td>
<td>24</td>
<td>1472</td>
<td></td>
</tr>
<tr>
<td>01580</td>
<td>24</td>
<td>1473</td>
<td></td>
</tr>
</tbody>
</table>

(Continued on next page)
MULTI-VARIABLE CORRELATION

R. Glaser
J. Taylor
General Electric Co.
Utica, New York

a. Purpose: Multi-variable Correlation Program computes the correlation of up to five variables simultaneously, one dependent and four independent from an $n^2$ order matrix of simple correlation coefficients.

b. Range: The order of the matrix $n \leq 13$.


d. Storage Required: Not given.

e. Remarks: The program is self-restoring, hence may be used to obtain highest square or correlation matrix in a manner such that it can be loaded with MA DIV III (IBM 5.0.01). B. N. Carr. (Continued on next page)
THE WHERRY-WINER METHOD

H. B. Bremer
Miss Frances Dallow
The Standard Oil Company of Ohio
Cleveland 15, Ohio

a. Purpose: This routine presents a method of analyzing variables on the basis of their inter-correlations to determine whether the variations represented can be accounted for adequately by a number of basic categories smaller than the number initially considered.
b. Range: Not given.
Accuracy: Not given.
Floating/Fixed: Fixed point arithmetic is used.
c. Mathematical Method: An iterative procedure is used for stabilizing commonalities.
d. Storage Required: Part 1 (containing observations' subset scores and correlation between subsets) requires approximately 2,000 locations.
Part 2 (containing item-subset correlations) requires 850 locations.
Part 3 (calculating projections on group centroid vectors) requires 100 locations.
Part 4 (an alternative procedure to obtain summation of inter-item correlations for each subset which failed to converge) requires 1,100 locations.
The number of passes through Part 2 equals the number of subsets.
Speed: Not given.
Relocatability: Not given.
e. Remarks: Maximum number of subsets 15
Maximum number of items in a subset 19
Maximum number of observations 300
f. IBM 650 System: One 533 required.

MULTIPLE REGRESSION ANALYSIS PROGRAMS: RAP; RAPA; TRAP

J. E. Nichols
Houston Research Laboratory
Shell Oil Company
Houston 1, Texas

a. Purpose: Three versions of the same regression analysis program, modified for use on different equipment, are included in this write-up.
RAP is for the basic 650 with the alphabetic device.
RAPA is for the 650 with the alphabetic device, IAS indexing registers, automatic floating decimal arithmetic, and on-line 533.
TRAP is for the 650 equipped as for RAPA, plus one 727 tape unit.

These programs offer improvements over the previous regression analysis program, File Number 6.0.018, in many important respects. Multiple transformation of variables as the data is entered permits more flexibility in the form of equations used. The programs also provide for the following:
1. Additional output, some of which is optional.
2. Error detection and correction features which check on the form of the data and of the equation.
3. An option to force the curve through the origin when certain physical situations require this.

Several modifications to the program logic have been made which reduce computing time.
b. Range: Data is entered as positive and/or negative four-digit floating decimal numbers. The programs provide for the entry of 36 variables and up to 999 observations. Nine dependent variables can be correlated in one pass in the RAPA and TRAP programs, while eight is the maximum number in the RAP program. The regression equation to be fitted may contain a maximum of 26 terms and dependent variables. Each term may be the product of up to five transformed variables, all raised to various powers ranging from 0.1 to 9.9. Variable transformations are done by means of codes and constants. The programs provide for thirty-two constants and thirty-two codes.

(Continued on next page)
Computing Center

Purpose: This program completes a multiple regression analysis and Mrs. Emma E. lulo

MULTIPLE REGRESSION ANALYSIS

IBM 650 Library Program Abstracts

FILE NO. 6.0.030

Accuracy: Not given.
Floating/Fixed: Floating decimal arithmetic is used. RAPA and TRAP utilize the automatic floating decimal arithmetic feature, and RAP uses the programmed floating decimal arithmetic in SIR.

Mathematical Method: Conventional least squares techniques are used, matrix inversion is done by Gaussian elimination.

Storage Required: Each program is divided into two parts. See the availability tables of each part in the program write-up for the storage requirements. The output from TRAP, Part I, is stored on magnetic tape, and the output from RAPA and RAP, Part I, is punched into cards. The output from Part I in any case is the input for Part II.

Speed: The speed of each program is a function of equation size, the number of observations, the number and type of transformations of the variables.

Relocatability: Not given.

Remarks: TRAP output contains the following:
1. Original least squares matrix
2. Inverse least squares matrix
3. A set of constants and coefficients for each dependent variable.
4. Total variation
5. Variation by regression
6. Correlation coefficient
7. Error variance and standard deviation
8. "F" and "T" test values for each term.
9. Table of residuals for each observed and calculated dependent variable.
10. Sum of residuals squared
11. Chi-square test values
12. Variance check to indicate round-off errors, if any.

RAPA and RAP outputs do not contain items 5, 10, 11 and 12.

RAP is further limited by not containing items 4, 5 and 6 in the above list.

IBM 650 System:
1. For RAP: One 533 and the alphabetic device.
2. For RAPA: One 533, indexing registers, SAS, indexing registers, automatic floating decimal arithmetic feature, and an on-line 4371.
3. For TRAP: Same system as for RAPA plus one 727 tape unit.

IBM 650 Library Program Abstracts

FILE NO. 6.0.031

Accuracy: Not given.
Floating/Fixed: Floating decimal.

Mathematical Method: The inverting part of the routine is that of D. W. Sweeney, Gaussian elimination is performed.

Storage Required: The entire drum is used for a 25 x 25 matrix.

Speed: The inversion, exclusive of input and output times, requires approximately 6.072n2 seconds, where n is the order of the matrix.

Relocatability: Not relocatable.

Remarks: None.

IBM 650 System: One 533 required.
Special Devices: Alphabetic device required.

IBM 650 Library Program Abstracts

FILE NO. 6.0.032

10 x 90 CORRELATION COEFFICIENTS

J. E. Farmer
Computing Center
State College of Washington
Pullman, Washington

Purpose: This program provides simple correlation coefficients and related data for up to ten dependent variables correlated with up to 90 independent variables.

Range: Maximum number of observations is 9999. Maximum size of any single variable is eight digits (positive or negative).

Accuracy: Not given.

Floating/Fixed: Floating decimal.

Mathematical Method: See the program write-up.

Storage Required: The entire drum.

Speed: For reading and computing, time required = \[2(1 + d + 5 + 6) + 5d \times 500\] seconds per observation, where d is the number of dependent variables and 1 is the number of independent variables.

For punching, time required = \((1 + d + 10)(8.6)\) seconds per problem.

Relocatability: Not relocatable.

Remarks: An unpacking routine must be written for each problem to place the data in particular locations in normalized form. Zero is treated as a significant observation.

IBM 650 System: One 533, indexing registers, and automatic floating decimal arithmetic feature are required.
ANALYSIS OF VARIANCE OR COVARIANCE AND ADJUST MEANS PROGRAM

IBM 650 Library Program Abstracts

File no. 6.0.035

G. Ingram
State College of Washington
Computing Center
Pullman, Washington

Purpose: This program computes either the complete analysis of variance or analysis of covariance, including F values. In addition, adjusted means may be computed for the analysis of covariance.

Range: Maximum number of variables is six. Maximum number of observations is 9999. Maximum number of sources of variation is 66. All output is in floating decimal form. There can be no missing observations.

Accuracy: Not given.

Floating/Fixed: Floating decimal.

Mathematical Method: The standard formulas are used, including F tests for F between effects. Options for computing variances, means, and appropriate partial analysis of variance tables are provided.

Remarks: Values of s2 supplied by the user. The Central Limit Theorem is utilized to produce a normal distribution for the mean of any single variable. The sequence is utilized to produce a sample mean, which is then added to produce a single random deviate having unit variance. These random deviates are then modified so that they have zero mean and variance s2. For most problems a value of 10 for N is sufficient. However, when sampling from the tails of the distribution is fairly important, N should be larger.

IBM 650 Library Program Abstracts

File no. 6.0.034

STATE COLLEGE OF WASHINGTON

ANALYSIS OF VARIANCE OR COVARIANCE AND ADJUST MEANS PROGRAM

IBM 650 Library Program Abstracts

File no. 6.0.035

G. Ingram
State College of Washington
Computing Center
Pullman, Washington

Purpose: This program computes either the complete analysis of variance or analysis of covariance, including F values. In addition, adjusted means may be computed for the analysis of covariance.

Range: Maximum number of variables is six. Maximum number of observations is 9999. Maximum number of sources of variation is 66. All output is in floating decimal form. There can be no missing observations.

Accuracy: Not given.

Floating/Fixed: Floating decimal.

Mathematical Method: The standard formulas are used, including F tests for F between effects. Options for computing variances, means, and appropriate partial analysis of variance tables are provided.

Remarks: Values of s2 supplied by the user. The Central Limit Theorem is utilized to produce a normal distribution for the mean of any single variable. The sequence is utilized to produce a sample mean, which is then added to produce a single random deviate having unit variance. These random deviates are then modified so that they have zero mean and variance s2. For most problems a value of 10 for N is sufficient. However, when sampling from the tails of the distribution is fairly important, N should be larger.

IBM 650 Library Program Abstracts

File no. 6.0.034

STATE COLLEGE OF WASHINGTON

ANALYSIS OF VARIANCE OR COVARIANCE AND ADJUST MEANS PROGRAM

IBM 650 Library Program Abstracts

File no. 6.0.035

G. Ingram
State College of Washington
Computing Center
Pullman, Washington

Purpose: This program computes either the complete analysis of variance or analysis of covariance, including F values. In addition, adjusted means may be computed for the analysis of covariance.

Range: Maximum number of variables is six. Maximum number of observations is 9999. Maximum number of sources of variation is 66. All output is in floating decimal form. There can be no missing observations.

Accuracy: Not given.

Floating/Fixed: Floating decimal.

Mathematical Method: The standard formulas are used, including F tests for F between effects. Options for computing variances, means, and appropriate partial analysis of variance tables are provided.

Remarks: Values of s2 supplied by the user. The Central Limit Theorem is utilized to produce a normal distribution for the mean of any single variable. The sequence is utilized to produce a sample mean, which is then added to produce a single random deviate having unit variance. These random deviates are then modified so that they have zero mean and variance s2. For most problems a value of 10 for N is sufficient. However, when sampling from the tails of the distribution is fairly important, N should be larger.

IBM 650 Library Program Abstracts

File no. 6.0.034

STATE COLLEGE OF WASHINGTON

ANALYSIS OF VARIANCE OR COVARIANCE AND ADJUST MEANS PROGRAM

IBM 650 Library Program Abstracts

File no. 6.0.035

G. Ingram
State College of Washington
Computing Center
Pullman, Washington

Purpose: This program computes either the complete analysis of variance or analysis of covariance, including F values. In addition, adjusted means may be computed for the analysis of covariance.

Range: Maximum number of variables is six. Maximum number of observations is 9999. Maximum number of sources of variation is 66. All output is in floating decimal form. There can be no missing observations.

Accuracy: Not given.

Floating/Fixed: Floating decimal.

Mathematical Method: The standard formulas are used, including F tests for F between effects. Options for computing variances, means, and appropriate partial analysis of variance tables are provided.

Remarks: Values of s2 supplied by the user. The Central Limit Theorem is utilized to produce a normal distribution for the mean of any single variable. The sequence is utilized to produce a sample mean, which is then added to produce a single random deviate having unit variance. These random deviates are then modified so that they have zero mean and variance s2. For most problems a value of 10 for N is sufficient. However, when sampling from the tails of the distribution is fairly important, N should be larger.
PAIRED COMPARISONS FROM BALANCED INCOMPLETE BLOCKS

H. Gulliksen
L. Tucker
Educational Testing Service
Princeton, New Jersey

a. Purpose: This program utilizes input data from a questionnaire involving 11 objects arranged in 31 blocks of six objects each, and gives the paired comparisons matrix and scale values determined from this matrix.

b. Range: The program will handle a maximum of 999 subjects in a single group.

Accuracy: Proportions are rounded to four decimals. The approximations for the normal deviate, arc sine, and logistic transform have a maximum discrepancy of 0.0005 for proportions between 0.05 and 0.95.

Floating/Fixed: Fixed decimal.

c. Mathematical Method: The least squares solution for paired comparisons computations for the total group requires approximately five seconds for each pair of comparisons.

Remarks: It is desirable to use the auxiliary checking program to insure that the input cards are in correct form. This program checks to see that the input cards are in consecutive numerical order and that each item contains no permutation of the rank orders 1 through 6. Errors here may produce misleading results.

d. IBM 650 System: One 533 required.

ORTHOGONAL POLYNOMIAL CURVE FITTER

E. McCleary
J. Kastner
Wayne State University
Detroit, Michigan

a. Purpose: The program fits least square polynomial of i points to degree m.

b. Range: 2 ≤ i ≤ 99; 1 ≤ m ≤ 19.

Accuracy: The coefficients output is computed to double precision accuracy.

Floating/Fixed: Input and output are in fixed decimal form.

c. Mathematical Method: Least squares curve fitting with orthogonal polynomials.

d. Storage Required: Program requires about 1990 locations; locations 0050-0999 are reserved for an optional weight computing subroutine.

Speed: Maximum time for curve fitting is 25 minutes.

Relocatability: Not given.

e. Remarks: Three methods of weighting may be used:
   1. Uniform weights.
   2. Weights arbitrarily assigned to each point.
   3. Weights as computed by any subroutine not longer than 100 words.

The complete routine consists of three sections:

1. Curve Fitter
2. Discriminator, which selects and evaluates best fitted curve.
3. Evaluator (in SOAP II form) which may be utilized to evaluate any polynomial 1 ≤ m ≤ 19) from section 1 above.

f. IBM 650 System: One 533 required.

Special Devices: Alphabetical device required if re-assembly of SOAP II deck is desired.

DETERMINING PROBABILITIES FROM A FITTED GAMMA DISTRIBUTION

R. O. Bartley
W. T. Lovell
Iowa State University
Ames, Iowa

a. Purpose: This program computes three decimal digit probabilities and is a sequel to "Fitting of the Gamma-Distribution to Rainfall Data" by R. O. Bartley and W. T. Lovell (file B.0.029).

b. Range: The parameter λ must be less than 100, but q must be greater than 0.1 and be less than 100.

Accuracy: Usually 3 decimal digits, but at the extremes, accuracy will be lost.

Floating/Fixed: Fixed decimal arithmetic is used.

c. Mathematical Method: For q ≤ 0.8 probabilities are computed from a stored table of the incomplete Gamma function. Linear and/or quadratic interpolation is used within the table. For q > 0.9, Wilson-Hilferty approximation, requiring a table of normal Probabilities, was used.

d. Storage Required: Entire drum is used.

Speed: About seven seconds for 20 probability values.

Relocatability: Not relocatable.

e. Remarks: Up to twenty probabilities are packed per output card. The levels at which the probabilities are calculated can be very easily changed.

f. IBM 650 System: One 533 required.

SEASONAL ADJUSTMENT OF ECONOMIC TIME SERIES

S. H. Bueckel
IBM, 650, R. L. Mo.

a. Purpose: This program was designed to isolate and remove the seasonal factor in time series.

b. Range: From five to ten years of monthly data may be adjusted at one time. Longer series may be broken down into ten-year periods and overlapped.

Accuracy: Does not apply.

Floating/Fixed: FOR TRANSIT floating decimal mode.


d. Storage Required: The entire drum is used.

Speed: Ten years of monthly data are processed in thirty minutes.

Relocatability: Not given.

e. Remarks: The original source program was written in FOR TRANSIT, and may be compiled on the 7090 series machines.

f. IBM 650 System: One 533 required.

Special Devices: None.

PROGRAM TO CALCULATE SEASONALLY ADJUSTED INDICES

W. Mendl
Prudential Life Insurance Company
Newark, New Jersey

M. Tartis
IBM, New York

a. Purpose: The program will adjust a time series, generally composed of a trend, cyclical movement, seasonal variations, and random or irregular fluctuations, to a form that shows primarily the non-seasonal movements.
MINMAX POLYNOMIAL APPROXIMATION ON A FINITE POINT SET

D. W. Marquardt
Mary Ann Beurling
E. L. deFont de Nenghow & Co., Inc.

Purpose: To compute the polynomial of specified degree which approximates in the minimum sense to a finite set of points (values of some function f(x) on a finite interval).

Range: Up to 100 values of f(x); where x = 1, 2, ..., N may be spaced as desired on any finite interval.

Accuracy: Program normalizes range of x's to -1 & 1, to minimize roundoff error. Accuracy is limited only by roundoff.

Floating/Fixed: Floating decimal arithmetic is used.

Mathematical Method: This program uses the iterative method of P. C. Curtis and W. L. Frank, as described in the Preprints of papers presented at the June 1965 meeting of the Association for Computing Machinery, pages 22-1 to 23-5.

Storage Required: Most of drum, all of immediate access storage.

Speed: Depends upon N, n, and number of iterations required.

Typical cases:

<table>
<thead>
<tr>
<th>N</th>
<th>n</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>33</td>
<td>3</td>
<td>3 min.</td>
</tr>
<tr>
<td>33</td>
<td>5</td>
<td>5 min.</td>
</tr>
<tr>
<td>33</td>
<td>7</td>
<td>7 min.</td>
</tr>
<tr>
<td>33</td>
<td>9</td>
<td>9 min.</td>
</tr>
</tbody>
</table>

Relocatability: Not relocatable.

Remarks: Output includes: coefficients of minimax polynomial, minimum error of the approximation, normalization constants. Utility board is used.

IBM 650 Library Program Abstracts

ANALYSIS OF VARIANCE PROGRAM FOR THE IBM 650

J. W. Johnson
Canadian Army Operational Research Establishment
Ottawa, Ontario
Canada

Purpose: This program calculates the analysis of variance table including the component of variance for crossed, nested, mixed experiments with three or fewer factors.

Range: The restrictions imposed by use of this program are:

- q < 500
- (number of digits in x) ≤ 10
- (number of digits in y) ≤ 10
- (number of replications) ≤ 10

Accuracy: Double precision arithmetic is used in summing squared terms to preserve accuracy. (Continued on next column)

Floating/Fixed: Fixed decimal arithmetic.

Mathematical Method: Double precision arithmetic is used. Computational techniques are those described in Bennett and Franklin, Statistical Analysis in Chemistry and the Chemical Industry, Wiley, New York.

Storage Required: Not given.

Speed: The example problem required about 75 seconds.

Relocatability: Not given.

Remarks: 1. The ratio of the number of levels in the sample to that in the corresponding population is entered as either 5 or 1. That is, finite random models cannot be analyzed with this program.

2. The program may be conditioned to punch the partial sums and means, and null sums of squares and variances.

IBM 650 System: One 533 required.

Special Device: Alphabetic device required.

IBM 650 Library Program Abstracts

COMPLETE PAIRED COMPARISONS SCHEDULE (PARCOPLET (2,2))

Harold Gulliksen
Psychology Department
Princeton University
Princeton, New Jersey

Purpose: This program stores input data from a paired comparison questionnaire of 21 objects or less (with or without the Like-Dislike section) and outputs the summary data for each subject and the scale values. The paired comparison matrix may be punched out or omitted as desired.

Range: Accuracy, Floating/Fixed: The program will handle a maximum of 300 subjects in a single group. Fixed input is used throughout. Proportions are rounded to four decimals. The approximation for the normal deviate, arcsine, and logistic have a maximum discrepancy of .005 for proportions between .01 and .99.

Mathematical Method: The least squares solution for scale values is used. Scale values are computed, using the normal deviate, the arcsine, and the logistic transform.

Storage Requirements, Speed, Relocatability: The analysis program utilizes fixed drum locations, and is not relocatable. Depending on the number of stimuli in the questionnaire the program processes each subject in about three to 15 seconds and the final paired comparisons computations for the total group take from one to five minutes.

Additional Remarks, Precautions or Restrictions: It is desirable to use the auxiliary checking program to ensure that the input cards are in correct form. This program chooses x to be the cards are in consecutive numerical order and that each item response is 1 or 2. Errors here may produce misleading results.

Equipment Specifications: It requires the minimum 650 installation and uses the standard 650-60 board for eight two-digit words for the 533 input-output.

IBM 650 Library Program Abstracts

MULTIPLE REGRESSION ANALYSIS

Numerical Computation Laboratory
Ohio State University Research Center
Columbus 12, Ohio

Purpose: This program performs the multiple regression analysis under the hypothesis

y = b_0 + b_1 x_1 + b_2 x_2 + ... + b_n x_n + \epsilon

The x_i are the observable independent variables, the y is the observable dependent variable, and the \epsilon, called the regression coefficients, are the constants to be estimated.

Range: Not given.

Floating/Fixed: All input data must be described by six digit (fixed point) numbers, and the format of the input data is not specified.

Mathematical Method: The method used is a standard one for multiple regression analysis. Details are contained in the program write-up.

Storage Required: This program utilizes the entire drum and high speed storage.

(Continued on next page)
**IBM 650 Library Program Abstracts**

**CALCULATION OF THE AUTO-CORRELATION FUNCTION AND THE SPECTRAL DENSITY**

Mrs. V. D. Minkait
Battelle Memorial Institute
505 King Avenue
Columbus, Ohio

**Purpose:**
This computer program computes the auto-correlation function and the spectral density. The program is divided into two phases as follows:

**Phase I:**
- **Part I:** Calculation of the mean value, \( T \)
- **Part II:** Calculation of the cross-correlation function, \( R(s) \)

**Phase II:**
- Calculation of the spectral density, \( W(f) \).

**Requirements:**
- **Phase I:**
  - The input data must not exceed 15,000 observations.
  - The number of observations must be less than 150.

**Accuracy:**
- Phase I: The mean value is calculated to the same number of significant digits as the given function.
- Phase II: The spectral density is evaluated to six more significant figures than the given input.

**Mathematical Method:**
Formulas are given in the write-up.

**Storage Requirements:**
- Phase I: Approximately 500 drum locations are needed.
- Phase II: About 1000 locations are needed.

**Speed:**
- Phase I: Computation speed of the computer program is dependent on the number of input data. Approximate formulae are given in the write-up.
- Phase II: The computation time is not given.

**Reliability:**
The program cannot be reentered.

**Remarks:**
- Phase I: None.
- Phase II: None.

**Special Devices Required:**
- One 533, indexing registers, floating point device, and three 800 cards per minute.

**IBM 650 Library Program Abstracts**

**GENERAL TABULATION PROGRAM**

V. H. Mitchell
Agricultural Marketing Service
U.S. Dept. of Agriculture
Washington, D. C.

**Purpose:**
The purpose of this program is to tabulate any desired field of 4 or less columns. As many as 10 fields or less may be tabulated at one time. No total must exceed 10 digits.

**Speed:**
By punching one control card, a control card can be shifted to any columns of the card and fields in any part of the data card may be tabulated.

**Accuracy:**
- Fixed decimal point is used throughout.

**Storage Requirements:**
- Storage required is approximately 800 locations.
- Program is written in one card FORTRAN II language and can be completely reprogrammed using one card.

**Speed:**
- The program can operate at rates from 150 to 600 cards per minute depending upon the number of columns tabulated.

**Remarks:**
- Can be used to tabulate fewer than 5 fields if desired.
- Requires minimum 650 equipment.

**IBM 650 Library Program Abstracts**

**CALCULATION OF THE CROSS-CORRELATION FUNCTION AND THE CROSS-SPECTRAL DENSITY**

Mrs. V. D. Minkait
Battelle Memorial Institute
505 King Avenue
Columbus, Ohio

**Purpose:**
This computer program computes the cross-correlation function and the cross-spectral density. The program is divided into two phases as follows:

**Phase I:**
- **Part I:** Calculation of the cross-correlation function, \( R(0) \) and \( R(s) \).
- **Part II:** Calculation of the cross-spectral density, \( W_{x,y}(f) \).

**Remarks:**
The two phases are used independently. The output of Phase I is the input for Phase II.

**Accuracy:**
- Phase I: The mean value is calculated to the same number of significant digits as the given function.
- Phase II: The spectral density is evaluated to six more significant figures than the given input.

**Mathematical Method:**
Formulas are given in the write-up.

**Storage Requirements:**
- Approximately 500 drum locations are needed.
- About 1000 locations are needed.

**Speed:**
- The computation speed of the computer program is dependent on the number of input data. Approximate formulae are given in the write-up.

**Reliability:**
The program cannot be reentered.

**Remarks:**
- Phase I: None.
- Phase II: None.

**Special Devices Required:**
- One 533, indexing registers, floating point device, and three 800 cards per minute.

(Continued on next page)
IBM 650 Library Program Abstracts

FITTING OF DATA TO THE TWO PARAMETER-GAMMA DISTRIBUTION WITH SPECIAL REFERENCE TO RAINFALL DATA
H. O. Hartley
Computing Group
Statistical Laboratory
Iowa State University
Ames, Iowa

a. Purpose: Calculates the two parameters \( \alpha \) and \( \beta \) for the Gamma distribution as well as the mean, variance and the covariance.
b. Range: Input = 4 digits or less and less than 20,000 observations. Output = \( \alpha \) and \( \beta \). Variance and covariance scaled 1.0.

Accuracy: If \( \alpha < 3712 \) maximum error \( q = 0.0510\% \) If \( \alpha > 3712 \) maximum error \( q = 0.0154\% \) for additional information see reference in the program description.

Floating/Fixed: All calculations in fixed.
c. Mathematical Method: Greenwood and Durand’s polynomial approximations in the maximum likelihood method.
d. Storage Requirements: Entire drum (5,000 words).

Speed: 4 digits input about 170/sec. 3 digits or less at 200/sec. Punch loop of about 2 seconds.

e. Remarks: Test example and answers contained in description.
f. IBM 650 System: One 535 required.

IBM 650 Library Program Abstracts

54 X 54 CORRELATION COEFFICIENTS
James R. Farmer
Computing Center
Washington State University
Pullman, Washington

a. Purpose: This program provides simple correlation coefficients and related statistics for all combinations of up to 54 variables. Zero is considered as a significant observation.
b. Range: Maximum number of variables 54. Maximum size of any variable is eight digits (positive or negative).

(Continued on next column)

Floating/Fixed: Floating Decimal.
c. Mathematical Method: See program write-up.

Speed: Timing approximation
Input--seconds/observation \( 0.05 \times (N^2 - N) \times 100 \)
Output--seconds/problem \( 0.5 (V)(V-1) \)
where \( N \) is the number of variables.

Relocatability: Not relocatable.
e. Remarks: Original data cards may be used as input. Eleven or more VARIANCE require the use of an unpacking routine.
f. IBM 650 System: One 535, 2000 word drum, indexing registers and automatic floating decimal arithmetic.

IBM 650 Library Program Abstracts

FOUR WAY ANALYSIS OF VARIANCE

Columbus 12, Ohio

a. Purpose: This routine produces the analysis of variance table as described in the detailed program write-up. All means on one, two, three, or four subscripts (i.e., replications are always averaged) together with estimates for the main effects and first and second interaction effects are computed.
b. Range: Not given.

Accuracy: Not given.

Floating/Fixed: Fixed point input and output. Included in the output is the error computation.
c. Mathematical Method: See program write-up.
d. Storage Requirements: Locations occupied: 1410-1999 (650 words)

Speed: Not given.

Relocatability: Not relocatable.

e. Remarks: This routine is easily adapted to any smaller dimensional analysis of variance, with or without replications. The replication subscript is always ---2.

The program card deck includes the loading and punching subroutines and the necessary control cards for these subroutines which are used by the program.
f. IBM 650 System: Minimum 650, no special equipment is needed.

IBM 650 Library Program Abstracts

TWO VARIABLE LINEAR REGRESSION AND CORRELATION

Philip J. Kinsler
Ohio State University

a. Purpose: This program fits a straight line:

\[ Y = a + bX \]

by the method of Least Squares. It also produces the arithmetic mean and standard deviation of each variable, the simple correlation coefficient and the standard error of estimate about the fitted line. If desired, the basic summations developed for calculating coefficient and the standard error of estimate about the fitted line. If desired, the basic summations developed for calculating these statistics can be punched out.
b. Restrictions, Range: Input data are limited to fixed decimal numbers of no more than 6 digits. The number of observations is essentially unlimited. (99,999 observations maximum). Output is in floating decimal notation.

c. Method: The Method of Least Squares is used for fitting the line. The standard deviations are computed as unbiased estimates.
d. Storage Requirements: Uses 39 instructions in instruction-per-card format. Data cards feed at 60 cards per minute. Punch-out occurs immediately after last data card is read. This program is not relocatable.

e. Remarks: Program deck includes the Erco Floating Decimal Point Subroutine (650 file 2, 6, 0.049) and the square root subroutine from the Triangle.

(Continued on next page)
**IBM 650 Library Program Abstracts**

### ANALYSIS OF VARIANCE, DISPROPORTIONATE SUBCLASS NUMBERS

**Purpose:** This program computes the statistics for an analysis of variance, allowing for disproportionate subclass numbers, and assuming that interactions are zero. The analysis is completed, and an F-value given for each factor tested.

**Method:** The method of "fitting constants" is used.

**Storage Requirements:** The entire 2000-word drum is used.

**Relocatability:** Not in relocatable form.

**Remarks:**
1. This routine used IBM 650 Library Program No. 05.2.02, Matrix Inversion Routine.
2. Special remarks are contained in the program write-up.

**IBM 650 System:** Three indexing accumulators and the floating decimal feature are used in the program.

### ANALYSIS OF VARIANCE OR COVARIANCE FOR NON-ORTHOGONAL DATA AND FOR ANY STATISTICAL DESIGN

**Purpose:** In writing a general analysis of variance program, one is confronted with the problems of (i) devising a general systematic scheme for retrieving from the computer storage the elements that occur to each of the many counts necessary for the analysis desired, (ii) making the program general enough to be useful for analyzing the data from as many types of experimental designs as possible and (iii) providing for the situation where there are missing data or unpalpable sub-class numbers. The purpose of this program is to analyze the variance in such a way that all three problems stated above are answered.

**Remarks:**
- All computations are in double precision fixed point arithmetic.
- Speed of square can be obtained at approximately 12 significant digits.
- Mathematical method: The mathematical method used in adjusting for disproportionate frequencies (obtaining a set of least squares equations) is an iterative scheme which does not require that the matrix of conditions be stored in the computer. For this reason up to 200 least squares equations in as many variables may be solved without using external storage.
- Storage Required: The program is non-relocatable, uses practically all of 2,000 word drum storage and is reasonably fast in execution.

(Continued on next column)
IBM 650 Library Program Abstracts

DISTRIBUTION PROGRAM GENERATOR

James E. Farmer
Computing Center
Washington State University
Pullman, Washington

Purpose: The purpose of this program is to provide a distribution program without programming effort. The generator provides a synthetic problem in EASP II input format, after being assembled the object program will provide the counts and percentages for simple distributions. Input to the generator consists of the number of items (questions) to be distributed and the highest numerical response to each item.

Restrictions: Range: Maximum number of items (questions) is 99. Maximum size of any item is 8 digits (positive response only).

Floating/Fixed: Not applicable.

Method: Not applicable.

Storage Requirements: Entire 2,000 word drum.

Speed: Approximately 2 to 6 minutes depending upon number of items.

Remarks: None.

IBM 650 System: One 533, 2,000 word drum and indexing registers.

EXPANDED SIMPLE CORRELATION COEFFICIENT ROUTINE

F. P. Fish
International Business Machines Corporation
524 Whitcraft Dr.
Los Angeles, California

Purpose: To provide the ability to obtain simple correlation coefficients of 8 dependent variables against several combinations of independent variables, to include linear terms, quadratic terms and interaction terms. This information will serve as an aid to Regression Analysis by giving the analyst more information on which to determine the form of the regression equation.

(Continued on next page)

IBM 650 Library Program Abstracts

ANALYSIS OF VARIANCE FOR PARTIALLY OR SINGLE REPLICATED 2^k FACTORIAL EXPERIMENT WITH OPTIONAL SINGLE CONFounding

Robert W. Naylor
Spencer Chemical Company
Research Center
5009 West 65th Street
Merriam, Kansas

Purpose: The program calculates the analysis of variance and F-test ratios for a 2^k (factorial experimental) wherein k may be any number of levels from 1 through 8, the first factor may have any number of additional factors to 1 through k. Fractional or single replication of such designs can be handled with or without single confounding in up to 8 blocks.

Restrictions: Range: The program runs in two parts and listing the Segment 7 output gives all two-way tables in conventional arrangement plus corrected sums of squares, main squares, and F-ratios along with degrees of freedom where they may be greater than one. Three-factor and higher interactions are combined into the residual for the F-test, but an external error estimate may be used instead.

Any number of measured value sets (temperature, pressure, yield, etc.) may be processed continuously for the same statistical experiment.

Method: Does not apply.

Storage Requirements: Dependent upon the statistical experiment being analyzed. Segment 1 requires about 3 minutes plus 60-100 seconds per seven experimental values fed. Segment 2 runs 8-10 minutes per set of measured values.

Remarks: Fortran I

IBM 650 System: Basic IBM 650.

CASP - A CORRELATION AND REGRESSION PROGRAM

R. E. Bacon
International Harvester Company
Wisconsin Steel Works
Chicago 17, Illinois

Purpose: The program computer computes means, standard deviations, simple correlation coefficients, partial correlation coefficients, and multiple regression coefficients. Provision is made for removing output to add or subtract observations, interchange and remove variables, and combine results of problems of equal dimensions.

Range: Up to 50 variables are permitted, of which any number may be designated as dependent.

Accuracy: Not given.

Floating/Fixed: Data may be entered in SIR floating-point or variables per card, or in standard 6-per-card FOR TRANSIT format. Internal operation and output are in SIR floating-point.
ROOTS OF A FUNCTION OF A REAL VARIABLE

F. Edelman

H.C.A., David Packoff 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. Wolostia 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 to successively halve this interval until the desired accuracy is obtained.
d) Storage required is 1200 locations, 0000 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 be detected. Program decks are available upon request from the author.
f) Minimum 650.

IBM 650 Library Program Abstracts

S. Rabushka

The Emerson Electric Mfg. Co., St. Louis 21, Missouri

a) Purpose: This program solves a linear or nonlinear equations in a unknowns for values of \( x \) equal to or less than 15.
b) Range: As noted above, values of \( n \leq 15 \).

Accuracy: Usually may be selected by the user.
Floating/Fixed: Floating decimal.
c) Mathematical Method: Newton-Raphson.
d) Storage Required: Locations 500-1494 are available for the programming of the equations that are to be solved. See the program write-up for more information.

Speed: Varies greatly with different problems.
Relocatability: Not given.
e) Remarks: The program fails in certain cases. However, these errors give additional information about the problem, as failure indicates one of the following:
1) Multiple solutions
2) Two or more solutions close together
3) No solutions in the neighborhood of the initial guess

These cases are indicated by an overflow stop with 34 1967 1553 in the program register or by the program running a long time without answers. However, it may be that in the latter case the accuracy demanded is simply too much.
f) IBM 650 System: One 533, indexing registers, and automatic floating decimal arithmetic feature.

BUNGEE-KUTTA ROUTINE FOR SOLVING DIFFERENTIAL EQUATIONS ON THE IBM 650

A. S. Rosenthal

Naval Air Development Center
Johnsville, Pennsylvania

a) Purpose: The programmer writes a SOAP II program for each of the derivatives beginning at one of a set of specified entry locations and exiting at a specified fixed location. Information such as number of equations, expected duration of program, allowable terminal error, and initial conditions is supplied to the system by the programmer. The system then computes, choosing its own time intervals and passing variables and derivatives at each time interval.
b) Range: The routine solves up to 35 simultaneous first order ordinary differential equations.

Accuracy: The routine provides automatic time interval control designed to keep the estimated accumulated errors in certain of the variables designated by the programmer.
Floating/Fixed: Floating decimal arithmetic is used.
c) Mathematical Method: Integration by standard Runge-Kutta formulas. Special formulas are derived for error estimation.
d) Storage Required: The main program uses 115 drum storage locations in addition to which seven locations are needed for the processing of each system variable. Two spaces are required to pass any auxiliary function in function which may be obtained from the system variables by algebraic manipulation alone.

The input-output routine (Real-Push "B") uses drum locations 1831-1999.

Speed: Processing time required is approximately 1 second per interval for each variable.
Relocatability: Not given.
e) Remarks: In addition to the main program the system contains an input-output routine Real-Push "B" which allows creating or passing any chosen number of words sequentially with any chosen number of drum locations as a fixed increment. This routine, which is extremely flexible, may be used independently, as well as with the system.
f) IBM 650 System: One 533 and indexing registers required.

Special Devices: None.
IBM 650 Library Program Abstracts

ZEROS OF COMPLEX POLYNOMIALS

Les Andrews
Technical Staff
Greenwich Engineering Division
AMF, Greenwich, Connecticut

a. Purpose: This SOAP II program will find the complex roots of the general
algebraic equation of the nth degree.

\[ a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0 = 0 \]

where the coefficients are complex numbers, \( a_n \neq 0 \) and \( a_0 \neq 0 \).
b. Range: \( N \) must be less than or equal to twenty.
c. Mathematical Method: Successive approximations toward a particular root
are attained by finding the nearest root of the quadratic which passes through
the last three iterates.
d. Storage Required: 649 locations.

Speed: Depends on the location of the roots.
Relocatability: Non-relocatable.
e. Remarks: None.
f. IBM 650 System: Minimum 650.

IBM 650 Library Program Abstracts

MATH FIN

Mr. Clay C. Rose, Jr.
Department of Mathematics
University of Kentucky
Lexington, Kentucky

a. Purpose: The program is designed to compute double-precision tables
of the following:
1. Amount of 1 at Compound Interest.
2. Present Value of 1 at Compound Interest.
3. Amount of an Annuity
4. The Annuity That 1 will Purchase
5. The amount of 1 at compound interest.

b. Range: The range of the argument must be:
for \( K_0 \), \( x \leq 5 \).00.
for \( K_1 \), \( x \leq 250 \).
for \( K_2 \), \( x \leq 5 \).00.
for \( K_3 \), \( x \leq 5 \).00.
for \( K_4 \), \( x \leq 5 \).00.

Accuracy: The accuracy of the subsequent terms are neglected.
Floating Point: The computation is done in fixed point arithmetic.
c. Mathematical method: Refer to "Tables of the Functions \( K_0(x) \) and
\( K_1(x) \)" by W. G. Bickley, D. L., and John Hayler, A. C. O. J.,
d. Storage Required: 100 locations (6500 through 6999) are used.

Speed: The average execution time is as follows:
for \( K_0 \): 1 sec.
for \( K_1 \): 2 sec.
for \( K_2 \): 4 sec.
for \( K_3 \): 6 sec.
for \( K_4 \): 8 sec.

Relocatability: Not relocatable.
e. Remarks: 650 Library Program P2.6.002 for \( x \) is incorporated as a subroutine.
f. IBM 650 System: Minimum 650.

IBM 650 Library Program Abstracts

NUMERICAL INTEGRATION OF THE DOUBLE INTEGRAL

A. A. Nakada
C. Church
Harvard University
Massachusetts Institute of Technology
Cambridge, Mass.

a. Purpose: To approximate the integral having the general form

\[ \int_0^1 \int_0^1 f(x,y) \, dx \, dy. \]

b. Restrictions, Range: None.
c. Method: Numerical integration over the annulus with outer
radius \( R \) and inner radius \( r \).
d. Storage Requirements: Not applicable.
e. Remarks: None.
f. IBM 650 System: Uses floating point and index register.

IBM 650 Library Program Abstracts

FLOATING POINT SQUARE ROOT SUBROUTINE

Charles Goldstein
IBM 650 Applied Programming
Time & Life Building
New York, New York

a. Purpose: This routine computes the square root of numbers in floating
point form using a floating iterative method. This program was designed to
accurately perform floating point square root operations.
b. **Range:** This routine accepts floating-point numbers of the form.
   \[ \pm 10^{\pm 100} \times (\pm 10^{-100} \text{ to } \pm 10^{100}) \]
   Answers are in floating-point form and all significant digits are exact.

   c. **Mathematical Method:** After taking an initial approximation, Newton's method is used to find the square root. With the initial approximation used, this method converges to eight significant figures in five iterations.

   d. **Storage Required:** 21 Permanent drum locations including a programmed step for negative arguments. 3 Temporary storage locations.

   e. **Speed:** 140 ms.

   f. **IBM System:** This routine uses index register D which is not required.

   i. **Purpose:** This package of programs is designed to facilitate arithmetic operations with complex numbers in the form \( a + b \cdot i \).

   ii. **Storage Requirements:** The program deck is written to be relocatable. It is used and the number of temporary storage locations are available for the program to be relocated when loaded.

   iii. **Accuracy:** Depends on the operation being performed. The accuracy is 2 parts in 8th decimal place.

   iv. **Frequently Used Subroutines:**

   a. **Purpose:** This package of programs is designed to facilitate arithmetic operations with complex numbers of the form \( a + b \cdot i \).

   b. **Range:** Depends on the operation being performed.

   c. **Mathematical Method:** All arithmetic operations are performed in floating-point arithmetic. In some operations a method of segment adjustment is used to prevent overflow and underflow.

   d. **Storage Requirements:** Drum locations 0000, 1389 through 1398.

   e. **Remarks:** The program incorporates a floating decimal arithmetic routine and a square root subroutine to perform necessary arithmetic operations.

f. **IBM System:** Minimum IBM 650.

**IBM 650 Library Program Abstracts**

---

**CLERSCHI-GORDON COEFFICIENT SUBROUTINE**

B. R. Chi

Bemidji Polytechnic Institute

a. **Purpose:** The subroutines compute the Clersch-Gordon or water-coupling coefficients \( \psi_j(l, \mu) = \chi_j(l, \mu) \).

b. **Range:** \( |\mu| = 0, 1, 2, \ldots, 8 \).

---

**PYRAMID OF RANOMANU**

John Bergau, Robert Keshell

IBM

a. **Purpose:** This program generates a set of random non-repeated numbers which span a predetermined range or field size.

b. **Range:** Up to 9,999 numbers may be generated for each computer pass. Any field size from a minimum of five “cella” may be used. Normal use of the program calls for a field size of OC columns \( 0 \leq CX \leq 99 \) by 10 rows, the “cella” being numbered 0000 to 1000.

c. **Mathematical Method:** Does not apply.

d. **Storage Required:** About 650 words of 160 memory optimally scattered in lower memory.

---

**COMPLEX I**

A. INTERPRETIVE PACKAGE FOR COMPLEX ARITHMETIC

(Column on next column)

---

**COMPLEX II**

AN INTERPRETIVE PACKAGE FOR COMPLEX ARITHMETIC

---

**SYMBOLIC INTERPRETIVE SYSTEM FOR THE IBM 650 - 655 (REAL AND COMPLEX ARITHMETIC)**

---

**IBM 650 Library Program Abstracts**
IBM 650 System: IBM 650 with alphanumeric device, one 533, automatic floating decimal, 16, indexing registers.

Remarks: None.

IBM 650 Library Program Abstracts

Present Value and Rate of Return (PVRA)

(For a Finite Chain of One Investment - Single Machine Horizon)

Martin B. Solomon, Jr.
University of Kentucky
Lexington, Kentucky

- Purpose: Will compute the present value of an investment at the end of each year of its useful life and the discounted rate of return over the whole life. It assumes an infinite chain of replacements.
- Range: Life can range from 1 to 50 years.
- Accuracy: Present value to eight significant digits. Rate of return to 1 percent computed.
- Mathematical Method: PV = \sum_{n=1}^{\infty} \frac{R_n}{(1+r)^n}
- Storage Required: Optimized by SOAP II as program is scattered throughout drum.
- Speed: Computes present value in a few seconds. Rate of return in 30 seconds.
- Relocatability: Not relocatable.
- Remarks: None.

IBM 650 Library Program Abstracts

Present Value and Rate of Return (PVRA)

(For a Finite Chain of One Investment - Single Machine Horizon)

Martin B. Solomon, Jr.
University of Kentucky
Lexington, Kentucky

- Purpose: Will compute the present value of an investment at the end of each year of its useful life and the discounted rate of return over the whole life. It assumes an infinite chain of replacements.
- Range: Life can range from 1 to 50 years.
- Accuracy: Present value to eight significant digits. Rate of return to 1 percent computed.
- Mathematical Method: PV = \sum_{n=1}^{\infty} \frac{R_n}{(1+r)^n}
- Storage Required: Optimized by SOAP II as program is scattered throughout drum.
- Speed: Computes present value in a few seconds. Rate of return in 30 seconds.
- Relocatability: Not relocatable.
- Remarks: None.
A RAY TRACING PROGRAM

J. May
Columbia University
Hudson Laboratories
Dobbs Ferry, N. Y.

a. Purpose: Traces the path of a ray in a layered inhomogeneous medium with regular boundaries.
b. Range: Maximum of 64 different velocity points.
d. Storage Requirements: Approximately 150 unused drum locations.
e. Remarks: None.

IBM 650 Library Program Abstracts

A SOLUTION OF HEAT DIFFUSION EQUATION

R. E. Stockey
Theoretical Physics Division
E. L. du Pont de Nemours & Co.
Savannah River Laboratory
Aiken, S. C.

a. Purpose: Equations and routines are presented to obtain the temperature distribution in a section of a nuclear heat source. The solution of the heat diffusion equation in 1-D geometry is approximated by the solution of a set of appropriate difference equations. Three regions with possible differences in heat conductivity or heat source are allowed in the radial direction, e.g., inner cladding, fuel, and outer cladding. Heat is transferred to a bulk coolant at each radial surface. The program can be used to study the effects of countending between regions and of inhomogeneities in the surface heat transfer and in the heat source.
b. Range: Floating.
d. Storage Requirements: 1000 locations. Speed depends on number of grid points used.
e. Remarks: Not given.
f. IBM 650 System: Model 2 with Floating decimal & index registers.

IBM 650 Library Program Abstracts

MOONSHINE

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

a) Solves the one-dimensional neutron diffusion equation. The multi-group diffusion equation is solved for the case of a sphere, cylinder, and slab.
b) A maximum of three different material regions and eight 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 850.

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 850.

ONE-SPACE-DIMENSIONAL MULTIGROUP

G. J. Baberley and V. A. Wahran
Ge, Knolls Atomic Power Lab, Schenectady

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 pamphlet which is supplied with the write-up and listing.
d) The entire drum is required. Timing is from 20 seconds to one minute per group for a 50-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.
b. Four approximations are provided with the inclusion of isotropic inelastic scattering, resonance capture, and fast fission. Fixed point arithmetic is used.

d) Solutions required two runs through the computer. The entire drum is used.

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

f) 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.
a) This code computes the one-velocity neutron flux distribution in concentric cylindrical geometry using a P2 spherical harmonics approximation to the neutron transport equation. Anisotropic scattering is included and each region must remain constant throughout the region.

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

c) The P-2 flux Code is an analytic solution of the P2 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 sections at large radii will cause a machine stop because the calculated Bessel functions become too large for the floating point representations.

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 distributions and will permit the code to operate.

The P-2 Flux Code will automatically compute the neutron flux distributions throughout the regions in the problem (the number of points computed is controlled) and will also provide average fluxes in each region.

f) Minimum 650.
IBM 650 Library Program

"P - 3 Flux Distribution," by J. W. Well and P. Cahal

Part 1 of the P - 3 program deck originally furnished to the library was discovered to contain erroneous multiple punches in column 70 in several cards. A number of copies of the deck were furnished to 650 installations before the errors were noted. Accordingly, it is recommended that any decks obtained from the library prior to August 1, 1958 be replaced. Decks mailed on or after that date have been corrected.

April 1958, Bulletin 18 - 31

650 LIBRARY PROGRAM ABSTRACT

FILE NUMBER 8.2.016

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) Iterative solution.

c) Mathematical Method: Integration is accomplished by the fourth order Runge-Kutta technique.

Accuracy: Given.

Floating/Fixed: Floating decimal.

REMARKS: Recipes are provided to reduce to several special cases of physical interest. Directions are given to allow addition of one more feedback loop.

Special Devices: None.

Speed: A representative problem using the full program takes about 4 minutes.

Relocatability: Not relocatable.

Storage: 450 locations.

Systems: One 533 required.

ANNUAL COST: $0.00

Special Notes: None.

9.2.016

IBM 650 Library Program Abstracts

FILE NUMBER 8.2.016

ARMOUR REACTOR KINETICS (ARK-I) CODE

T. Engelhart
W. E. Loewe
Armour Research Foundation of Illinois Institute of Technology
Chicago 16, Illinois

a) Purpose: This routine is used to obtain the transients of neutron flux in response to a change in reactivity of a nuclear reactor. The routine is a modification of the Savannah River Laboratory K-code (IBM 650 Library Program 8.2.018), from which it differs in the following respects: (1) driven changes in reactivity remain arbitrary functions of time, but must occur as a result of a change in the average neutron absorption cross section; (2) temperature coefficients are restricted to those affecting \( \Sigma_b \); (3) the feedback equations are slightly more general; and (4) a substantial saving in running time is realized. This last difference results from the fact that integration is accomplished by a fourth order Runge-Kutta technique.

b) Range: Six delayed groups of neutrons and two reactivity feedback loops are allowed.

Accuracy: Not given.

Floating/Fixed: Computation is in the floating decimal mode as described by G. R. Trimble in Technical Newsletter 8, pp. 37 - 41.

c) Mathematical Method: Integration is accomplished by the fourth order Runge-Kutta technique.

d) Storage Required: Approximately 1930 storage locations are required.

Speed: A representative problem using the full program takes about 1 hour.

Relocatability: Not relocatable.

e) Remarks: Recipes are provided to reduce to several special cases of physical interest. Directions are given to allow addition of one more feedback loop.

f) System: One 533 required.

Special Devices: None.

Speed: A representative problem using the full program takes about 4 minutes.

Relocatability: Not relocatable.

Storage: 450 locations.

Systems: One 533 required.

ANNUAL COST: $0.00

Special Notes: None.

9.2.019

IBM 650 Library Program Abstracts

FILE NUMBER 8.2.019

Physical Sciences

ARMOUR REACTOR KINETICS (ARK-I) CODE

T. Engelhart
W. E. Loewe
Armour Research Foundation of Illinois Institute of Technology
Chicago 16, Illinois

a) Purpose: This routine is used to obtain the transients of neutron flux in response to a change in reactivity of a nuclear reactor. The routine is a modification of the Savannah River Laboratory K-code (IBM 650 Library Program 8.2.018), from which it differs in the following respects: (1) driven changes in reactivity remain arbitrary functions of time, but must occur as a result of a change in the average neutron absorption cross section; (2) temperature coefficients are restricted to those affecting \( \Sigma_b \); (3) the feedback equations are slightly more general; and (4) a substantial saving in running time is realized. This last difference results from the fact that integration is accomplished by a fourth order Runge-Kutta technique.

b) Range: Six delayed groups of neutrons and two reactivity feedback loops are allowed.

Accuracy: Not given.

Floating/Fixed: Computation is in the floating decimal mode as described by G. R. Trimble in Technical Newsletter 8, pp. 37 - 41.

c) Mathematical Method: Integration is accomplished by the fourth order Runge-Kutta technique.

d) Storage Required: Approximately 1930 storage locations are required.

Speed: A representative problem using the full program takes about 1 hour.

Relocatability: Not relocatable.

e) Remarks: Recipes are provided to reduce to several special cases of physical interest. Directions are given to allow addition of one more feedback loop.

f) System: One 533 required.

Special Devices: None.

Speed: A representative problem using the full program takes about 4 minutes.

Relocatability: Not relocatable.

Storage: 450 locations.

Systems: One 533 required.

ANNUAL COST: $0.00

Special Notes: None.

9.2.019

IBM 650 Library Program Abstracts

FILE NUMBER 8.2.019

Physical Sciences

ARMOUR REACTOR KINETICS (ARK-I) CODE

T. Engelhart
W. E. Loewe
Armour Research Foundation of Illinois Institute of Technology
Chicago 16, Illinois

a) Purpose: This routine is used to obtain the transients of neutron flux in response to a change in reactivity of a nuclear reactor. The routine is a modification of the Savannah River Laboratory K-code (IBM 650 Library Program 8.2.018), from which it differs in the following respects: (1) driven changes in reactivity remain arbitrary functions of time, but must occur as a result of a change in the average neutron absorption cross section; (2) temperature coefficients are restricted to those affecting \( \Sigma_b \); (3) the feedback equations are slightly more general; and (4) a substantial saving in running time is realized. This last difference results from the fact that integration is accomplished by a fourth order Runge-Kutta technique.

b) Range: Six delayed groups of neutrons and two reactivity feedback loops are allowed.

Accuracy: Not given.

Floating/Fixed: Computation is in the floating decimal mode as described by G. R. Trimble in Technical Newsletter 8, pp. 37 - 41.

c) Mathematical Method: Integration is accomplished by the fourth order Runge-Kutta technique.

d) Storage Required: Approximately 1930 storage locations are required.

Speed: A representative problem using the full program takes about 1 hour.

Relocatability: Not relocatable.

e) Remarks: Recipes are provided to reduce to several special cases of physical interest. Directions are given to allow addition of one more feedback loop.

f) System: One 533 required.

Special Devices: None.
Aiken, a. Purpose: This code computes the distribution in energy from zero to d. b. Range: Not given. c. Mathematical Method: The equation for the conservation of neutrons is expressed in difference form as the matrix equation $N \cdot V = K \cdot V$, which is solved by iteration. d. Storage Required: The program consists of 204 instructions and one constant. e. Remarks: Transient terms, important for the first six periods only, are neglected. All material constants are fixed for any one run. The program is written in the Bell Telephone Laboratories L2 General Purpose System, IBM 650 Library Program 2. 0. 006. f. 650 System: One 533 required. Special Devices: None.

**IBM 650 Library Program Abstracts**

NEUTRON ENERGY SPECTRA IN WATER

J. C. English
E. I. du Pont de Nemours and Company
Aiken, South Carolina

a. Purpose: This code computes the distribution in energy from zero to 2.5 ev. It includes the effects of moderator motion and chemical binding.

b. Range: Not given.

c. Mathematical Method: The equation for the conservation of neutrons is expressed in difference form as the matrix equation $N \cdot V = K \cdot V$ which is solved by iteration.

d. Storage Required: Not given.

e. Remarks: The code as written assumes that the input parameters are in the range of those for $\text{H}_2\text{O}$ and $\text{D}_2\text{O}$ moderators.

f. 650 System: One 533 required.

Special Devices: None.

**IBM 650 Library Program Abstracts**

ENSIGN CODE

B. L. Anderson
H. Bohl, Jr.
Bettis Atomic Power Division
Westinghouse Electric Corporation
Pittsburgh 35, Pennsylvania

a. Purpose: ENSIGN is a few-group, one-dimensional code designed to handle asymmetric slabs, nonsymmetric slabs, and cylinders.

b. Range: Problems may not exceed 4 groups, 10 regions, and 100 points.

c. Mathematical Method: Fluxes and eigenvalues are computed by means of an iterative scheme in which it is necessary to make an initial guess. At either of the outer boundaries, there may be a flux of zero or a derivative of the flux equal to zero. The balance check method is used for crossing internal boundaries.

d. Storage Required: The program requires about 2000 words of storage.

e. Remarks: None.

**IBM 650 Library Program Abstracts**

RAYTHEON REACTOR SURVEY CODES 2G 2R, 2G 3R, AND 2G 3R

L. Holway
Research Division
Raytheon Manufacturing Company
Waltham, Massachusetts

a. Purpose: These routines will find the critical radius or the critical value of the infinite multiplication constant using two energy group diffusion theory in thermal reactors with two or three regions.

b. Range: Includes all values of core radius greater than 15 centimeters in 2G 2R and all values of $k_{eff}$ greater than 1.1 in 2G 3R and 2G 3R.

c. Mathematical Method: The continuity conditions joining the analytic solutions at a boundary produce a determinant which is solved by an iterative process for that value of the radius (2G 2R) or $k_{eff}$ (2G 3R and 2G 3R) which makes the determinant equal to zero.

d. Storage Required: Approximately 575 storage locations for 2G 2R and 2G 3R approximately 700 storage locations for 2G 3R.

e. Remarks: None.

f. 650 System: One 533 required.

Special Devices: None.

**IBM 650 Library Program Abstracts**

AN IBM 650 PROGRAM TO CALCULATE THE NEUTRON ATTENUATION IN A WATER-METAL REACTOR SHIELD

(Continued on next page)
IBM 650 Library Program Abstracts

MUL TIGROUP
J. C. English
Savannah River Laboratory
E. L. du Pont de Nemours & Co.
Aiken, S. C.

a. Purpose: This program solves the one-dimensional neutron diffusion equation by means of the associated difference equations in several energy groups. The program is essentially the WAPD "Lil' Abner" code rewritten for the Model 2 IBM 650. A grid in speed of a cell of five over "Lil' Abner" is retained.

b. Restrictions, Range: Floating point arithmetic is used.

c. Method: Difference equations which approximate the set of coupled differential equations

\[ \frac{d}{dx} \left( \sum_{g=1}^{G} D_{g} \phi_{g} \right) = \sum_{g=1}^{G} \sum_{i=1}^{m} S_{i} \phi_{g} \left( \chi_{g}^{i} / \chi_{g} \right) - \sum_{g=1}^{G} \sum_{i=1}^{m} \sum_{j=1}^{m} \frac{1}{\sum_{g=1}^{G} D_{g} \chi_{g}^{i} / \chi_{g} \chi_{g}^{j} / \chi_{j}} \phi_{g} (\xi_{g}^{i} / \xi_{g}^{j}) \left( \phi_{g}^{i} / \phi_{g}^{j} \right) \]

are used to obtain the solution for each speed group. Here \( \phi_{g} \) is the transverse building, \( i \) is the group index, \( D_{g} \), and \( \chi_{g} \) are the diffusion constants, absorption cross section, and the removal cross section respectively.

d. Storage Requirements: Approximately 1750 storages are required, including input data allocation. The program is supplied in SOAP II format and deck.

e. Remarks: Requires automatic floating decimal feature and index registers.

f. IBM 650 System: Not given.

IBM 650 Library Program Abstracts

LOC SURFACE FITTING PROGRAM FOR BASIC 650
W. C. Krumbein
Department of Geology
Northwestern University
Evanston, Illinois

a. Purpose: To fit linear, quadratic, and cubic surfaces to map data where the points of observation are distributed irregularly over the map area, rather than on a rectangular grid.

b. Restrictions, Range: The program handles as many as four mapped variables at a time for an indefinite number of map points, inasmuch as the computations are in floating point.

c. Method: Least squares polynomial fitting.

Speed: Part I computes basic 10 x 10 cubic matrix and four 10 x 1 vectors at the rate of one data card per 9 seconds. The output is in the form of 10 x 10, 6 x 1, and 3 x 3 matrices and their corresponding vectors.

Part II computes 6 x 6, and 3 x 3 matrices and computes the coefficients at the rate of 10 minutes per mapped variable.

Part III computes 3 curves, points per data card every 6 seconds (Observed values, computed values, and deviations). Some of squares vanish at end.

e. Remarks: Full description of data and output cards in program write-up.

f. IBM 650 System: Basic 650 and 533.
STRUCTURE FACTORS

B. Shiono
University of Pittsburgh
Pittsburgh 13, Pa.

a. Purpose: The programs compute structure factors of triclinic, monoclinic and orthorhombic space groups. The output cards of these programs are used as the input cards for "Differential Fourier Synthesis" program (File No. 8.4.001). Six individual programs were prepared for triclinic and non-centric space groups of the three classes respectively, and the modifications for any particular space group are made by addition of a few cards.

b. Range: The following upper limits are given:

- Number of independent atoms (at a time) 50
- Number of different kinds of atoms 8
- Number of temperature factors:
  1. Isotropic temp. factor for each kind 8
  2. Individual anisotropic temp. factor 50
- Number of different kinds of atoms 50
- Number of reflexions:
  1. Centrosymmetric no limit
  2. Non-centrosymmetric 99

Accuracy: Not given.

Floating/Fixed: Fixed point.

Mathematical Method: Geometrical structure factors are computed by R. W. Blaylock and J. M. Kibbee.

Storage Required: Most of the 1000 storage locations are used.

Speed: The following examples of speed are given:

- P 2 1 2 1
  9 atoms, 2 kinds
  3.5 sec/reflexion
- P 2 1 2 1
  7 atoms, 7 kinds
  8 sec/reflexion
- P 1
  28 atoms, 2 kinds, anisotropic temp. factors
  20 sec/reflexion

Relocatability: Not given.

Remarks: The necessary modification cards for each space group are listed. The programs are written in SOAP.

IBM 650 System: One 533 required.

 RIIBM 650 Library Program Abstracts

The following corrections have been submitted in the listing of the writing of the above program:

PAGE LOCATION LINE WORD WORD
60 0437 233 09 9191 0493 should be 65 9193 0994
60 0322 200 02 0314 0404 should be 65 3156 0442

Differential Fourier Synthesis

B. Shiono
University of Pittsburgh
Pittsburgh 13, Pa.

a. Purpose: This program uses the output cards from the program "Structure Factors" (File No. 8.4.001) as the input cards. It computes the electron densities, their nine derivatives of observed and calculated structure factors, and solves the shift from them. The modifications for each space group are made by the addition of a few cards.

b. Range: There is no limit to the number of reflexions.

Accuracy: Not given.

Floating/Fixed: Fixed point.

(Continued on next column)
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., 1952.)

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 6 members can meet at any given point.

f) Minimum 650.

**TRUSS ANALYSIS**

A. A. Aucoin
J. D. Hutchinson
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 1125 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.

**CONNECTOR AND REDUNDANCY PROGRAMS**

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 1125 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.
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.


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.

(Continued on next column)

b) Will solve any system with up to 192 zones. All data is in fixed point.


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
R. 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. Humphries.

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 stresses due to vertical as well as horizontal loading, and soil pressure for ( ) positions along a pile divided into i sections (i ≤ 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.
IBM 650 Library Program Abstracts

**PROGRAM ABSTRACT**

**FILE NUMBER** 9.2.013

**Title:** Analysis of Laterally Loaded Piles, by C. R. Seider, Sr., C. R. Boley, E. J. Opara.

The following correction has been submitted for the listing of the writeup. Page 10, location 0974, should be changed from 10 1411 0974 + to 11 1411 0974 +.

This change affects only those cases where the slope of the pile at the top is other than zero.

Also note that the one per card listing is the writeup should be ignored. Only the five per card deck listing should be considered reliable.

---

**650 LIBRARY PROGRAM ABSTRACT**

**FILE NUMBER** 9.2.015

**REVISED TRAVERSE AND TRAVERSE ADJUSTMENT COMPUTATION**

**J. A. Haller**
California Division of Highways
Sacramento, California

**a)** This program 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 64 for sine, and cosine, arc tangent, and arc sine.

**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 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.

---

**650 LIBRARY PROGRAM ABSTRACT**

**FILE NUMBER** 9.2.016

**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 1150 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.

---

**650 LIBRARY PROGRAM ABSTRACT**

**FILE NUMBER** 9.2.017

**FREeways 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.

---

**650 LIBRARY PROGRAM ABSTRACT**

**FILE NUMBER** 9.2.018

**CURVED BRIDGE PROGRAM**

**Texas Highway Department**
Austin, Texas

**a)** This program receives the detailer of much of the laborious computations 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 stops.

**e)** Only 30 bents may be computed at one time. The values of radius are limited to less than 10,000. Other limitations given in write-up.

**f)** Minimum 650.

---

**IBM 650 Library Program Abstracts**

**COMPOSITE BEAM**

**B. E. Shields**
**J. A. Haller**
California Division of Highways
Sacramento, California

(Continued on next page)
TRAVEL AND COORDINATE PROGRAM

K. F. Kohler
E. R. DeClerk
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 bearers, Stations, Length of Courses, Course, Lat., and Dept., and Coordinates of angle points are computed. Using P.1. 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, Bearers N. or S., DDMMSS E. or W., Stations DDMMSS, Deflection Angles DDMMSS E. or L., Delta Angles DDMMSS, P.1. Numbers PP, PPP, and Course Lengths LLL, LLL, LL, L, LLL, LL, when using coordinates as input. The subroutine used are SC-5 (Square Root), SC-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 aseasonic device.

IBM 650 Library Program Abstracts

FILE NUMBER 9.2.022

EARTHWORK LINE SHIFT

C. Travis
S. R. Cash
Computer Section
Washington Department of Highways
Olympia, Washington

a) Purpose: Shifts the center line on earthwork cross-section and interpolates a rod for the new center line if the new center line is located at a point for which a rod reading was given.

b) Range: Makes both left and right shifts of any size which will not cause the final distances to exceed four digits.

c) Mathematical Method: The interpolation is a straight line interpolation.

d) Storage Required: 416 drum locations.

e) Speed: Program runs at almost punch speed.

f) Floating/Fixed: The program is in fixed point arithmetic.

g) Special Devices: None.

IBM 650 Library Program Abstracts

FILE NUMBER 9.2.023

SPEED CHECK ANALYSIS

C. Travis
Computer Section
Washington Department of Highways
Olympia, Washington
(Continued on next page)
SLOPE TOPOG PROGRAM

K. F. Kohler
R. R. DeClark
Bureau of Public Roads
Portland, Oregon

a. Purpose: Converts cross section slope topog (slope in percent or degrees) to percent or degrees.

b. Range: Handles speeds from 5 to 80 MPH with as many observations as desired. Six groups may be read in for each station.


Standard deviation by the following equation:

\[ \sigma = \sqrt{\frac{1}{N-1} \sum (y_i - \bar{y})^2} \]

Variance = \( \sigma^2 \)

d. Storage Required: Program leaves 329 locations available.

- Speed: Requires about 2 minutes per problem.
- Relocatability: Program is non-relocatable.

- Remarks: Self loading five instructions per card deck is available.

- IBM 650 System: One 533 required.

- Special Devices: None.

---

SURVEY TRAVERSE PROGRAM

S. E. LaMarcha
Ohio Department of Highways
Columbus, Ohio

a. Purpose: Using as input the following survey traverse information:

1) Course length
2) Course angle
3) Bearing
4) Deflection
5) Asimuth

The program computes and supplies as output the latitude, departure, station coordinates, and components of closure error.

b. Range: In the case of a closed traverse, the number of courses must be less than one hundred.

(Continued on next page)
Accuracy: Output data is accurate to the nearest one-tenth foot.
Floating/Fixed: Computation is made in fixed point arithmetic.

b. Mathematical Method: The angle is first converted to an azimuth and then
added to the previous sum. Latitude and departures are computed with the
use of the sine-cosine subroutine, SC 2.

c. Storage Required: Memory locations 1 - 50 and 200 - 600 approximately,
are used.

Storage: Speed: Speed is approximately the maximum for card reading and punch­
ing.

Relocatability: The program is relocatable.

e. Remarks: None.

f. 650 System: One 533 required.
Special Devices: None.

IBM 650 Library Program Abstracts

ROD READING CONVERSION PROGRAM

M. Gold
Ohio Department of Highways
Columbus, Ohio

a. Purpose: The program reduces rod readings to elevations for use in the
Road Design Program IBM 650 Library Program 9.2.029.

b. Range: The maximum X value is 999.9 feet. The maximum B value is
99.9 feet.

Accuracy: Values are rounded to the nearest tenth from the field notes.

Floating/Fixed: The decimal is fixed in all calculations.

Mathematical Method: Simple arithmetic is used.

Storage Required: 348 memory locations in the first eight bands of the
Drum.

Speed: Data is processed at card reading speed.

Relocatability: The program is relocatable in multiples of fifty.

Remarks: None.

f. 650 System: One 533 required.
Special Devices: None.

IBM 650 Library Program Abstracts

ROAD DESIGN PROGRAM

B. T. Wade
Ohio Department of Highways
Columbus, Ohio

a. Purpose: Computes coordinates of the road design template from the
shoulder to the slope stakes according to design criteria.

b. Range: The range of input is as follows: 0.00 to 5 station 2 999,999.99;
200 to 2 e of 999.99 0.05 elevation; 6,999.98; 0.005 profile grade 5 999.99; 0.005
shoulder slope 5 99.99; 0.05 ditch slopes 5 9.9.
The range of the output is the same as input except that elevations are not
punched but rather distances above or below profile grade which have the
same range as the offsets.

Accuracy: Values are computed to the nearest tenth foot.

Floating/Fixed: Values are computed in fixed point arithmetic.

c. Mathematical Method: The methods used incorporate analytical geometry
plus comparisons at design criteria.

d. Storage Required:

0000 - 0399 Tables
0400 - 1719 Program
1820 - 1999 Constant and temporary storage locations.

LD 1 occupies 2900 - 1999 but is wiped out by the program.

(Continued on next column)
**IBM 650 Library Program Abstracts**

**DESIGN TEMPLATE PROGRAM**

C. R. Caylor  
Ohio Department of Highways  
Columbus, Ohio

- **Purpose**: Computes the design template for any given station.
- **Range**: The maximum X value is 1000 feet. The maximum Y value is 10,000 feet.
- **Accuracy**: The coordinates are computed to the nearest tenth of a foot.
- **Floating/Fixed**: Computation is in fixed point arithmetic.
- **Mathematical Method**: Trigonometry.
- **Storage Required**: 1099 consecutive memory locations.
- **Speed**: Not given.
- **Relocatability**: Program is relocatable by multiples of fifty.
- **Remarks**: Self-loading five instructions per card deck is available. Written in SOAP using 650.

**MOMENT DISTRIBUTION AND INFLUENCE LINE CALCULATION**

P. Yengay  
L. C. McReynolds  
E. D. Lee  
Computer Section  
Washington State Highway Department  
Olympia, Washington

- **Purpose**: Computes final end moments in beams and columns tops of single story continuous frames. The beams may be integral with the columns. Computes influence line ordinates for loads at the 0.3, 0.5, and 0.7 points. These ordinates are to be computed. When influence line ordinates are to be computed, a table of fixed end moment coefficients must be supplied only if the beams are not prismatic.
- **Range**: Will distribute fixed end moments for any single story continuous frame structure with up to 15 spans. This program will also compute influence line ordinates for a structure with up to 5 spans.
- **Accuracy**: Not given.
- **Floating/Fixed**: All data is in fixed point.
- **Mathematical Method**: Uses the Hardy Cross method of moment distribution.
- **Storage Required**: Program occupies 1869 positions of memory storage.
- **Speed**: Not given.
- **Relocatability**: Program is not relocatable.
- **Remarks**: Self-loading five instructions per card deck is available. Written in SOAP.

**SUSPENSION BRIDGE ANALYSIS**

- **Purpose**: Computes deflections, moments and shears in stiffening truss columns and the anchor spans, side spans suspended or not suspended.
- **Range**: Computes values for three span suspension bridge with or without anchor spans.
- **Accuracy**: Not given.
- **Floating/Fixed**: Input and output is in floating point.
- **Mathematical Method**: Uses Exact (Triple Series) Method wherein deflected structure is represented by a Fourier series.
- **Storage Required**: Program is split into two parts with 1218 locations available in the first part and 49 locations available in the second part.
- **Speed**: Speed is approximately 15 minutes for the first loading and 12 minutes for successive loadings.
- **Relocatability**: Not given.
- **Remarks**: Self-loading 5 instruction per card deck is available. Written in SOAP using 650.

**APPORXIMATION OF FUTURE TRIP TRANSFERS**

E. A. Radstiff  
California Division of Highways  
Sacramento, California

- **Purpose**: The program utilizes the Fratar Method to compute one or more successive approximations of future trip transfers between zones. Input data consist only of a set of initial trip transfers and (per zone) trip end growth factors. Trip transfers will be approximated for all pairs of zones up to a maximum of 70 zones.
- **Range**: Initial and approximated trip transfers have a range up to 9999.9 but any transfer which is initially zero will remain zero. Growth factors may range up to 99,999. Initial or approximate trip ends (per zone) may not exceed 100,000.
- **Accuracy**: Not given.
- **Floating/Fixed**: Fixed point arithmetic is used throughout.
- **Storage Required**: Essentially the entire drum is used by the program. Only 460 locations are used for instructions or constants, but 1488 fixed locations are required for storage of data.
- **Speed**: Time for loading and punching blocks in normal machine speed. Calculation time varies with the number of zones (N) and the number of non-zero initial trip transfers (M). A rough time formula (in minutes) is \[(5N^2 + 38) + (7M)\] per approximation.
- **Remarks**: All data are first loaded and then one or more approximations may be obtained (in succession at the programmer's option). Optional percentage criteria (in terms of approximated trip ends as compared to expected trip ends) are available to define the standard of accuracy of the final approximations.

**IBM 650 Library Program Abstracts**

Firm 9.2.022  
Engineering Applications

**APPROXIMATION OF FUTURE TRIP TRANSFERS**

- **Purpose**: Computes deflections, moments and shears in stiffening truss columns and the anchor spans, side spans suspended or not suspended.
- **Range**: Computes values for three span suspension bridge with or without anchor spans.
- **Accuracy**: Not given.
- **Floating/Fixed**: Input and output is in floating point.
- **Mathematical Method**: Uses Exact (Triple Series) Method wherein deflected structure is represented by a Fourier series.
- **Storage Required**: Program is split into two parts with 1218 locations available in the first part and 49 locations available in the second part.
- **Speed**: Speed is approximately 15 minutes for the first loading and 12 minutes for successive loadings.
- **Relocatability**: Not given.
- **Remarks**: Self-loading 5 instruction per card deck is available. Written in SOAP using 650.

**SUSPENSION BRIDGE ANALYSIS**

- **Purpose**: Computes deflections, moments and shears in stiffening truss columns and the anchor spans, side spans suspended or not suspended.
- **Range**: Computes values for three span suspension bridge with or without anchor spans.
- **Accuracy**: Not given.
- **Floating/Fixed**: Input and output is in floating point.
- **Mathematical Method**: Uses Exact (Triple Series) Method wherein deflected structure is represented by a Fourier series.
- **Storage Required**: Program is split into two parts with 1218 locations available in the first part and 49 locations available in the second part.
- **Speed**: Speed is approximately 15 minutes for the first loading and 12 minutes for successive loadings.
- **Relocatability**: Not given.
- **Remarks**: Self-loading 5 instruction per card deck is available. Written in SOAP using 650.

**APPORXIMATION OF FUTURE TRIP TRANSFERS**

- **Purpose**: The program utilizes the Fratar Method to compute one or more successive approximations of future trip transfers between zones. Input data consist only of a set of initial trip transfers and (per zone) trip end growth factors. Trip transfers will be approximated for all pairs of zones up to a maximum of 70 zones.
- **Range**: Initial and approximated trip transfers have a range up to 9999.9 but any transfer which is initially zero will remain zero. Growth factors may range up to 99,999. Initial or approximate trip ends (per zone) may not exceed 100,000.
- **Accuracy**: Not given.
- **Floating/Fixed**: Fixed point arithmetic is used throughout.
- **Storage Required**: Essentially the entire drum is used by the program. Only 460 locations are used for instructions or constants, but 1488 fixed locations are required for storage of data.
- **Speed**: Time for loading and punching blocks in normal machine speed. Calculation time varies with the number of zones (N) and the number of non-zero initial trip transfers (M). A rough time formula (in minutes) is \[(5N^2 + 38) + (7M)\] per approximation.
- **Remarks**: All data are first loaded and then one or more approximations may be obtained (in succession at the programmer's option). Optional percentage criteria (in terms of approximated trip ends as compared to expected trip ends) are available to define the standard of accuracy of the final approximations.

**SUSPENSION BRIDGE ANALYSIS**

- **Purpose**: Computes deflections, moments and shears in stiffening truss columns and the anchor spans, side spans suspended or not suspended.
- **Range**: Computes values for three span suspension bridge with or without anchor spans.
- **Accuracy**: Not given.
- **Floating/Fixed**: Input and output is in floating point.
- **Mathematical Method**: Uses Exact (Triple Series) Method wherein deflected structure is represented by a Fourier series.
- **Storage Required**: Program is split into two parts with 1218 locations available in the first part and 49 locations available in the second part.
- **Speed**: Speed is approximately 15 minutes for the first loading and 12 minutes for successive loadings.
- **Relocatability**: Not given.
- **Remarks**: Self-loading 5 instruction per card deck is available. Written in SOAP using 650.

**APPORXIMATION OF FUTURE TRIP TRANSFERS**

- **Purpose**: The program utilizes the Fratar Method to compute one or more successive approximations of future trip transfers between zones. Input data consist only of a set of initial trip transfers and (per zone) trip end growth factors. Trip transfers will be approximated for all pairs of zones up to a maximum of 70 zones.
- **Range**: Initial and approximated trip transfers have a range up to 9999.9 but any transfer which is initially zero will remain zero. Growth factors may range up to 99,999. Initial or approximate trip ends (per zone) may not exceed 100,000.
- **Accuracy**: Not given.
- **Floating/Fixed**: Fixed point arithmetic is used throughout.
- **Storage Required**: Essentially the entire drum is used by the program. Only 460 locations are used for instructions or constants, but 1488 fixed locations are required for storage of data.
- **Speed**: Time for loading and punching blocks in normal machine speed. Calculation time varies with the number of zones (N) and the number of non-zero initial trip transfers (M). A rough time formula (in minutes) is \[(5N^2 + 38) + (7M)\] per approximation.
- **Remarks**: All data are first loaded and then one or more approximations may be obtained (in succession at the programmer's option). Optional percentage criteria (in terms of approximated trip ends as compared to expected trip ends) are available to define the standard of accuracy of the final approximations.

**SUSPENSION BRIDGE ANALYSIS**

- **Purpose**: Computes deflections, moments and shears in stiffening truss columns and the anchor spans, side spans suspended or not suspended.
- **Range**: Computes values for three span suspension bridge with or without anchor spans.
- **Accuracy**: Not given.
- **Floating/Fixed**: Input and output is in floating point.
- **Mathematical Method**: Uses Exact (Triple Series) Method wherein deflected structure is represented by a Fourier series.
- **Storage Required**: Program is split into two parts with 1218 locations available in the first part and 49 locations available in the second part.
- **Speed**: Speed is approximately 15 minutes for the first loading and 12 minutes for successive loadings.
- **Relocatability**: Not given.
- **Remarks**: Self-loading 5 instruction per card deck is available. Written in SOAP using 650.
GENERAL FREEWAY ASSIGNMENT

M. Brubaker
R. Bieber
California Division of Highways
Sacramento, California

a. Purpose: The purpose of this routine is to compute time and distance on a freeway system and then compare it to an existing system to determine if the proposed freeway system would be adequate.

b. Range: The routine can handle any ten routing cards per routing.

c. Mathematical Method: Does not apply.

d. Storage Required: The entire drum is used. 100 locations are used to store cumulative time and distances between zones. For problems not requiring this many zonal interchanges, additional locations can be made available.

e. Accuracy: Not given.

f. Remarks: Total vehicle miles and minutes for each alternate processed are punched out at the end of the problem by the use of the end of file card. The program was written in SOAP.

Special Devices: Alphabetic device.

IBM 650 Library Program Abstracts

"General Freeway Assignment," by M. Brubaker and R. Bieber.

The following additions should be made to the program write-up:

An error has been discovered in the Freeway Assignment Program due to rounding the computed trips assigned to the basic best freeway and second best freeway routes.

In block 430 of the program the trips assigned to the basic route were computed by multiplying the per cent time by the number of trips and rounding the result. The trips assigned to the second best freeway route were obtained by subtracting the sum of the basic and second best assignment from the total number of trips. This was done to ensure assigning all the trips and never to assign more than the total number of trips. However, if all of the trips fall into the two computed categories and values are such that each computation is rounded up by one half a trip, the two computed categories have one more than the total number of trips assigned to the best freeway route.

This was done to insure assigning all the trips and never to assign more than the total number of trips.

In Block #4 the sum of the basic and second best assignment is rounded up by one half a trip and the number of trips assigned to the best freeway trips becomes a minus 1. The following corrections should be made in the program to use decimal accumulation and avoid the result stated above.

Delete from the program the following instructions:

Block Card Code Loc. Instruction
430 431 0 1702 65 118 1552
430 431 0 1652 16 128 1452
430 431 0 1602 16 128 1452
430 431 0 1552 20 1027 1412
430 431 0 1502 65 1146 1152
430 431 0 1452 16 1030 1504
430 431 0 1402 16 1030 1504
430 431 0 1352 16 1030 1504
430 431 0 1302 16 1030 1504
430 431 0 1252 16 1030 1504
430 431 0 1202 16 1030 1504
430 431 0 1152 16 1030 1504
430 431 0 1102 16 1030 1504
430 431 0 1052 16 1030 1504
430 431 0 1002 16 1030 1504
430 431 0 0 1052 20 1031 1877

IBM 650 Library Program Abstracts

LOADOMETER W-6 TABLE

J. H. Harbour
California Division of Highways
Sacramento, California

a. Purpose: Edit data and calculate per cent of overload on total weight and each axle of trucks and truck combinations with one or more axles 10,000 pounds or more, and single unit trucks weighing 13 tons or more per California Wheel Base Law and "AASHO," American Association of State Highway Officials recommendations.

b. Range: A minimum of 7 axles per vehicle.
Accuracy: Per cent deviation to 1/10 of one per cent which is converted to code.
Floating/Fixed: Fixed decimal point.
Mathematical Method: Arithmetic.
Storage Required: 2000-word drum.
Speed: Approximately 700 operations per hour.
Relocatability: Not given.
Remarks: Minor changes in program may be required subject to changes in State Wide Base Law and "AASHO", American Association of State Highway Officials, recommendations.
IBM 650 System: One 533 required.
Special Devices: None.

STRESS ANALYSIS OF OPEN-WEB STRUCTURES

C. W. Zahler
United States Steel Corporation

J. E. O'Keefe
American Bridge Division
Pittsburgh, Pennsylvania

a. Purpose: Several specific computer programs concerned with obtaining the axial stresses in members of an open-web system, together with their relative geometry, provide a basis for a brief sketch of the various phases of development of the system from conception to utilization.
b. Range: Simple web, 99 panels; Subdivided, 65 panels; "R" type, 88 panels.
Accuracy: All lengths and distances are computed to three decimal places.
Mathematical subroutines include:
SINE, COSINE, SINH, COSH, LOG, ARCSINE, ARCTAN, \sqrt{N}, \sqrt{X}.
In the right triangle a, b, c, any of the following are computed, with or without their natural functions: \sqrt{a^2 + b^2}, \sqrt{a^2 - b^2}, \sqrt{a^2 + b^2 + c^2}.
Also, \sqrt{a^2}, \sqrt{b^2}, \sqrt{a^2 - b^2}.
c. Mathematical Method: Coordinate transformations and trigonometry are used.
d. Storage Required: HA-1, 2, 3, and 4 are loaded together. There are 533 required.
Relocatability: Not relocatable.
e. Remarks: Minor changes in program may be required subject to changes in State Wide Base Law and "AASHO", American Association of State Highway Officials recommendations.
f. IBM 650 System: One 533 required.

IBM 650 Library Program Abstracts

DIGITAL TERRAIN MODEL SYSTEM HORIZONTAL ALIGNMENT PROGRAMS

a. Purpose: HA-1, DTM Basic Horizontal Alignment Program: Computes the geometry of a highway centerline defined by coordinates of P, I, "s", and the radii of the curves, relates the DTM Terrain Data Sections to this centerline and computes the terrain elevation at the centerline for each section.
HA-2, DTM Even Station Interpolation Program: Takes the centerline terrain elevations (which are on odd centerline stations) and interpolates for elevations on even stations.
HA-3, DTM Parallel Offset Alignment Program: Takes the same input as HA-1, includes the same output but also computes the data for two parallel offset lines.
HA-4, DTM Special Alignment Geometry Program: The same as HA-1 except that it computes only centerline geometry. It can be used independently of the DTM System.

b. Range: Maximum number of horizontal curves is 50. Maximum number of points per cross section is 200.
Accuracy: All lengths and distances are computed in three decimal places.
Floating/Fixed: Fixed point arithmetic is used.
Mathematical Method: Coordinate transformations and trigonometry are used.
Storage Required: HA-1, 2, 3, and 4 are loaded together. There are 200 locations available.
Speed: Not given.
Relocatability: Not given.
Remarks: HA-3 and HA-4 are options of HA-1. HA-2 is a separate program but is loaded with HA-1.
IBM 650 System: One 533 required.

IBM 650 Library Program Abstracts
### IBM 650 Library Program Abstracts

**DIGITAL TERRAIN MODEL SYSTEM PRELIMINARY EARTHWORK PROGRAM EW-2**

Massachusetts Department of Public Works
C. L. Miller
R. Bieber

**Purpose:** This is the basic program for computing earthwork quantities in location studies. A simplified template is used for the efficient evaluation of a number of trial lines. The input is the template definition data, the DTM terrain data deck, and the output of the DTM VA-1 program. The output is the template definition data for each section and the volumes at each section.

<table>
<thead>
<tr>
<th>Range:</th>
<th>Maximum number of points per cross section is 200.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy:</td>
<td>Volumes are computed to the nearest cubic yard.</td>
</tr>
<tr>
<td>Floating/Fixed:</td>
<td>Fixed point arithmetic is used.</td>
</tr>
<tr>
<td>Mathematical Method:</td>
<td>The average end area method is used to compute the volumes.</td>
</tr>
<tr>
<td>Storage Required:</td>
<td>VA-1 and VA-2 are loaded together and use 600 locations.</td>
</tr>
<tr>
<td>Speed:</td>
<td>Not given.</td>
</tr>
<tr>
<td>Relocatability:</td>
<td>Not in relocatable form.</td>
</tr>
<tr>
<td>Remarks:</td>
<td>None.</td>
</tr>
<tr>
<td>IBM 650 System:</td>
<td>One 533 required.</td>
</tr>
</tbody>
</table>

---

**SAN DIEGO FREEWAY ASSIGNMENT**

M. Brubaker
R. Bieber
California State Division of Highways
Sacramento, California

| Purpose: | This routine computes time and distance on a freeway system and compares this data with that of a basic system to determine whether the proposed freeway system would be adequate. |
| Range: | Not given. |
| Accuracy: | Not given. |
| Floating/Fixed: | Fixed point arithmetic is used. |
| Mathematical Method: | Not applicable. |
| Storage Required: | The entire drum is used. Cumulative time and distance between zones are stored in 1599 locations. For a problem not requiring many zone interchanges, additional locations can be made available to the routine. |
| Speed: | Not given. |
| Relocatability: | Not relocatable. |

---

**IBM 650 Library Program Abstracts**

**Notes:**

Addenda/Errata


The following additions should be made to the program write-up:

An error has been discovered in the Freeway Assignment Program due to rounding the computed trips assigned to the basic best freeway and second best freeway routes.

In Block 430 of the program the trips assigned to the basic route were computed by multiplying the per cent times the number of trips and rounding the result. The trips assigned to the second best freeway route were obtained in the same manner. Trips assigned to the best freeway route were obtained by subtracting the sum of the basic and second best assignment from the total number of trips. This was done to insure assigning all the trips and never to assign more than the total number of trips. However, if all of the trips fall into the two computed categories and values are such that such computation is rounded up by one half of a trip, the two computed categories have one more than the total trips to be assigned, and the number of trips assigned to the best freeway trips becomes minus 1. The following corrections should be made in the program to use decimal accumulation and avoid the result stated above.

Delete from the program the following instructions:

<table>
<thead>
<tr>
<th>Block</th>
<th>Card</th>
<th>Code</th>
<th>Loc.</th>
<th>Instruction</th>
</tr>
</thead>
<tbody>
<tr>
<td>430</td>
<td>010</td>
<td>0</td>
<td>1478</td>
<td>69 1109 1052</td>
</tr>
<tr>
<td>420</td>
<td>035</td>
<td>0</td>
<td>1852</td>
<td>24 1931 1994</td>
</tr>
<tr>
<td>410</td>
<td>046</td>
<td>0</td>
<td>1994</td>
<td>69 1427 1280</td>
</tr>
<tr>
<td>430</td>
<td>070</td>
<td>0</td>
<td>1461</td>
<td>60 0002 1356</td>
</tr>
<tr>
<td>420</td>
<td>081</td>
<td>0</td>
<td>1546</td>
<td>30 0002 1436</td>
</tr>
<tr>
<td>430</td>
<td>085</td>
<td>0</td>
<td>1646</td>
<td>15 1902 1996</td>
</tr>
<tr>
<td>430</td>
<td>085</td>
<td>0</td>
<td>1596</td>
<td>20 1902 1996</td>
</tr>
<tr>
<td>430</td>
<td>071</td>
<td>1</td>
<td>1509</td>
<td>21 1810 1145</td>
</tr>
<tr>
<td>430</td>
<td>341</td>
<td>0</td>
<td>1477</td>
<td>45 1370 1471</td>
</tr>
</tbody>
</table>

Add to the program the following instructions:

<table>
<thead>
<tr>
<th>Block</th>
<th>Card</th>
<th>Code</th>
<th>Loc.</th>
<th>Instruction</th>
</tr>
</thead>
<tbody>
<tr>
<td>430</td>
<td>010</td>
<td>0</td>
<td>1478</td>
<td>69 1109 1052</td>
</tr>
<tr>
<td>420</td>
<td>035</td>
<td>0</td>
<td>1852</td>
<td>24 1931 1994</td>
</tr>
<tr>
<td>410</td>
<td>046</td>
<td>0</td>
<td>1994</td>
<td>69 1427 1280</td>
</tr>
<tr>
<td>430</td>
<td>070</td>
<td>0</td>
<td>1461</td>
<td>60 0002 1356</td>
</tr>
<tr>
<td>420</td>
<td>081</td>
<td>0</td>
<td>1546</td>
<td>30 0002 1436</td>
</tr>
<tr>
<td>430</td>
<td>085</td>
<td>0</td>
<td>1646</td>
<td>15 1902 1996</td>
</tr>
<tr>
<td>430</td>
<td>085</td>
<td>0</td>
<td>1596</td>
<td>20 1902 1996</td>
</tr>
<tr>
<td>430</td>
<td>071</td>
<td>1</td>
<td>1509</td>
<td>21 1810 1145</td>
</tr>
<tr>
<td>430</td>
<td>341</td>
<td>0</td>
<td>1477</td>
<td>45 1370 1471</td>
</tr>
</tbody>
</table>

---

**IBM 650 Library Program Abstracts**

**Notes:**

Addenda/Errata

This is a revision of the block diagram for Block 430 to replace page 12 of the program write-up.

---

**IBM 650 System:** One 533 required.

**Special Devices:** Alphanumeric device, 10 extra punch selectors (for a total of 24) and 12 extra punches (for a total of 20) are required.
a. **Purpose:** The basic purpose of this program is to compute the coordinates of the point of intersection of a given line with a line offset a given distance from a Talbot spiral, the radial bearing at this point and the distance along the offset line from the beginning of the spiral. It will also compute the length and bearing of lines joining successive sets of coordinates. The coordinates developed in one problem may be stored for use in later problems.

b. **Range:** Only one spiral at a time may be used, but an unlimited number of problems based on this spiral may be calculated. An unlimited number of distances and bearings computations is possible.

c. **Accuracy:** Distances are given to thousands of a foot and bearings to seconds.

Floating/Fixed: Input and output are in fixed point; floating point is used within the program.

**Mathematical Method:** Intersection is found by iteration.

**Storage Required:** The program occupies 1762 storage locations.

**Speed:** The computations for each intersection require approximately 330 seconds. Distance and bearing computations proceed at about 30 per minute.

Relocatability: Not relocatable.

e. **Remarks:** The program is written in SOAP I form. It uses portions of SOAP I Intersective Routine, File No. 1, 6, 601.

f. **IBM 650 System:** One 533 required.

Special Devices: Alphabetic device required.

---

**IBM 650 Library Program Abstracts**

**FILE NO. 9, 2, 044**

**Title:** Earthwork Data Check

**Authors:** K. F. Kohler and R. R. DeClark

**Institution:** Bureau of Public Roads, Portland, Oregon

**Description:** This program indicates and locates all probable major errors, omissions or deviations contained in design earthwork data. When an error or significant deviation is detected, an error card is punched which indicates and locates the deviation or error.

**Range:** Minor errors are not detected. The break-point between major errors and minor errors may be designated by the design engineer. This program does not contain program stops. The amount of input or output is unlimited. The routine checks Earthwork Design Data Cards in any of the following arrangements:

1. Type "N" or "V" or "S" separately
2. Type "U" combined with type "1" or type "2"

**Accuracy:** Not given.

**Floating/Fixed:** Fixed point.

**Mathematical Method:** Simple arithmetic is used.

**Storage Required:** The program and data use 1960 storage locations.

**Speed:** The program operates at approximately 3/4 read speed, depending on the number of points in the section and the number of errors detected.

**Relocatability:** Not given.

**Remarks:** This program is designed to be used in conjunction with B.P.R. revised version of the IBM Library Program, File No. 9, 2, 044. Error cards contain the location of the error and a 20-character statement identifying the type of error.

**IBM 650 System:** One 533 required.

**Special Devices:** Alphabetic device required.

---

**IBM 650 Library Program Abstracts**

**FILE NO. 9, 2, 045**

**Title:** Talbot Spiral Intersections

**Authors:** J. Petersen

**Institution:** Washington Department of Highways, Olympia, Washington

**Abstract:** This routine computes gradients between points of Isoparametric curves developed in one problem may be stored for use in later problems.

**Range:** The program will handle up to 98 changes of grade.

**Accuracy:** To hundredths for all factors except grade, which is to ten thousandths.

**Floating/Fixed:** Fixed point.

**Mathematical Method:** Intersection is found by iteration.

**Storage Required:** The program occupies 1762 storage locations.

**Speed:** The routine operates at full punch speed.

**Relocatability:** Not given.

**Remarks:** None.

**IBM 650 System:** One 533 required.

**Special Devices:** Ten extra pilot selectors (for a total of 20) are required.

---

**IBM 650 Library Program Abstracts**

**FILE NO. 9, 2, 046**

**Title:** Profile Grade

**Authors:** J. Oakes and T. L. Yates

**Institution:** Oregon State Highway Department, Salem, Oregon

**Description:** This routine computes gradients between P.T.'s and profile grade elevations for either defined incremental stations or selected stations. The program will compute for either plus or minus stations and in either ascending or descending order. It will handle both horizontal and vertical equations caused by changes in datum or differences in depth of surfacing.

**Range:** The program will handle up to $999,999.99$ changes of grade.

**Accuracy:** To hundredths for all factors except grade, which is to ten thousandths. Stationing may be selected to either the nearest foot or the nearest hundredth of a foot.

**Mathematical Method:** Standard.

**Storage Required:** The program requires approximately 1950 storage locations.

**Speed:** The routine operates at full punch speed.

**Relocatability:** Not given.

**Remarks:** None.

**IBM 650 System:** One 533 required.

**Special Devices:** Ten extra pilot selectors (for a total of 20) are required.

---

**IBM 650 Library Program Abstracts**

**FILE NO. 9, 2, 047**

**Title:** Contract Bid Computations

**Authors:** J. Oakes and T. L. Yates

**Institution:** Oregon State Highway Department, Salem, Oregon

**Description:** This routine checks the contractors' bid extensions and totals. It arranges the job bids in order by amount.

**Range:** Unit bids from $0.0001 to $999,999.99. Item and job totals up to $9,999,999,999. This routine can handle up to 95 items and 30 bidders per job.

**Accuracy:** As indicated above.

**Floating/Fixed:** Not given.

**Mathematical Method:** Does not apply.

(Continued on next page)
**Sacramento, California Division of Highways**

**TREND ANALYSIS AND PREDICTION**

R. A. Bieber  
California Division of Highways  
Sacramento, California

**Purpose:** This routine is designed to reapproximate values A and B for the equation \( Y = AB^X \) using an initial approximation obtained by other methods. A standard error of estimate is calculated from calculated \( Y \) using the new approximations. \( Y \) for future years is also calculated.

**Range:** The program is not designed to handle decreasing trends.

**Accuracy:** All output is in fixed point. Floating/Fixed: DOPSIR, the double-precision floating point routine, is used for nearly all mathematical operations.

**Mathematical Method:** The method of solution of normal equations is used but with modifications as to scaling of the \( X \) power. The standard error of estimate is calculated by the normal method.

**Storage Required:** The program, including DOPSIR, requires approximately 1750 storage locations.

**Speed:** The speed is relatively slow due to the use of DOPSIR. For analyzing 20 years of data plus predicting 10 years, approximately 3 minutes are required.

**Relocatability:** Not given.

**Remarks:** The program has been designed to handle reapproximations of its own approximations for up to three approximations, or until desired accuracy is obtained. Better the approximation used for input, the better the computed \( Y \)'s and standard error.

**IBM 650 System:** One 533 required.

---

**IBM 650 Library Program Abstracts**

### BID SUMMARIES

**T. L. Yates**  
Oregon State Highway Department  
Salem, Oregon

a. **Purpose:** This routine is designed to summarize the item and total bids on a job.

b. **Range:** See IBM 650 Library Program "Contract Bid Computations" (File No. 9.2.047).

c. **Accuracy:** Not given.

d. **Mathematical Method:** Does not apply.

e. **Storage Required:** This routine requires 1945 storage locations.

f. **Devices:** Alphabetic device; one read and punch speed.

**Purpose:** This routine is designed to reapproximate values \( A \) and \( B \) for the equation \( Y = AB^X \) using an initial approximation obtained by other methods. A standard error of estimate is calculated from calculated \( Y \) using the new approximations. \( Y \) for future years is also calculated.

**Range:** The program is not designed to handle decreasing trends.

**Accuracy:** All output is in fixed point numbers of at most ten figures.

Floating/Fixed: DOPSIR, the double-precision floating point routine, is used for nearly all mathematical operations.

**Mathematical Method:** The method of solution of normal equations is used but with modifications as to scaling of the \( X \) power. The standard error of estimate is calculated by the normal method.

**Storage Required:** The program, including DOPSIR, requires approximately 1750 storage locations.

**Speed:** The speed is relatively slow due to the use of DOPSIR. For analyzing 20 years of data plus predicting 10 years, approximately 3 minutes are required.

**Relocatability:** Not given.

**Remarks:** The program has been designed to handle reapproximations of its own approximations for up to three approximations, or until desired accuracy is obtained. Better the approximation used for input, the better the computed \( Y \)'s and standard error.

**IBM 650 System:** One 533 required.

---

**IBM 650 Library Program Abstracts**

### TIME SERIES TREND EQUATIONS

**R. A. Bieber**  
California Division of Highways  
Sacramento, California

**Purpose:** This program is designed to solve the equations \( Y = A + Bx \), \( \log Y = A + Bx \), and \( Y = AB^x \) for a value of \( A \) and \( B \) and using this value determine a \( Y \) for the years of trend plus some desired years in the future. In addition, a standard estimate of error is determined for each type of trend. The \( Y \)'s which are calculated may be punched out for each year or for any interval of years desired.

**Range:** The linear equation may be based on increasing or decreasing trends. The semilog equation may be based on increasing or decreasing trends along as long as the values of \( Y \) do not become negative. The exponential may only be solved for increasing trends.

**Accuracy:** The log and antilog routines used are accurate to \( 2 \times 10^{-7} \) and the square root routine is accurate to \( 10^{-5} \).

Floating/Fixed: DOPSIR, the double-precision floating point routine, is used. All output, however, is in fixed point.

**Mathematical Method:** The linear and semilog equations are solved by the method of least squares and the exponential is solved by a set of normal equations modified for flexibility.

**Storage Required:** The program requires the entire 2000 storage locations.

**Speed:** The time required for solving the three types of equations is approximately 4-1/2 minutes.

**Relocatability:** Not relocatable.

---

**Remarks:** The program has been designed to solve the three equations as a unit or in different combinations.

**IBM 650 System:** One 533 required.

---

**IBM 650 Library Program Abstracts**

### WATER SURFACE PROFILE PARAMETERS

**P. D. Douet**  
Soil Conservation Service  
U. S. Department of Agriculture  
Beltsville, Maryland

a. **Purpose:** This program computes the following:

1) The parameters used in the graphical solution of water surface profiles in natural streams for any discharge
2) Critical discharge
3) Cross-sectional area
4) Top width
5) Conveyance values based on Manning's formula.

b. **Range:** Top width of 9999 feet; hydraulic radius of 99 feet. A maximum of 20 profiles and 6 segments may be used to define the cross section. No two consecutive points defining the cross section may have the same elevation.

**Accuracy:** Vertical and horizontal distances may be given to the nearest 0.1 of a foot and 1.0 foot respectively.

Floating/Fixed: Not given.

c. **Mathematical Method:** Escoffier's method is modified to correct for changes in velocity head.

d. **Storage Required:** The program uses the entire 2000 storage locations.

**Speed:** The time in seconds for one cross section is approximately:

\[ T = 2a + bc \]

where \( a \) = number of points in cross section;
\( c \) = number of elevations for which the computer calculates a set of parameters.

**IBM 650 System:** One 533 required.

---

**IBM 650 Library Program Abstracts**

**TIME SERIES TREND EQUATIONS**

**R. A. Bieber**  
California Division of Highways  
Sacramento, California

**Purpose:** This program is designed to solve the equations \( Y = A + Bx \), \( \log Y = A + Bx \), and \( Y = AB^x \) for a value of \( A \) and \( B \) and using this value determine a \( Y \) for the years of trend plus some desired years in the future. In addition, a standard estimate of error is determined for each type of trend. The \( Y \)'s which are calculated may be punched out for each year or for any interval of years desired.

**Range:** The linear equation may be based on increasing or decreasing trends. The semilog equation may be based on increasing or decreasing trends along as long as the values of \( Y \) do not become negative. The exponential may only be solved for increasing trends.

**Accuracy:** The log and antilog routines used are accurate to \( 2 \times 10^{-7} \) and the square root routine is accurate to \( 10^{-5} \).

Floating/Fixed: DOPSIR, the double-precision floating point routine, is used. All output, however, is in fixed point.

**Mathematical Method:** The linear and semilog equations are solved by the method of least squares and the exponential is solved by a set of normal equations modified for flexibility.

**Storage Required:** The program requires the entire 2000 storage locations.

**Speed:** The time required for solving the three types of equations is approximately 4-1/2 minutes.

**Relocatability:** Not relocatable.

(Continued on next column)
IBM 650 Library Program Abstracts

AUTOMATIC MINIMUM WEIGHT DESIGN OF STEEL FRAMES
R. L. Stone
Division of Applied Mathematics
Brown University
Providence, Rhode Island

a. Purpose: Given the centerlines dimensions of a plane structure and the loads acting upon it, this program computes the bending moment distribution which minimizes the structural weight.

b. Range: Frames up to and including 3-bay, 4-storey or 6-bay, 3-storey.

Accuracy: Not given.

Floating/Fixed: Fixed point.

Mathematical Method: A method which was devised by J. Heyman and W. Prager of the Division of Applied Mathematics of Brown University.

Storage Required: The entire drum is used.

Speed: Varies considerably with the size of the frame being designed. The following examples are typical:
   1. A one-bay, one-storey frame was designed in 3 minutes.
   2. A two-bay, two-storey frame was designed in one hour and 45 minutes.
   3. A three-bay, three-storey frame was designed in slightly over 4 hours.

Relocatability: Not relocatable.

Remarks: The program is completely automatic, requiring no intermediate intervention by the operator. It consists of 15 subroutines (a total of about 2400 instructions).

IBM 650 System: One 533 required.

IBM 650 Library Program Abstracts

DPR REVISION OF OREGON HORIZONTAL ALIGNMENT PROGRAM
K. F. Kohler
C. L. Horst
Bureau of Public Roads
Portland, Oregon

a. Purpose: This program will compute curve and spiral data, and stationing and coordinates, for curve points of a projected alignment when the coordinates of the P. L.'s are scaled from a detail map and the degree of curve and length of spirals are assigned.

b. Range: Stationing (SSS/SS555), all distances, and coordinates are full normal range and to two decimal places; angles (DD000000) and bearing (DDMMSS) are either as indicated or selectable to the nearest 30 seconds or minutes.

Accuracy: Consistent with normal manual methods.

Floating/Fixed: Computations are in floating point; input and output are in fixed point.

Mathematical Method: Based on Talbot's spiral using "Arc" definition of circular curve.

Storage Required: Approximately 1888 storage locations are used.

Speed: Computing time is approximately 18 seconds per simple curve and 12 seconds per spiraled curve.

Remarks: The program is written in FOR (2.0, 0.01).

IBM 650 System: One 533 required.

Special Devices: Alphabetic device is required.

IBM 650 Library Program Abstracts

LAND AREA - SURVEY TRAVERSE

(Continued on next column)
THREE-POINT SOLUTION

D. Geister
Oregon State
b.

D. Geister
IBM 650 Library
MOMENT AND REACTION INFLUENCE LINE
b. Range: From three to nine known points are acceptable in the input data. The output will include every combination of three points.

Accuracy: Not given.
Floating/Fixed: Floating decimal.
Mathematical Method: Three-point solution; see the program write-up.
Remarks: Subroutines used in SIR.
Storage Required: 1,700 storage locations.
Speed: Not given.
Relocatability: Not given.

Remarks: Some precautions which should be observed are:
1. Negative information must be identified by a negative overpunch in the unit position of the appropriate input word.
2. A plus sign need not be punched for any value other than in the first word of data cards 3 and 4 (column 8). In these words, the overpunch serves to identify the card so having two words of information in it.
3. Of course, one cannot exceed the problem format. Any D distance cannot exceed 99,999 feet.

f. IBM 650 System: One 533 required.

IBM 650 Library Program Abstracts

MOMENT AND REACTION INFLUENCE LINE ORDNATE FOR SYMMETRICAL 1-SPAN OR 4-Span CONTINUOUS GIRDER BRIDGES

J. W. Chambers
C. Cook
b.

J. W. Chambers
B. Williams
Alabama State Highway Department
Montgomery, Alabama

Purpose: This program calculates moment and reaction influence line ordinates for symmetrical 1-span or 4-span continuous girder bridges with constant moment of inertia, or for symmetrical 3-span or 4-span continuous concrete girder bridges with parabolic haunches at the intermediate supports [with limitations as stated in program write-up].

Range: See the program write-up.
Accuracy: All machine calculations are rounded to five decimal places.
Floating/Fixed: Fixed decimal.

IBM 650 System: One 533 required.

IBM 650 Library Program Abstracts

STRAIGHT LINE BRIDGE GRID SYSTEM

D. L. Marxist
Ohio Department of Highways
Columbus, Ohio
(Continued on next column)

Purpose: This program computes the necessary information needed for designing a tangent bridge. The information calculated includes the following:
1. The station of a point.
2. The P. C. elevation of a point.
3. A longitudinal distance back to the preceding point.
4. A skewed distance along the centerline of a substructure element, from one point to the next succeeding point.
5. A final surface elevation.
6. A total skewed distance from a point to the centerline of survey.

Range: The maximum number of points on any substructure element is 20. Any number of substructure elements are allowed.

Accuracy: All calculations are accurate to at least three decimal places.
Floating/Fixed: Fixed decimal.
Mathematical Method: Elementary arithmetic, algebra and trigonometry.
Storage Required: The program requires the first 725 drum storage locations; subroutines included require about 350 additional locations.
Speed: The time required by the program is approximately as follows: 58 = 0.5s, where n is the number of points to be computed.
Relocatability: Not given.

Remarks: Some precautions which should be observed are:
1. Negative information must be identified by a negative overpunch in the unit position of the appropriate input word.
2. A plus sign need not be punched for any value other than in the first word of data cards 3 and 4 (column 8). In these words, the overpunch serves to identify the card so having two words of information in it.
3. Of course, one cannot exceed the problem format. Any D distance cannot exceed 99,999 feet.

f. IBM 650 System: One 533 required.
Special Devices: None required.

IBM 650 Library Program Abstracts

CIRCULAR CULVERT ANALYSIS

S. N. Doden
Ohio Department of Highways
Columbus, Ohio

Purpose: This program determines the proper method of analysis for a culvert acting under a given set of conditions and determines the most economical size of circular section.

Range: Maximum design discharge is 9999 cfs; maximum length of conduit is 999 feet. Circular pipe sizes analyzed by the program range from 12 in. to 108 in.

Accuracy: Not given.
Floating/Fixed: Fixed decimal arithmetic is used.
Mathematical Method: Primarily, algebra and trigonometry. Manning’s Equation is used to compute the hydraulic values of conduits flowing full. Chezy’s Formula is the basis for computing the hydraulic elements of partially full conduits.
Storage Required: 999 drum storage locations are reserved for tables, subroutines and loading routines; 1034 locations are required for the program. This leaves seven remaining storage locations; however, additional drum storage space may be found within the area reserved for the Square Root Subroutine.
Speed: This is a function of the type of analysis chosen by the program to simplify the hydraulic elements of the conduit.
Relocatability: Not relocatable.

Remarks: The program is primarily designed for checking culvert designs; however, an additional feature is included whereby a culvert may be designed providing certain conditions exist. SIR symbolic deck listing is included.

f. IBM 650 System: One 533 required.
Special Devices: Alphabetic device. However, the program can very easily be revised to operate without this device.

IBM 650 Library Program Abstracts

1-SPAN CURVED CONCRETE SLAB BRIDGE PROGRAM

B — 650
a. Purpose: This program is designed to generate and compute a station number; a profile grade elevation; an X and Y coordinate; and a final surface elevation for a number of specified and given points on the abscissas and plays of a 3-span curved concrete slab bridge.

b. Range: The range of the important portion of the input data is as follows:

For $R_3 - R_1$, incl. 0.01 ft. $R > 516256$, 06 ft.

$c. \theta - 120^\circ \leq \theta \leq 299^\circ - 599^\circ$,

where $D$ = Degree of Curvature

For $S_3 - S_1$, incl. 0.000 ft. 59.999

Accuracy: The accuracy of the station, the profile grade and the final surface elevations calculations are to 0.01 ft. of a foot. The X and Y coordinates are accurate to at least three decimal places.

Floating/Fixed: Computations are made in fixed decimal arithmetic.

c. Mathematical Method: Primarily, trigonometry is used. In block 21 of the flow diagram, there is a formula stated as $T_n = \frac{Y_n - Y_{n-1}}{X_n - X_{n-1}}$. There were several methods of computing $Y$ at this point. This method was chosen mainly for its ease of handling and its relative simplicity. Another way of accomplishing the same task might be to obtain $Y$ as the quotient of $X_n - X_{n-1}$, convert that to an angle $\theta$ in degrees, convert $\theta$ in degrees to radians, obtain the cosine and multiply by a particular radius.

There are two methods for computing the bridge limit on the center line of survey. The method that was used is discussed more fully in Section V of the write-up. The other method is similar to that used for the inner and outer guard rail lengths and is based on the fact that $R > 60$. 

Using this, we may compute D.L. Survey $= (\theta - R_n) R_n$. This is obviously the easier of the two but was discarded in lieu of the standard method to produce a more accurate answer.

**IBM 650 Library Program Abstracts**

**PROFILE GRADE**

S. E. LaMacchia
H. R. Sharp
Ohio Department of Highways
Columbus, Ohio

a. Purpose: This program computes elevations along the profile grade of a proposed highway for both tangent sections and vertical curves.

b. Range: The maximum number of stations and equations are as follows: 400.

Accuracy: Percent grade is accurate to the nearest 0.01 ft. Other values are accurate to the nearest 0.01 ft.

Floating/Fixed: Fixed decimal.


d. Storage Required: 1954 locations.

e. Speed: Not given.

f. Remarks: None.

IBM 650 System: One 533 required.

Special Devices: None.

**IBM 650 Library Program Abstracts**

**DIGITAL TERRAIN MODEL SYSTEM FOUR POINT POLYNOMIAL INTERPOLATION PROGRAM DA-2**

Massachusetts Department of Public Works
C. L. Miller
Photogrammetry Laboratory
Massachusetts Institute of Technology
Cambridge, Massachusetts

a. Purpose: This program interpolates centerline terrain elevations on even stations from a profile given on odd stations. Four point polynomial interpolation is used giving a better representation of the terrain than straight line interpolation (used in the DTM HA-1 Program, IBM 650 Library Program File Number 9.2, 649).

b. Range: 1. The increment between even stations may be any positive, nonzero number.

2. A profile having any number of points may be used.

c. Accuracy: The output has as many significant digits as the input.

Floating/Fixed: Fixed decimal arithmetic is used.

d. Mathematical Method: Aitken's method of iteration is used to compute the polynomial.

e. Storage Required: About 200 locations are required for program and storage. However, the program is spread over locations 0000 to 1200 and uses the read and punch areas in the 1530 band.

f. Remarks: The program has been written to use a standard DTM card format and the standard DTM control panel. However, the program is not dependent on control panel wiring and any card format may be used providing a corresponding control point is used.

IBM 650 System: One 533 required.

Special Devices: None.
b. Range: This program was written for bridges having spans of from 15 to 200 ft.
   Accuracy: Moments are generally accurate to 0.1 ft-kip. Shears are generally accurate to 0.1 kip.
   Floating/Fixed: Fixed decimal.

c. Mathematical Method: Influence lines are used to calculate end moments, and each span is then treated as a free body.

d. Storage Required: All 533 locations.
   Speed: The program will compute approximately 29 problems per minute.
   Relocatability: Not relocatable.

e. Remarks: This program was written for bridges using AASHTO loading and specifications. It is recommended that this program be used in conjunction with the Washington State Highways Department’s “Moment Distribution and Influence Line Calculation” program. (IBM 650 Program Library File Number 9.2.05).

f. IBM 650 System: One 533 required.

Special Devices: None.

### IBM 650 Library Program Abstracts

#### GEOIDMETER COMPUTATIONS

P. E. Mishler
California Division of Highways
Sacramento, California

a. Purpose: This program takes the readings from the Model 3 Geodimeter and a vertical angle from a theodolite, computes a slope distance and reduces this distance to horizontal and vertical components.

b. Range: Not given.
   Accuracy: Computes to nearest 0.01 ft.
   Floating/Fixed: Fixed decimal arithmetic.

c. Mathematical Method: The mathematics used follows closely the hand-calculated procedure making numerous decisions following standard rules of the problem.

d. Storage Required: 415 drum storage locations exclusive of the read and punch locations.
   Speed: The program will compute approximately 29 problems per minute.
   Relocatability: Not given.

e. Remarks: The program utilizes the IBM 650 Program Library SIN routine.

f. IBM 650 System: One 533 required.

Special Devices: Alphabetic device was used, but is not necessary.

### IBM 650 Library Program Abstracts

#### CONTINUOUS BRIDGE ANALYSIS

T. L. Yates
Oregon State Highway Department
Salem, Oregon

a. Purpose: This program encompasses three independent routines used in the analysis and design of continuous beam type structures. The three routines are: (1) Analysis of Continuous Beams and Frames, (2) Live Load and Total Moments Due to H-8 Loading, and (3) Deflections.

b. Range: Two to five span structures are accommodated.
   Accuracy: In calculating dead load moments, an error of approximately 1/5% exists.
   Floating/Fixed: Not given.


b. Storage Required: All but six storage locations are used in the routine Live Load and Total Moments Due to H-8 Loading.
   Speed: A complete frame analysis, including total moments and deflections, requires approximately 15 minutes per span.

(Continued on next column)
OVERHAUL PROGRAM

Kathy Brown
Charlene Travis
Dept. of Highways
Olympia, Washington

a. Purpose: To compute overhaul quantities.
b. Range: 123 even stations for each haul area.
c. Mathematical Method: Does not apply.
d. Storage Required: 1230 storage locations were used.

IBM 650 System: One 333 required.

IBM 650 Library Program Abstracts

STAGE CONSTRUCTION PROGRAM

G. J. Kellichiens
Washington State Highway Dept.
Olympia, Washington

a. Purpose: Given the cross-section template and catch points, this program will calculate a new cross-section card giving the cross-section readings outside the catch points, the catch points and template readings in elevations.
b. Range: Will handle 100 cross-section readings, 100 template readings and give 50 stations on new cross-sections.
d. Storage Required: This program uses 1010 drum storage locations.

IBM 650 System: One 533 required.

IBM 650 Library Program Abstracts

STAGE CONSTRUCTION PROGRAM

T. L. Yates
State Highway Department of Oregon
Salem, Oregon

a. Purpose: This program summarizes truck weight violation data from the W-6 table for issuance with Bureau of Public Roads requirements.
b. Range: The program as written, will handle a maximum of 300 vehicles; it can be readily expanded, however.
c. Mathematical Method: Does not apply.
d. Storage Required: 500 storage locations.

IBM 650 System: One 533 required.

IBM 650 Library Program Abstracts

DATA TABLE SUMMARY

(Continued on next column)
IBM 650 Library Program Abstracts

PROFILE COMPARISON AND STATISTICAL ANALYSIS PROGRAM DA-1
C. L. Miller - Project Director
R. A. Laffine - Programming Supervisor
D. F. Hsberg - Programmer
Photogrammetry Laboratory
Department of Civil and Sanitary Engineering
Massachusetts Institute of Technology
Cambridge, Mass.

a. Purpose: Compare elevations obtained from contour maps to field data on the same profile. Four point polynomial interpolation is used to obtain the map elevation at the same point as the field data. Differences between the two elevations and a statistical analysis of the differences are computed for each profile individually and for all profiles collectively.

b. Range: (i) A map data profile cannot exceed 600 points.
(ii) The field data profile will be computed for only those points which are beyond the first two and before the last two map data points.

c. Accuracy: (i) Differences have as many significant digits as the input data.
(ii) Statistics are rounded to two decimal places.

Floating/ Fixed: Fixed.

d. Mathematical Method: Althoen's method of iteration is used to compute the polynomials.

e. Storage Required: 600 locations are reserved for the map profile and the program occupies the remaining 1449 locations.

f. Speed: Differences are computed in 2 seconds, therefore 30 points per minute are compared and punched. Profile or map statistics require 25 seconds, independent of the number of points in the profiles.

Relocatability: Not relocatable.

e. Remarks: Input uses eight digit words, however, the output requires special control panel wiring. Output is designed for setting on a 650, with all 6's and 0's.

f. IBM 650 System: Minimum 650.

Special Devices: Alphabetic Device.

IBM 650 Library Program Abstracts

COMPUTATION OF BRIDGE SCREEN ELEVATIONS
Z. L. Moh
C. E. Cooper
Bridge Bureau
State Highway Department of Indiana
Indianapolis 4, Indiana

a. Purpose: This program computes the elevations for setting anchors for concrete slabs on continuous steel beam or steel girder bridges.

b. Range: Elevations are given at ten foot intervals along four screen lines. Successive spans are considered one at a time with no limitation on the number of spans.

(Continued on next column)
The program is intended to be used with special shift. A basic unit of time is one minute. The program is written in SOAP 1 and may be used.

f. Remarks: The purpose of this program is to compute time and distance on a freeway system. Speed is approximately 55 mph on input cards per hour. The program is written in SOAP 1 and may be used.

e. Method: Fixed point arithmetic is used.

b. Restrictions, Range: Fixed point arithmetic is used.

a. Purpose: Freeway Assignment. The purpose of this program is to compute time and distance on a freeway system and then compare it to a basic system to determine if the proposed system would be adequate.

IBM 650 Library Program Abstracts

Felix D. Geseler
Pennsylvania Department of Highways
North Office Building
Harrisburg, Pennsylvania

TRACING A MINIMUM PATH BETWEEN ZONE CENTERS OVER A ROAD NETWORK

Marvin Brodsky
California Division of Highways
120 N Street
Sacramento, California

a. Purpose: The purpose of the program is to obtain mechanical routing as input to a freeway assignment program in place of the present manual methods. Also to obtain the time or distance between zone centroids for use in forecasting trips between zones.

b. Restrictions, Range: The program uses fixed point arithmetic. Accuracy is not a problem.

c. Method: The mathematical method is a minimum path algorithm which beings all possible routes between a pair of zones for a road network and selects the minimum path between zones using time, distance or some other value for each segment of the road network.

d. Storage Requirements: All locations of a 1600-word drum are used except T. The program is in SOAP format and completely relocatable. The speed depends upon the number of zones in the road network. It is maximum 60 words per hour. If the proposed system would be adequate, a big increase in speed can be obtained by reducing the number of locations reserved in the tables and relocating in the size of the problem.

f. Remarks: The table of reference allows for a backlog of zones which seems to be sufficient, but will cause a machine stop if inadequate. Two zones must be identified. Notes must be numbered on the map so that going from a link described by consecutive node numbers to a link described by consecutive node numbers and vice versa out of order for which a penalty will be assessed in determining the route. This is to avoid unnecessary figuring in the selection of the path. Through a grid type network it is inevitable that some penalties will be assessed where they should not be and vice versa. All link values must be greater than zero and less than 100. Links with larger values must be split.

b. Range: The program is in SOAP 1 and may be used.

d. Storage Requirements: The table of reference allows for a backlog of zones which seems to be sufficient, but will cause a machine stop if inadequate. Two zones must be identified. Notes must be numbered on the map so that going from a link described by consecutive node numbers to a link described by consecutive node numbers and vice versa out of order for which a penalty will be assessed in determining the route. This is to avoid unnecessary figuring in the selection of the path. Through a grid type network it is inevitable that some penalties will be assessed where they should not be and vice versa. All link values must be greater than zero and less than 100. Links with larger values must be split.

f. Remarks: The table of reference allows for a backlog of zones which seems to be sufficient, but will cause a machine stop if inadequate. Two zones must be identified. Notes must be numbered on the map so that going from a link described by consecutive node numbers to a link described by consecutive node numbers and vice versa out of order for which a penalty will be assessed in determining the route. This is to avoid unnecessary figuring in the selection of the path. Through a grid type network it is inevitable that some penalties will be assessed where they should not be and vice versa. All link values must be greater than zero and less than 100. Links with larger values must be split.

b. Range: The program is in SOAP 1 and may be used.

d. Storage Requirements: The table of reference allows for a backlog of zones which seems to be sufficient, but will cause a machine stop if inadequate. Two zones must be identified. Notes must be numbered on the map so that going from a link described by consecutive node numbers to a link described by consecutive node numbers and vice versa out of order for which a penalty will be assessed in determining the route. This is to avoid unnecessary figuring in the selection of the path. Through a grid type network it is inevitable that some penalties will be assessed where they should not be and vice versa. All link values must be greater than zero and less than 100. Links with larger values must be split.

f. Remarks: The table of reference allows for a backlog of zones which seems to be sufficient, but will cause a machine stop if inadequate. Two zones must be identified. Notes must be numbered on the map so that going from a link described by consecutive node numbers to a link described by consecutive node numbers and vice versa out of order for which a penalty will be assessed in determining the route. This is to avoid unnecessary figuring in the selection of the path. Through a grid type network it is inevitable that some penalties will be assessed where they should not be and vice versa. All link values must be greater than zero and less than 100. Links with larger values must be split.
a. Purpose: This routine adjusts traverses by the compass or the transit rule, as required by the engineer. Input is in the form of one course per card and output is also in the form of one course per card. Areas for closed traverses may be obtained.

b. Restrictions, Range: Each traverse may have a maximum of 90 regular courses. All linear measurements are given to thousands of feet and bearings are computed to seconds. All trigonometric functions are computed to nine decimal places.

c. Method: The trigonometric functions are calculated from Technical Newsletter 650. Area is calculated using the cross–cross method.

d. Storage Requirements: One hundred locations each are required for storage of unadjusted latitude deviation and distance. Three hundred locations are required for storage of the description. Program and temporary storage requirements are approximately 610 650 more locations. Speed is approximately 2500 courses per hour. The program is written for SOAP I form.

e. Remarks: No provision has been made for computing areas of stray segments because no provision has been made to keep certain course constant.

f. IBM 650 System: A 650 with half–time emitter and alphabetic device is used.

IBM 650 Library Program Abstracts

REVIEWED TRAVERSE AND HORIZONTAL ALIGNMENT

S. F. Pergolin
J. Zien
California Division of Highways
1120 N. Street
Sacramento, California

a. Purpose: The routine will calculate traverses with two unknowns or with one unknown in each traverse, input is in the form of one course per card. Hints are punched one course per card and show identification, distance, latitude, latitude deviation, and coordinates for regular courses and also course error. Area for closed figures and segments areas are computed. Although two solutions are calculated, results are only tabulated. When both solutions are obtained, the solution with the smallest difference of bearings is used as base and the other solution is used as check. All angles are calculated using the cross–cross method.

b. Range: Each traverse may have a maximum of 20 regular courses.

c. Accuracy: Distances are given to thousandths of feet and bearings to seconds. Functions are computed to nine decimal places. Area is calculated to square feet and thousandths of acres.

d. Floating Point: Does not apply.

e. Mathematical Method: Library subroutines are from Technical Newsletter 650 letter A14 for sine, cosine, and tangents. Area is calculated using the cross–cross method.

f. Storage Requirement: One hundred ninety–nine storage locations are required for regular data storage. Eighty locations are required for interdependency table storage. Other program and temporary storage requirements are the same as two thousand–five locations.

g. Speed: Speed is approximately two thousand courses per hour. The program is considered optimum and should not be reckoned although the program in SOAP I terminology.

h. Remarks: The program has several routines which test for invalid data.

IBM 650 Library Program Abstracts

DATA PROCESSING CENTER

a. Purpose: This routine calculates traverses by the compass or the transit rule, as required by the engineer. Input is in the form of one course per card and output is also in the form of one course per card. Areas for closed traverses may be obtained.

b. Restrictions, Range: Each traverse may have a maximum of 90 regular courses. All linear measurements are given to thousands of feet and bearings are computed to seconds. All trigonometric functions are computed to nine decimal places.

c. Method: The trigonometric functions are calculated from Technical Newsletter 650. Area is calculated using the cross–cross method.

d. Storage Requirements: One hundred locations each are required for storage of unadjusted latitude deviation and distance. Three hundred locations are required for storage of the description. Program and temporary storage requirements are approximately 610 650 more locations. Speed is approximately 2500 courses per hour. The program is written for SOAP I form.

e. Remarks: No provision has been made for computing areas of stray segments because no provision has been made to keep certain course constant.

f. IBM 650 System: A 650 with half–time emitter and alphabetic device is used.

IBM 650 Library Program Abstracts

MODEL 2 GEODIMETER

Virginia T. Greenfield
Division of Highways
Planning Survey Department
Division of Highways
Sacramento, California

a. Purpose: To take readings from the Model 2 Geodimeter and compute the slope distance between two points. Using the vertical angle measured (Continued on next column)
IBM 650 Library Program Abstracts  

**THERMODYNAMIC PROPERTIES AND PHASE BEHAVIOR OF LIGHT HYDROCARBON MIXTURES**

W. Edwards  
E. I. Organic  
L. Larrey  
Computing Center  
University of Houston  
Houston, Texas  

- **Purpose:** Computes density, compressibility factor, enthalpy, entropy, and equilibrium ratios of single and two phase systems.  
- **Range:** Handles mixtures with up to nine components.  
- **Accuracy:** Not given.  
- **Floating/Fixed:** Single precision floating point with input and output data supplied in fixed point (Humble floating point interpretive routines).  
- **Mathematical Method:** Rigorous thermodynamic solution based on:  
  2. Zero pressure thermal properties of pure components.  
- **Storage Required:** Approximately 100 unused drum locations.  
- **Speed:** Speed depends upon number of phases, number of components, and on option to compute enthalpy and entropy.  
- **Relocatability:** Program is non-relocatable.  
- **Remarks:** None.  
- **IBM System:** One 533 required.  
- **Special Devices:** None.  

**CALCULATION OF THE LEAST-SQUARES BEST HALF-WAVE POTENTIAL AND SLOPE OF A POLAROGRAPHIC WAVE**

D. L. McMasters  
W. B. Schaap  
Indiana University  
Bloomington, Indiana  

- **Purpose:** This program calculates the half-wave potential and slope of a polarographic wave,  
  \[ E = E_{1/2} + \frac{0.0625 \log (i_{lim})}{n} \]  
  by the method of least squares using current-voltage data taken from a polarogram.  
- **Range:** This program is set up to analyze only polarographic reduction waves.  
- **Accuracy:** Not given.  
- **Floating/Fixed:** Floating decimal arithmetic is used in the Bell Labs System.  
- **Mathematical Method:** See a. above.  
- **Storage Required:** Most of the locations from 0100 through 0400 are used by the entire program.  
- **Speed:** The entire routine requires just 15 seconds for each complete calculation.  
- **Relocatability:** The program would be difficult to relocate.  
- **Remarks:** This program, written in the Bell Labs Interpretive System (see TNL No. 11), was designed for polarograms recorded by the Sargent Model XXI Visible Recording Polarograph, however, with only a few obvious and minor changes in the recording of the data (and not in the program), this program can be adapted to other manually and electronically recorded polarograms.  
- **IBM System:** One 533 required.  

**PLATE-TO-PLATE CALCULATIONS** (Continued on next column)
b. Restrictions. Range: The program calculates the above properties of any polyatomic non-linear molecular system in the ideal gaseous state for the rigid rotor - simple harmonic model. The contributions for hindered internal rotation cannot be gained by this program. The mathematical notation is C. 0.0015 unit.

c. Method: The calculations of the exponential and the logarithmic functions are made by the use of the sub-routine.

d. Storage Requirements: The number of storages used for the whole computation is 550. When the number of the translational frequencies is nine, the time required for the computation for an assigned temperature is 1.5 sec.

e. Remarks: Either the vibrational contribution or the sum of the translational and rotational contributions may be calculated separately.

f. IBM 650 System: Minimum, IBM 650.

---

**IBM 650 Library Program Abstracts**

**FIFTY BUS LOAD FLOW PROGRAM**

B. J. Brown
W. F. Tinney
Bonneville Power Administration
Portland, Oregon

a. Purpose: This program is designed to solve electric utility power network flow problems for systems of no more than 50 buses and seven lines per bus.

b. Range: The scaling was determined experimentally to accommodate the range of data in problems solved at Bonneville. This scaling may not be satisfactory for all other systems. A power base of 1 pu = 100 MVA is used.

Accuracy: Not given.

Floating/Fixed: Arithmetic is in fixed point.

c. Mathematical Method: The Gauss-Seidel method is used.

d. Storage Required: The program uses almost all drum locations.

Speed: Approximately one hour is required for an average system.

Relocatability: Program is not relocatable.

e. Remarks: Considerable study is necessary for effective operation of the system.

f. IBM 650 System: One 533 required.

Special Devices: None.

---

**IMPROVED DOUBLE SHORT CIRCUIT SOLUTION OF POWER SYSTEM NETWORKS**

M. J. Lasto
Bonneville Power Administration
Portland, Oregon

a. Purpose: Precalculates short circuit currents at various possible locations in the system.

b. Range: Solves a 20 x 20 matrix which is equivalent to a network having 45 impedance elements.

Accuracy: Not given.

Floating/Fixed: Floating point.

c. Mathematical Method: Loop equations are used to reduce matrix size.

d. Storage Required: Not given.

Speed: Solution time per fault is approximately .0025 min. where 

N is the matrix size.

Relocatability: Not given.

e. Remarks: None.

f. IBM 650 System: One 533 required.

Special Devices: None.

---

**99-8455 LOAD FLOW PROGRAM**

W. F. Tinney
Bonneville Power Administration
Portland, Oregon

a. Purpose: Solves AC load flow problems for power systems with up to 99 buses and 199 branches.

b. Range: As above.

Accuracy: Any degree of precision desired.

(Continued on next page)
Floating/Fixed: Fixed point arithmetic is used.

c. Mathematical Method: The nodal iterative method of solution is used.

d. Storage Required: Almost entire drum.

Speed: A function of desired precision. Approximately 0.5 seconds per

bus per iteration, exclusive of input and output time. One-half to one

and one-half hours over-all computing time for full capacity problem.

Relocatability: Non-relocatable.

e. Remarks: Input data are prepared and punched from convenient standard

forms. Output consists of complete load flow information including bus

voltage and angles, real and reactive flow into and out of each branch,

losses in each branch, and total system losses.

f. IBM 650 System: One 533 required.

Special Devices: Alphabetic device.

IBM 650 Library Program Abstracts

PROBABILITY OF LOSS OF LOAD

H. D. Limmer
Public Service Electric & Gas Co.
Newark, New Jersey

Purpose: Calculates the probability of loss of load (due to lack of

sufficient generation or interconnections) of a power system.

b. Range: Will handle at least 50 machines.

Accuracy: Not given.

Floating/Fixed: Not given.

c. Mathematical Method: Based on method outlined in AIEE paper 58-139,

d. Storage Required: Not given.

Speed: Running time varies with size of system. A 35-machine system

takes about 4 hours. Program can be re-run in 4 minutes if only the

characteristics of the load or firm interconnection capacity are changed.

Relocatability: Not in relocatable form.

e. Remarks: None.

f. IBM 650 System: One 533 required.

Special Devices: None.

IBM 650 Library Program Abstracts

CALCULATION OF ELECTRIC POWER SYSTEM SHORT-CIRCUIT CURRENTS

L. W. Coombe
The Detroit Edison Company
Detroit, Michigan

Purpose: This program computes the total fault current and the currents in

the line connected to the faulted bus. The real and imaginary components

and the magnitude of the currents are punched out together with the X/R

ratio. The input data can be arranged so that the location of the fault can

be changed automatically.

b. Range: The program will accommodate networks of up to 96 buses and/or

100 lines.

Accuracy: Depends on the convergence tolerance specified.

Floating/Fixed: Fixed point arithmetic is used.

c. Mathematical Method: A nodal analysis is used to form a set of simultaneous

equations with complex coefficients. These equations are formed by the

program and solved by the Gauss-Jordan iteration method with relaxation.

d. Storage Required: Not given.

Speed: Requires approximately 0.858 seconds per iteration, where B is

the number of buses. The number of iterations required depends on the

system and accuracy desired, usually ranging between 6 and 60 iterations

per fault.

Relocatability: Not given.

e. Remarks: A routine is included to convert the form of the input from

impedances to admittances. The program may also be used to determine

system driving-point and transfer admittances (equivalent circuits). It

does not handle mutual impedances.

f. IBM 650 System: One 533 required.

IBM 650 Library Program Abstracts

OVERHEAD ELECTRICAL DISTRIBUTION SYSTEMS ANALYSIS

J. B. Jones
E. J. Farrow
IBM, Houston, Texas

G. W. Ogroa
Houston Lighting and Power Company
Houston, Texas

a. Purpose: This program calculates voltage drops at various load points

along a given circuit, based on total loading of circuits, physical and
electrical design, and customer demand at designated load points.

b. Range: Maximum of 40 load points per circuit.

Accuracy: Not given.

Floating/Fixed: Fixed point arithmetic is used.

c. Mathematical Method: Does not apply.

d. Storage Required: The entire drum is required for instructions and data.

Speed: About 3 seconds per point.

Relocatability: Not relocatable.

e. Remarks: Both absolute and SOAP listings are included.

f. IBM 650 System: One 533 required.

Special Devices: Alphabetic device required.

IBM 650 Library Program Abstracts

ECONOMIC CONDUCTOR STUDY

N. F. Thomas
Consumers Power Company
Jackson, Michigan

a. Purpose: This program is designed to determine the economic conductor

size for a proposed electrical transmission line.

b. Range: \[ 2 \Delta i \times 10^3 \cdot \text{amps} \], where \[ 15 \Delta i < 10 \text{ and } -50 \Delta i \leq 49 \text{,} \]

Accuracy: Eight decimal digits.

Floating/Fixed: Bell Labs Floating Decimal Interpretive System (TNL # 11)
is used.

c. Mathematical Method: The equations used in calculating the electrical

characteristics of transmission lines are those equations commonly used
to calculate impedances, sending-end and receiving-end power, etc., based
upon a symmetrical pi equivalent circuit.

d. Storage Required: This program uses 1253 storage locations.

Speed: The running time for one conductor size is approximately 100

seconds.

Relocatability: Not given.

a. Remarks: Conduit format, control panel and operating instructions are as

prescribed by the interpretive system used (see par. b. above). An
exception is that the Programmed switch is set to the "Run" position.

f. IBM 650 System: One 533 required.

IBM 650 Library Program Abstracts

CORRECTION OF COAL MOISTURE MEASUREMENTS

N. Savage
The Detroit Edison Company
Detroit, Michigan

a. Purpose: This program calculates the constants of a linear equation which

relates percentage moisture in coal at two different locations in a power

plant. Then, for 120 equal increments of percentage moisture at one

(continued on next page)
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 2560 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 IBM 650 Program Library.

f) Minimum 650.

PIPE STRESS ANALYSIS

W. S. Pickrell
J. H. Rogers
L. S. Woon
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 its 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 Kellog Method is used for the calculations, while the stresses and the anchor reactions are computed according to the ASME 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.

ENGINEERING APPLICATIONS

KINEMATIC SYNTHESIS OF PATH GENERATING MECHANISMS

G. H. Sander
TIME, Inc.
Springfield, Connecticut

(Continued on next page)
IBM 650 Library Program Abstracts

**EVALUATING COMPRESSOR PERFORMANCE**

J. W. Evans
L. L. Smith
Sinclair Oil and Gas Company
Tulsa, Oklahoma

- **Purpose:** Sinclair’s purpose is writing a compressor program is to enable engineers to design for maximum efficiency of compressor application with a minimum of engineering time in each new compressor application. A method of computing data for horsepower and capacity curves has been developed which presents a wide range of operating characteristics of the compressor in question for engineering analysis.

---

**CO-ORDINATE ROUTINE**

R. A. Semrad

- **Range:** Values of \( d \), the nearest hundredth of a degree. Accuracy: Better than \( 10^{-3} \) to the five specified places.

---

**CAM LEADER CO-ORDINATE ROUTINE**

M. T. Connolly
Henry M. Scalatt
United Shoe Machinery Corporation
Research Division
Beverly, Massachusetts

- **Purpose:** Calculates the cam leader follower roll center \( x \) and \( y \) co-ordinates for any angular position of the cam from the outer most position of the roll. This subroutine is designed for use with the interpretive system developed at Bell Telephone Laboratories and described in IBM Technical Newsletter 42.

---

**PERFORMANCE**

J. T. Ahlin and G. E. Mitchell

- **Storage Required:** Including the interpretive system, the program occupies locations 0000-0199. However, the program is so constructed that only those decks which are pertinent to the individual problem need be used. The interpretive system occupies locations 1000-1999. It takes approximately 2 to 4 seconds to calculate the co-ordinates for each degree of cam rotation.

---

**REDUCTION**

- **Mathematical Method:** A seven-term approximation is used for the arc tangent. The first value of the iteration is obtained from a table included in the program.

---

**WEIL BORE DEVIATION RECORD**

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

- **Storage Required:** In about 600 locations between 0000 and 0999. Speed is about 60 stations per minute.

---

**B. Range:** Not given.

---

**Accuracy:** Not given.

(Continued on next column)
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 metric units. Accuracy depends on quantity considered.
b) Fixed point arithmetic with different scaling for English and metric units.
c) Uses Benedict equation. Exponential and logarithmic routines are employed.
d) Program scattered optimum over the whole drum. A temperature-pressure combination takes 3-4 seconds, depending on number of iterations required.
e) None.
f) Minimum 530.

EQUILIBRIUM FLASH CALCULATION

M. E. Klcoka
R. Y. Souder
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 ± 0.001 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 5-6 minutes per complete calculation.
e) These 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) 530 equipped with alphabetic device.

OPTIMUM SEPARATOR PRESSURE

John M. Tyler
Citizens Service Research & Development Co.
Tulsa, Oklahoma

a. Purpose: To determine optimum separator pressure for a series separation consisting of two separators and one stock tank.
b. Range: Not given.
Accuracy: Optimum pressure is determined with a precision of one psi. Actual accuracy depends on the accuracy of the K values.
Floating/Fixed: Floating point arithmetic is used.
d. Storage Required: All storage locations other than 1400-1499 are utilized.
Speed: 3 minutes to 1 hour depending on accuracy of first estimates.
Relocatability: Not given.
e. Remarks: The operating pressure is determined by the first estimate for the separator pressure. As the user acquires familiarity with the program his estimate will become better, thereby reducing computer time. Output may be modified so that a special character device is not necessary.
f. IBM 650 System: One 530, automatic floating decimal arithmetic feature, indexing registers.
Special Devices: Special character device required unless output is modified to 533 terminals.

POROSITY CALCULATION FROM RADIOACTIVITY LOG INTERPRETATION

Charles D. Woodward
Sunray Mid-Continent Oil Company
Tulsa, Oklahoma

a. Purpose: This program calculates the following from the neutron curve of the radioactivity log and the water saturation curve (water saturation vs. subsea depth) and interval feet, porosity, porosity feet, (L-CW) determined from the water saturation vs. subsea depth curve, hydrocarbon porosity feet, and average hydrocarbon porosity.
b. Range: The total interval being evaluated must be less than 10,000 feet. A maximum of fifty points may be used to define the water saturation vs. subsea depth curve.
Accuracy: Not given.
Floating/Fixed: Fixed decimal arithmetic is used.
c. Mathematical Method: The evaluation of the water saturation curve is determined by a linear interpolation of the curve points.
d. Storage Required: This program requires 165 drum storage locations.
Speed: Not given.
Relocatability: Not relocatable.

(Continued on next page)
BUCKEY ROD PUMP DESIGN

H. E. Osborne & C. E. Thomas
Core Laboratories, Inc.
Dallas, Texas

a. Purpose: The program calculates the design features of a sucker rod pump to fit a set of conditions by investigating the effect of changing each of the variables throughout its full range of possibilities.

b. Range: Not given.
Accuracy: Not given.
Floating/Fixed: Not given.

Mathematical Method: Not given.

Storage Required: Not given.

Remarks: Theoretical producing rate, actual plunger strokes, load stress, peak stress, percent of each rod size to allow equal stress at the top of each section of the rod string are used.

Speed: Up to 300 cases may be computed in an hour.
Relocatability: Not given.

IBM 650 System: One 533 required.

RESIDUALS AND DERIVATIVES OF GRAVITY

J. E. Ward
Atlantic Refining Co.
Dallas, Texas

a. Purpose: This program computes several residuals and second derivatives of gravity at each regularly spaced grid intersection where sufficient data exist.

b. Range: Maximum size of each map is limited to 100 rows by 800 columns.
Accuracy: Not given.
Floating/Fixed: Not given.

Mathematical Method: Not given.

Storage Required: The program requires 1472 storage locations, of which 792 are for data storage, 288 for program instructions, 100 for temporary storage, and the remaining 107 are for constants, corrections, read and punch, etc.
Speed: Average running time for each datum point is .014 minutes. A map of 70 rows by 70 columns should run in about 11 hours.
Relocatability: Not given.

Remarks: Input data is punched into cards as four-digit positive values at each intersection, up to 10 per card. Output results are punched one card per grid intersection with six residuals and four derivatives at this point if all necessary data exist.

f. IBM 650 System: One 533 required.

IBM 650 Library Program Abstracts
File no. 2.6.007
Engineering Applications
IBM 650 Library Program Abstracts

FIVE LAND SURVEYING PROGRAMS

Shell Oil Company
Houston, Texas
a. Purpose: To convert hand calculations on land surveying problems for use with the IBM 650.
b. Range Accuracy: Self checks are built into the programs.
c. Mathematical Method: Given in write-up.
d. Storage Requirements, Speed, Relocatability: N/A
e. Remarks: None
f. IBM 650 System: 650 with alphanumeric device and a 533.

IBM 650 Library Program Abstracts

A PROGRAM FOR PARTITIONING OF ARBITRARILY SHAPED AREA

D. C. Schillers
Shell Oil Company
Houston, Texas
a. Purpose: Given an area bounded by straight lines with known intersections, the program will partition it with a horizontal line parallel to the X-axis into any desired ratio.
b. Range Accuracy: N/A
c. Mathematical Method: Given in write-up.
d. Storage Requirements, speed: Not given
e. Remarks: Two limitations exist. First, no more than 99 intersections can be counted around any area. Second, the area in square varas and the distances in varas may not exceed 99,999,999,99.
f. IBM 650 System: N/A

IBM 650 Library Program Abstracts

A PROGRAM FOR THE GAUSS-SOUTHWELL RELAXATION METHOD

G. C. Carney
D. C. Schillers
Shell Oil Company
Houston, Texas
a. Purpose: To illustrate a method used to solve the systems of simultaneous equations derived in the adjustment of survey nets such as found in land and geophysical surveys.
b. Range: The method will be applicable to other systems if the conditions of sparseness and convergence are met.
c. Mathematical Method: N/A
d. Storage Requirements: The complete system and needed control words use approximately 9,600 storage words where 1 is the number of off diagonal elements.
e. Remarks: The program is divided into three parts.
SEISMOGRAM SYNTHESIS FROM CONTINUOUS INTERVAL VELOCITY (CVL)

E. J. Assiter
D. H. Eckhardt
W. Williams
Mobil De Venezuela
Caracas, Venezuela

a. Purpose: This program is designed to perform the convolution of the three major components of a seismogram: (1) Seismic pulse, (2) Instrument Impulse Response, (3) Interval velocity function (CVL). In addition, a six trace seismogram is plotted, on line, by the IBM 407.
b. Range and Accuracy: Not given.
d. Storage: Not given.
e. Remarks: None.
f. 650 System: 533, 653 (core and indexing), on line 407.

NORMAL MOVEMENT COMPUTATIONS FOR LINEAR INCREASE OF VELOCITY WITH DEPTH

E. J. Assiter
D. H. Eckhardt
W. Williams
Mobil De Venezuela
Caracas, Venezuela

a. Purpose: This program solves the "Slovenian Equation" for the case of circular ray paths.
b. Range: Not given.
d. Storage: Not given.
e. Remarks: None.
f. 650 System: 533, 653 (Index Registers).

IBM 650 Library Program Abstracts

UNIT OPERATIONS SIMULATOR

(Continued on next page)
GAS NETWORK ANALYSIS PROGRAM
P. L. Duffy
The Cincinnati Gas & Electric Co.
Cincinnati, Ohio

a. Purpose: This program provides a very flexible method for computing the million of low, intermediate or high pressure gas networks. Variations in network conditions to arrive at the optimum system development may be handled with a minimum of effort.

b. Range: Networks with 100 main sections may be analyzed and any flow formula which can be reduced to the form
   \[ h = f(P_1, P_2, \ldots, P_n, Q) = a \]
   can be used. The main length and flow may be in any unit whatsoever.

Accuracy: The network may be balanced to a predetermined limit of accuracy.

Floating/Fixed: Computations are in a fixed point.

Mathematical Method: Iterative procedure based on a modified Hardy-Cross Method is used.

Storage Required: Storage varies for the separate sections of the program. Maximum storage requirement is 125 locations.

Speed: Speed is dependent on accuracy desired.

Relocatability: Not given.

e. Remarks: There are some limitations on size and length; see program write-up.

f. IBM 650 System: One 533 required.

Special Devices: None.

IBM 650 Library Program Abstracts
Fileno. 9.7.001
Engineering Applications

HYDRAULIC NETWORK ANALYSIS

Continued on next column
IBM 650 Library Program Abstracts

LIQUID VOLUMES IN FLAT END HORIZONTAL CYLINDRICAL TANKS

A. J. Ballier
Ventil, Incorporated
R. Louis, Missouri

a. Purpose: This program calculates the volume of liquid, at height of liquid h, contained in a flat-end horizontal cylindrical tank.

b. Range: Depends on system of units selected to measure dimensions of tank.

c. Accuracy: Greatest possible error = \( \pm 0.25\% \).

Floating/Fixed: Fixed decimal arithmetic is used.

d. Mathematical Method: Does not apply.

e. Storage Required: 110-120 storage locations.

Speed: About 5 minutes for a tank 10' in diameter and 170' in length.

Relocatability: Not given.

e. Remarks: None.

f. IBM 650 System: One 333 required.

IBM 650 Library Program Abstracts

GAS FLOW ANALYSIS

(Continued on next column)
b. A comparison of the largest correction ( Q) with the desired limit of accuracy, causes the program to perform additional iterations or print results. This feature permits the problem to be solved during other than prime machine times. A patch of the largest ( Q) at the end of each iteration provides a check on convergence. When the desired accuracy is attained, flows and pressures drop are punched for all pipes in the network, including dead-end pipes.

c. The modified Hardy Cross Method is used in the program. This technique is used throughout the industry. The Sprott's coefficients, which are supplied with the program deck, may be changed easily.

d. The program was arbitrarily limited to 400 drum locations, providing 1600 locations for data storage. These locations are normally reserved for 700 pipe sections and 900 items of loop data. Division of the 1600 locations may be altered to specific problem requirements.

The program was written in machine language and may not be relocated. Optimum locations were initially assigned.

e. Remarks: None.

f. The program was written for the basic 650. Wring it for the 533.

---

**IBM 650 Library Program Abstracts**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>B. D. Blosser</td>
<td>Firestone Tire &amp; Rubber Co. Los Angeles, California</td>
</tr>
<tr>
<td>F. A. Vandenberg</td>
<td>Portland, Oregon</td>
</tr>
</tbody>
</table>

**PROGRAM: PARALLAX REDUCTION**

- **Purpose**: This program determines the transient behavior of a control system as a result of changes in loop gain, component time constants, and stabilizing network configurations.
- **Range**: Degree of forward and feedback loop must be less than 14.
- **Accuracy**: Seven significant figures.
- **Method**: Mathematical Method: Root Loci, Cross, Gain. System: Operates at approximately 9/10 full read speed depending on the number of points in the section.

**IBM 650 Library Program Abstracts**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>H. F. Smith</td>
<td>IBM, Chicago</td>
</tr>
</tbody>
</table>

**LINEAR PROGRAMMING**

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 ≥ 20, N ≥ 50 and (M+N+1) ≤ 1400 (these values pertain to the system after the slack vectors and artificial vectors have been adjacently.)

c. Method not given.

d. The entire drum is used. Time required is approximately 0.89 MN seconds for one iteration.

e. Input consists of matrix elements, cost coefficients, indices of basis, and constraints. 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.
By changing one instruction it is possible to reduce this cumulative rounding error below its present level.

The instruction in location 8068 new reads: 30 9909 8297. It should be changed to read: 30 9909 8727.

This change may be made in the following manner:

1. Place a correction card just before the last card of part 5 of the program deck. Part 5 consists of those cards in the program deck which follow the matrix elements and which precede the constants.

2. The correction card contains:

<table>
<thead>
<tr>
<th>Column</th>
<th>1-10</th>
<th>11-20</th>
<th>21-30</th>
<th>31-40</th>
<th>41-50</th>
<th>51-60</th>
</tr>
</thead>
<tbody>
<tr>
<td>Content</td>
<td>0000 0000</td>
<td>20 0020</td>
<td>0000</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(End of correction).

3. Naturally this change is only of consequence when the right-hand positions of the data fields contain significant digits.

IBM 650 Library Program Abstracts

LINEAR PROGRAMMING

By H. F. SMITH

On Page 2, Section B, Scrolling, the third sentence now reads:

"The cost coefficients must be scaled so they are all less than 1."

This sentence should be changed to read:

All cost coefficients except the artificial cost coefficients must be scaled so they are all less than 1.

650 LIBRARY PROGRAM ABSTRACT

FILE NUMBER 10.1.002

EReTATA

650 Library Program - File No. 10.1.003

"Transportation Problem," by S. Poley

It has been discovered that the copies of the program deck for Program III (Alternate Optimal) 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.

650 LIBRARY PROGRAM ABSTRACT

FILE NUMBER 10.1.004

TRANSPORTATION PROBLEM

By S. Poley

IBM, New York

March 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 10^m + 3000 with m < 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.

650 LIBRARY PROGRAM ABSTRACT

FILE NUMBER 10.1.003

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 xxxx.xxxx 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, 27, 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.

650 LIBRARY PROGRAM ABSTRACT

FILE NUMBER 10.1.005

LINEAR PROGRAMMING

R. L. Graves

Standard Oil, Indiana

March 29, 1956

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 Trumble-Kubic interpretive system is used for the floating-point arithmetic, see Technical Newsletter No. 8.

f) IBM 650 System: One 533 required.

IBM 650 Library Program Abstracts

LINEAR PROGRAMMING CODE FOR THE AUGMENTED IBM 650

O. B. Perry
IBM, Los Angeles 5, California

Purposes: This routine provides a method to find optimal solutions for relatively large linear programming problems with flexibility of input and output, while maintaining simplicity and speed in operation.

b) Range: Not given.

c) Mathematical Method: Composite Algorithm; reduces to Simplex Algorithm when feasibility has been achieved.

d) Storage Required: This routine uses the entire drum; however, if the problem is less than maximum size a large portion of the drum will be available for other use.

e) Speed: Computing speed depends on several factors; as an example, in a problem where \( M = 17 \) and \( N = 57 \), the speed is approximately 20 seconds per iteration.

f) Remarks: Input and output are in fixed point with automatic conversion to floating point for computation. The ability to make changes in the problem specifications without repetition of preliminary iterations is provided. Shadow prices and ranges on shadow prices and cost coefficients are provided.

IBM 650 Library Program Abstracts

RENT OR BUY ANALYSIS

L. Kunitz
S. Perl
IBM, White Plains, New York

a) Purpose: This program is designed to assist management in making a rent or buy decision on a capital investment. It will compute a rate of return from one to fifteen years. The Present Value Method is utilized because it considers the time distribution of an irregular pattern of savings occurring in the future. In addition toleasing considerations, this program will make special evaluations for utilities, lease, insurance companies and nonprofit organizations. The program will also evaluate new assets and assets purchased under a special option plan. While the program description refers specifically to the purchase of IBM data processing equipment it is sufficiently general to be easily adapted for any type of capital asset.

b) Range: Not given.

c) Accuracy: Not given.

(Continued on next column)
LINEAR PROGRAMMING FORCED INVERSION VECTOR PARTITIONING CODE FOR THE AUGMENTED 650
P. F. Fisher
International Business Machines Corporation
Western Region Programming Systems
340 Wiltshire Drive
Los Angeles, California

a. Purpose: The program is designed for use with the Linear Programming
Code for the Augmented 650. It has the following features:
1. It is completely compatible with the existing version of Linear
Programming and Vector Partitioning Code for the Augmented
650.
2. Allows the analyst to pre-select the final basis variables. Selected
non-basis vectors in the matrix are forced into the basis and non-basis
vectors outside the matrix are updated and placed in the matrix if
they are in the Forced Inversion directory.
3. Experience has indicated, if a proper selection is made, the time
in completing a partitioned problem can be reduced to one-third of
the former times.

b. Accuracy: Single precision floating point.

c. Method: Vectors outside the matrix during inversion are updated by the
inversion of the previous basis. Updated vectors that are in the Forced
Inversion Directory are placed in the matrix and other vectors are
pulled out of the update from Forced Inversion continued until all
vectors have been forced into the basis. The positions in then checked
or optimized by the conventional simplex and partitioning programs.

d. Storage Requirements: The entire storage will ordinarily be required.

e. Remarks: The simulation of a system that consists of a network of
queues.

f. Equipment Specifications: Basic 650 with index registers, floating
point and I/O.

IBM 650 Library Program Abstracts

THE CORNELL RESEARCH SIMULATOR
R. H. Curylop
M. M. Johnston
W. L. Maxwell
Cornell University
Ithaca, New York

a. Purpose: To simulate the operation of a system that consists of a network of
queues.

b. Range: The minimum number of operations per job with the basic program is seven.

Accuracy: Not given.
Floating/Fixed: Not given.

Mathematical Method: Not given.

Storage Required: One hundred eight storage locations are available for records
of jobs in process.

Speed: The speed of the program varies greatly with the size of the program.
One hundred jobs are normally done in less than 30 minutes.

Reliability: Not reliable.

Remarks: The construction of a group of mathematical models of the assembly is
simulated by this program. The user must supply a sub-program describing the
assembly under study and the distributions of given dimensions.

IBM 650 System: One 533 required.

IBM 650 Library Program Abstracts

TOLERANCE SIMULATION PROGRAM
J. E. Monsma
IBM Corporation
Poughkeepsie, Illinois

a. Purpose: This program is intended to aid in the choice of tolerance values for a
manufactured item. Assembly of the item is simulated within the computer.

(Continued on next column)
IBM 650 Library Program Abstracts

**SCHEDULING**

**Node Numbering**

IBM

**PRODUCTION DAY CALENDAR**

R. L. Freeman
Portsmouth Naval Shipyard
Portsmouth, New Hampshire

- **Purpose**: This program is written to be used as a subroutine for scheduling events which are based upon normal productive working days.
- **Range**: The sample calendar is for a five-year period beginning January 1962 and ending December 1969.
- **Accuracy**: Does not apply.
- **Floating/Fixed**: Fixed decimal.
- **Mathematical Method**: Table lookup method is used.
- **Storage Required**: The calendar requires 242 storage locations, and the program requires 203 locations.
- **Speed**: Not given.
- **Relocatability**: Relocatable. See program write-up.
- **Remarks**: The program is built around two features of the IBM 650: TLS and Branch on Distributor codes. For correct input, error designations are provided which do not stop the 650 but allow the programmer to take such action as is necessary. The range of the calendar may be extended merely by relocating either the program or the table.
- **IBM 650 System**: One 533 required.
- **Special Devices**: Alphanumeric device required for the SOAP I version.

**IBM 650 Library Program Abstracts**

**LES**

Frederick Bacher, Jr.
IBM Applied Science
Dallas, Texas

- **Purpose**: The program is designed to answer the question, "At what time will a set of jobs be completed?" The program accepts an initial time for each job and calculates the total idle time on the entire line. The method employed has shown a substantial savings over hand methods.
- **Range**: Fixed point.
- **Mathematical Method**: Method employed has shown a substantial savings over hand methods. The total idle time on the entire line has been exceeded by the maximum allowable idle time on the entire line.
- **Storage Required**: The entire drum is used.
- **Speed**: Averages about 0.8 of the cases run to date.
- **Remarks**: The program is built around two features (Continued on next column)
IBM 650 Library Program Abstracts

LEAST COST ESTIMATING & SCHEDULING - SCHEDULING PHASE ONLY

M. G. Finck
Special Representative
Manufacturing Industries
3625 Wilshire Blvd.
Los Angeles, California

a. Purpose: The program has been given information about the relationship and duration of individual jobs in a project; computes project duration and develops a schedule for the project.
b. Restrictions: Range: Since integers are operated on throughout in fixed point, and time only by addition and subtraction, accuracy is assumed.
d. Storage Requirements: Almost the entire drum is used. Data (one card per job) is read at 333 read speed, schedule computations vary with project size and complexity, and the schedule is printed at punch speed (one card per job). A project of 93 jobs scheduled in 25 seconds.
e. Remarks: Projects are limited to 500 jobs or less; durations limited to four digits or less.
f. IBM 650 System: Basic 650 with 533 (99-48 beard); 407 off line for arranging and listing output.

IBM 650 Library Program Abstracts

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 memory 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.

IBM 650 Library Program Abstracts

HUMAN REACTION TIME DEMONSTRATION ROUTINE

B. M. Taylor, Jr.
North Carolina State College
Raleigh, North Carolina

a) Purpose: This program permits an operator to test his reaction time by awaiting, for rectangularly-distributed random waiting times, a signal from the console using the operator to press the program reset key. The program start key is used to initiate a new trial. A card is punched for each trial, recording a serial number, the random waiting time in hundredths of a second, and the reaction time in ten-thousandths of a second. The reaction time is also displayed on the console.
b) Range: Does not apply.
c) Does not apply.
d) Relocatability: Not given.
e) Remarks: The program deck consists of four cards.
f) IBM 650 System: One 533 required.

IBM 650 Library Program Abstracts

GENERAL PURPOSE CALENDAR PROGRAM

N. Jaspen
National League for Nursing, Inc.
New York 19, New York

a) Purpose: This program has been designed to calculate the following:
   1) The day of the week corresponding to any date in the Gregorian calendar.
   2) The difference in days between two dates.
   3) The date that is a given number of days before or after a given date.
b) Range: The program has been written on the assumption that the year can be expressed in four digits, ranging from 0001 to 9999 AD.
d) Storage Required: The program uses the first, third, fifth, and seventh read bands, and the first 3 storage locations of the 1977 punch band.
e) Remarks: The conventions used in applying the formulas are explained in the write-up when applying the formulas.
f) IBM 650 System: One 533 required.
IBM 650 Library Program Abstracts

COMPUTER AUTOMATED MUSIC
Norman V. Wiley
University of Rochester Computing Center
IBM Applied Science
Rochester, N.Y.

1. Background: The CAM program is a two phase program to produce actual musical tones via a Digital to Audio Converter connected to the operating lights of the IBM 650 computer. The first phase, the CAM Compiler, codes each note into an appropriate language for Phase II, the CAM Tune Program. Once coded, Phase II, a short program in IAS, is sufficient to produce the song again and again. The tone produced resembles a conventional organ sound and is completely successful in reproducing the musical score selected. Percussion effects, such as 650 type-horn looming to simulate drum rolls or symbolic crayons can be incorporated into the selection to enhance the musical effect.

2. Equipment: The above routines, except the last one, are written in IBM 650 System including IAS and Index Registers Digital to Audio Converter (General Electric Company, Schenectady, N.Y.).

3. Remarks: None

IBM 650 Library Program Abstracts

GO SOAP II
F. D. Greenley
P. L. Govearns
American Trust Company
San Francisco, California

GO SOAP II is a 407 pre-assembly procedure which makes the benefits of SOAP II available to those using a 650 system without the alphabetic device. The procedure requires a 407 with summary punch. No changes from SOAP II are necessary. (Rev SOAP II Reference Manual, C28-4280; formerly 12-76446)

IBM 650 Library Program Abstracts

402 CONTROL PANEL FOR SOAP II, 8-WORD LST, AND 650 LOAD CARDS
Mrs. Margaret Crowley
Computing Laboratory
The University of Oklahoma
Norman, Oklahoma

This paper describes the control panel wiring, function, and application of a control panel for the IBM 640 Accounting Machine designed for coding SOAP II input and output, 650 load cards, and eight-word output cards.

IBM 650 Library Program Abstracts

650 SOAP CONTROL PANEL WIRING SUGGESTION
O. A. De Vito
R. E. Van Allen
General Electric Company
Schenectady 5, New York

This paper describes additional wiring to the IBM 533 SOAP II control panel to detect double punches and blank columns when assembling a 650 program using SOAP II.

IBM 650 Library Program Abstracts

AUTOMATIC INFORMATION RETRIEVAL PROGRAM
J. Y. Atkin
Manager, DP Information Retrieval
IBM
R. E. Post Rd.
Palo Alto, N. Y.

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.
IBM 0704 PROGRAM LIBRARY ABSTRACT

0704 0S5UAIW1 AVAILABLE PRIOR TO JANUARY 1962

MATRIX INVERSION.

INVERTS A MATRIX STORED IN CORE STORAGE. USES AN ELIMINATION METHOD. THE STARTING ELEMENT IS THE LARGEST IN THE COLUMN, BUT THE COLUMNS ARE USED IN ORDER FROM LEFT TO RIGHT. THE ORIGINAL MATRIX IS DESTROYED AND IS REPLACED IN STORAGE BY THE INVERSE.

THE ROUTINE REQUIRES 172 CELLS PLUS 89 COMMON.

0704 0S5UAIW2 AVAILABLE PRIOR TO JANUARY 1962

FLOATING EXponential

EVALUATES FLOATING X TO FLOATING X FOR X ABSOLUTE LESS THAN OR EQUAL TO 97.3 ACCURATE TO 0 OR +1 IN EIGHTH DECIMAL DIGITS. MAXIMUM TIME ABOUT 1.22 MILLION SECONDS. USES 63 STORAGE CELLS 63 COMMON.

0704 0S5UAIW3 AVAILABLE PRIOR TO JANUARY 1962

COMPUTES FLOATING NATURAL LOG OF FLOATING X FOR X GREATER THAN 0.001. 75 SEQUENCES WITH ERROR RETURN FOR AN X OF ZERO OR LESS. ACCURATE TO 1 OR +2 IN EIGHTH DECIMAL DIGITS. MAXIMUM TIME ABOUT 1.22 MILLION SECONDS. USES 59 STORAGE CELLS 59 COMMON.

0704 0S5UAIW4 AVAILABLE PRIOR TO JANUARY 1962

READ BCD TAPE OR ON-LINE CARD READER.

READS BCD TAPE OR CARD PROGRAM READ HEADERS, SEQUENCES OR DETERMINED BY SENSE SWITCH. INFORMATION IS STORED IN CORE IN BCD FORM. ROUTINE REQUIRES 167 CELLS PLUS 9 COMMON.

0704 0S5UAIW5 AVAILABLE PRIOR TO JANUARY 1962

READS 16 CORE TAPE AND DCar LOADER.

READS CORE TAPE OR CARD PROGRAM OR COMPLEX, STORED ROW-WISE. Converted THIS INFORMATION TO BINARY, FIXED OR FLOATING DECIMAL NUMBERS BEING CONVERTED TO FIXED OR FLOATING BINARY NUMBERS. DECIMAL OR OCTAL INTEGERS ARE CONVERTED TO BINARY UNITS. 44 STORAGE CELLS 44 COMMON.

0704 0S5UAIW6 AVAILABLE PRIOR TO JANUARY 1962

DECOMPLEX.

REAL, OCTAL, BCD LOADER USED WITH UA TSM 2 OR UA CH 2. CONTROLS TAP PROGRAM UA TSM 2 OR TAP CARD PROGRAM UA CH 2 TO READ BCD INFORMATION INTO CORE. CONVERTS THIS INFORMATION TO BINARY, FIXED OR FLOATING DECIMAL NUMBERS OR COMPLEX. EACH ROW PRECEDED BY 128-BIT SEQUENCE.

0704 0S5UAIW7 AVAILABLE PRIOR TO JANUARY 1962

READ HEADER REMOVAL.

READS REAL OR COMPLEX MATRIX 50 HEADERS ARE ELIMINATED, RESULTING ELEMENTS STORED CONSECUTIVELY. USES 66 STORAGE CELLS PLUS 4 COMMON.

0704 0S5UAIW8 AVAILABLE PRIOR TO JANUARY 1962

MATRIX PUNCH.

CODES REAL OR OCTAL DECIMAL CARDS ON-LINE OR PREPARE REAL OR OCTAL DECIMAL TAPE FOR TAPE PUNCH UNIT. CARDS ACCEPTABLE TO CL MRU1. REQUIRES 400 STORAGE CELLS 400 COMMON.

0704 0S5UAIW9 AVAILABLE PRIOR TO JANUARY 1962

MATRIX EXPAND.

SHIFTS ROWS OR ROWS OF REAL OR COMPLEX MATRIX TO GIVE STORAGE FOR HEADERS, AND FORMS HEADER.. ELEMENTS IN CONSECUTIVE LOCATIONS IN ROW ORDER. USES 66 STORAGE CELLS PLUS 4 COMMON.

0704 0S5UAIW10 AVAILABLE PRIOR TO JANUARY 1962

MATRIX INTERCHANGE OF ROWS AND COLUMNS.

INTERCHANGES, DELETES, INSERTS ROWS OR COLUMNS. EITHER REAL OR COMPLEX. EACH ROW OR COLUMN MUST BE USED IN SEQUENTIALLY. REQUIRES 291 STORAGE CELLS PLUS 26 COMMON.

0704 0S5UAIW11 AVAILABLE PRIOR TO JANUARY 1962

MATRIX INVERSE.

CODES REAL, INVERTS REAL SQUARE MATRIX, USES 390 STORAGE CELLS PLUS 21 COMMON.
**IBM 0104 PROGRAM LIBRARY ABSTRACT**

**0704 166CLAS1** AVAILABLE PRIOR TO JANUARY 1962
Arc Sine and Arc Cosine
Ancillaries of Floating Point Argument, Square Root
Routine Using 3 Common Must Be Assembled Concurrently.
Requires 17 Storage Cells + 7 Common.

**0704 166CLAS2** AVAILABLE PRIOR TO JANUARY 1962
Divided Difference Interpolation
Finds Functions Y for Arguments X Using Table of Divided Differences Formed by CL 0104. Requires 136 Storage Cells + 14 Common.

**0704 166CLAS3** AVAILABLE PRIOR TO JANUARY 1962
Determinant Table Formation
Forms Divided Difference Table Up Through N-th Order, 0-1 To 6-7, From Table of Arguments and Functions. Requires 91 Storage Cells + 6 Common. Used with CL 0312.

**0704 166CLAS4** AVAILABLE PRIOR TO JANUARY 1962
Determinate and Eigenvector for Complex Matrix

**0704 166CLAS5** AVAILABLE PRIOR TO JANUARY 1962
Integral Eval., Trapez, Rule Fd, Interval
Integral and Values of Function in Floating Point. Requires 28 Storage Cells + 1 Common.

**0704 166CLAS6** AVAILABLE PRIOR TO JANUARY 1962
Least Squares Polynomial Fit

**0704 166CLAS7** AVAILABLE PRIOR TO JANUARY 1962
Least Squares S0, of Simultaneous Equations

**0704 166CLAS8** AVAILABLE PRIOR TO JANUARY 1962
Relatives Symbolic Deck
Relatives Symbolic Deck Consists of Two Decks Designated as REL and REL2. Reproduce Symbolic Deck with Location Symbol Relative to First Output Is to Tape for Off-Line Punching Only. Usead Similar to EPA in MANY RESPECTS. Uses Core and Tapes 1 and 4, and Tape 4 of Output From Tape, Revised Dist. 236

**0704 166CLAS9** AVAILABLE PRIOR TO JANUARY 1962
Simultaneous Real Equations, Determinant

**0704 166CLAS10** AVAILABLE PRIOR TO JANUARY 1962
Simultaneous Equations Complex
K Vector Solutions of N Simultaneous Equations. Requires 304 Storage Cells + 21 Common.

**IBM 0104 PROGRAM LIBRARY ABSTRACT**

**0704 166CLAS11** AVAILABLE PRIOR TO JANUARY 1962
Simultaneous Real Equations
K Vector Solutions of N Simultaneous Equations. Requires 124 Storage Cells + 7 Common.

**0704 166CLAS12** AVAILABLE PRIOR TO JANUARY 1962
Tangent
Tan X For X in Radians. Requires 63 Storage Cells + 4 Common.

**0704 1210I0MA** AVAILABLE PRIOR TO JANUARY 1962
Harmonic Analysis Subroutine

**0704 1306CNLI** AVAILABLE PRIOR TO JANUARY 1962
Random Number Generator

**0704 14115S05** AVAILABLE PRIOR TO JANUARY 1962
Solution of General Matrix Equation A X = B
Given an Array of P Columns and N Rows. M Greater Than N, B of Elements Stored Row-Wise at 1 Where X is N and B is M-N. S 995 Finds the Solution Matrix, A., of Dimension N-M, N. The Solution Matrix is Stored Row-Wise at 1. The Program Is Generally Most Useful When It Is a Column Matrix So That K is the Solution to a System of N Linear Equations in N Unknowns, Or When X is the Identity Matrix So That X is the Inverse of A, Or to Get Both the Solution and the Inverse. S 889 Uses 703 Cells and 1 Common.

**IBM 0104 PROGRAM LIBRARY ABSTRACT**

**0704 166CLAS13** AVAILABLE PRIOR TO JANUARY 1962
Integral Eval., Simpson Rule Fd, Interval
Integral and Values of Function in Floating Point. Requires 24 Storage Cells + 2 Common.

**0704 166CLAS14** AVAILABLE PRIOR TO JANUARY 1962
Least Squares Polynomial Fit

**0704 166CLAS15** AVAILABLE PRIOR TO JANUARY 1962
Integration of Special Form of 2nd Order Equ.
For Differential Equations of Second Order with First Solution Given and Order of Integration One. Proceeds to Calculate the Value of the Second Derivative. Starting Conditions for the Integration Must Be Available. S 887 Uses 80 Cells and 1 Word for Common.

**0704 166CLAS16** AVAILABLE PRIOR TO JANUARY 1962
Differential Equation Solving System
Solves a System of Ordinary Differential Equations, Any Number of Equations, of Any Order, Linear or Non-Linear May Be Solved. A Series of Table Lookups with Several Parts of the Routine. Necessity to Mules Formulas are Used After a Special Set of Starting Formulas Computes the First N Points. Requires 1494 Storage Cells. Corr. 175, 269

**0704 166CLAS17** AVAILABLE PRIOR TO JANUARY 1962
Characteristic Roots and Vectors
Computes in Floating Point Single Precision All Characteristic Roots and Vectors of a Real Symmetric Matrix. Uses a Modified Jacobi Iterative Method. Accepts Either 10 Digit Decimal Input Data Having 10 Decimal Places or 5 Bit Fractional Binary Data Having 5 Binary Places Which is 20 Scans That Within the Normal Number of the Matrix Can Be Floating Point or Fixed Point. Output Roots and Vectors Also Puts Out the Output If Desired

**0704 166CLAS18** AVAILABLE PRIOR TO JANUARY 1962
Pre-Assembly Program
Does Bookkeeping Work for North American Tape Assembly System

**0704 166CLAS19** AVAILABLE PRIOR TO JANUARY 1962
Lagrangian Interpolation Subroutine
Computes P Equals P of X Or E Equals P of X and Y, Table Values at Equal Intervals of X and Y. All Floating Point, Extrapolates for X Outside Table, Tying Independent of Table Size or Location of Point. Requires 119 Storage Cells Plus 17 Common.
COMPUTES FLOATING POINT SIMULTANEOUS EQUATIONS.

CONTINUED FRACTION SUBROUTINE

EXPONENTIAL FLOATING

IMPROVED MATRIX BY ROWS

INTEGRATION SUBROUTINE

INCOMPLETE ELLIPTIC INTEGRALS

INTEGER INTEGRATION

POLYNOMIAL COEFFICIENT REDUCTION

POLYNOMIAL COEFFICIENTS

POLYNOMIAL ARE ASSUMED TO BE COMPLEX

POLYNOMIAL COEFFICIENTS ARE REDUCED AND

POLYNOMIAL COEFFICIENTS ARE REDUCED AND

POLYNOMIAL COEFFICIENTS ARE REDUCED AND

POLYNOMIAL COEFFICIENTS ARE REDUCED AND

POLYNOMIAL COEFFICIENTS ARE REDUCED AND

POLYNOMIAL COEFFICIENTS ARE REDUCED AND

POLYNOMIAL COEFFICIENTS ARE REDUCED AND

POLYNOMIAL COEFFICIENTS ARE REDUCED AND

POLYNOMIAL COEFFICIENTS ARE REDUCED AND

POLYNOMIAL COEFFICIENTS ARE REDUCED AND

POLYNOMIAL COEFFICIENTS ARE REDUCED AND

POLYNOMIAL COEFFICIENTS ARE REDUCED AND

POLYNOMIAL COEFFICIENTS ARE REDUCED AND
IBM 0704 PROGRAM LIBRARY ABSTRACT

0704 264EMICS1 AVAILABLE PRIOR TO JANUARY 1962
INPUT-OUTPUT SYSTEM AN EXECUTIVE ROUTINE WHICH CONTROLS MULTIPLE-INPUT, MULTIPLE-OUTPUT OFF-LINE OPERATION. THE 074. OPERATES IN THREE PHASES: (1) CONVERTS ALL JOBS FROM BCD TO BINARY, (2) CONDUCTS SEQUENCING OF JOBS DURING PROGRAM EXECUTION AND (3) CONVERTS BINARY OUTPUT TO BCD FOR ALL JOBS. ALSO PROVIDES INPUT-OUTPUT SYSTEM MANAGEMENT ROUTINES, A SELECTION OF DEBUGGING ROUTINES AND JOB REJECT. REQUIRES 6 TAPES, 3 CORE, DRUM 1 AND A PROGRAMMABLE CLOCK/STOPWATCH.

0704 264MPPXX AVAILABLE PRIOR TO JANUARY 1962
PERIPHERAL CARD VERIFYER VERIFIES AN N CHARACTER BCD TAPE RECORD OF M FIELDS ON SELECTED INPUT ANP PCRS OR OUTPUT ANP PERC/2 TAPE, SUB PROGRAM OF THE M. Y. INPUT-OUTPUT SYSTEM. USES 125 LOCATIONS.

0704 264MPPXX AVAILABLE PRIOR TO JANUARY 1962
PERIPHERAL LINE PRINTER VERIFYER TO VERIFY AN N CHARACTER BCD RECORD OF M FIELDS ON A SELECTED OUTPUT TAPE FOR PERIPHERAL PRINTING.

0704 264P4N4 AVAILABLE PRIOR TO JANUARY 1962
MURA FIXED POINT ARC TANGENT ROUTINE COMPUTES ARC TANGENT OF A FIXED POINT FRACTION. REQUIRES 27 WORDS PLUS 2 COMMON. TIMEING 0.5 MS.

0704 264MPFIF AVAILABLE PRIOR TO JANUARY 1962
MURA FIXED PUNCH ROUTINE PUNCHES A BLOCK OF N WORDS FROM CORE STORAGE INTO ABSOLUTE BINARY CARDS. LOADING ADDRESS ON CARD SAME AS LOCATIONS IN STORAGE. PARAMETERS PLM MUST BE ENTERED INTO THE MUL. 45 WORDS OF PROGRAM, THE PUNCH OPERATES AT FULL SPEED /JUG CANSUP-. SELF-LOADING.

IBM 0704 PROGRAM LIBRARY ABSTRACT

0704 268XQLSL AVAILABLE PRIOR TO JANUARY 1962
24 WORD PER CARD BINARY LOADER A ONE CARD SELF-LOADING PROGRAM. THIS ROUTINE CONSECUTIVELY LOADS ABSOLUTE BINARY CARDS WITH 24 WORDS PER CARD. A PROGRAM STOP ALLOWS THE USER TO ENTER MANUALLY AN INITIAL LOADING ADDRESS INTO THE PL. THIS ADDRESS MUST BE LARGER THAN 7.

0704 268MUOZ AVAILABLE PRIOR TO JANUARY 1962
MURA FIXED DECIMAL INTEGRALS ROUTINE READS ONE OR TWO DECIMAL INTEGERS FROM A CARD AND PLACES THEM IN CORE STORAGE. STORES REQUIRED 48 WORDS PROGRAM & 6 COMMON. EXIT IS AFTER EACH CARD WITH 12R IN AC. FOR FULL-READ SPEED 24.9 NS. ARE AVAILABLE FOR COMPUTATION BETWEEN EXIT AND RE-ENTRY.

0704 268MUW AVAILABLE PRIOR TO JANUARY 1962
READ OCTAL NUMBER ROUTINE READS OCTAL ADDRESSES AND WORDS FROM CARDS, CONVERTS TO DECIMAL ADDRESSES AND PLACES THEM INTO THEIR SPECIFIED LOCATIONS. EITHER A SELF-LOADING PROGRAM OR A CLOSED SUBROUTINE WITH EXIT TO CARDS. UP TO FOUR OCTAL WORDS PER CARD ARE ALLOWED. CARD READER RATE OF 250 CARDS PER MINUTE IS MAINTAINED.

0704 268MUZON AVAILABLE PRIOR TO JANUARY 1962
MURA FIXED POINT SQUARE ROOT ROUTINE COMPUTES THE SQUARE ROOT OF A SINGLE OR DOUBLE PRECISION FIXED POINT FRACTION. REQUIRES 10 WORDS PLUS 1 COMMON. TIMEING 7.09 MINS.

0704 268MUC5 AVAILABLE PRIOR TO JANUARY 1962
MURA FIXED DECIMAL SQUARE ROUTINE COMPUTES THE SQUARE ROOT OF A SINGLE OR DOUBLE PRECISION FIXED POINT FRACTION. REQUIRES 10 WORDS PLUS 1 COMMON. TIMEING 7.09 MINS.

0704 268MUS4 AVAILABLE PRIOR TO JANUARY 1962
MURA FIXED DECIMAL ROOT ROUTINE COMPUTES THE SQUARE ROOT OF A SINGLE OR DOUBLE PRECISION FIXED POINT FRACTION. REQUIRES 10 WORDS PLUS 1 COMMON. TIMEING 7.09 MINS.

0704 268MUS4 AVAILABLE PRIOR TO JANUARY 1962
MURA FIXED DECIMAL SQUARE ROUTINE COMPUTES THE SQUARE ROOT OF A SINGLE OR DOUBLE PRECISION FIXED POINT FRACTION. REQUIRES 10 WORDS PLUS 1 COMMON. TIMEING 7.09 MINS.

0704 264MUS5 AVAILABLE PRIOR TO JANUARY 1962
MURA FIXED DECIMAL ROOT ROUTINE COMPUTES THE SQUARE ROOT OF A SINGLE OR DOUBLE PRECISION FIXED POINT FRACTION. REQUIRES 10 WORDS PLUS 1 COMMON. TIMEING 7.09 MINS.

0704 264MUS6 AVAILABLE PRIOR TO JANUARY 1962
MURA FIXED DECIMAL SQUARE ROUTINE COMPUTES THE SQUARE ROOT OF A SINGLE OR DOUBLE PRECISION FIXED POINT FRACTION. REQUIRES 10 WORDS PLUS 1 COMMON. TIMEING 7.09 MINS.

0704 264MUS7 AVAILABLE PRIOR TO JANUARY 1962
MURA FIXED DECIMAL ROOT ROUTINE COMPUTES THE SQUARE ROOT OF A SINGLE OR DOUBLE PRECISION FIXED POINT FRACTION. REQUIRES 10 WORDS PLUS 1 COMMON. TIMEING 7.09 MINS.

0704 264MUS8 AVAILABLE PRIOR TO JANUARY 1962
MURA FIXED DECIMAL SQUARE ROUTINE COMPUTES THE SQUARE ROOT OF A SINGLE OR DOUBLE PRECISION FIXED POINT FRACTION. REQUIRES 10 WORDS PLUS 1 COMMON. TIMEING 7.09 MINS.

0704 264MUS9 AVAILABLE PRIOR TO JANUARY 1962
MURA FIXED DECIMAL ROOT ROUTINE COMPUTES THE SQUARE ROOT OF A SINGLE OR DOUBLE PRECISION FIXED POINT FRACTION. REQUIRES 10 WORDS PLUS 1 COMMON. TIMEING 7.09 MINS.

0704 264MUS0 AVAILABLE PRIOR TO JANUARY 1962
MURA FIXED DECIMAL SQUARE ROUTINE COMPUTES THE SQUARE ROOT OF A SINGLE OR DOUBLE PRECISION FIXED POINT FRACTION. REQUIRES 10 WORDS PLUS 1 COMMON. TIMEING 7.09 MINS.

0704 264MUS1 AVAILABLE PRIOR TO JANUARY 1962
MURA FIXED DECIMAL ROOT ROUTINE COMPUTES THE SQUARE ROOT OF A SINGLE OR DOUBLE PRECISION FIXED POINT FRACTION. REQUIRES 10 WORDS PLUS 1 COMMON. TIMEING 7.09 MINS.

0704 264MUS2 AVAILABLE PRIOR TO JANUARY 1962
MURA FIXED DECIMAL SQUARE ROUTINE COMPUTES THE SQUARE ROOT OF A SINGLE OR DOUBLE PRECISION FIXED POINT FRACTION. REQUIRES 10 WORDS PLUS 1 COMMON. TIMEING 7.09 MINS.
IBM 0104 PROGRAM LIBRARY ABSTRACT

0704 28991945 AVAILABLE PRIOR TO JANUARY 1962
0704 ASSEMBLER OF 709 PROGRAMS MODIFICATION OF 04 SAFP TO ASSEMBLE 709 SYMBOILIC PROGRAMS ON THE 704.

0704 28991918 AVAILABLE PRIOR TO JANUARY 1962
MURA FLOATING POINT CORE ROUTIN1. COMPUTES CORE ROOT OF A NORMALIZED FLOATING POINT NUMBER RESIDING IN THE ACCUMULATOR. UPON EXIT THE NORMALIZED RESULT IS AGAIN PLACED IN THE ACCUMULATOR. REQUIRE5 50 WORDS PLUS 3 COMMON. TIMING 5.1 MS.

0704 28991944 AVAILABLE PRIOR TO JANUARY 1962
MURA FLOATING POINT DOUBLE PRECISION ADDITION ADDS TWO DOUBLE PRECISION FLOATING POINT NUMBERS, ONE LOCATED IN 104 AND THE OTHER IN COMMON AND COMPOUND. THE MAP OF EACH NUMBER MUST BE NORMALIZED. 32 WORDS OF PROGRAM PLUS 4 COMMON. TIMING 0.3-3.4 MS.

0704 28991902 AVAILABLE PRIOR TO JANUARY 1962
MURA FIXED POINT LOGARITHM, BASE 2 GIVEN A FIXED POINT FRACTION 1 MORE THAN ZERO AND LESS THAN 1, LOGARITHM & BASE 2 IS COMPUTED. MAXIMUM ERROR 2EXP-.34, MAXIMUM TIME 13.2 MS. 16 WORDS PROGRAM & 5 COMMON.

0704 28991900 AVAILABLE PRIOR TO JANUARY 1962
MURA FIXED POINT LOGARITHM, BASE 10 COMPUTES THE LOGARITHM OF A FIXED POINT NUMBER IN AN INTEGER FORM AND ANOTHER IN A DOUBLE PRECISION FRACTION. MAXIMUM ERROR 1.2 X 10-15. PLUS AUXILIARY STORAGE 480 WORDS COMMON.

0704 28991929 AVAILABLE PRIOR TO JANUARY 1962
MURA FIXED POINT SINE COMPUTES THE SINE OF AN ANGLE EXPRESSED IN RADIANS. ENTER ANGLES IN RADIAN SPACE. USES LINEAR EXPANSIONS IN ALL. EXIT WITH 1/2 SINE IN AC. MAXIMUM ERROR ± 2EXP-35. 30 WORDS PROGRAM & 3 COMMON. TIMING 3.1 MS.

IBM 0104 PROGRAM LIBRARY ABSTRACT

0704 28991925 AVAILABLE PRIOR TO JANUARY 1962
MURA FIXED POINT TANGENT COMPUTES THE TANGENT OF AN ANGLE EXPRESSED IN RADIANS. ENTER ANGLES IN RADIAN SPACE. USES LINEAR EXPANSIONS IN ALL. EXIT WITH TANGENT IN AC. MAXIMUM ERROR ± 2 EXP-35. 30 WORDS PROGRAM & 3 COMMON. TIMING 3.1 MS.

0704 28991935 AVAILABLE PRIOR TO JANUARY 1962
MURA FIXED POINT COSINE COMPUTES THE COSINE OF AN ANGLE EXPRESSED IN RADIANS. ENTER ANGLES IN RADIAN SPACE. USES LINEAR EXPANSIONS IN ALL. EXIT WITH COSINE IN AC. MAXIMUM ERROR ± 2 EXP-.34. RMS ERROR 2 EXP-.35. 34 WORDS PROGRAM & 5 COMMON. TIMING 3.1 MS.

0704 28991949 AVAILABLE PRIOR TO JANUARY 1962
MURA FIXED POINT COTANGENT COMPUTES THE COTANGENT OF AN ANGLE EXPRESSED IN RADIANS. ENTER ANGLES IN RADIAN SPACE. USES LINEAR EXPANSIONS IN ALL. EXIT WITH COTANGENT IN AC. MAXIMUM ERROR ± 12EXP-.34. RMS ERROR 2 EXP-.35. 34 WORDS PROGRAM & 5 COMMON. TIMING 3.1 MS.

0704 28991968 AVAILABLE PRIOR TO JANUARY 1962
MURA FLOATING POINT SQUARE ROUTINE COMPUTES THE SQUARE OF A SINGLE OR DOUBLE PRECISION FLOATING POINT NUMBER. FIXED POINT FRACTION. MAXIMUM ERROR 2EXP-34. PLUS AUXILIARY STORAGE 61 COMMON & 10 RESPECTIVELY. ACCURACY GOOD.

0704 28991972 AVAILABLE PRIOR TO JANUARY 1962
MURA FLOATING POINT SQUARE ROUTINE COMPUTES THE SQUARE ROOT OF A SINGLE OR DOUBLE PRECISION FLOATING POINT NUMBER. MAXIMUM ERROR 2EXP-35. PLUS AUXILIARY STORAGE 15 COMMON AND 28 RESPECTIVELY.

IBM 0104 PROGRAM LIBRARY ABSTRACT

0704 28991974 AVAILABLE PRIOR TO JANUARY 1962
MURA FLOATING POINT DOUBLE PRECISION ROUTINE FOR THE NUMBER IN CORE AT THE SPECIFIED ADDRESS. EXIT IS UPON END OF FILE OR ON 12 RIGHT WITH 12 RIGHT IN THE ACCUMULATOR AS A LOGICAL ERROR. STORAGE REQUIRED. 36 WORDS & 10 COMMON.

0704 28991983 AVAILABLE PRIOR TO JANUARY 1962
MURA FLOATING POINT DOUBLE PRECISION ROUTINE READS A DECIMAL ADDRESS AND FRACTION FROM A CARD AND PLACES THEM IN COMMON AND CORE RESPECTIVELY. ACURACY IS QUITE GOOD. STORES 20 WORDS PLUS 3 COMMON. TIMING 5.1 MS. ARE AVAILABLE BETWEEN EXIT AND RE-ENTRY.
0704 303YVWCD AVAILABLE PRIOR TO JANUARY 1962
MONITOR SUBROUTINE
PRINTS ONLINE IN OCTAL THE CONTENTS OF ANY SPECIFIED CORE LOCATION ALONG WITH ANY DESIRED OCTAL INFORMATION. THIS SUBROUTINE MAY BE USED TO MONITOR PROGRAMS, i.e., TO PRINT OUT THE CONTENTS OF A VARIABLE CONTROL WORD UPON ENCOUNTERING AN ERROR.

0704 303YVWNE AVAILABLE PRIOR TO JANUARY 1962
MONITOR SUBROUTINE AND OUTPUT PROGRAM
PRINTS ONLINE IN OCTAL THE CONTENTS OF ANY SPECIFIED CORE LOCATION ALONG WITH ANY DESIRED OCTAL INFORMATION. THIS SUBROUTINE MAY BE USED TO MONITOR PROGRAMS, i.e., TO PRINT OUT THE CONTENTS OF A VARIABLE CONTROL WORD UPON ENCOUNTERING AN ERROR. MON2 CONTAINS NP OUT WHICH MAY BE USED INDEPENDENTLY.

0704 3D1NOWGN AVAILABLE PRIOR TO JANUARY 1962
RANDUM NUMBER GENERATOR
GENERATES FIXED OR FLOATING POINT UNIFORM RANDOM NUMBERS.

0704 3D1NOWSP AVAILABLE PRIOR TO JANUARY 1962
MURA SIX COLUMN FRACTION CATHODE RAY TUBE DISPLAY
SCORE LOCATION-PRESERVING LOCATIONS AT SUCCESSIVE CORE MEMORY LOCATIONS R16 THE LINE. 133 PROGRAM PLUS 7 COMMON WORDS. TIMING 555 MS 1 LINE.

0704 3K10M0M5 AVAILABLE PRIOR TO JANUARY 1962
THE TRANSCENDENTAL FUNCTIONS MU AND NU
COMPUTATION OF THE TRANSCENDENTAL FUNCTIONS MU AND NU USED IN THE HERTZ STRESS FORMULAS, GIVEN COS thumb, MU AND MU ARE COMPUTED BY A FIFTH OR NINTH DEGREE POLYNOMIAL APPROXIMATION, REQUIRES KLIX (BASIS) ON USER(S) WITH AN ERROR RETURN, 107 CELLS & 13 COMMON WORDS.

0704 3K11MCRT3 AVAILABLE PRIOR TO JANUARY 1962
MURA FIXED POINT CUBE ROOT
COMPUTES THE CUBE ROOT OF A SINGLE OR DOUBLE PRECISION FIXED POINT FRACTION. REQUIRES 28 WORDS PROGRAM PLUS 3 TEMPORARY. TIMING IS 1.25 MS PER ITERATION.

0704 3K11MUPPA AVAILABLE PRIOR TO JANUARY 1962
MURA SIX COLUMN FRACTION PRINT
TO PRINT SIX FIXED-POINT FRACTIONS ON ONE LINE OF THE 161A PRINTER. THE LOCATION OF THE FIRST FRACTION IS GIVEN IN THE CALLING SEQUENCE. A MAXIMUM ERROR OF 5 IN THE ELEVENTH DECIMAL PLACE IS INTRODUCED DURING CONVERSION. THE SHARE PRINTER BOARD NO. 1 IS USED. 134.5 MS OF CALCULATING TIME IS AVAILABLE BETWEEN SUCCESSIVE ENTRIES WITHOUT REDUCING THE PRINTER SPEED OF 150 LINES PER MINUTE.

0704 3K11MUAR3 AVAILABLE PRIOR TO JANUARY 1962
MURA FLOATING POINT HUNGE-HUTT
SOLVES A SET OF N SIMULTANEOUS FIRST ORDER DIFFERENTIAL EQUATIONS. 116 WORDS OF PROGRAM (4 WORDS TEMPORARY) & WORDS OF STORAGE. TIMING 75MLA 116VOSFM 5 & AUXILIARY SUBROUTINE TIME/MS. PER INTEGRATION STEP.

0704 3K11MUSEP3 AVAILABLE PRIOR TO JANUARY 1962
GENERAL ALPHANUMERICAL CATHODE RAY TUBE DISPLAYS ALPHANUMERIC MESSAGES ON THE 716 OUTPUT RECORDER. 144 WORDS PROGRAM & 7 WORDS COMMON. FILE ABOUT 0.5 MILLION CHARACTERS PER SECOND.

0704 3K11NARZ29 AVAILABLE PRIOR TO JANUARY 1962
PACT II SAMPLE PROGRAM
PROVIDED AN EXAMPLE OF PACT II INPUT AND OUTPUT AND PROVIDES A SIMPLE TEST OF COMPILER OPERATION ON ANY MACHINE CONFIGURATION. PROGRAM IS WRITTEN IN PACT LANGUAGE.

0704 3K11NBR03 AVAILABLE PRIOR TO JANUARY 1962
DOUBLE-PRECISION FLOATING BINARY MATRIX CONVERSION PROGRAM
TO CONVERT A MATRIX OR VECTOR IN FLOATING DECIMAL ON A BCD TAPE TO DOUBLE-PRECISION FLOATING BINARY ON A BINARY TAPE, ZEROS INSERTED WHERE NECESSARY.

0704 3K11NM090 AVAILABLE PRIOR TO JANUARY 1962
SMOOTH AND DIFFERENTIATE UNSEQUENTIALLY SPACED DATA POINTS TO SMOOTH N POINTS, WHERE N IS EQUAL TO OR LESS THAN 7, WHICH MAY BE UNSEQUENTIALLY SPACED. THE METHOD IS LEAST SQUARES OPTIMIZATION RANDOM ERRORS+. DISCARD WILD POINTS AND TO DIFFERENTIATE ARE PROVIDED. THIS ROUTINE DEPENDS ON THE LEFT IN THAT THE FIRST DATA POINT IS ANCHORED. I.E., UNCHANGED, SO THAT THE CURVE WILL ALWAYS PASS THROUGH THIS POINT. REQUIRES 148 WORDS PLUS 56 COMMON.

0704 3K11NMT02 AVAILABLE PRIOR TO JANUARY 1962
TAP EDITOR AND DUPLICATOR WITH COMPARE
TO TRANSFER AND/OR COMPARE IN ANY ORDER, ANY RECORDS OR ANY FILES FROM ANY TAPE TO ANY OTHER TAPE. AVAILABLE PRIOR TO JANUARY 1962.

0704 3K11NMT03 AVAILABLE PRIOR TO JANUARY 1962
TAP EDITOR AND DUPLICATOR WITH COMPARE
TO TRANSFER AND/OR COMPARE IN ANY ORDER, ANY RECORDS OR ANY FILES FROM ANY TAPE TO ANY OTHER TAPE. AVAILABLE PRIOR TO JANUARY 1962.

0704 3K11NMT04 AVAILABLE PRIOR TO JANUARY 1962
TAP EDITOR AND DUPLICATOR WITH COMPARE
TO TRANSFER AND/OR COMPARE IN ANY ORDER, ANY RECORDS OR ANY FILES FROM ANY TAPE TO ANY OTHER TAPE. AVAILABLE PRIOR TO JANUARY 1962.

0704 3K11NMT05 AVAILABLE PRIOR TO JANUARY 1962
TAP EDITOR AND DUPLICATOR WITH COMPARE
TO TRANSFER AND/OR COMPARE IN ANY ORDER, ANY RECORDS OR ANY FILES FROM ANY TAPE TO ANY OTHER TAPE. AVAILABLE PRIOR TO JANUARY 1962.

0704 3K11NMT06 AVAILABLE PRIOR TO JANUARY 1962
TAP EDITOR AND DUPLICATOR WITH COMPARE
TO TRANSFER AND/OR COMPARE IN ANY ORDER, ANY RECORDS OR ANY FILES FROM ANY TAPE TO ANY OTHER TAPE. AVAILABLE PRIOR TO JANUARY 1962.
BINARY DECK MINIMIZER
REQUIRES THE SIZE OF A RELOCATABLE BINARY DECK OR AN
ABSOLUTE BINARY DECK CONTAINING PATCH CARDS BY PUNCHING
WHEN ABSOLUTE DECK, USES CELLS 0-35

DOUBLE PRECISION INPUT SCALING
FRANK MAZORI CONVERTS A GIVEN DOUBLE PRECISION BINARY
INTEGER TO A SCALING-FLOATING AND NORMALIZED DOUBLE PRECISION
BINARY NUMBER X WITH COMPATIBLE SIGNS AND CHARACTERISTIC OF L
SH EQUAL CHARACTERISTIC OF MAN LESS 1
SPACE REQUIRED 103 CELLS

DOUBLE PRECISION OUTPUT SCALING
FRANK MAZORI EFFECTS A DOUBLE PRECISION SCALING OF BINARY
NUMBER TO A DOUBLE PRECISION BINARY INTEGER FOR OUTPUT.
SPACE REQUIRED 100 CELLS

MOVING AVERAGES OF TIME-SERIES DATA
TO ANALYZE A SET OF NON-STATIONARY TIME-SERIES DATA FOR
PERIODIC AND TREND COMPONENTS, MOVING AVERAGES OF THE DATA
ARE USED TO MEASURE THE TREND OR NON-STATIONARY COMPONENTS
AND PERIODIC COMPONENTS. MOVING AVERAGES INDICATE SHORTER SHIFTS IN
PERIODIC AVERAGES GIVE AN ESTIMATE OF THE PERIODIC COMPONENTS
IN THE ORIGINAL DATA. THE OUTPUT OF MOVING AVERAGES AND
DEVIATIONS MAY BE USED DIRECTLY AS INPUT WITH NYB F22. IT WILL
HANDLE UP TO 3200 OBSERVATIONS.

EIGENVALUE SOLUTION, REAL
TO FIND THE HIGHEST EIGENVALUE AND CORRESPONDING EIGENVECTOR
OF A REAL SYMMETRIC MATRIX. IF A SUB OR SUB 1 IF IS THE
ASSOCIATED EIGENVECTOR OF THE MATRIX A.,THE MATRIX
EIGENVALUES, COMPS MUST BE ASSEMBLED CONCURRENTLY
REQUIRES 451 WORDS PLUS COMMON THROUGH COMMON 460 PLUS
THE MATRIX MULTIPLY ROUTINES, DRUMS 2,3,4 AND TAPE 2

ATMOSPHERIC DATA SUBROUTINE
THIS SUBROUTINE EFFECTIVELY REPRODUCES PORTIONS OF THE
ATMOSPHERIC DATA BASED ON THE ARDC MODEL ATMOSPHERE
FOR X UP TO 35 MILESTOMES.
GIVEN ALTITUDE, FIND CORRESPONDING TEMPERATURE IN
DECIMEL MKS. PRESSURE, DENSITY RATIO AND WEIGHT DENSITIES
REQUIRES A SQUARE ROOT, LOGARITHM AND EXPONENTIAL
SUBROUTINES. SPACE REQUIRED 120 STORAGE CELLS PLUS 2 COMMON NEEDED
FOR SQRT, EXP, AND LOG. SUBROUTINES, TIME APPROX 12.5MS.

TABLE SEARCH ROUTINE
ROUTINE USES BINARY SEARCH TECHNIQUE TO FIND AN ENTRY
IN A TABLE CONTAINING 999 ITEMS. THE ROUTINE USES A CENTRAL
SEARCH LOOP CONSUMES NINE CYCLES FOR EACH ENTRY EXAMINED, TABLE LENGTH MAY VARY
FROM ONE WORD TO ALL OF STORAGE. MEN SEARCH TIME FOR A
1000 WORD TABLE 151.260 MS. RS 0166 REQUIRES 65
SOURCE CELLS PLUS TWO COMMON, ROUTINE IS NON-STANDARD IN THE SENSE THAT THE RESULT APPEARS IN INDEX 1.

THIS SUBROUTINE SAVES THE CONSOLE SYNCHRONIZED, AC AND TO OVERFLOW, DEVICE CHCK, TAPE CHCK, A SENSE LIGHTS,
THE CONSOLE TEAR OFF LIGHTS, THE CONSOLE TEAR OFF LIGHTS, A SELF LOADING TAPE, PHM TAPE WILL LOAD ITSELF, RESTORE CORES
AND THE CONSOLE AND RETURN CONTROL TO THE MAIN PROGRAM.

THIS SUBROUTINE SAVES THE CONSOLE SYNCHRONIZED, AC AND TO OVERFLOW, DEVICE CHCK, TAPE CHCK, A SENSE LIGHTS,
THE CONSOLE TEAR OFF LIGHTS, THE CONSOLE TEAR OFF LIGHTS, A SELF LOADING TAPE, PHM TAPE WILL LOAD ITSELF, RESTORE CORES
AND THE CONSOLE AND RETURN CONTROL TO THE MAIN PROGRAM.

SHARE ASSEMBLER
ASSEMBLES PROGRAMS WRITTEN IN SYMPLIC FORM, INPUT AND OUTPUT
MAY BE EITHER DUMP-OR BINARY OR SCALED FLOATING POINT, THE OUTPUT INCLUDES THE
GIVEN PROGRAM IN SYMPLIC AND THE ASSEMBLED PROGRAM IN
DETAIL. OUTPUT IS ALSO DUMPED ON BINARY CARDS OR IT MAY BE
WRITTEN ON TAPE IN BINARY CARD IMAGE FORM. DECIMAL, OCTAL,
AND BINARY DATA MAY BE USED. A LIBRARY OF STANDARD SUB-
ROUTINES IS PROVIDED AS AN Appendix, WHICH IS PERFORMED. IN ASCII 3-7 SUMERSE ASC 12-2, CORR. 31457,
WRITE-UP DISK 654, CMX 176

THE F SYSTEM
THIS IS AN EXECUTIVE PROGRAM THAT CONTROLS PROGRAM
TO ALLOW MULTI-JOB-MULTI-FUNCTION OPERATION, ANY
COMPARISON OF COMPLEX, EXECUTE, OR COMPARE AND EXECUTE
JAPS MAY BE PLACED ON THE INPUT TAPE. NORMAL OPERATION
IN QUEST OF THE SYSTEM BREAKS MAY BE Terminated BY A PERIPHERAL
SIGNAL, SUCH AS A PRINT END, IT WILL WRITE-UP DISK 654, CORR.
176

BINARY DECK MINIMIZER
REQUIRES THE SIZE OF A RELOCATABLE BINARY DECK OR AN
ABSOLUTE BINARY DECK CONTAINING PATCH CARDS BY PUNCHING
WHEN ABSOLUTE DECK, USES CELLS 0-35

DOUBLE PRECISION INPUT SCALING
FRANK MAZORI CONVERTS A GIVEN DOUBLE PRECISION BINARY
INTEGER TO A SCALING-FLOATING AND NORMALIZED DOUBLE PRECISION
BINARY NUMBER X WITH COMPATIBLE SIGNS AND CHARACTERISTIC OF L
SH EQUAL CHARACTERISTIC OF MAN LESS 1
SPACE REQUIRED 103 CELLS

DOUBLE PRECISION OUTPUT SCALING
FRANK MAZORI EFFECTS A DOUBLE PRECISION SCALING OF BINARY
NUMBER TO A DOUBLE PRECISION BINARY INTEGER FOR OUTPUT.
SPACE REQUIRED 100 CELLS

MOVING AVERAGES OF TIME-SERIES DATA
TO ANALYZE A SET OF NON-STATIONARY TIME-SERIES DATA FOR
PERIODIC AND TREND COMPONENTS, MOVING AVERAGES OF THE DATA
ARE USED TO MEASURE THE TREND OR NON-STATIONARY COMPONENTS
AND PERIODIC COMPONENTS. MOVING AVERAGES INDICATE SHORTER SHIFTS IN
PERIODIC AVERAGES GIVE AN ESTIMATE OF THE PERIODIC COMPONENTS
IN THE ORIGINAL DATA. THE OUTPUT OF MOVING AVERAGES AND
DEVIATIONS MAY BE USED DIRECTLY AS INPUT WITH NYB F22. IT WILL
HANDLE UP TO 3200 OBSERVATIONS.

EIGENVALUE SOLUTION, REAL
TO FIND THE HIGHEST EIGENVALUE AND CORRESPONDING EIGENVECTOR
OF A REAL SYMMETRIC MATRIX. IF A SUB OR SUB 1 IF IS THE
ASSOCIATED EIGENVECTOR OF THE MATRIX A.,THE MATRIX
EIGENVALUES, COMPS MUST BE ASSEMBLED CONCURRENTLY
REQUIRES 451 WORDS PLUS COMMON THROUGH COMMON 460 PLUS
THE MATRIX MULTIPLY ROUTINES, DRUMS 2,3,4 AND TAPE 2

ATMOSPHERIC DATA SUBROUTINE
THIS SUBROUTINE EFFECTIVELY REPRODUCES PORTIONS OF THE
ATMOSPHERIC DATA BASED ON THE ARDC MODEL ATMOSPHERE
FOR X UP TO 35 MILESTOMES.
GIVEN ALTITUDE, FIND CORRESPONDING TEMPERATURE IN
DECIMEL MKS. PRESSURE, DENSITY RATIO AND WEIGHT DENSITIES
REQUIRES A SQUARE ROOT, LOGARITHM AND EXPONENTIAL
SUBROUTINES. SPACE REQUIRED 120 STORAGE CELLS PLUS 2 COMMON NEEDED
FOR SQRT, EXP, AND LOG. SUBROUTINES, TIME APPROX 12.5MS.

TABLE SEARCH ROUTINE
ROUTINE USES BINARY SEARCH TECHNIQUE TO FIND AN ENTRY
IN A TABLE CONTAINING 999 ITEMS. THE ROUTINE USES A CENTRAL
SEARCH LOOP CONSUMES NINE CYCLES FOR EACH ENTRY EXAMINED, TABLE LENGTH MAY VARY
FROM ONE WORD TO ALL OF STORAGE. MEN SEARCH TIME FOR A
1000 WORD TABLE 151.260 MS. RS 0166 REQUIRES 65
SOURCE CELLS PLUS TWO COMMON, ROUTINE IS NON-STANDARD IN THE SENSE THAT THE RESULT APPEARS IN INDEX 1.

THIS SUBROUTINE SAVES THE CONSOLE SYNCHRONIZED, AC AND TO OVERFLOW, DEVICE CHCK, TAPE CHCK, A SENSE LIGHTS,
THE CONSOLE TEAR OFF LIGHTS, THE CONSOLE TEAR OFF LIGHTS, A SELF LOADING TAPE, PHM TAPE WILL LOAD ITSELF, RESTORE CORES
AND THE CONSOLE AND RETURN CONTROL TO THE MAIN PROGRAM.

THIS SUBROUTINE SAVES THE CONSOLE SYNCHRONIZED, AC AND TO OVERFLOW, DEVICE CHCK, TAPE CHCK, A SENSE LIGHTS,
THE CONSOLE TEAR OFF LIGHTS, THE CONSOLE TEAR OFF LIGHTS, A SELF LOADING TAPE, PHM TAPE WILL LOAD ITSELF, RESTORE CORES
AND THE CONSOLE AND RETURN CONTROL TO THE MAIN PROGRAM.

SHARE ASSEMBLER
ASSEMBLES PROGRAMS WRITTEN IN SYMPLIC FORM, INPUT AND OUTPUT
MAY BE EITHER DUMP-OR BINARY OR SCALED FLOATING POINT, THE OUTPUT INCLUDES THE
GIVEN PROGRAM IN SYMPLIC AND THE ASSEMBLED PROGRAM IN
DETAIL. OUTPUT IS ALSO DUMPED ON BINARY CARDS OR IT MAY BE
WRITTEN ON TAPE IN BINARY CARD IMAGE FORM. DECIMAL, OCTAL,
AND BINARY DATA MAY BE USED. A LIBRARY OF STANDARD SUB-
ROUTINES IS PROVIDED AS AN Appendix, WHICH IS PERFORMED. IN ASCII 3-7 SUMERSE ASC 12-2, CORR. 31457,
WRITE-UP DISK 654, CMX 176

THE F SYSTEM
THIS IS AN EXECUTIVE PROGRAM THAT CONTROLS PROGRAM
TO ALLOW MULTI-JOB-MULTI-FUNCTION OPERATION, ANY
COMPARISON OF COMPLEX, EXECUTE, OR COMPARE AND EXECUTE
JAPS MAY BE PLACED ON THE INPUT TAPE. NORMAL OPERATION
IN QUEST OF THE SYSTEM BREAKS MAY BE Terminated BY A PERIPHERAL
SIGNAL, SUCH AS A PRINT END, IT WILL WRITE-UP DISK 654, CORR.
176
0704 355CMAT81  AVAILABLE PRIOR TO JANUARY 1962
TABLE INTERPOLATION
ALL FLOATING-POINT VALUES, GIVEN X COMPUTES Y EQUALS F OF X FROM A TABLE. REQUIRED, USUAL X, Y, AND A SEQUENCE WITH RETURN TO X, Y. REQUIRED 99 STORAGE CELLS COMMON DEPENDING UPON TABLE SIZE. EXTRAPOLATES FOR X OUTSIDE TABLE. 8MB/408

0704 356 CA0015  AVAILABLE PRIOR TO JANUARY 1962
DOUBLE PRECISION SIMULTANEOUS REAL EQUATIONS.
K VECTOR SOLUTIONS AND DETERMINANT OF SIMULTANEOUS EQUATIONS. REQUIRES 54 STORAGE PLUS 8 COMMON.

0704 357MUL00  AVAILABLE PRIOR TO JANUARY 1962
NUA FIXED POINT LOGARITHM BASE 2.
GIVEN A FIXED POINT FRACTION X MORE THAN 0 AND LESS THAN 1, LOGARITHM BASE 2 IS COMPUTED. MINIMUM ERROR 0.003
MINIMUM TIME 1.45 MS, MAXIMUM TIME 19.9 MS. 38 WORDS PROGRAM OF 6 COMMON.

0704 358MUNIC2  AVAILABLE PRIOR TO JANUARY 1962
HC1F FIXED POINT MATLAB-4 QUADRATURE
APPROXIMATES THE VALUE OF AN INTEGRAL OF THE FORM \( \int f(x) \, dx \) BETWEEN \( x = a \) AND \( x = b \). THE VALUES FOR \( a \) AND \( b \) ARE FIXED AT 0 AND 1, RESPECTIVELY. THE FUNCTION \( f(x) \) IS TYPICALLY A POLYNOMIAL OR A RATIONAL FUNCTION. OCCUPIES 37 STORAGE CELLS PLUS 10 TEMORARY. TIMING IS ABOUT 6 MS PER INTEGRATION STEP.

0704 359MUPS25  AVAILABLE PRIOR TO JANUARY 1962
MURA VARIABLE COLUMN FRACTION PRINT
THIS ROUTINE PRINTS ON LINE, ONE TO FIVE FIXED POINT FRACTIONS PLUS A FIVE DIGIT INTEGER LINE LABEL. THE MURA PRINTER BOARD 1 IS REQUIRED. ACCURATE TO -3 IN THE ELEVENTH DECIMAL PLACE. THE PROGRAM USES 82 WORDS STORAGE PLUS 20 WORDS COMMON.

0704 359MUPR2F  AVAILABLE PRIOR TO JANUARY 1962
MURA VARIABLE COLUMN FRACTION PRINT
THIS ROUTINE PRINTS ON LINE, ONE TO FIVE FIXED POINT FRACTIONS PLUS AN INTEGER LINE LABEL. THE MODIFIED SHARE 1 BOARD IS REQUIRED. ACCURATE TO -3 IN THE ELEVENTH DECIMAL PLACE. THE PROGRAM USES 81 WORDS STORAGE PLUS 26 WORDS COMMON.

0704 359MUSE29  AVAILABLE PRIOR TO JANUARY 1962
SCOPE GRID PLOTTER
TO DISPLAY ON THE TV OUTPUT RECORD A GRID OF HORIZONTAL AND VERTICAL LINES. PROVISION IS MADE FOR PLOTTING CERTAIN SPECIFIED LINES HEAVIER THAN OTHERS. PROGRAM REQUIRES 55 WORDS STORAGE PLUS 2 TEMORARY.

0704 359OLSMP1  AVAILABLE PRIOR TO JANUARY 1962
BDC ADD-SUBTRACT
ADDS OR SUBTRACTS TWO SIGNED 12 DIGIT BCD NUMBERS. ADDS 6 DIGIT BCD NUMBERS PLUS ELECTRONIC ADDER, TO RESTORE CORRECT BCD FORM. 62 STORAGE LOC PLUS 1 COMMON MINIMUM TIMING 1.0 MS, MAXIMUM OVERALL 2.5 MS.

0704 359OLSMP2  AVAILABLE PRIOR TO JANUARY 1962
BDC ARITHMETIC CORRECT
ADDS OR SUBTRACTS TWO SIGNED 12 DIGIT BCD NUMBERS PLUS ELECTRONIC ADDER, TO RESTORE CORRECT BCD FORM. 62 STORAGE LOC PLUS 1 COMMON. MINIMUM TIMING 340 MICROSEC, MAXIMUM 390 MICROSEC.

0704 359OLSMP3  AVAILABLE PRIOR TO JANUARY 1962
BINARY TO PACKED BCD CONVERTER
CONVERTS SIGN-BINARY NUMBERS IN CONSECUTIVE LOCATIONS TO EQUIVALENT BCD NUMBERS ALSO IN CONSECUTIVE LOCATIONS. SIGNS MAY BE IGNORED IF DESIRED.

0704 359OLSMP4  AVAILABLE PRIOR TO JANUARY 1962
GENERAL SORT ROUTINE
TO SORT A TABLE IN WHICH THE UNIT RECORD IS LONGER THAN ONE FOR WORD, MASKS MAY BE USED TO SELECT THE Bits OF A RECORD TO BE USED IN SORTING.

0704 361MNAL01  AVAILABLE PRIOR TO JANUARY 1962
WRITE 6-4 DIGIT DECIMAL INTEGER AND SIGN ON CAT.
K. SHIMIZU: WRITE 6-DIGIT DECIMAL INTEGER WITH BINARY SCALE 35 AT SPECIFIED LOCATION ON CRT. WILL PRINT MINUS SIGNS AND SUPPRESSES PLUS SIGNS. SPACE REQUIRED = 50 LOCATIONS PLUS 6 WORDS OF A MODIFIED VERSION OF NA-209 WHICH INCLUDES A TABLE OF TEN CHARACTERISTIC WORDS.

0704 362NYAN02  AVAILABLE PRIOR TO JANUARY 1962
AUTOCORRELATION ANALYSIS
NYAN2 PERMITS A AUTOCORELATION ANALYSIS TO BE PERFORMED UPON THE RESULTS OF AN AUTOCORRELATION ANALYSIS, THE AUTOCORRELATION ANALYSIS IS PERFORMED BY NYAN1. THE AUTOCORRELATION ANALYSIS IS PERFORMED BY CERTAIN PARTS OF NYAN2. THE NYAN2 PROGRAM HAS BEEN SO MODIFIED THAT ITS OUTPUT MAY BE DIRECTLY UTILIZED BY THE AUTOCORRELATION PARTS OF NYAN2.

0704 363NYAN03  AVAILABLE PRIOR TO JANUARY 1962
AUTOCORRELATION ANALYSIS
NYAN3 PERMITS A AUTOCORRELATION ANALYSIS TO BE PERFORMED UPON THE RESULTS OF AN AUTOCORRELATION ANALYSIS, THE AUTOCORRELATION ANALYSIS IS PERFORMED BY NYAN2. THE AUTOCORRELATION ANALYSIS IS PERFORMED BY CERTAIN PARTS OF NYAN3. THE NYAN3 PROGRAM HAS BEEN SO MODIFIED THAT ITS OUTPUT MAY BE DIRECTLY UTILIZED BY THE AUTOCORRELATION PARTS OF NYAN3.

0704 364NYAN04  AVAILABLE PRIOR TO JANUARY 1962
AUTOREgression ANALYSIS

0704 365NYAN05  AVAILABLE PRIOR TO JANUARY 1962
AUTORESSON ANALYSIS
NYAN5 PERMITS A AUTORESSON ANALYSIS TO BE PERFORMED UPON THE RESULTS OF AN AUTORESSON ANALYSIS, THE AUTORESSON ANALYSIS IS PERFORMED BY NYAN4. THE AUTORESSON ANALYSIS IS PERFORMED BY CERTAIN PARTS OF NYAN5. THE NYAN5 PROGRAM HAS BEEN SO MODIFIED THAT ITS OUTPUT MAY BE DIRECTLY UTILIZED BY THE AUTORESSON PARTS OF NYAN5.

0704 366NYAN06  AVAILABLE PRIOR TO JANUARY 1962
AUTORESSON ANALYSIS

0704 367MNAT2  AVAILABLE PRIOR TO JANUARY 1962
GENERAL MATRIX ABSTRACT FROM TAPE.
USED IN CONJUNCTION WITH NA-107 FOR MATRIX MANIPULATIONS WHERE EITHER BOTH OF THE MATRICES A AND B ARE TOO LARGE FOR AVAILABLE C.S. PERFORMS THE FOLLOWING MATRIX OPERATIONS ON REAL OR COMPLEX MATRICES.

1. ADD
2. MULTIPY
3. TRANSPONSE

0704 368NAX2740  AVAILABLE PRIOR TO JANUARY 1962
SINGLE INTEGRATION SUBROUTINE
ROGER MILLS COMPUTES A SINGLE INTEGRAL FUNCTION OVER A FINITE RANGE. USES COTES NUMBERS AS WEIGHTING COEFFICIENTS. SPACE REQUIRED = 59 LOCATIONS PLUS 5 COMMON.

0704 369NAX2750  AVAILABLE PRIOR TO JANUARY 1962
DOUBLE INTEGRATION SUBROUTINE
ROGER MILLS COMPUTES A TWICE ITERATED INTEGRAL OF A SINGLE INTEGRABLE FUNCTION OF A SINGLE VARIABLE OVER A FINITE RANGE. USES COTES NUMBERS AS WEIGHTING COEFFICIENTS. SPACE REQUIRED = 56 LOCATIONS PLUS 6 COMMON.

0704 369NAX2760  AVAILABLE PRIOR TO JANUARY 1962
TRIPLE INTEGRATION SUBROUTINE
ROGER MILLS COMPUTES A THREE ITERATED INTEGRAL OF A SINGLE INTEGRABLE FUNCTION OF A SINGLE VARIABLE OVER A FINITE RANGE. USES COTES NUMBERS AS WEIGHTING COEFFICIENTS. SPACE REQUIRED = 69 LOCATIONS PLUS 8 COMMON.

0704 370DSS103  AVAILABLE PRIOR TO JANUARY 1962
NORMALIZED ADD-EXTENDED RANGE FLOATING BINARY ARITH.
TO ADD OR SUBTRACT TWO NUMBERS EXPRESSED IN EXTENDED RANGE FLOATING BINARY. EACH NUMBER OCCUPIES 2 MEMORY LOCATIONS PLUS 96 BIT EXPONENT, 63 CELLS & 2 CELLS OF COMMON.
IBM 0704 PROGRAM LIBRARY ABSTRACT

0704 065PFQCON
Available prior to January 1962

SIX ROUTINES FORM A PACKAGE WHICH INCLUDES SOFTWARE TO FLEXIBLE READ-WRITE TAPE PROGRAM.

0704 049PSPCB2
Available prior to January 1962

CALCULATES THE COMPLEX POLYNOMIAL WITH REAL OR COMPLEX COEFFICIENTS.

0704 045SPFECB2
Available prior to January 1962

CALCULATES THE COMPLEX POLYNOMIAL WITH REAL OR COMPLEX COEFFICIENTS.

0704 045PFCB2
Available prior to January 1962

CALCULATES THE COMPLEX POLYNOMIAL WITH REAL OR COMPLEX COEFFICIENTS.

0704 045PFQPR
Available prior to January 1962

CALCULATES THE COMPLEX POLYNOMIAL WITH REAL OR COMPLEX COEFFICIENTS.

0704 045PFQPS
Available prior to January 1962

CALCULATES THE COMPLEX POLYNOMIAL WITH REAL OR COMPLEX COEFFICIENTS.

0704 045PFQPR
Available prior to January 1962

CALCULATES THE COMPLEX POLYNOMIAL WITH REAL OR COMPLEX COEFFICIENTS.

0704 045PFQPP
Available prior to January 1962

CALCULATES THE COMPLEX POLYNOMIAL WITH REAL OR COMPLEX COEFFICIENTS.

0704 045PFQPP
Available prior to January 1962

CALCULATES THE COMPLEX POLYNOMIAL WITH REAL OR COMPLEX COEFFICIENTS.

0704 045PFQPP
Available prior to January 1962

CALCULATES THE COMPLEX POLYNOMIAL WITH REAL OR COMPLEX COEFFICIENTS.

0704 045PFQPP
Available prior to January 1962

CALCULATES THE COMPLEX POLYNOMIAL WITH REAL OR COMPLEX COEFFICIENTS.

0704 045PFQPP
Available prior to January 1962

CALCULATES THE COMPLEX POLYNOMIAL WITH REAL OR COMPLEX COEFFICIENTS.

0704 045PFQPP
Available prior to January 1962

CALCULATES THE COMPLEX POLYNOMIAL WITH REAL OR COMPLEX COEFFICIENTS.

0704 045PFQPP
Available prior to January 1962

CALCULATES THE COMPLEX POLYNOMIAL WITH REAL OR COMPLEX COEFFICIENTS.

0704 045PFQPP
Available prior to January 1962

CALCULATES THE COMPLEX POLYNOMIAL WITH REAL OR COMPLEX COEFFICIENTS.

0704 045PFQPP
Available prior to January 1962

CALCULATES THE COMPLEX POLYNOMIAL WITH REAL OR COMPLEX COEFFICIENTS.

0704 045PFQPP
Available prior to January 1962

CALCULATES THE COMPLEX POLYNOMIAL WITH REAL OR COMPLEX COEFFICIENTS.

0704 045PFQPP
Available prior to January 1962

CALCULATES THE COMPLEX POLYNOMIAL WITH REAL OR COMPLEX COEFFICIENTS.

0704 045PFQPP
Available prior to January 1962

CALCULATES THE COMPLEX POLYNOMIAL WITH REAL OR COMPLEX COEFFICIENTS.

0704 045PFQPP
Available prior to January 1962

CALCULATES THE COMPLEX POLYNOMIAL WITH REAL OR COMPLEX COEFFICIENTS.

0704 045PFQPP
Available prior to January 1962

CALCULATES THE COMPLEX POLYNOMIAL WITH REAL OR COMPLEX COEFFICIENTS.

0704 045PFQPP
Available prior to January 1962

CALCULATES THE COMPLEX POLYNOMIAL WITH REAL OR COMPLEX COEFFICIENTS.

0704 045PFQPP
Available prior to January 1962

CALCULATES THE COMPLEX POLYNOMIAL WITH REAL OR COMPLEX COEFFICIENTS.

0704 045PFQPP
Available prior to January 1962

CALCULATES THE COMPLEX POLYNOMIAL WITH REAL OR COMPLEX COEFFICIENTS.

0704 045PFQPP
Available prior to January 1962

CALCULATES THE COMPLEX POLYNOMIAL WITH REAL OR COMPLEX COEFFICIENTS.

0704 045PFQPP
Available prior to January 1962

CALCULATES THE COMPLEX POLYNOMIAL WITH REAL OR COMPLEX COEFFICIENTS.

0704 045PFQPP
Available prior to January 1962

CALCULATES THE COMPLEX POLYNOMIAL WITH REAL OR COMPLEX COEFFICIENTS.

0704 045PFQPP
Available prior to January 1962

CALCULATES THE COMPLEX POLYNOMIAL WITH REAL OR COMPLEX COEFFICIENTS.

0704 045PFQPP
Available prior to January 1962

CALCULATES THE COMPLEX POLYNOMIAL WITH REAL OR COMPLEX COEFFICIENTS.

0704 045PFQPP
Available prior to January 1962

CALCULATES THE COMPLEX POLYNOMIAL WITH REAL OR COMPLEX COEFFICIENTS.

0704 045PFQPP
Available prior to January 1962

CALCULATES THE COMPLEX POLYNOMIAL WITH REAL OR COMPLEX COEFFICIENTS.

0704 045PFQPP
Available prior to January 1962

CALCULATES THE COMPLEX POLYNOMIAL WITH REAL OR COMPLEX COEFFICIENTS.

0704 045PFQPP
Available prior to January 1962

CALCULATES THE COMPLEX POLYNOMIAL WITH REAL OR COMPLEX COEFFICIENTS.

0704 045PFQPP
Available prior to January 1962

CALCULATES THE COMPLEX POLYNOMIAL WITH REAL OR COMPLEX COEFFICIENTS.

0704 045PFQPP
Available prior to January 1962

CALCULATES THE COMPLEX POLYNOMIAL WITH REAL OR COMPLEX COEFFICIENTS.

0704 045PFQPP
Available prior to January 1962

CALCULATES THE COMPLEX POLYNOMIAL WITH REAL OR COMPLEX COEFFICIENTS.
DATA PRODUCES TO DUMP STORAGES, CORE, DRUM, AND TAPES. THIS IS A MODIFICATION OF NO DUMP WHICH WILL DUMP CORES, DRUMS, AND TAPES, NOT REQUIRING THE USE OF A LOGICAL DUMP FOR SAVING THE FIRST 2069 WORDS OF CORE MEMORY. A MAGNETIC TAPE LOGICAL DUMP OF A/II IS USED FOR SAVING INSTANT THE SAME SENSE OPTION AS WOULD BE USED TO SELECT THE TAPE, WITH CS OSI IT IS POSSIBLE TO SAVE ONE OF CORE AND ALL OF UNCHANGED, PERIPHERAL STORAGE, INCLUDING THE MACHINE, SELF-LOADER, INTEGERS, ETC., REQUIRED MINIMUM 70A 711 CARD READER, 727 TAPE, AND TAPE PRINTERS OR AN ADDITIONAL 727 TAPE. SUPERSEDED BY CS-052, CS-157, 494.

ANALYSIS OF VARIANCE COMPUTES MEAN SQUARES OF SQUARES, DEGREES OF FREEDOM AND F FACTOR FOR UP TO 15 VARIABLE ANALYSIS ANY NUMBER OF VARIABLES ONE WAY AND ANY AMOUNT OF DATA MAY BE USED.

PODUMPPOUT POP-OUT ORGANIZATION. PURPOSE PRINT OUT SUBROUTINE WHICH SUBROUTINE IS A MODIFICATION OF SLOUT-2 CAPABLE OF PERIPHERAL AND/OR 0-200. THE PRINTING AND/OR PRINTING OF UP TO 120 CHARACTERS. OTHER DIFFERENCES WITH SLOUT-2 ARE:
1. ONLINE PRINTING IS NOT ECHOED.
2. TAPE WRITING IS NOT CHECKED BY RE-READING.
3. THE END-OF-TAPE TEST IS MADE.
4. THE 20-LETTER DISPLAY IN THE SUBROUTINE USES 437 INSTRUCTION CELLS 61 ERASABLE CELLS.

DOUBLE PRECISION ARC TANGENT INSTRUCTION COMPUTER DOUBLE PRECISION ARG OF A DOUBLE PRECISION ARGUMENT AS DESCRIBED IN IBM TNP. REQUIRE 395 AND 20 COMMON STORAGE. 85 4TH Requires 2) STORAGE LOCATIONS.

BINARY TO TG0 CONVERSION OF UNRESTRICTED INTEGERS. CONVERTS A BINARY INTEGER TO A FIXED OOR INTEGER OF 12 CHARACTERS, 20 WILL APPEAR AS LEFT MOST CHARACTER. RIGHT; AN INT. ANY PLUS OR MINUS BINARY INTEGER SUCH FOR BINARY DOES NOT EXCEED THE CAPACITY OF A 70A WOULD USE 33 STORAGE CELLS PLUS 4 COMMON.

INTEGRATION BY GAUSSIAN QUADRATURE INTEGRATION OVER INTERVAL [A,B] BY N + 49 1/3 1/3 1/3 1/3 1/3, OR 32 POIN QUADRATURE. WILL DREAM APART INTO 32 INTERVALS IF DESIRED. REQUIRE 917 STORAGE.

INTEGRATION BY HERMIT QUADRATURE INTEGRATES FROM MINS INFINITY TO PLUS INFINITY BY 56, 169, 1010, OR 20 POIN QUADRATURE. REQUIRE 192 STORAGE.


CARO TO TAPE, BINARY IS A SELF-LOADER TO WRITE A SINGLE BINARY FILE ON TAPE 1 FROM NON-RELOCATABLE BINARY CARDS WITH WS TRANSFER. IT CONVERTS A PROGRAM TO CARDS TO TAPE 1 ARE READ BY MEAN, NO LOCATION A TRANFER INTO WHICH WS 1282 WILL LOAD THE RECORD ARE SPECIFIED ON CONTROL CARDS AND MAY INCLUDE ALL EFFECT- VELY LEADING ADDRESSES IN THE DECK BETWEEN THE CONTROL CARDS AND WAS TRANSFER CARDS. CONTROL CARDS CAN WRITE TAPE LOADS ON WS 1282 BETWEEN PROGRAM RECORDS. ABSOLUTE BINARY CARDS AND TAPE CHECKS ARE AXED CHECKED, ALSO TEST IS USED.

SERVICES TO TAPE GENERATOR WRITES A SERVICE TAPE CONSISTING OF SERVICE ROUTINES, DEBUGGING ROUTINES, AND PRODUCTION PROGRAMS. THE ROUTINE PUNCHES OUT ONE CARD LOADS WHICH ARE TO CALL THE PRODUCTION PROGRAMS FROM THE SERVICE TAPE.

BINARY TAPE LOADER IS A SELF-LOADER THAT LOADS THE NEXT RECORD ON TAPE 1, IN THE WS 1282 FORMAT, INTO LOCATIONS A THRU C AS SPECIFIED BY WORDS 3 AND 4 OF THE RECORDS. TRANSfers CONTROL TO THE LOCATION IN THE RECORD OF WORD D HELD WILL NOT LOAD OVER ITSELF AND SO MAY BE REFERENCED TO LOAD SUBSEQUENT RECORDS WITHOUT RESTART FEATURES. IT CAN BE ASSUMED FOR CASES ENTERED BY BOTH CHECKSUM AND E1 TESTS.

TAPE TO TAPE COPY WITH CHANGES COPIES PROGRAM AND DATA TAPE WITH WS FORMAT AND PRODUCE A MEANS OF CORRECTING A SPECIFIED RECORDS.

THI:ERMODYNAMIC PROPERTIES OF STEAM AND WATER A SET OF SUBROUTINES TO BE USED IN VARIOUS COMBINATION WITH ONE ANOTHER TO PRODUCE VALUES FOR THERMODYNAMIC PROPERTIES OF STEAM AS TABULATED BY KEEFE AND KEEFE. RESULTS CAN BE COMPUTED FOR PRESSURE, TEMPERATURE, ENTHALPY, ENTROPY, VISCOSITY, SPECIFIC VOLUME AND QUALITY IN TERMS OF ONE OR TWO OF THE OTHER PARAMETERS IN THE WT, DW, SATURATED, OR LIQUID REGIONS WHEREVER APPLICABLE.

RANDOM NUMBER GENERATOR UNIFORM AND NORMAL RAMO NUMBER GENERATOR- PRODUCES UNIFORM DISTRIBUTED WITHIN 10 OF AND NORMAL DISTRIBUTED WITHIN 10 OF THE MEAN, IF ENTERED WITH A SIGN "+" TO THE MEAN.

NURU MATRIX MULTIPLY-FLOATING POINT,
MULTIPLIES AN M X N MATRIX OF AN N X M MATRIX TO GIVE AN M X M MATRIX. THE ELEMENTS OF EACH MATRIX ARE SEQUENTIALLY LOCATED BY ROWS. REQUIRE 65 WORDS PROGRAM PLUS 1 TEMPORARY.
THIS COMPUTER

0704 432MUMTR AVAILABLE PRIOR TO JANUARY 1962

SQUARE MATRIX TRANSPOSE ON ITSELF TO SUPPLY THE TRANSPOSE OF A MATRIX STORED ROW-WISE IN CORE STORAGE AND PLACE IN THE SAME LOCATIONS AS THE ORIGINAL MATRIX. PROGRAM REQUIRES 33 WORDS PLUS 4 TEMORARY, AN ADDED 33 BY 33 MATRIX TRANSPOSED IN LESS THAN 800 MICROSECSCONDS. CORR. 437A

0704 433MUR04 AVAILABLE PRIOR TO JANUARY 1962

MURA UPPER RELLOCATABLE BINARY LOADER ONE CARD LOADS STANDARD RELLOCATABLE BINARY CARDS WITHOUT ALTERNATION OF LOADING ADDRESSES. EXECUTES TRANSFER CARDS. OCCUPIES LAST 22 WORDS OF MEMORY. SELF LOADING.

0704 433MUM04 AVAILABLE PRIOR TO JANUARY 1962

MUR REFLECTED 704 CAUSES THE 704 TO BEHAVE LIKE A 407 IN ITS ROLE AS A READER AND PRENTREIP OF CARDS. 50 WORDS PROGRAM PLUS 21 WORDS FOR LOWER BINARY LOADER. READER AND PRENTREIP OPERATE AT FULL SPEED. SUPERSEDES MY TON 0437A 253.

0704 433MUSC01 AVAILABLE PRIOR TO JANUARY 1962

SCOPE GRID PLOTER TO DISPLAY ON THE 740 OUTPUT RECODER A GRID OF HORIZONTAL AND VERTICAL LINES. PROGRAM 15 MODE FOR PLOTTING CERTAIN SPECIFIED LINES HEAVIER THAN OTHERS. PROGRAM REQUIRES 53 WORDS STORAGE PLUS 2 TEMPORARY.

0704 433MUTG1 AVAILABLE PRIOR TO JANUARY 1962

ITERATION, ONE OR TWO VARIABLES GIVEN A-F, X, Y, Z, V, B, D, E TO FIND A VALUE FOR X AND Y WITHIN A GIVEN EPSILON OF RELATIVE ERROR. REQUIRES 269 WORDS PLUS 34 ERASABLE STORAGE. CORR. 4492

0704 433MUC00 AVAILABLE PRIOR TO JANUARY 1962

DETERMINANT EXPANSION THIS ROUTINE CALCULATES THE CHARACTERISTIC EQUATION OF M BY M SQUARE MATRIX. PROGRAM REQUIRES 419 WORDS OF STORAGE PLUS 9 MWORDS OF STORAGE FOR THE MATRIX. CORR. 1024

0704 435MAM04 AVAILABLE PRIOR TO JANUARY 1962

MATRIX MULTIPLICATION MULTIPLIES TWO MATRICES OF THE FORM A X B X C IN FLOATING POINT ARITHMETIC REQUIRES 177 WORDS OF STORAGE PLUS 64 WORDS OF COMMON STORAGE.

0704 435MUM04 AVAILABLE PRIOR TO JANUARY 1962

POLYNOMIAL EXPANSION COMPUTES THE POLYNOMIAL RESULTING FROM THE MULTIPLICATION OF LINEAR AND QUADRATIC FACTORS Requires 179 WORDS OF STORAGE PLUS 62 WORDS OF COMMON STORAGE.

0704 436MAAM02 AVAILABLE PRIOR TO JANUARY 1962

ATMOSPHERIC DATA SUBROUTINE GIVEN A GEOGRAPHIC AltITUDE IN THE RANGE 0 TO 299,000 FEET, COMPUTE THE FOLLOWING VALUES 1. TEMPERATURE IN DEGREES KELVIN, 2. DENSITY RATIO, 3. PRESSURE RATIO, 4. VELOCITY OF SOUND (FT./SEC.., ROUTINE REQUIRES 158 CELLS PLUS COMMON STORAGE AS NEEDED FOR 5-14 SUBROUTINE.

0704 439MA290 AVAILABLE PRIOR TO JANUARY 1962

GENERAL CATHODE RAY TUBE COUPLE SUBROUTINE. THIS SUBROUTINE WILL DRAW A SUBDIVIDED GRID, WRITE A TEXT TITILE AT THE TOP OF GRID WRITE A LABEL IN APPROPRIATE SCALE LABELS, AND PLACE POINTS OR SYMBOLS FOR POINTS ON THE 740 CRT OUTPUT.

0704 441 CSTO AVAILABLE PRIOR TO JANUARY 1962

TVAC (PSYCO) COMPUTER/ SIMULATOR THIS COMPUTER IS DESIGNED IN THE IBM DIGITAL COMPUTER PROGRAMMING BY D. J. MC CARRE.
FORTRAN 0704

FORTRAN LINEAR PROGRAMMING COU, HAS SIZE, 51 ROWS BY 91 COLUMNS INCLUDING ALL FUNCTIONAL BUT EXCLUDING ARTIFICIAL COLUMNS AND RIGHT HAND SIDE. DESIGN IS MODULAR WITHIN LIMITS OF FORTRAN ALGORITHM INCLUDES PHASE I MATRIOS TRANSFORMATION AND COMPOSITE ALGORITHM. SPEED QUITE GOOD BUT PRECISION LIMITATIONS AIM TO BIT IN INACCURACY OF SINGLE PRECISION FLOATING POINT. THE TOLERANCE IN STATEMENT 108 MAY BE CRITICAL, MAKING IT LARGE HAS EFFECT OF BYPASSING COMPOSITE ALGORITHM. Compile Time About 15 MINS

0704 4BOCE650S AVAILABLE PRIOR TO JANUARY 1962

SIMULATE BASIC AND COMPUTER WITH 70%. CODED FOR RF BUT SHOULD WORK ON RF IF ONLY 100 LOCATIONS USED FOR ASC PROG, USES CE 650 TO SIMULATE 650 INPUT PLUGBOARD. TAPE INPUT IS SHOWN WORK ON SIMULATE BASIC 650 COMPUTER SHOULD WORK ON SIMULATE BASIC 650 COMPUTER.

0704 483NA0296 AVAILABLE PRIOR TO JANUARY 1962

THE OBTAINS FUNCTIONAL VALUES FOR SIMULATIONS THE TIN FUNCTION AND ITS FIRST DERIVATIVES. REQUIRES DOUBLE PRECISION INTERNALLY IN CALCULATING THE DEPENDENT VARIABLES, THE USER MUST PROVIDE AN AUXILIARY SUBROUTINE WHICH EVALUATES THE FIRST ORDER DERIVATIVES. INITIALLY, THE USER MUST PROVIDE THE VALUES OF THE FIRST ORDER DERIVATIVES, REQUIRES 159 PLUS ON CELLS.

0704 491R5W65F AVAILABLE PRIOR TO JANUARY 1962

LIFE SQUARES ANALYSIS OF VARIANCE COMPUTES ALL SUMS OF SQUARES FOR A POLYNOMIAL PARTIALIZATION ARE FLOATING, FIXED, HOLLERITH, OR OCTAL. PROGRAM OCCUPIES 295 LOCATIONS, PLUS TEMPORARY STORAGE FOR COMMON.

0704 493LASB58 AVAILABLE PRIOR TO JANUARY 1962


0704 4956H6101 AVAILABLE PRIOR TO JANUARY 1962

DUMP STORAGE, CORE, DRUM, AND TAPES FORTRAN SUBROUTINE WHICH DUMP CORES, DRUMS AND TAPES, NOT REQUIRING THE USE OF A LOGICAL DRUM FOR SAVING. THE FIRST 640 WORDS OF CORE MEMORY IN QUATRINARY. A MODIFIED TAPESIVAL 1 TO 64 IS USED FOR SAVING INSTEAD, THE SAME SENSE SWITCH OPTION AS WOULD BE USED TO SELECT THE TAPE, WITH 64 OR 128 IS POSSIBLE TO DUMP ALL OF CORE AND DRUM MEMORY WITH ONE PASS ON THE MACHINE. SELF LOADING HUBRIT UPCARD READER 727 TAPE AND 716 PRINTER ON AN ADDITIONAL 727 TAPE.

0704 4956S502 AVAILABLE PRIOR TO JANUARY 1962

POLYNOMIAL APPROXIMATION FOR A GIVEN ARGUMENT USING THE RESULTS OF 04-.296

0704 492LAO297 AVAILABLE PRIOR TO JANUARY 1962

LAMINAR SQUARES ANALYSIS OF VARIANCE FOR A POLYNOMIAL PARTIALIZATION ARE FLOATING, FIXED, HOLLERITH, OR OCTAL. PROGRAM OCCUPIES 295 LOCATIONS, PLUS TEMPORARY STORAGE FOR COMMON.

0704 496MC055 AVAILABLE PRIOR TO JANUARY 1962

0704 497L5S03 AVAILABLE PRIOR TO JANUARY 1962

LOGARITHM OF THE GAMMA FUNCTION FOR COMPLEX ARGUMENTS. THIS SUBROUTINE COMPUTES THE REAL AND IMAGINARY PARTS OF THE GAMMA FUNCTION.

0704 496MC05S AVAILABLE PRIOR TO JANUARY 1962

POLYNOMIAL APPROXIMATION FOR A GIVEN ARGUMENT USING THE RESULTS OF 04-.296

0704 497L5S50 AVAILABLE PRIOR TO JANUARY 1962

LAMINAR SQUARES ANOVA COMPUTES ALL SUMS OF SQUARES FOR A POLYNOMIAL PARTIALIZATION ARE FLOATING, FIXED, HOLLERITH, OR OCTAL. PROGRAM OCCUPIES 295 LOCATIONS, PLUS TEMPORARY STORAGE FOR COMMON.

0704 497L5S53 AVAILABLE PRIOR TO JANUARY 1962

GENERAL PURPOSE OUTPUT PROGRAM WRITES ONE VARIABLE-FORMAT LINE ON TAPE, PRINTER, OR FNCHE, RESULTS ARE FLOATING, FIXED, HOLLERITH, OR OCTAL. REPEATING, INCREASING ADDRESSING AND CHECKING OF OUTPUT ARE OPTIONS IN CALLING SEQUENCE, ANY NUMBER OF OUTPUT MODES POSSIBLE FROM ONE CALLING SEQUENCE, TAPE OR PRINTER USE SAME INHIBIT CONTROL CODES. USES 460 CELLS PLUS 46 COMMON.
**IBM 0704 PROGRAM LIBRARY ABSTRACT**

0704 690PCDCP

ON LINE OCTAL DUMP
TO BE READ ON LINE AFTER A PROGRAM STOP, AND TO DUMP A
BLOCK OF CORE IN LOGICAL OCTAL WORDS. REQUIRES 95 CELLS.

0704 500B58F8

LEAST SQUARES ALLOCATION ERROR POLYNOMIAL FIT
FINDS THE POLYNOMIAL P OF GIVEN N ORDER DEGREE N THAT MINI-
MIZES THE MAXIMUM ALLOCATION ERROR AT A GIVEN SET OF K DATA
POINTS. P IS PRESENTED AS A SUM OF POWERS AND AS A SUM OF
CHORDS, IF THE MODEL IS PREDICTED. FLOATING
POINTER ARITHMETIC. REQUIRES 6 IN AND OUTPUT SUBROUTINES AND
THEM COMMON CELLS, 357 CELLS FOR THE CODE AND 1650/PSL/
CELLS FOR DATA.

0704 500D25G

BCD OUTPUT SUBROUTINE
PRINT A BCD RECORD OF ARBITRARY LENGTH ON THE ON-LINE
PRINT PROCEEDS CONVERTING MAIN PROGRAM MAY SWITCH TO
DOUBLE SPACE, PRINTING PRINTING WITHOUT ECHO OR SHORT
FORMAT USES 106 CELLS PLUS 31 COMMON.

0704 503A1631

ARGON TAP LOWER BINARY LOADER
SELF-LOADING LOAD TAPE KEY READS SHARE ABSOLUTE BINARY
PROGRAM RECORDS INTO CORE AND EXECUTES TRANSFER RECORDS.
LOADS REQUIRED LUG. E.G. 391253 CARD TO BINARY TAPE
LOADER, TO PREPARE TAPE, OCCUPIES CELLS 0-23.

0704 503A2112

ARGON DECK TO BINARY TAPE LOADER
DECK BY DECK DECK TAPE TAPE LOADER FOR COMPLETE SELF-
LOADING PROGRAMS BINARY PROGRAM CARDS AND TRANSFER CARDS
INTO CORE CONSTRUCTS IN SINE WRITE CORRESPONDING BINARY
TAPE RECORDS WITH BIC CMM. RESULT MAY BE LOADED BY LOAD TAPE
KEY IF TAPE BINARY LOADER PRECEEDS ON TAPE RECORDS. CAN ORDER
REQUIRED OCCUPIES CELLS 0-139.

**IBM 0704 PROGRAM LIBRARY ABSTRACT**

0704 500PCRI

CONTOUR PLOT PROGRAM
PLOTS CONTOUR LINES OF FUNCTION OF TWO VARIABLES ON CATYODR
RAY TUBE.

0704 504F46C2

CONTOUR PLOT PROGRAM
PLOTS DEFINED CONTOUR LINES OF FUNCTION OF TWO VARIABLES ON
CATYODR RAY TUBE, USED WHEE MICR.

0704 5080SP1L

GENERAL PROGRAM LOADER
COMBINES UP TO 60, x AND y BY BIL. L. LOBS ABSOLUTE BINARY,
RECALLABLE BINARY TRANSFER, REL CONTROLL AND FOUR-WORD
CARD CARD.-SELF-LOADS INTO 0-206 OCTAL.

0704 5080TPICCI

TAPE CORRECTOR
DUPLICATES A BCD TAPE AND MAKES INSERTIONS, DELETIONS, OR
CHANGES. CORRECTIONS MAY BE READ ON-LINE OR OFF-LINE.

0704 5100BEP

FIXED POINT EXPONENTIAL SUBROUTINE
TYPICAL, 0.44; 12 PM, 71 LOCATIONS. 10 DIGIT ACCURACY.
CORR./65

0704 5101MCAP

CAPACITIVELY NETWORK FLOW PROGRAM
THE PROGRAM DETERMINES A FLOW PATTERN OVER A GENERAL NETWORK
SO THAT A LINEAR COST FUNCTION OF THE BRANCH FLOWS IS
MINIMIZED. THE COSTS ARE ASSUMED TO BE INCREASED AS THE
BRANCHES, AND FLOW INTO AND OUT OF THE NODES IS

**IBM 0704 PROGRAM LIBRARY ABSTRACT**

0704 512MCVVT

BCD TO MODIFIED BCD CONVERSION ROUTINE
TO CONVERT A SET OF BCD WORDS TO MODIFIED BCD.

0704 513DPOPD1

DATA PROCESSING OUTPUT ROUTINE
TO SET UP AND PRINT ONE LINE OF OUTPUT ON AN ON-LINE PRINTER
IF SW. 2 IS ON OR OFF-LINE ON TAPE 2 IF SW. 2 IS OFF. THIS
ROUTINE CONVERTS BOTH FLOATING AND FIXED POINT BINARY NUMBERS
TO FIXED POINT OUTPUT AND PRINTS MODIFIED HOLLERITH
MODIFIED HOLLERITH INFORMATION.

0704 510MPK2

GENERAL PUNCHED OUTPUT ROUTINE
TO SET UP AND PRINT ONE LINE OF OUTPUT ON OFF-LINE PUNCH OR TO SET UP CARD IMAGE IN CORE. THIS ROUTINE
CONVERTS BOTH FLOATING AND FIXED POINT BINARY NUMBERS TO
FIXED POINT OUTPUT AND PRINTS MODIFIED HOLLERITH
HOLLERITH INFORMATION.

0704 5136152

INTERPRETER FOR BCD PROGRAMS
INTERPRETS SYSTEMS PROGRAMS WRITTEN ACCORDING TO IBM TECHNICAL
NEWSLETTER NO. 11. ACCEPTS GIVING PROGRAM DIAMOND WITH PUNCH MODIFICATION, PRINTS THE SAME OUTPUT CARDS AFTER TAPE-
CARDS. PRINTS UP TO A 30 TO 1 SPEED INCREASE OVER 65.
CORR./665.

0704 5138542

MAKE OCTAL
WHEN LOADED USING THE SAP 3-7 PB1 3 PSEUDO-OPERATION,
THE SAP IS CHANGED TO OCTAL-TERNARY, ALL NUMBERS
IN THE SYMBOLIC DEC AND THEREFORE STORED AS OCTAL.
EXCEPT THOSE IN THE VARIABLE FIELD OF DEC CARDS. THIS
PATH TO SAP IS ONLY USEFUL FOR ASSEMBLING
PROGRAM CORRECTIONS.

**IBM 0704 PROGRAM LIBRARY ABSTRACT**

0704 514440299

DETERMINATION EVALUATION AND ROOT EXTRATION
P. GALLO
THIS ROUTINE EVALUATES A DETERMINANT WITH
POLYNOMIAL ELEMENTS AND REPEATS THE ROOTS OF 155 RESULTING
POLYNOMIAL. THE ORDER OF THE DETERMINANT MAY VARY FROM 2 TO
60. THE ROUTINE WILL GENERATE UP TO 155 VALUES FROM 50 UNPREDICTED THAT PLUS TIMES 200 SQ EQUA TO 2 TO
LESS THAN 1200. THE ROUTINE IS MODIFIED FOR A 60TH DEGREE POLYNOMIAL IN ADDITIVE ROUTINE MAY BE USED
TO EVALUATE A DETERMINANT ONLY EXTRACT THE ROOTS OF A
POLYNOMIAL END.

0704 5164582

INCOMPLETE GAMMA FUNCTION
GIVEN A AND X, THIS ROUTINE WILL COMPUTE THE INCOMPLETE
FUNCTION AS P עד טO THE INTEGRAL FROM X TO INFINITY OF
EXP(-x)/x TO THE A^-1 POWER DU.

0704 521WAF1

FACTOR ANALYSIS
GEOMETRIC METHOD OF THOMPSON, ANALYSIS OF A CORRELATION MATRIX,
EXTRACTION OF SUCCESSIVE FACTORS AND
COMPUTATION OF CORRELATIONAL
MAX. MIN. OF MATRIX IS 64.
INPUT BY CARDS OR BY TEXI OUTPUT ON TAPE.

0704 522PE6

COMPLEX LINEAR SYSTEM SOLUTION PROGRAM
SIMPLE PRECISION SOLUTION OF COMPLEX LINEAR SYSTEMS AND
SOLUTION OF COMPLEX PARTS. HIGHEST ORDER OF MATRIX IS 64.
HIGHEST NUMBER OF MEMBER VECTORS IS 10. OFF-LINE OUTPUT.
JORDAN S METHOD

0704 5235C5P

MUSH DATA ASSEMBLER AND PRINT ROUTINE
PRONES INPUT AND OUTPUT FOR SC-MUSH USES A SLIGHTLY
MODIFIED RAN LD INPUT TAPE DEC/2. OUTPUT FORMAT
SIMILAR TO THAT OF RAN.
SOLVES PROBLEM WITH UP TO 104523 SIMULTANEOUS EQUATIONS AVAILABLE PRIOR TO JANUARY 1962.

IBM 0704 PROGRAM LIBRARY ABSTRACT

0704 525PMARK AVAILABLE PRIOR TO JANUARY 1962

FLOATING-POINT PROJECTED SIMULTANEOUS EQUATIONS OR A SINGLE ROOT OR A SIMULTANEOUS EQUATION, WHERE N IS AN INTEGER LESS THAN 150. METHOD IS ITERATED SINGLE PRECISION FLOATING-POINT OF EQUATIONS APPROXIMATELY .07 N MS. 51 CELLS. CONV. .120

0704 525PMSINP AVAILABLE PRIOR TO JANUARY 1962

DOUBLE-PRECISION PROJECTED SIMULTANEOUS EQUATIONS OR A SINGLE ROOT OR A SIMULTANEOUS EQUATION, WHERE N IS AN INTEGER LESS THAN 150. METHOD IS ITERATED DOUBLE PRECISION FLOATING-POINT OF EQUATIONS APPROXIMATELY .07 N MS. 51 CELLS. CONV. .120

0704 525PMSINTO AVAILABLE PRIOR TO JANUARY 1962

INTERPRETIVE DOUBLE-PRECISION FLOATING-POINT ARITHMETIC SUBROUTINE. READS AND EXECUTES CONSECUTIVE MACHINE LANGUAGE INSTRUCTIONS OR WHICH 20 ARE PERFORMED IN THEIR DOUBLE-PRECISION FLOATING POINT ANALOG. PACKAGE IS COMPOSED OF PK INTE, PK INTD, AND PKDUP. REQUIRES 499 STORAGE CELLS.

0704 525PMSINTO AVAILABLE PRIOR TO JANUARY 1962

INTERPRETIVE DOUBLE-PRECISION FLOATING-POINT ARITHMETIC SUBROUTINE. READS AND EXECUTES CONSECUTIVE MACHINE LANGUAGE INSTRUCTIONS OR WHICH 20 ARE PERFORMED IN THEIR DOUBLE-PRECISION FLOATING POINT ANALOG. PACKAGE IS COMPOSED OF PK INTE, PK INTD, AND PKDUP. REQUIRES 499 STORAGE CELLS.

0704 525PMSINTO AVAILABLE PRIOR TO JANUARY 1962

INTERPRETIVE DOUBLE-PRECISION FLOATING-POINT ARITHMETIC SUBROUTINE. READS AND EXECUTES CONSECUTIVE MACHINE LANGUAGE INSTRUCTIONS OR WHICH 20 ARE PERFORMED IN THEIR DOUBLE-PRECISION FLOATING POINT ANALOG. PACKAGE IS COMPOSED OF PK INTE, PK INTD, AND PKDUP. REQUIRES 499 STORAGE CELLS.
0704 5294SD0D AVAILABLE PRIOR TO JANUARY 1962
FLOATING-POINT DOUBLE-PRECISION SQUARE ROOT
COMPUTES THE SQUARE ROOT OF A DOUBLE-PRECISION FLOATING-
POINT NUMBER. NORMAL T4 SEQUENCE, ERROR RETURN FOR
NEGATIVE ARGUMENT WITH SQUARE ROOT OF THE ABSOLUTE VALUE IN
AC-PL. 52 BIT ACCURACY, REQUIRES 42 STORAGE CELLS PLUS 5 COMMON.
TIMING 2.736 MS.

0704 5294TVD0A AVAILABLE PRIOR TO JANUARY 1962
TIME SERIES DECOMPOSITION AND ADJUSTMENT
FORTRAN PROGRAM TO ADJUST SEASONAL AND IRREGULAR TIME SERIES
TO A FORM THAT SHOWS PRIMARILY THE TENDENCY-TOWARD MOVEMENT.
SEASONAL FACTORS, IREGULAR FLUCTUATIONS AND MANY SUMMARY
MEASURES USEFUL IN TIME SERIES ANALYSIS ARE COMPUTED IN THE
PROCESS. USES 16K DRUMLESS MACHINE.

0704 5295S6W0 AVAILABLE PRIOR TO JANUARY 1962
BCD OUTPUT PROGRAM
WRITES A BCD RECORD ON TAPE AND/OR PRINTS IT ON THE ON-LINE
PRINTER, AS DETERMINED BY SENSE SWITCHES. REQUIRES 79 CELLS
PLUS 20 COMMON.

0704 5295SB0T2 AVAILABLE PRIOR TO JANUARY 1962
DOUBLE PRECISION FLOATING POINT PRINT SUBROUTINE
COMPUTES AND PRINTS A BLOCK OF DOUBLE PRECISION FLOATING-
POINT NUMBERS TO THE OUTPUT UNIT. THE DATA POINTS OCCUPY 6
CONSECUTIVE CORE LOCATIONS. THE FORM OF OUTPUT IS
VARIABLE UNDER CONTROL OF A FORMAT. MODIFICATION FOR OTHER
CONVERSIONS IS POSSIBLE, USES 85 HOT OR UA SMPS. REQUIRES 333
STORAGE CELLS PLUS 50 COMMON.

0704 5305CSHHZ AVAILABLE PRIOR TO JANUARY 1962
HANKEL FUNCTION Routines
COMPUTES THE HANKEL FUNCTION HSUB/FX FOR ALL INTEGER ORDERS
FROM 0 TO N FOR POSITIVE X, X REQUIRE 132 RND AND ANY LS
AND EX ROUTINES WITH ERROR RETURN. ACCURACY IS QUESTIONABLE FOR
X GREATER THAN 1. SUPERLEDS ES 9100 DIST. 401.

0704 5306B0ADP AVAILABLE PRIOR TO JANUARY 1962
DOUBLE PRECISION ARCSIN/HACOS SUBROUTINE
TO COMPUTE A DOUBLE PRECISION FLOATING POINT ARC SIN OR ARC
COSINE, IN RADIANS, FROM A DOUBLE PRECISION FLOATING POINT
ARGUMENT. REQUIRES 232 STORAGE CELLS PLUS COMMON THROUGH
COMMON0.

0704 5306B0AKZ AVAILABLE PRIOR TO JANUARY 1962
GAUSS QUADRATURE 15 POINTS METHOD. THIS IS A MODIFICATION OF
SAR SUBROUTINE 61 GALS. THE SUBROUTINE DIVIDES THE INTERVAL
[1,6.] INTO 15 EQUAL INTERVALS AND BY THE PROPER TRANSFORMATION
EACH INTERVAL IS INTEGRATED OVER THE INTERVAL 0,1.] COMMON0.

0704 5404CS0C AVAILABLE PRIOR TO JANUARY 1962
ONE CARD TAPE COPR SUBROUTINE
CONTROL MODE TO EDU IF 153 UP AND BINARY IF DOWN. MODE CAN
BE CHANGED DURING RUNG

0704 5484SD05 AVAILABLE PRIOR TO JANUARY 1962
TRIPLE PRECISION COMPLEX ARITHMETIC PACKAGE
TRIPLE PRECISION COMPLEX ARITHMETIC PACKAGE PERFORMS BASIC
ARITHMETIC OPERATIONS ON TRIPLE PRECISION FLOATING POINT
COMPLEX NUMBERS, REAL AND IMAGINARY PARTS OF THE COMPLEX
NUMBERS ARE REPRESENTED AS A SIGNED 1600 FRACTION AND A
SIGNED 1600 EXPONENT, USES 122 CELLS PLUS 30 CELLS OF
COMMON.

0704 5484SD05 AVAILABLE PRIOR TO JANUARY 1962
TRIPLE PRECISION COMPLEX ARITHMETIC PACKAGE
TRIPLE PRECISION COMPLEX ARITHMETIC PACKAGE PERFORMS BASIC
ARITHMETIC OPERATIONS ON TRIPLE PRECISION FLOATING POINT
COMPLEX NUMBERS, REAL AND IMAGINARY PARTS OF THE COMPLEX
NUMBERS ARE REPRESENTED AS A SIGNED 1600 FRACTION AND A
SIGNED 1600 EXPONENT, USES 122 CELLS PLUS 30 CELLS OF
COMMON.

0704 5556PLOT AVAILABLE PRIOR TO JANUARY 1962
POLAR POINT PLOT SUBROUTINE
TO REPRESENT NUMERICAL DATA BY GRAPHICAL METHODS. A 120 BCD
CHARACTER HOLLERITH FORMAT IS SET UP FOR EACH LINE TO BE
PLOTTED. IT CAN HANDLE UP TO 500 CURVES SIMULTANEOUSLY.

0704 5565C0AD2 AVAILABLE PRIOR TO JANUARY 1962
ZEROES EXTENDED RANGE POLYNOMIAL/JERRY.
THIS SUBROUTINE DETERMINES THE ROOTS, REAL OR COMPLEX, OF A
POLYNOMIAL OF DEGREE N WITH REAL COEFFICIENTS, USING EXTENDED
RANGE ARITHMETIC. USES 150 EXTENDED RANGE PDS, AND 100 EXTEN-
DED RANGE COMPLEX PKG. TIMING APPROX. 5 SECS/ROOT. STORAGE
660 CELLS X COMMON THRU COMMON 6 32.

0704 5565C0AD3 AVAILABLE PRIOR TO JANUARY 1962
ZEROES AXIARY FUNCTION/JERRY.
THIS SUBROUTINE DETERMINES A REAL OR COMPLEX ROOT OF AN AX-
INARY FUNCTION USING TRIPLE PRECISION ARITHMETIC. USES 945 PDS
AND 945 COMMON THRU COMMON 5 32.

0704 5565C0AD8 AVAILABLE PRIOR TO JANUARY 1962
TRIPLE PRECISION COMPLEX SQUARE ROOT
THIS SUBROUTINE TAKES THE SQUARE ROOT OF A TRIPLE PRECISION
COMPLEX NUMBER. REQUIRES 963 AND 963. TIMING 150 MS/ROOT.
STORAGE 78 CELLS X COMMON THRU COMMON 5 32.

0704 5568LC0C2 AVAILABLE PRIOR TO JANUARY 1962
A MODIFIED NEUTON-RAPHSON POLYNOMIAL ROOT-FINDER--
WITH NUMERICAL CONVERGENCE--
THIS SUBROUTINE CALCULATES THE COMPLEX ROOTS OF POLYNOMIALS
HAVING REAL COEFFICIENTS, INCLUDING ANY MULTIPLE ROOTS, WITH
SINGLE PRECISION ACURACY. ELC0C2 SHOULD BE REPLACED BY THIS
ENHANCED SUBROUTINE.

139 0704 PROGRAM LIBRARY ABSTRACT
*********************************************************************
IBM 0704 PROGRAM LIBRARY ABSTRACT

0704 500ORQRTS AVAILABLE PRIOR TO JANUARY 1962
SORT, ALGEBRAC, KEY AND ITEM LENGTH - 1 WORD, OPEN, NO. ITEMS MUST BE A POWER OF 2. VARYS STE-2, NO. ITEMS, REASONABLY FAST OPEN SUBROUTINE REQUIRING 49 CELLS.

0704 500ORQRTZ AVAILABLE PRIOR TO JANUARY 1962
SORT, ALGEBRAC, KEY AND ITEM LENGTH - 1 WORD, OPEN, LENGTH OF STRING TO BE SORTED MUST BE A POWER OF 2, REQUIRES STORAGE TWICE LENGTH OF STRING, REASONABLY FAST, 60 CELLS.

0704 500ORQRT0 AVAILABLE PRIOR TO JANUARY 1962
SORT, ALGEBRAC, MULTIPAD KEYS, WHOLE WORD KEYS ONLY, NO. ITEMS A POWER OF 2, 2 WORD KEYS WHICH GIVE LOC OR KEYS ARE ORDERED TO MATCH SORTED KEYS, ONLY CHIEVES MOVES, WORDS OF KEY MUST BE ALIGNED, WORDS STE-2, NO. CELLO 90 CELLS.

0704 5702FCQDC AVAILABLE PRIOR TO JANUARY 1962
ABSOLUTE AND CORRECTION CARD LOADER, ONE CARD LOADER OF ABSOLUTE BINARY AND CORRECTION CARDS.

0704 570FCF013 AVAILABLE PRIOR TO JANUARY 1962
GENERALIZED, PACKAGED-O/W LINE INPUT-OUTPUT SUBROUTINE LOADS DECIMAL DATA FROM VARIABLE FIELD CARDS DIRECTLY INTO CORE STORAGE WITH AUTOMATIC CONVERSION, CONVERSION MAY BE FIXED-TO-FIXED, FIXED-TO-FLOATING, FLOATING-TO-FIXED, OR FLOATING TO FLOATING, PAGE IDENTIFICATION HANDLED AUTOMATICALLY AND COLUMN HEADINGS ARE OPTIONALLY AUTOMATICAL, REQUIRES 1150 CELLS 6% COMMON.

0704 570FCF095 AVAILABLE PRIOR TO JANUARY 1962
SYMPLETIC MATRIX INVERSION, INVERSION OF NON-SINGULAR SYMMETRIC MATRICES OF ORDER EQUAL TO OR LESS THAN 22, SELECTS MATRIX FROM DECIMAL CARDS AND INVERSE IT IN CORE 5 X CORE MEASURED IS REQUIRED.

IBM 0704 PROGRAM LIBRARY ABSTRACT

0704 5745CTUKS AVAILABLE PRIOR TO JANUARY 1962

0704 57505F123 AVAILABLE PRIOR TO JANUARY 1962
END OF FILE FUNCTION TO ACCOMPLISH A TRANSFER TO ANY DESIRED STATEMENT WITHIN A FORTRAN PROGRAM WHENEVER AN END OF FILE IS ENCOUNTERED WHILE READING A BINARY TAPE, REQUIRES 192 CELLS, NO COMMON.

0704 575055Q50 AVAILABLE PRIOR TO JANUARY 1962
EXTENDED TRANSFER FUNCTION TO ACCOMPLISH A TRANSFER FROM A FORTRAN PROGRAM TO A SHARE, OR OTHER, PROGRAM EVEN WHEN THE FORTRAN OBJECT PROGRAM USES AN INDEX REGISTER TO COMPUTE THE EFFECTIVE ADDRESS OF THE TRANSFER, ROUTINE REQUIRES 25 CELLS, NO COMMON.

0704 57505ORQZ AVAILABLE PRIOR TO JANUARY 1962
TRANSFER FUNCTION TO ACCOMPLISH A TRANSFER FROM A FORTRAN PROGRAM TO A SHARE, OR OTHER, PROGRAM AND RETURN IF DESIRED, ROUTINE REQUIRES 15 LOCATIONS, NO COMMON.

0704 5750NWF02 AVAILABLE PRIOR TO JANUARY 1962
AUTOCORRELATION FUNCTION GENERATING FLOATING, TO COMPUTE ONE POINT OF EITHER THE AUTOCORRELATION FUNCTION GIVEN A SET OF TIME SERIES DATA FOR EQUALLY-SPACED POINTS, 29 LOC. 6% COMMON.

IBM 0704 PROGRAM LIBRARY ABSTRACT

0704 7570WDP02 AVAILABLE PRIOR TO JANUARY 1962
DOUBLE PRECISION INPUT, READS 16 DIGIT DECIMAL FLOATING POINT NUMBERS WITH CORRESPONDING DECIMAL SCALES AND CONVERT TO DOUBLE PRECISION FLOATING POINT NUMBERS, INPUT CARD IS COMPRISED OF 4 FIELDS, 16 COLUMNS TO A FIELD, OF WHICH THE FIRST 16 COLUMNS CONTAIN THE FRATIONAL PART AND THE LAST 2 COLUMNS SPECIFY THE CORRESPONDING DECIMAL SCALES, SIGNS ARE DISEMPLOYED OVER A 1 DIGIT INTEGER, THE FRACTIONAL PART WILL BE NORMALIZED AND ROUNDED, CORR.578

0704 7570WDP03 AVAILABLE PRIOR TO JANUARY 1962
DOUBLE PRECISION OUTPUTS, OUTPUTS 6 TO 16 DIGIT DOUBLE PRECISION FLOATING POINT NUMBERS WITH DECIMAL SCALES AND IF DESIRED, RED WORDS, NUMBERS AND CHARACTERS ARE POSITIONED IN A LINE OF OUTPUT AS SPECIFIED IN THE CALLING SEQUENCE, UNDER POINT WHEN CONTROL METHOD POINTS ARE KEM TO BE IMMEDIATELY TO THE LEFT OF THE LEFT MOST DIGIT, BUT NOT PRINTED, THE EXP. OF THE RADOX IS PRINTED TO THE RIGHT AND APPEARS AS A 2 DIGIT INTEGER, THE FRACTIONAL PART WILL BE NORMALIZED AND ROUNDED, CORR.578

0704 7570WDP04 AVAILABLE PRIOR TO JANUARY 1962
POWER SPECTRAL DENSITY FUNCTION, FLOATING TO COMPUTE THE POWER SPECTRAL DENSITY FUNCTION, GIVES ESTIMATES OF THE AUTOCORRELATION FUNCTION FOR EQUALLY SPACED POINTS, 150 LOC. 17% COMMON.

0704 7570WDP05 AVAILABLE PRIOR TO JANUARY 1962
SINE AND COSINE FLOATING COMPUTES SINE AND COSINE OF THE PHASE ANGLE THE SINE, WHERE THE SINE AND COSINE ARE GIVEN IN RADIANS IN FLOATING POINT, TYPICAL LIMITS 0.1 SIN TO 157.05 DEG, TO LOC. 34 ERASABLE, INCLUDES SSDC/SINE-COSINE/ SUBROUTINE.

0704 7570WDP06 AVAILABLE PRIOR TO JANUARY 1962
NORMALLY DISTRIBUTED PSEUDO-RANDOM NUMBERS, EACH ENTRANCE PRODUCES THE NEXT NUMBER IN FIXED POINT IN A SET OF NUMBERS REALIZED AS DEVIATION DISTRIBUTED NUMBERS WITH ZERO MEAN AND UNIT STANDARD DEVIATION, REQUIRES 25 CELLS AND 5+7 MILLISECONDS.
TRAP TRAC, GI TRAP, Computes floating point quotient of 2 floating point numbers with proper quadrant allocation in range -Pi to Pi. Requires 17 storage cells. Replaces 100 in the accuracy of the numbers computed with CA 011. Uses 34 storage cells & 10 Common.

FORTRAN SNAP SHOT ROUTINE. TO TAKE SNAP SHOTS AT THE SPECIFIED PLACES IN A FORTRAN PROGRAM.

ANEL A/B, FORTRAN II VERSION/SNAP CODED. FUNCTION SUBROUTINE FOR FORTRAN II LIBRARY. COMPUTES PL. POINT VALUES IN RANGE -Pi to Pi. USES IBATN1, Requires 117 storage cells in E-MH-NC.

SIMPSON TRAP, SIMPSON XX TRAP trace, TO SIMPLIFY Trace Instruction. USES IBATN1. Computes X values in range 0 to Pi. Requires 23 instructions plus 2 working storage. Timing is 21.25 RS per transfer.

SIMPSON SNAP, AVAILABLE PRIOR TO JANUARY 1962

PORTRAN SNAP SHOT ROUTINE. TO TAKE SNAP SHOTS AT THE SPECIFIED PLACES IN A FORTRAN PROGRAM.

.documentation/
IBM 0104 PROGRAM LIBRARY ABSTRACT

0704 62902450 AVAILABLE PRIOR TO JANUARY 1962

DETERMINANT EVALUATOR FOR NEARLY TRIANGULAR MATRICES
THIS FORTRAN SUBPROGRAM EVALUATES THE DETERMINANT OF A MATRIX A
NEARLY TIMES 1, WHERE A IS A NEARLY TRIANGULAR MATRIX OF DIMENSION N TIMES N AND ALPHA IS A SCALAR.
IT HAS A DIMENSION STATEMENT OF A /50/50 AND B /50/50 WHICH CAN BE CHANGED ACCORDING TO NEEDS OF THE PROGRAM.
INPUT MATRIX A IS NOT DESTROYED BY THE PROGRAM.
216 CELLS EXCLUDING ARRAYS A AND B ARE REQUIRED.

0704 6330606 AVAILABLE PRIOR TO JANUARY 1962

REAL EIGENVALUES OF REAL MATRICES
THIS FORTRAN SUBPROGRAM DETERMINES THE N REAL EIGENVALUES OF A REAL SIMILARITY TRANSFORMATION MATRIX M.
It has a dimension statement of A /50/50 and B /50/50 which can be changed according to needs of the programmer.
The program requires 2150 cells excluding arrays A, B, C, and D and the square root routines.

0704 6350785 AVAILABLE PRIOR TO JANUARY 1962

GENERAL LEAST-SQUARES FORTRAN SUBPROGRAM
GIVES THE LEAST SQUARE SOLUTION TO A SYSTEM OF OVER-DETERMINED LINEAR EQUATIONS A X = C, WHERE A IS AN N TIMES M MATRIX WITH N GREATER THAN M, OR TO A SYSTEM OF M TIMES N EQUATIONS B X = D, WHERE B IS AN M TIMES M SQUARE MATRIX, AND X AND C ARE N-TIMES-1 VECTORS.
IT HAS A DIMENSION STATEMENT OF A /50/50, B /50/50, C /50/50, AND D /50/50 WHICH CAN BE CHANGED ACCORDING TO NEEDS OF THE PROGRAMMER.
INPUT DATA IS DESTROYED DURING COMPUTATION.
REQUIRES 545 CELLS EXCLUDING ARRAYS A, B, C, D, X, AND 11 AND THE SQUARE ROOT ROUTINES.

0704 6510818 AVAILABLE PRIOR TO JANUARY 1962

GENERAL ROOT FINDER FORTRAN SUBROUTINE
THIS FORTRAN SUBPROGRAM FINDS THE REAL ZEROS OF ANY ANALYTIC FUNCTION FA.
IT HAS A DIMENSION STATEMENT OF A /50/50 WHICH CAN BE CHANGED TO SUIT NEEDS OF THE PROGRAMMER.

0704 6570445 AVAILABLE PRIOR TO JANUARY 1962

LINEAR EQUATION SOLVER
THIS FORTRAN SUBPROGRAM FINDS THE SOLUTION X OF A LINEAR EQUATION AX = C WHERE A IS AN N TIMES N SQUARE MATRIX, AND X AND C ARE N-TIMES-1 VECTORS.
IT HAS A DIMENSION STATEMENT OF A /50/50 AND C /50/50 WHICH CAN BE CHANGED ACCORDING TO NEEDS OF THE PROGRAMMER.
INPUT DATA IS DESTROYED DURING COMPUTATION.
REQUIRES 418 CELLS EXCLUDING ARRAYS A AND C ARE REQUIRED.

0704 6570418 AVAILABLE PRIOR TO JANUARY 1962

NEARLY TRIANGULARIZATION OF A MATRIX SUBROUTINE
THIS FORTRAN SUBPROGRAM TRANSFORMS A REAL MATRIX INTO A NEARLY TRIANGULAR FORM BY SIMILARITY TRANSFORMATIONS.
IT HAS A DIMENSION STATEMENT OF A /50/50 AND B /50/50 WHICH CAN BE CHANGED ACCORDING TO NEEDS OF THE PROGRAMMER.
INPUT DATA IS DESTROYED DURING COMPUTATION.
339 CELLS EXCLUDING ARRAYS A AND B ARE REQUIRED.

0704 6570410 AVAILABLE PRIOR TO JANUARY 1962

EIGENVECTOR DETERMINANT SUBROUTINE
GIVEN A REAL EIGENVECTOR ALPHA OF A MATRIX A OF ORDER N TIMES N THIS FORTRAN SUBPROGRAM DETERMINES THE CORRESPONDING REAL EIGENVECTOR v. IT HAS A DIMENSION STATEMENT A /50/50 AND V /50/50 WHICH CAN BE CHANGED ACCORDING TO NEEDS OF THE PROGRAMMER.
INPUT DATA IS DESTROYED DURING COMPUTATION.
345 CELLS EXCLUDING ARRAYS A AND V ARE REQUIRED.

0704 6660387 AVAILABLE PRIOR TO JANUARY 1962

DESSL FUNCTIONS OF ORDER ZERO
COMPUTES J ZERO AND Z ZERO OF X FROM ASYMPTOTIC FORMULAS.
REQUIRES 45 CELLS PLUS 10 COMMON.
514, SQUARE ROOT AND LOG ROUTINES INCLUDED.

0704 6660388 AVAILABLE PRIOR TO JANUARY 1962

DESSL FUNCTIONS OF ORDER ONE
COMPUTES J ONE AND Z ONE OF X FROM ASYMPTOTIC FORMULAS.
REQUIRES 235 CELLS PLUS 10 COMMON.
514, SQUARE ROOT AND LOG ROUTINES INCLUDED.

0704 6660389 AVAILABLE PRIOR TO JANUARY 1962

EIGENVALUES SOLVABILITY REAL TO FIND THE HIGHEST EIGENVALUE AND CORRESPONDING EIGENVECTOR OF THE MATRIX EQUATION AX = 0.
SPACE N X N LOCATIONS.

0704 6660390 AVAILABLE PRIOR TO JANUARY 1962

READ WRITE GRAM
ROUTINE UTILIZES MULTIPLE RECORD FEATURES FOR OPTIMIZING STORAGE USE.
THE CONTENTS OF UNUSUALLY DISTRIBUTED INPUT LOCATIONS INTO THE CONTENTS OF EQUALLY DISTRIBUTED CORE REGIONS IS MEASURABLE.
A CORE DRUM IS USED AND 2 IN CORE MUST BE EQUALLY SPACED, BUT THE SPECIFIC SPACING OF THE AFFECTED LOCATIONS ON THE DRUM NEED NOT BE THE SAME AS FOR THE CORE.

0704 6660391 AVAILABLE PRIOR TO JANUARY 1962

SELECT OF COMBINATIONS OF INPUT DATA.
ALL DATA ITEMS TO BE USED MUST BE STORED IN CORES, AND FROM THESE DATA ITEMS IN AN ORDERED FASHION TO CORE DATA, THE CRAMER, ASSIGN A COMBINATION TO EACH CORE.
THE USER MAY SPECIFY THE NUMBER OF CORES TO BE USED FOR EACH COMBINATION.
NO CORE DATA IS transferred TO NORMAL RETURN WHERE CORE PROCESSING PROGRAM SHOULD RESUME.
AT THE END OF THE LAST PROGRAM THE USER TRANSFERS BACK TO SUBRT, WHICH SELECTS NEXT COMBINATION, ETC., WHEN ALL COMBOS. PROCESSED SINCE WILL TRA TO FIRST RAN.

201
0704 674WOP09
AVAILABLE PRIOR TO JANUARY 1962
ABSOLUTE AND CORRECTION TRANSFER CARD LOADER.
LOADS SHARE STANDARD ABSOLUTE BINARY AND CAT CARDS.
ALL CARDS MAY BE CHECKED VERIFYING REQUIRED LOCATIONS
AND INDEX REGISTER 4. MACHINE MUST NOT BE IN TRAPPING MODE.

0704 674WOKPA0
AVAILABLE PRIOR TO JANUARY 1962
ELLIPITIC PARTIAL DIFFERENTIAL EQUATIONS.
THIS PROGRAM FINDS THE APPROXIMATE SOLUTION OF A SET
OF EQUATIONS WITH PRESCRIBED BOUNDARY CONDITIONS
BY THE METHODS OF INFINITE DIFFERENCES AND SUCCESSIVE OVER-
RELAXATION. THE REGION MAY BE ARBITRARY IN SHAPE AND MAY
INCLUDE INTERFACES AND HOLES. THE BOUNDARY CONDITIONS MAY
BE FIXED. THE MAIN PROGRAM REQUIRE THE USER MUST SUPPLY.
EXCLUSIVE OF THE THREE SUBROUTINES THE USER MUST SUPPLY, CORR. 699

0704 674WOKPA1
AVAILABLE PRIOR TO JANUARY 1962
Tape and Rock Simulation of the 714 Card to Tape.
Requires Non-Standard 714 Ctl. Panel. Extra Cards in Code if
Reading 80 Ef. No Checking Home. Uses EF 14, NY 01.

0704 674WOKPA2
AVAILABLE PRIOR TO JANUARY 1962
THERMAL ANALYZER
THIS IS A SIMULATION TO SHARE SUBROUTINE 3 WHICH SOLVE
5 THE GENERAL PROBLEM OF STEADY STATE AND TRANSIENT HEAT TRAN-
SFER. MULTIPLE CASES CAN BE HANDLED WITH OTHER PARTIAL SCAM
ITER REPLACEMENT OR DOING A COMPLETE NEW PROBLEM.

0704 674WOKPA3
AVAILABLE PRIOR TO JANUARY 1962
NON-LINEAR ESTIMATION /PRECISION/IBM-
Given a functional relation and data for m observed values of
A A single dependent variable, m corresponding values for k
INDEPENDENT VARIABLES, AND INITIAL VALUES FOR k PARAMETERS,
A PROCEDURE PROVIDES ESTIMATES FOR THE PARAMETERS A/S PROVIDES
STATISTICAL INFORMATION TO ASSESS THE NORTH OF THE ESTIMATED
PARAMETERS. USE OF THE PROGRAM FOR MORE THAN ONE DEPENDENT
A VARIABLE IS POSSIBLE. THE FUNCTIONAL RELATION IS NOT
NON-LINEAR OR LINEAR IN THE PARAM. & INDEP. VAR. CORR. 046

IP 0704 PROGRAM LIBRARY ABSTRACT

0704 674WOKPA4
AVAILABLE PRIOR TO JANUARY 1962
TAPE MACHINING ROUTINE.
This is a tape copy routine with a number of subroutines which
A permit record manipulation and modification in any of several
WAYS. THESE INCLUDE INDIVIDUAL WORD CHANGES AND CHECKSUM
ACCOUNTING AS WELL AS RECORD REWRITE FROM CARDS WHILE
COPYING, THIS CHECKING METHOD MAKES IT A LITTLE SLIGHTER
THAN SMITH OR BLOOR IN SOME RESPECTS, BUT WHERE HERRING
OF SEVERAL PAPERS IS DESIRED, IT IS PASTER.

0704 674WOKPA5
AVAILABLE PRIOR TO JANUARY 1962
BINARY OCTAL CARD OR TAPE LOADER.
Five card high order self loading program to load
ASSISTED SHARE STANDARD AND CASE BINARY, OCTAL & OCTAL
TRANSFER CARDS. OPTION AVAILABLE FOR WRITING A SELF
LOADING RECORD FROM CORE BEFORE EXECUTING TRANSFER CARD.

0704 674WOKPA6
AVAILABLE PRIOR TO JANUARY 1962
N ROOT ROUTINE.
COMPUTES THE nth ROOT OF A NORMALIZED FLOATING POINT
NUMBER, MUMIP IN THE ACUMULATOR AND N IN INDEX
RECORD 1 UPON ENTRY. RESULT IN ACCUMULATOR UPON RETURN.
ERROR RETURN IF COMPLEX ROOT.

0704 674WOKPA7
AVAILABLE PRIOR TO JANUARY 1962
TAPE INPUT/OUTPUT
TO RECORD WHERE A VARIABLE LENGTH OCTAL OR DEC RECORD
WITH OR WITHOUT CHECKING AND CHECK FOR AN END OF FILE OR
END OF TAPE CONDITION.

0704 674WOKPA8
AVAILABLE PRIOR TO JANUARY 1962
LAGRANGEAN INTERPOLATION ROUTINE.
GIVEN A TABLE OF N PAIRS OF X AND F/X AND A GIVEN VALUE
OF X, THE ROUTINE WILL USE P-1 THE FORM OF FLOATING POINT
TO COMPUTE F/X. LAGRANGE COEFFICIENT FUNCTIONS ARE USED.
REQUIRES 7 STORAGE LOCATIONS FOR PROGRAM AND NGS AT COMMON.

1H 0704 PROGRAM LIBRARY ABSTRACT

0704 674WOKPA9
AVAILABLE PRIOR TO JANUARY 1962
FLOATING POINT UNIVERSE SEARCH.
GIVEN A BLACK BOX ROUTINE COMPUTING F/X FROM A GIVEN
X, THE SEARCH ROUTINE VARIES C TO OBTAIN A DESIRED VALUE
OF F/X. REQUIREs 200 SBES STORAGE LOCATIONS INCLUDIING UP GHAT,
REQUIRES 6 LOCATIONS AT COMMON.

0704 674WOKPB0
AVAILABLE PRIOR TO JANUARY 1962
FLOATING POINT ADVANCED SEARCH.
GIVEN A BLACK BOX ROUTINE WITH TWO INPUT AND TWO OUTPUT
PARAMETERS, THIS ROUTINE ADJUSTS THE INPUT PARAMETERS TO THE
DESIRED VALUES OF THE OUTPUT PARAMETERS. THIS IS DONE BY
APPROXIMATION TO THE FIRST PARTIAL DERIVATIVES.
REQUIRES 200 LOCATIONS AND 6 SPACES AT COMMON.

0704 674WOKPB1
AVAILABLE PRIOR TO JANUARY 1962
ZEROS OF COMPLEX POLYNOMIALS.
COMPUTES THE ZERO OF A POLYNOMIAL WITH COMPLEX
COEFFICIENTS USING A SINGLE PRECISION DIAGONALIZATION.
METHOD. STORAGE LOCATIONS 487 & 38 ERASABLE & ZM615/

0704 674WOKPB2
AVAILABLE PRIOR TO JANUARY 1962
DOUBLE PRECISION MATRIX MULTIPLICATION.
MULTIPLES TWO REAL MATRICES WHOSE ELEMENTS ARE STORED
CONSECUTIVELY IN CORE STORAGE USING DOUBLE PRECISION.
COEFFICIENTS. THE PRODUCT IS STORED IN THE SAME MANNER.
IN CORE STORAGE. MEASURES ANY STORAGE PLUS
IN CORE. DP8 AND CL DPV MUST BE ASSEMBLED
CONCURRENTLY.

1H 0704 PROGRAM LIBRARY ABSTRACT

0704 674WOKPB3
AVAILABLE PRIOR TO JANUARY 1962
DISEL FUNCTION Y SUB N /F/A.
GIVEN 10 AND 1, THIS SUBROUTINE FINDS Y SUB N /F/A OR
ALL VALUES Y SUB D /F/A TO Y SUB N /F/A.

0704 674WOKPB4
AVAILABLE PRIOR TO JANUARY 1962
TAP-VECTOR-
This subroutine may be used on a secondary or tertiary tape.
These subroutines are intended to be used in conjunction with
TAP-VECTOR.

0704 674WOKPB5
AVAILABLE PRIOR TO JANUARY 1962
TAP-VECTOR-
This subroutine may be used on a secondary or tertiary tape.
These subroutines are intended to be used in conjunction with
TAP-VECTOR.

0704 674WOKPB6
AVAILABLE PRIOR TO JANUARY 1962
TAP-VECTOR-
This subroutine may be used on a secondary or tertiary tape.
These subroutines are intended to be used in conjunction with
TAP-VECTOR.
0704 PROGRM LIBRARY ABSTRACT

0704 TSW9PSM2
AVAILABLE PRIOR TO JANUARY 1962

3D SELECTIVE MONITOR TRACE SYSTEM.
TO BE USED AT EXECUTION TIME BY MEANS OF CONTROL CARDS TO
PROVIDE A DETAIL PRINTOUT OF LOG ON, OFF MONITOR, ELECTRONIC
COUNTERS, OF THE FREQUENCY OF EVERY INSTRUCTION, OR A TRAP
TRAYING EACH EXECUTABLE TRANSFER OR FILE TRANSFER ON ALL
SELECTED TYPES. THIS SYSTEM MAY BE USED IN CONJUNCTION WITH
EVER ANY SELECTED PORTIONS OF PROGRAM BEING CHECKED. TRACES
PROGRAMS WHICH OPERATE IN TRAP MODE, AS WELL AS 150 OPERATIONS
BY SIMULATION. PLACED AC AND PC ON-O LINE PRINT ONLY.
VOS STORAGE CELLS ADAPTABLE.

0704 TSW9CZ1
AVAILABLE PRIOR TO JANUARY 1962

FLOATING POINT COMPLEX ARITHMETIC ABSTRACTIONS
TO FACILITATE EXECUTION OF A PROGRAM BY USING EITHER REAL
OR COMPLEX ARITHMETIC WITHOUT MODIFICATION OF THE PROGRAM.
CONTAINS MESH GENERATOR, ITERATION, OUTPUT PRINTER,
INTERPOLATOR AND OTHER AUXILIARY PROGRAMS.

0704 TSW9C6C0
AVAILABLE PRIOR TO JANUARY 1962

10 TRANSPORTATION CODE.
10 TRANSPORTATION CODE USING JAMES MUNFORD ALGORITHM JOURNAL
MARCH 1957Y. REQUIRES 1 CORE, 4 DRUMS AND AT LEAST
1 TAPE UNIT.

0704 TSW9ASG0
AVAILABLE PRIOR TO JANUARY 1962

DOUBLE PREC. FLOATING PT. SQUARE-ROOT SUBROUTINE.
RELATIVE ERROR LESS THAN 2.3616X10^-6. 2.02 M1. 54 LOCATIONS 5 4
COMMON.

0704 TSW9PC01
AVAILABLE PRIOR TO JANUARY 1962

READING OF FORMATTER STATEMENTS AT EXECUTION TIME.
FORTRAN-2 SUBROUTINE TYPE PROGRAM.

0704 TSW9F03
AVAILABLE PRIOR TO JANUARY 1962

TAPE COPY PROGRAM. BINARY OR DCO MODE MAY BE IMPOSED AS WELL AS INTEGRAL COPY
OR NUMBER OF FILES OR NUMBER OF RECORDS TO BE COPIED CAN BE
PRESET. CHECKSUM AND OPTIONAL RTI VERIFICATION IS EFFECTED.

0704 TSW9PH03
AVAILABLE PRIOR TO JANUARY 1962

TAPE PROGRAM AND LOADER SUBROUTINE. THIS IS A B LOADER THAT CREATES A PROGRAM TAPE FOR PROGRAMS
COMPiled BY FORTRAN-2 AND EXCEEDING STORAGE CAPACITY.
SUBROUTINE F03 IS USED TO CALL IN THE PROGRAM TAPE.

0704 TSW9F02
AVAILABLE PRIOR TO JANUARY 1962

FLOATING TRAP SIMULATION.
FORTRAN-2 SUBROUTINE PERFORMING FLOATING OVERFLOW UNDERFLOW
AND OTHER CODE DETECTION CONSOLE 0346S DETAILED
IMPLEMENTATION TO CONTINUE BY AUTOMATIC RECONSTRUCTION OF RESULT.

0704 TSW9F01
AVAILABLE PRIOR TO JANUARY 1962

DINARY SUBROUTINE IDENTIFICATION AND MEMORY ALLOCATION
READS IN DINARY PROGRAM DECK LISTING ON-LINE OR OFF-LINE
THE SUBROUTINES IN THE DECK, ALSO VERSION LENGTH ENTRIES
COMMON REQUIREMENTS. UPON FINDING A TRAP TRANSFER CARD, STATUS
ACTUAL MAN AVAILABLE CELL AND LOWEST COMMON CELL REFERENCED
IN PROGRAM, MAKES NO CHECK FOR MISSED SUBROUTINES.

0704 TSW9F00
AVAILABLE PRIOR TO JANUARY 1962

LINEAR EQUATION SOLVER
GIVEN A LINEAR MATRIX EQUATION AX = B, WHERE A HAS THE
DIMENSIONS M X N AND C IS A COLUMN VECTOR OF DIMENSION
M X 1, THIS ROUTINE FINDS THE SOLUTION X IN THE LEAST
SQUARES SENSE. REQUIRES 146 CELLS OF PROGRAM AND CONSTANTS
FINDINGS Au AND BOS, PLUS N3 CELLS OF COMMON.

0704 TSW9FL03
AVAILABLE PRIOR TO JANUARY 1962

GENERAL LEAST SQUARE CURVE FITTING ROUTINE.
SOLVES THE VECTOR X IN LEAST SQUARES SENSE, REQUIRES
PROVIDE OF M X N CELL PROGRAM OF CONSTANTS FINDINGS LEAF.
Au, AND DOU PLUS N3 CELLS OF COMMON.

0704 TSW9DZ00
AVAILABLE PRIOR TO JANUARY 1962

RANDOM NUMBER GENERATOR, AZIMUTHAL ANGLE. FIXED POINT.

0704 TSW9CD00
AVAILABLE PRIOR TO JANUARY 1962

RANDOM NUMBER GENERATOR, CAUCHY DISTRIBUTION, FT. PT.

0704 TSW9REXFR
AVAILABLE PRIOR TO JANUARY 1962

RANDOM NO. GENERATOR, EXPONENTIAL DISTRIBUTION, FT. PT.

0704 TSW9DF10
AVAILABLE PRIOR TO JANUARY 1962

PLATE A PARTITION CONVERTS A PARTITION TO FLOATING POINT FORMAT.

0704 TSW9F00
AVAILABLE PRIOR TO JANUARY 1962

RANDOM NUMBER GENERATOR, FLOATING POINT.

0704 TSW9F00
AVAILABLE PRIOR TO JANUARY 1962

RANDOM NUMBER GENERATOR, FIXED POINT.

0704 TSW9F00
AVAILABLE PRIOR TO JANUARY 1962

READER OF FORMATTER STATEMENTS AT EXECUTION TIME.
FORTRAN-2 SUBROUTINE TYPE PROGRAM.

0704 TSW9F00
AVAILABLE PRIOR TO JANUARY 1962

REQUIRES 146 CELLS OF PROGRAM AND CONSTANTS FINDINGS LEAF.
Au, AND DOU PLUS N3 CELLS OF COMMON.

0704 TSW9F00
AVAILABLE PRIOR TO JANUARY 1962

DOUBLE PRECISION MATRIX ADDITION AND SUBTRACTION.
ADDS OR SUBTRACTS TWO REAL MATRICES, WHOSE ELEMENTS ARE STORED
CONSECUTIVELY BY ROWS IN CORE STORE USING DOUBLE PRECISION
STORED IN THE SAMEMANNER IN CORE STORAGE. REQUIRES OF
STORAGE PLUS B COMMON. DL DPA MUST BE ASSEMBLED
CONCURRENTLY.

0704 TSW9F00
AVAILABLE PRIOR TO JANUARY 1962

MULTIPLE REGRESSION SOLUTION PROGRAM.
TO PROVIDE BACK SOLUTIONS FOR THE RESULTS OF THE MULTIPLE
REGRESSION CODE SCRAB.
IBM 0704 PROGRAM LIBRARY ABSTRACT

0704 74VSCB1M
AVAILABLE PRIOR TO JANUARY 1962
INPUT EDITOR FOR MULTIPLE REGRESSION CODE SCRAP.
THIS 0704 PROGRAM USES FORTRAN TO CALCULATE FUNCTION VARIABLES FROM OBSERVED VARIABLES AND PLACE THEM IN THE FORMAT REQUIRED FOR THE MULTIPLE REGRESSION CODE SCRAP.

0704 74VSCRAF
MULTIPLE REGRESSION & CORRELATION ANALYSIS PROGRAM.
PROVIDES MULTIPLE REGRESSION COEFFICIENTS, STANDARD ERROR OF ESTIMATES, MEANS, STANDARD DEVIATIONS, REGRESSION COEFFICIENTS AND T-TABLE ENTRIES FOR UP TO 39 INDEPENDENT VARIABLES WITH AS MANY KS AND OBSERVATIONS PER VARIABLE REQUIRES 4K FOR WITH 1 DRUM AND AT LEAST 4 TAPE. CORR/94

0704 752SMFYA
AVAILABLE PRIOR TO JANUARY 1962
FORTRAN ERROR PACKAGE: A FORTRAN II SUBRoutines WITH SEVERAL ENTRIES TO PROVIDE ERROR ESTIMATES, MEANS, STANDARD DEVIATIONS, REGRESSION COEFFICIENTS AND T-TABLE ENTRIES FOR UP TO 39 INDEPENDENT VARIABLES WITH AS MANY KS AND OBSERVATIONS PER VARIABLE REQUIRES 4K FOR WITH 1 DRUM AND AT LEAST 4 TAPE. CORR/94

0704 753MUIEPF
AVAILABLE PRIOR TO JANUARY 1962
EXPONENTIAL INTEGRAL, COMPLEX IFIX, EXP-IFIX OR SIG/LOG/EXP AND SUBROUTINES. ALSO EXISTS AS FORTRAN II SUBROUTINES.

0704 755MUIEPF
EXPONENTIAL INTEGRAL, COMPLEX IFIX, EXP-IFIX OR SIG/LOG/EXP, FORTRAN II SUBROUTINE VERSION OF NU EXP-I IF OR RELOCATABLE BINARY CARDS INCLUDING LOG AND EXP SUBROUTINES. 29305 COMMON STORAGE. 7704 1962

IBM 0704 PROGRAM LIBRARY ABSTRACT

0704 754CEFEL
AVAILABLE PRIOR TO JANUARY 1962
GENERATE A FORTRAN II PROGRAM TAPE OR ABSOLUTE BINARY CARDS LOADS A FORTRAN II PROGRAM OR A BIN-TAPE AS ONE RECORD WITH A BOOTSTRAP PREFACE OF LOAD THE PROGRAM ON ABSOLUTE BINARY CARDS OR DRUM.

0704 756KINSNS
DECIMAL, OCTAL, DEC LOADER READS DATA FROM BINARY ENCYCLOPEDIA IF SENSE SWITCH 1111. OR ELLIPTIC PUNCH CARDS ON-LINE IF 01–11. ALSO ACCEPTS NUMBERS IN BINARY CODE WHICH SEEMS TO BE ACCEPTABLE TO USER HAS BEEN EXTENDED SO THAT INPUT PREPARED IN ANY FORMULA MAY BE ACCEPTED IN MANUSCRIPT, WHICH WILL BE PROCESSED. TECHNIQUES REQUIRE 688 WORDS OF CORE. ALL TEMPORARY STORAGE IS SELF-CONTAINED.

0704 756KINSNS
DECIMAL, OCTAL, DEC LOADER ALL SEQUENTIAL INPUT WITH A SINGLE CALL STATEMENT, AND ALL SEQUENTIAL DATA WHICH WERE NOT ORIGINALLY READ FROM BINARY ENCYCLOPEDIA, STORED ON 16K RELATIVE STORAGE WITH ALL TEMPORARILY STORED CORESELF-CONTAINED. CORR/81

0704 754MPD5M
DOUBLE PRECISION MATRIX SCALAR MULTIPLICATION MULTIPLES A REAL MATRIX WHOM ELEMENTS ARE STORED CONSECUTIVELY 9 TIMES A SCALAR IN CORE STORAGE USING DOUBLE PRECISION ARITHMETIC. THE ELEMENTS OF THE PRODUCT MATRIX ARE STORED IN THE SAME MANNER IN CORE STORAGE. REQUIRES 22 STORAGE OF COMMON, ORI OPMI MUST BE ASSEMBLED CONCURRENTLY.

0704 756KINSNS
CONTINUOUS DERIVATIVE INTERPOLATION SUBROUTINE COMPUTES Y AS A FUNCTION OF X FROM A TABLE OF X AND Y VALUES SUCH THAT THE FUNCTION Y IS AND ITS FIRST AND SECOND DERIVATIVES ARE CONTINUOUS IN THE RANGE OF X IN THE TABLE WRITTEN AS 2 FORTRAN II SUBROUTINES.

0704 762FAGE
DIFFERENTIAL EQUATION SOLUTION OF A FIRST ORDER DIFFERENTIAL EQUATION USING THE EULER-CAYLEY METHOD. SOLUTIONS FOR EQUATIONS WITH PREDICTED STEP SIZE. REQUIRES 160 CELLS, 2 COMMON AND A BLOCK OF 2000 CELLS.

0704 762FAGEO
AVAILABLE PRIOR TO JANUARY 1962
LAGRANGIAN INTERPOLATION AND/OR DIFFERENTIATION GIVEN M TABLES OF Y=FX WHERE M IS greater than or equal to KTH ORDER INTERPOLATION AND/OR DIFFERENTIATION OF THE LAGRANGIAN FORMULA IS PERFORMED ON ALL TABLES AS MUST BE OF SAME FORMAT. REQUIRES 274 CELLS AND COMMON TO COMMUNICATE.

0704 765AGCGZ
AVAILABLE PRIOR TO JANUARY 1962

0704 765AUGS3
AVAILABLE PRIOR TO JANUARY 1962
PLON TRACE PROGRAM = UA 1962.
0704 766RAGJF
AND/OR OFF-LINE OF-OFF LINES AND A PRINT AFTER EXECUTION OF EACH TRANSFER. 256 SUBROUTINES ARE AVAILABLE. CONDITIONAL AND/OR UNCONDITIONAL. ENTRANCE TO AND EXIT FROM TRAPPING MODIFIED BY CONTROLLED BY INDEX REGISTER CONTENTS. CORE STORAGE LOCATION CONTENTS COUNT-DOWN IN NUMBER OF TRANSFERS OF SOME CORE STORAGE LOCATION, OR MANUALLY BY THE SETTING OF A SENSE Switch. CORE STORAGE LOCATION ASSIGNMENTS 00000-007777.

0704 770AGJF
AVAILABLE PRIOR TO JANUARY 1962
DECIMAL-TO-BINARY CONVERSION PROGRAM = UA 1962 FIXED POINT, FLOATING POINT, INTEGER OR REAL, VARIABLE FIXED FIELD FORMAT IF FORTRAN FLOATING FIELD MAY BE SPECIFIED TO CASE INTERPRETATION. 32-CHARACTER BINARY NUMBERS MAY BE SCALING, REPLICATING, STOPPED, ETC. LOADING IS BY BLOCK, BUT THE INTERRUPT ALLOWS INPUT TO BE LOADED INTO ANY N SIZED CORE STORAGE. THE USE OF US 6575 FOR READ TAPE OR CARD, OCCUPIES 4451 CORE STORAGE LOCATION AND 40 WMP OF COMMON STORAGE.

IBM 0704 PROGRAM LIBRARY ABSTRACT

0704 770JFZTP
AVAILABLE PRIOR TO JANUARY 1962
FORTRAN II AND/OR FORTRAN II TO SELF-LOADING TAPE THIS PROGRAM MAKES A SELF-LOAD TAPE TAPE OF ANY NUMBER OF INDEPENDENT FORTRAN II AND OR FORTRAN II PROGRAMS. A LOAD FUNCTION IS REQUIRED IF MORE THAN ONE PROGRAM IS TO BE LOADED. THIS FUNCTION IS DESCRIBED IN APPENDIX A OF THE WRITING OF TRC.

0704 772JFZ0
AVAILABLE PRIOR TO JANUARY 1962

0704 775JFZ0
AVAILABLE PRIOR TO JANUARY 1962
FLOATING PT. CORREL. 22ND SUM. RUNG-EUTTA INTEGRATION OF SECOND-ORDER EQUATIONS. SOLVES A SET OF N SIMULTANEOUS SECOND-ORDER ORDINARY DIFFERENTIAL EQUATIONS IN WHICH FIRST DERIVATIVES MAY OR MAY NOT APPEAR.

0704 775W85C
AVAILABLE PRIOR TO JANUARY 1962
GENERAL LEAST SQUARE CURVE FITTING ROUTINE GIVEN AN N X M MATRIX A, AN M DIMENSIONAL ROW VECTOR B AND AN N X N DIAGONAL MATRIX S/STORED AS A ROW THIS ROUTINE FINDS AN M DIMENSIONAL ROW VECTOR V. IF THE USER SETS ALL S = 0 SOLVES V IN THE LEAST SQUARES SENSE.

0704 775W85D
AVAILABLE PRIOR TO JANUARY 1962
GENERAL ANALYSIS OF VARIANCE TO COMPUTE AND PRINT ALL SUMS OF SQUARES ASSOCIATED WITH FACTORIAL EXPERIMENTATION. ALL SUMS OF OBSERVATIONS ENTERING INTO EACH SUM OF SQUARES ARE ALSO PRINTED, POLYNOMIAL PARTITIONING OF MAIN EFFECT SUMS OF SQUARES IS OPTIONAL. ANY NUMBER OF FRATIONAL REPPLICATIONS CAN BE HANDLED AS WELL AS A HIGH DEGREE DEEPEST REPLICATION. CORR/84
0704 7B3WOD2 AVAILABLE PRIOR TO JANUARY 1962

COLUMN DISASSEMBLY PROGRAM ONTO COMPUTED FOR EACH RECORD. A CHECK SUM IS COMPUTED FOR EACH RECORD.

0704 7B3WOD43 AVAILABLE PRIOR TO JANUARY 1962

SELF LOADING TAPE WRITING ROUTINE

TO LOAD THE INFORMATION FROM A PORTION OBJECT PROGRAM ONTO A MASTER PROGRAM TAPE, TO BE USED WITH ALL BUT THE DECK WHICH MAKES UP THE FINAL RECORD.

0704 7B2PPCR3 AVAILABLE PRIOR TO JANUARY 1962

CORRELATION AND REGRESSION ANALYSIS.

CALCULATIONS ARE PERFORMED AS SPECIFIED BY A CONTROL CARD. OPTION: OUTPUT FORW, PROVISIONS ARE MADE FOR PROGRAM INTERRUPT AND RESTART. ADDITIONAL COMPUTATION MAY BE INTRODUCED. MAXIMUM NUMBER OF VARIABLES IS 110. SINGLE PRECISION OR DOUBLE PRECISION, NUMBER OF OBSERVATIONS IS 2*20*1.

0704 7B6EGDR5 AVAILABLE PRIOR TO JANUARY 1962

COLUMN DISASSEMBLY PROGRAM

THIS PROGRAM WILL READ A COLUMN BINARY ABSOLUTE OR RELOCATABLE DECK AND TRANSLATE THE INFORMATION BACK TO SYMBOLIC FORM, SEE 6 AD51.

0704 7B6EGD56 AVAILABLE PRIOR TO JANUARY 1962

ROW DISASSEMBLY PROGRAM

THE PROGRAM WILL READ A BINARY ABSOLUTE OR RELOCATABLE DECK WITH BINARY TRANSITION-CORRECTION CARDS AND TRANSLATE THE INFORMATION BACK TO SYMBOLIC FORM WHICH WOULD BE ACCEPTABLE TO A BYPASS-7 ASSEMBLER.

0704 7B6EES5M AVAILABLE PRIOR TO JANUARY 1962

ERRO EPOR FOR 7B3W043

THE INCLUSION OF THE STANDARD ERROR PROCEDURE FOR FORTAN 4 INCORPORATION OF AN ERROR ROUTINE AND A REVISION OF THE LIBRARY SUBROUTINES TO MAKE USE OF ERROR RETURN. FORTAN LIBRARY SUBROUTINES WERE MODIFIED, AND IN SOME CASES REPLACED BY BETTER ROUTINES. CORR/ 657

0704 7B7PMW62 AVAILABLE PRIOR TO JANUARY 1962

COMPUTATION OF A MINIMUM TWO-LEVEL AND-OR-SWITCHING CIRCUIT. GENERATES A MINIMUM TWO-LEVEL SWITCHING CIRCUIT WHERE ONE LEVEL IS ALL ANS AND THE OTHER LEVEL IS ALL ORS. DONT-CARE CONDITIONS AND MULTIPLE OUTPUT PROBLEMS ARE HANDLED. THE CIRCUIT IS DESCRIBED AS A BOOLEAN FUNCTION IN NORMAL FORM, AND TO THE MINIMIZATION OF TOPOLOGICAL COVERS OF CYCLIC COMPLEXES. PROGRAM MAY BE RUN ON A MACHINE WITH 2 ON 8 7372 OR A 750 MEMORY FRAME. IT ALSO RUNS SIX TAPE AND FOUR LOGICAL UNITS. CORR/ 884

0704 7B7BAS1 AVAILABLE PRIOR TO JANUARY 1962

ADD OR SUBTRACT TWO FOURIER SERIES.

IN CANONICAL REPRESENTATION OBTAINING AS THE RESULT A FOURIER SERIES IN CANONICAL REPRESENTATION.

0704 7B7BAS1 AVAILABLE PRIOR TO JANUARY 1962

ADD OR SUBTRACT TWO FOURIER SERIES.

IN CANONICAL REPRESENTATION OBTAINING AS THE RESULT A FOURIER SERIES IN CANONICAL REPRESENTATION.

0704 7B7BAS1 AVAILABLE PRIOR TO JANUARY 1962

ADD A TERM TO A FOURIER SERIES.

IN CANONICAL REPRESENTATION OBTAINING AS THE RESULT A FOURIER SERIES IN CANONICAL REPRESENTATION.
IBM 0704 PROGRAM LIBRARY ABSTRACT

0704 T83067SP1 AVAILABLE PRIOR TO JANUARY 1962

SPLITS A FOURIER SERIES IN CANONICAL REPRESENTATION.
FOR THE COEFFICIENTS OF A SPECIFIED TERM, TAKING IF P IS THE
EXCLUSION INDEX AND THE OR WAVEFORM SIMILARITY OF THE TYPE BEING XEARCHED FOR IN THE SERIES, EXECUTION TIME DOES NOT EXCEED 556 RP CYCLES.

0704 T83067SP2 AVAILABLE PRIOR TO JANUARY 1962

UNPACKS THE INDICES FROM FOURIER SERIES INDEX WORDS, CONVERTS THEM TO NORMALIZED FLOATING-POINT FORM, AND CONVERTS THE INDEXES 1 TO 6, WHERE I AND K ARE THE INDICES, AND D IS AN ARBITRARY PARAMETER SPILL IS DESIGNED FOR USE AS A SUBROUTINE OF ESPH.

0704 T83067SP2 AVAILABLE PRIOR TO JANUARY 1962

COMPUTES A SPECIAL FUNCTION F OF THE INDICES
IN ONE TERM OF A FOURIER SERIES, USES UHT AS A SUBROUTINE.

0704 T83067SP1 AVAILABLE PRIOR TO JANUARY 1962


0704 T83067SP1 AVAILABLE PRIOR TO JANUARY 1962

UNPACKS UP TO 4 INDICES INTO AN INDEX WORD, OF A FOURIER SERIES IN CANONICAL REPRESENTATION AND CONVERTS THEM TO NORMALIZED FLOATING-POINT NUMBERS.

0704 T83067SP1 AVAILABLE PRIOR TO JANUARY 1962

WRITES A FOURIER SERIES AS ONE BINARY RECORD ON TAPE, WITH LOGICAL CHECK SUM AS THE LAST WORD ON THE RECORD.

IBM 0704 PROGRAM LIBRARY ABSTRACT

0704 T83067SP1 AVAILABLE PRIOR TO JANUARY 1962

CONVERTS A FOURIER SERIES IN CANONICAL REPRESENTATION, TO QCD AND WRITING THE ACO SERIES ON ANY DESIRED TAPE. PRINTING IS OPTIONAL.

0704 T83067SP1 AVAILABLE PRIOR TO JANUARY 1962

MACHINE LOADING PROBLEM OF LINEAR PROGRAMMING
SOLVES A GENERALIZATION OF THE TRANSPORTATION PROBLEM IN WHICH EACH TERM OF ROW AND COLUMN SUMS MAY BE WEIGHTED BY ARBITRARY NON-UNITARY COEFFICIENTS. A PRINTING LISTING DISTRIBUTED IN S.O. 883.

0704 T83067SP1 AVAILABLE PRIOR TO JANUARY 1962

OPTIMIZED TAPE READ FOR FORMATS 1376-6, THIS FORTRAN II SUBROUTINE READS FROM TAPE 6 CONVERTS, AT OPTIMIZED SPEED, DATA PUNCHED IN THE FORMAT 1376-6. IT ALLOWS RECORDS AND CONVERSION TO PROCEED AT ESSENTIALLY THE SAME SPEED NORMALLY REQUIRED FOR READING ALONE, THEREBY ELIMINATING THE STOP-START TIME AT INTER-RECORD GAPS.

0704 T83067SP1 AVAILABLE PRIOR TO JANUARY 1962

FLOATING POINT FOR VARIATE PROBABILITY INTEGRAL
OBTAINS THE PROBABILITY INTEGRAL FOR X/J LESS THAN OR
EQUAL TO 18 LESS THAN OR EQUAL TO 6 VARIABLES OF THE NORMAL
FUNCTION OVER POLYNOMIAL REGIONS, REQUIRES 27X CELLS FOR PROGRAM AND CONSTANTS PLUS 14 COMMON-CONM.1208.

0704 T83067SP1 AVAILABLE PRIOR TO JANUARY 1962

AUTOMATIC CHECK POINT AND RECOVERY
THIS PROGRAM KEEPS A RUNNING RECORD OF THE MAIN PROGRAM BY DUMPING THE CONTENTS OF MEMORY, TAPE UNIT INFORMATION AND ALL INDICATORS ON THE TAPE UNIT, WHEN DUMPED INTO A MEMORY TAPE. THIS AVOIDS A MEAN OF RESTORING A PROGRAM ANY POINT PREVIOUSLY REACHED WITH A MINIMUM OF LOST TIME.
0704 0524-014E

DIFFERENTIAL EQUATIONS SOLVER
SOLVES SIMULTANEOUS DIFFERENTIAL EQUATIONS WITH INTERRUPTIBLE INTEGRATION ON EITHER THE INDEPENDENT OR THE DEPENDENT VARIABLES. METHOD USES A FOURTH-ORDER RUNGE-KUTTA. STORAGE REQUIREMENTS ARE 452 WORDS FOR PROGRAM, PLUS 6 WORDS OF COMMON.

0704 0525-014F

GENERAL INTEGRAL EVALUATOR
GENERATES THE SIMPSON RULE APPROXIMATIONS FOR ANY TYPE OF INTEGRAL EXCEPT ON, WHETHER ITERATED INTEGRALS, MULTIPLE INTEGRAL, VECTOR VALUED (INTEGRAL FROM A VECTOR VALUED FUNCTION OR THE INTEGRAL OF A FUNCTION OF OTHER INTEGRALS. REQUIRE THIS WORDS PLUS 1 COMMON.

0704 0362-014G

PRINT RSS LOADER DIAGNOSTICS
PRINTS A 10% SAP-CODED FORTRAN II SUBPROGRAM TO SUPPLY ON-LINE DIAGNOSTIC COMMENTS ON THE ACTIVATED ERROR STOPS OF M8522 LOADER.

0704 0363-014H

OCTAL CONNECTION CARD READER
MITS-1C A 10% SAP-CODED FORTRAN II SUBPROGRAM TO LOAD RELATABLE OR ABSOLUTELY CORRECT OCTAL CONNECTION CARDS AND COMMENT CARDS. CORRECTIONS AND COMMENTS MAY BE LOGGED ON OUTPUT TAPE Z.

0704 0364-014I

WRITE RSS LOADER STORAGE MAP
MISTP-A 10% SAP-CODED FORTRAN II SUBPROGRAM THAT WRITES ON TAPE Z THE CORE MEMORY STORAGE MAP FORMED BY THE M8522 LOADER.

0704 0365-014J

WRITE CORE IMAGE ON TAPE
MISTP-A 10% SAP-CODED FORTRAN II SUBPROGRAM THAT WRITES THE CONTENTS OF CORE MEMORY AS A SINGLE SELF-LOADING RECORD ON TAPE 4.

0704 0366-014K

COMPLEX NUMBER INTERPRETIVE SYSTEM FLOATING POINT?
A TWO-ADDRESS INTERPRETIVE SYSTEM PROPOSED TO WORK WITHIN SAP PROGRAMS. IT OFFERS A TOTAL OF TWELVE ALGEBRAIC OPERATIONS, FOUR CONTROL OPERATIONS AND THREE TRAIL OPERATIONS. INDEXING IS AVAILABLE BUT IS LIMITED TO ONE INDEX REGISTER.
IBM 0704 PROGRAM LIBRARY ABSTRACT

0704 BITOREC2P

AVAILABLE PRIOR TO JANUARY 1962

OFFSET CIRCLE PROBABILITY FUNCTION.
COMPUTES THE OFFSET CIRCLE PROBABILITY FUNCTION, P(A,Z),
EQUAL TO THE INTEGRAL FROM Z TO 1.5 TIMES THE
PRODUCT OF SQUARED PLUS X SQUARED TIMES THE
MODIFIED BESSEL FUNCTION OF THE FIRST KIND OF ORDER Z
OF TIMES X FOR PARAMETER VALUES A AND Z WHERE Z IS GREAT
ER THAN OR EQUAL TO ZERO.

0704 BTOGROWN

AVAILABLE PRIOR TO JANUARY 1962

BINARY INTEGER TO ROMAN NUMERAL CONVERSION.
A PORTABLE BINARY INTEGER IS CONVERTED TO A BCD ROMAN NUMERAL

0704 BTOGCOLO

AVAILABLE PRIOR TO JANUARY 1962

704 SURGE OBJECT LOADER
LOAD IS A ONE CARD LOADER USED TO LOAD 704 SURGE OBJECT PROGRAMS.

0704 BTOGCS500

AVAILABLE PRIOR TO JANUARY 1962

704 SURGE SYSTEM START
THE 5500 CARD IS USED TO INITIATE A 704 SURGE COMPILATION.

0704 BTOGSURG

AVAILABLE PRIOR TO JANUARY 1962

704 SURGE SYSTEM
THE 704 SURGE SYSTEM IS A SELF-CONTAINED COMPILER DESIGNED
FOR DATA PROCESSING TYPE PROGRAMS. THE SYSTEM CONVERTS A
FIXED FORMAT SOURCE PROGRAM TO AN 80-BINARY-80 CHARACTERS.
Either on 80-BINARY-80 CARDS OR ON TAPE, THE BINARY SYSTEM
DECK MAY BE USED ON ANY 704 SERIES 8, 216, 417, 418, OR 770
MACHINES WITHOUT REQUIRING ANY MODIFICATIONS. THE SYSTEM USES 365,060
PREVIOUSLY LOADABLE NAME, ALPHABETICALLY FORMED IN
PATHWAY, AND ONLINE EQUIPMENT OPERATES.

0704 BTOGREMPX

AVAILABLE PRIOR TO JANUARY 1962

EXTREMUM OF UNIMODAL FUNCTIONS OF ONE VARIABLE
ANY NUMBER OF FUNCTIONALS MAY BE MINIMIZED MINIMIZED.
THE DESIRED ACCURACY MAY BE SPECIFIED, OR THE NUMBER
OF FUNCTIONALS TO BE USED MAY BE SPECIFIED AND
THE PROGRAM WILL CALCULATE THE EXTREMUM TO THE BEST
ACCURACY THEN POSSIBLE. THE PROGRAM HAS ADDITIONAL
ERROR PRINTOUTS.

IBM 0704 PROGRAM LIBRARY ABSTRACT

0704 BTOGREM501

AVAILABLE PRIOR TO JANUARY 1962

ESTIMATION FROM DUPLICATED TRUNCATION SAMPLES
ESTIMATES THE HAJN AND STANDARD DEVIATION OF THE
NORMAL DISTRIBUTION FROM A DUPLICATE SAMPLE
OF A HAJN DISTRIBUTION. THE TRUNCATION POINTS ARE
SHOWN AND THE TRUNCATION POINTS ARE KNOWN.

0704 BTOGICOC

AVAILABLE PRIOR TO JANUARY 1962

MANIPULATE BCD-CODED DATA, INCLUDING
704 BCD-CODED COMMON SUBPROGRAMS.

0704 BTOGINT1

AVAILABLE PRIOR TO JANUARY 1962

INTERVAL ARITHMETIC SUBROUTINE
AN ARBITRARY SEQUENCE OF THE ARITHMETIC OPERATIONS
PERFORMED ON INTERVALS BY INTERPOLATION OF THE
CALLING SEQUENCE, ROUND-OFF ERROR IS INCLUDED IN THE
RESULTANT INTERVALS. EACH INTERVAL IS REPRESENTED BY
ITS TWO ENDPOINTS. EACH ENDPOINT IS IN SINGLE-
PRECISION NORMALIZED FLOATING POINT FORM. UNDERFLOW IS
AUTOMATICALLY ELIMINATED. OVERFLOW RESULTS IN
PROGRAMMED INTERRUPTION. REQUIRES 40 LOCATIONS.
AVERAGE EXECUTION TIME ABOUT 1.7 MS PER OPERATION.

0704 BTOGRFRP1

AVAILABLE PRIOR TO JANUARY 1962

REAL ROOTS OF A REAL POLYNOMIAL USING INTERVAL ARITH.
PROGRAM IS IN THE FORM OF AN INTERVAL SUBROUTINE
OUTPUT IS A SEQUENCE OF CLOSED FINITE INTERVALS, EACH
CONTAINING AT LEAST ONE, IF POSSIBLY ONLY ONE, REAL
ROOT OF THE POLYNOMIAL. THE INTERVALS ARE MADE
SMALLER AND SMALLER WITH ACCOUNTING FOR ALL
ROUND-OFF ERROR. COEFFICIENTS OF THE POLYNOMIAL MAY
ALSO BE INTERVALS. USES 1 INT FOR INTERVAL ARITH.
REQUIRES 40 LOCATIONS EXCLUSIVE OF INTL.

IBM 0704 PROGRAM LIBRARY ABSTRACT

0704 BTOGRFRQ2

AVAILABLE PRIOR TO JANUARY 1962

REAL ROOTS OF A REAL POLYNOMIAL USING INTERVAL ARITH
PROGRAM IS SELF-LOADING AND PROVIDES EXTERNAL DECIMAL
INPUT AND OUTPUT. OTHERWISE IT IS LIKE IN FRP1, WHICH
IS USED AS A SUBROUTINE.

0704 BTOGMSM2

AVAILABLE PRIOR TO JANUARY 1962

SOLUTION OF MATRIX EQUATION AX = B USING INTERVAL ARITH.
PROGRAM IS IN THE FORM OF AN INTERVAL SUBROUTINE. THE
EQUATIONS OF OUTPUT MATRIX A ARE CLOSED FINITE INTERVALS
WHICH CONTAIN THE ELEMENTS OF THE EXACT SOLUTION.
ROUND-OFF ERROR ACCOUNTED FOR. USEFUL FOR MATRICES
OF SMALL ORDER, IFX IS ON LESS. USES FORM OF GAUSS
ELIMINATION. EMPLOYS 10 INT FOR INTERVAL ARITHMETIC.
REQUIRES 40 LOCATIONS EXCLUSIVE OF INTL.
EXECUTION TIME ABOUT 49K/197277 MILLI-SECONDS,
WHERE 8 IS M ноя AND 8 IS N.

0704 BTOGMSM3

AVAILABLE PRIOR TO JANUARY 1962

SOLUTION OF MATRIX EQUATION AX = B USING INTERVAL ARITH.
PROGRAM IS SELF-LOADING AND PROVIDES EXTERNAL DECIMAL
INPUT AND OUTPUT. OTHERWISE IT IS LIKE IN FRP1, WHICH
IS USED AS A SUBROUTINE.

0704 BRIMHSMP

AVAILABLE PRIOR TO JANUARY 1962

ARC ATMOSPHERIC SUBROUTINE
COMPUTES 5 ATMOSPHERIC PROPERTIES; DENSITY, SPEED OF SOUND,
TEMPERATURE, MOLECULAR-DENSITY TEMPERATURE-DENSITY,
COEFFICIENT OF VISCOSITY, AND MOLECULAR WEIGHT AS FUNCTIONS
OF ALTITUDE, BASED ON THE 1959 AND 1964 ARC MODELS. REAL
VALUES DIFFER FROM THE 1959 VALUES ONLY ABOVE 60,000 FEET.
NONE OF THE 1959 MOLECULAR WEIGHT VALUES GREATER THAN
300,000 FEET CAUSES MOLECULAR WEIGHT AND TEMPERATURE
TO VARY FROM THE 1959 MODEL. A REQUIRED EXP. LG. AND
SCOT SUBROUTINES. 176 STORAGE CELLS 6 704 CPU.

0704 BRIMHSMM

AVAILABLE PRIOR TO JANUARY 1962

ESTIMATES AND ESTIMATIONS OF A HERMITIAN MATRIX.
JACOBS METHOD IS USED. THE MATRIX ELEMENTS ARE SINGLE-PREC
NORMALIZED FLOATING-POINT NUMBERS. THE ELEMENTS MAY
BE GIVEN IN EITHER RECTANGULAR OR POLAR FORM AND THE OUTPUT
MAY BE OBTAINED IN EITHER FORM. THE SUBROUTINE REQUIRES
LOCATIONS PLUS 29 LOCATIONS OF COMMON AND 7/256 1/24
LOCATIONS PROVIDED BY USER.

EHP 0704 PROGRAM LIBRARY ABSTRACT

0704 BROMHSMM

AVAILABLE PRIOR TO JANUARY 1962

KEY WORD IS CONTEXT
EACH WORD IN A SERIES OF BIBLIOGRAPHY TITLES IS SEARCHED FOR
A TABLE TO DETERMINE ITS STATUS AS EITHER A KEY WORD OR A
COMMON WORD. FOR EACH KEY WORD FOUND AS A COMPONENT OF
THE SURROUNDING TITLE AS PUT OUT WITH THE EMBEDDED KEY WORD IS
A BREAKING OF THE 240 CHARACTERS. THE TOTAL KEY WORDS IN COMMON
OUTPUT MAY BE STORED TO PRODUCE AN INDEX FOR THE BIBLIOGRAPHY
AUTHOR AND SOURCE INFORMATION ATTACHMENT TO EACH TITLE IS CON
MOVED IN A STANDARD FASHION TO 11 CHARACTERS FOR OUTPUT WITH
EACH KEY WORD IN THE CORRESPONDING TITLE.

0704 BRMRVYNA

AVAILABLE PRIOR TO JANUARY 1962

METER FIXED POINT HUNGE-OIUTA
SOLVES A SET OF M DIFFERENTIAL EQUATIONS. 48 WORDS OF PROGRAM PLUS 3 COMMON PLUS IN WORDS OF
STORAGE, 14 WORDS OF AUXILIARY TIME/FPI
PER INTEGRATION STEP.

0704 BRTAYV00

AVAILABLE PRIOR TO JANUARY 1962

VEPP INSERT LEADING BLANKS.
MODIFIES BCD FIELD FORM LEFT TO RIGHT UNTIL END OF FIELD OR
ENCOUNTERING CHARACTERS OTHER THAN ZERO, BLANK,
PLUS ZERO, MINS ZERO, PLUS SIGN OR MINS SIGN.
REFERENCE MC CV VEPP.

0704 BRTAE003

AVAILABLE PRIOR TO JANUARY 1962

ERROR FUNCTION
EVALUATES ERROR FUNCTION 18.55 M AND FOR NORMAL
PROBABILITY FUNCTION 1/40 MS.
REQUIRES 40 LOCATIONS PLUS 2 COMMON. TURNS OFF AC OVERFLOW INDICATOR. Voids 436

0704 BRTA0051

AVAILABLE PRIOR TO JANUARY 1962

POWER SPECTRUM SPECTRUM
THE SUBROUTINE COMPUTES THE RMS-ARITHMETIC MEAN, AND THE
POWERS AT A SPECIFIC FREQUENCY INTERVALS OF DATA
THE NUMBER OF DATA POINTS AND THE TIME INCREMENT AT WHICH
THE POINTS ARE OBTAINED ARE REQUIRED. THE SUBROUTINE USES 246
CELLS.
RATLICAL FORTRAN ASS LOADER

TESTED FOR OVERFLOW
REDUCED TO LOWEST TERMS.

RATLICAL FORTRAN ASS LOADER

TESTED FOR OVERFLOW
REDUCED TO LOWEST TERMS.

RATLICAL FORTRAN ASS LOADER

TESTED FOR OVERFLOW
REDUCED TO LOWEST TERMS.

RATLICAL FORTRAN ASS LOADER

TESTED FOR OVERFLOW
REDUCED TO LOWEST TERMS.

RATLICAL FORTRAN ASS LOADER

TESTED FOR OVERFLOW
REDUCED TO LOWEST TERMS.

RATLICAL FORTRAN ASS LOADER

TESTED FOR OVERFLOW
REDUCED TO LOWEST TERMS.

RATLICAL FORTRAN ASS LOADER

TESTED FOR OVERFLOW
REDUCED TO LOWEST TERMS.

RATLICAL FORTRAN ASS LOADER

TESTED FOR OVERFLOW
REDUCED TO LOWEST TERMS.

RATLICAL FORTRAN ASS LOADER

TESTED FOR OVERFLOW
REDUCED TO LOWEST TERMS.

RATLICAL FORTRAN ASS LOADER

TESTED FOR OVERFLOW
REDUCED TO LOWEST TERMS.

RATLICAL FORTRAN ASS LOADER

TESTED FOR OVERFLOW
REDUCED TO LOWEST TERMS.

RATLICAL FORTRAN ASS LOADER

TESTED FOR OVERFLOW
REDUCED TO LOWEST TERMS.

RATLICAL FORTRAN ASS LOADER

TESTED FOR OVERFLOW
REDUCED TO LOWEST TERMS.

RATLICAL FORTRAN ASS LOADER

TESTED FOR OVERFLOW
REDUCED TO LOWEST TERMS.

RATLICAL FORTRAN ASS LOADER

TESTED FOR OVERFLOW
REDUCED TO LOWEST TERMS.

RATLICAL FORTRAN ASS LOADER

TESTED FOR OVERFLOW
REDUCED TO LOWEST TERMS.

RATLICAL FORTRAN ASS LOADER

TESTED FOR OVERFLOW
REDUCED TO LOWEST TERMS.

RATLICAL FORTRAN ASS LOADER

TESTED FOR OVERFLOW
REDUCED TO LOWEST TERMS.

RATLICAL FORTRAN ASS LOADER

TESTED FOR OVERFLOW
REDUCED TO LOWEST TERMS.

RATLICAL FORTRAN ASS LOADER

TESTED FOR OVERFLOW
REDUCED TO LOWEST TERMS.

RATLICAL FORTRAN ASS LOADER

TESTED FOR OVERFLOW
REDUCED TO LOWEST TERMS.

RATLICAL FORTRAN ASS LOADER

TESTED FOR OVERFLOW
REDUCED TO LOWEST TERMS.

RATLICAL FORTRAN ASS LOADER

TESTED FOR OVERFLOW
REDUCED TO LOWEST TERMS.

RATLICAL FORTRAN ASS LOADER

TESTED FOR OVERFLOW
REDUCED TO LOWEST TERMS.
DOUBLE PRECISION SIN-COS ROUTINE COMPUTES A DOUBLE PRECISION FLOATING POINT SIN OR COSINE OF A DOUBLE PRECISION FLOATING POINT ARGUMENT. THE ARGUMENT MUST BE IN RADIANS.

CORE ROOT SUBROUTINE EVALUATES THE CORE ROOT OF A NORMALIZED FLOATING POINT NUMBER.

MERCURY COMPARISON CUMP COMPARES PROGRAM ON CARDS OR TAPE WITH SAME PROGRAM IN CORE.

RCEMPEEPFD DOUBLE PRECISION FLOATING POINT EXPONENTIAL ROUTINE.

W-STEP TRAPEZOIDAL RULE INTEGRATION/EQUAL INTERVALS - A SHARE TYPE SUBROUTINE FOR THE EVALUATION OF F(X) FOR THE N VALUES OF X LINING IN THE INTERVAL MUST BE PROVIDED. SUBROUTINE CAN BE CONVENIENTLY USED WITH PM ITEM TO OBTAIN TRAPEZOIDAL RULE INTEGRATION.

MURP-INVRS SUBROUTINE APPLICATION OF RA-INVRS TO A SQUARE MATRIX.

21H 070N PROGRAM LIBRARY ABSTRACT

0704 9314PTIFM AVAILABLE PRIOR TO JANUARY 1962

TPF SORTER FOR THE 21H 70N.

0704 9204PTIFM AVAILABLE PRIOR TO JANUARY 1962

TPF SORTER, FIRST PHASE OF A GENERAL PURPOSE.

0704 0704921 926TIPM AVAILABLE PRIOR TO JANUARY 1962

THE IBM 704. SECOND PHASE IS TAPED.

0704 0704920 931PKMTL AVAILABLE PRIOR TO JANUARY 1962

THE IBM 704. SECOND PHASE IS TAPED.

0704 07049375ERCONV AVAILABLE PRIOR TO JANUARY 1962

LP/90 TO SCROL 704 INPUT CONVERTER.

0704 07049355ERCONV AVAILABLE PRIOR TO JANUARY 1962

LP/90 TO SCROL 704 INPUT CONVERTER.

0704 0704930GMDY AVAILABLE PRIOR TO JANUARY 1962

BASSO'S 80S.

0704 0704935USC AVAILABLE PRIOR TO JANUARY 1962

TO STORE CARD INPUT.

0704 0704935USC AVAILABLE PRIOR TO JANUARY 1962

TO STORE CARD INPUT.

0704 0704935USC AVAILABLE PRIOR TO JANUARY 1962

TO STORE CARD INPUT.

0704 0704935USC AVAILABLE PRIOR TO JANUARY 1962

TO STORE CARD INPUT.
IBM 0704 PROGRAM LIBRARY ABSTRACT

0704 0101 JPOMI AVAILABLE PRIOR TO JANUARY 1962
ZERO, MINIMUM SOLVER 1 SOLVES THE CLASS OF PROBLEMS WHICH CAN BE STATED AS MINIMIZATION OF A CONTINUOUS FUNCTION WHERE ANY COMBINATION OF ZEROS AND/OR MINIMUMS ARE POSSIBLE TO SOLVE SIMULTANEOUSLY.

0704 0102 JPSDF AVAILABLE PRIOR TO JANUARY 1962
RENDERS COEFFICIENTS COMPUTE CI=fi-(CI+P);... (CI+M/N-1)..., BY USING STIRLING APPROXIMATIONS. LA 5500, LA 5550, AND 6500 MUST BE ASSEMBLED WITH 5150 150 STORAGE LOCATIONS ARE USED.

0704 0103 JPSCH AVAILABLE PRIOR TO JANUARY 1962
SIMULTANEOUS PARTIAL DIFFERENTIAL EQUATIONS SOLVER SOLVES THE PROBLEM: \( \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = f(x,y) \). STORAGE REQUIRED IS 456 WORDS N 8 WORDS OF COMMON.

0704 0104 JPSNO AVAILABLE PRIOR TO JANUARY 1962
GAUSS APPROXIMATE SOLVER THIS SUBROUTINE IS CAPABLE OF GENERATING THE GAUSS APPROXIMANT FOR ANY TYPE OF INTEGRAL EXPRESSION WHETHER IT BE AN INTEGRAL, VECTOR VALUE, OR INTEGRAL OF A FUNCTION OF OTHER INTEGRALS, OR ANY COMBINATION OF THESE. USES 271 LOCATIONS.

0704 0105 JPSPI AVAILABLE PRIOR TO JANUARY 1962
QUADRATIC PROGRAMMING CODE THE CODE WILL SOLVE THE QUADRATIC PROGRAMMING PROBLEM OF MINIMIZING A QUADRATIC FUNCTION OF NONNEGATIVE VARIABLES SUBJECT TO LINEAR CONSTRAINTS. THE NUMBER OF CONSTRAINTS PLUS VARIABLES MUST BE LESS THAN 256. THE PROGRAM WILL OPERATE ON A 7090 WITH A MINIMUM OF 8K, 4 DRUMS, AND 6 TAPE. THE CODE, WITH THE ADDITION OF TWO CARDS, CAN RUN ON A 7040 WITH COMPATIBILITY.

IBM 0704 PROGRAM LIBRARY ABSTRACT

0704 0106 JPSQF AVAILABLE PRIOR TO JANUARY 1962
ANALYZING SYSTEM FAILURE DATA THIS 704 PROGRAM WAS WRITTEN TO IMPLEMENT THE STATISTICAL ANALYSIS OF THE FAILURE PROPERTIES OF COMPUTER SYSTEMS WHICH IS GIVEN IN "THE THEORY OF MEASUREMENT OF COMPUTER SYSTEM RELIABILITY- IN PRESS."

0704 0107 JPSSTP AVAILABLE PRIOR TO JANUARY 1962

0704 0108 JPSSTP AVAILABLE PRIOR TO JANUARY 1962
PI-STAR SUBROUTINE SUBROUTINE TO TRANSFORM AN IB0209 06.6 A BOOLEAN FUNCTION OR FUNCTIONS INTO A NORMAL FORM EXPRESSION OR EXPRESSIONS. OTHERWISE EXPRESSED, IT GIVES THE FUNCTION OR FUNCTIONS DESCRIBED BY A BOOLEAN TREE OR GRAPH.

0704 0109 JPSSTY AVAILABLE PRIOR TO JANUARY 1962

0704 0110 JPSWUXR AVAILABLE PRIOR TO JANUARY 1962
FLOATING POINT TRAP ROUTINE FOR FORTRAN VAP USES THIS SUBROUTINE PROVIDES ENTRY TO THE FLOATING-POINT TRAP MODE AND SETS THE INNER PROGRAM FOR DETERMINING WHETHER A FLOATING-POINT ERROR OCCURRED OR UNDERFLOW TOOK PLACE AND THE ACTION TO BE TAKEN. THE ROUTINE IS ALREADY FOR AN EXIT FROM THE FLOATING-POINT TRAP MODE.

0704 0111 JPSWUXR AVAILABLE PRIOR TO JANUARY 1962
SOLUTION OF RADIAL SCHRÖDINGER EQUATION THIS IS A FORTRAN PROGRAM TO CALCULATE THE EIGENVALUES AND EIGENFUNCTIONS OF THE RADIAL SCHRÖDINGER EQUATION.
COMPUTES A SH OF GENERATES A REDUCES 4 SEQUENTIAL MACHINE AND OUTPUT SEQUENCES. OR 0104 1092RSMlAS 070'1 l081LROSRA IWAILABLE PRIOR TO JANUARY 1962.

GREATMENT OF GENERA FORTRAN EDIT DECK PROGRAM 11 ORARY ABS TRAC OR EQUIVALENT.

NUMERICAL PROGRAMMING SYSTEM NUMERICAL EQUATIONS.

COEFFICIENTS AND P1FORMATION ARE IN IBM 0704 PROGRAM LIBRARY ABSTRACT.

DIFFERENTIAL VARIABLES.

THE IBM 0704 PROGRAM LIBRARY ABSTRACT

THE PROGRAM GIVES AN ITERATIVE MULTIPLE REGRESSION TECHNIQUE THE LEAST SQUARE ESTIMATES OF THE SHOWN.

THESE ROUTINES CONSTITUTE A AUGMENTATION OF THE RSFM1 ROUTING FOR LINEAR PROGRAMMING. THEY PERMIT THE FINDING OF ALL OPTIMAL SOLUTIONS OF A LINEAR PROGRAMING PROBLEM OR OF ALL VERTICES OF A POLYHEDRON GIVEN IN INEQUALITIES. AN EFFICIENT NON-EXHAUSTIVE ALGORITHM IS USED.

IF NOT BUILT IN 1401 HARDWARE, 74/MAY NEED A SUB ROUTINE TO PROVIDE TAPE ERRORS. CORR 1145.

A SUBROUTINE IS AVAILABLE PRIOR TO JANUARY 1962.

A SUBROUTINE DESCRIBING THE PROGRAM FORTAN LIBRARY.

FORTRAN INPUT/OUTPUT PACKAGE PROVIDES GREATER INPUT AND OUTPUT FLEXIBILITY WITH 704.

THE DEGREE OF INTERPOLATION IS VARIABLE IN BOTH DIRECTIONS FROM 1 TO 7. LAGRANGE INTERPOLATION IS USED THROUGHOUT THIS ROUTINE. FUNCTIONS MAY BE EITHER CONTINUOUS OR DISCONTINUOUS.

AUTOMATIC TOOL PATH GENERATION FOR NUMERICAL CONTROL OF MULTIPLE FORMAT STATEMENTS ARE USED IN DESCIBING TAPE RECORDS. INCLUDES 5000 WORDS OF UPPER STORAGE FOR 1/0 BUFFER.

THE AUTOMATIC TOOL PATH GENERATION FOR NUMERICAL CONTROL OF MULTIPLE FORMAT STATEMENTS ARE USED IN DESCIBING TAPE RECORDS. INCLUDES 5000 WORDS OF UPPER STORAGE FOR 1/0 BUFFER.

THE AUTOMATIC TOOL PATH GENERATION FOR NUMERICAL CONTROL OF MULTIPLE FORMAT STATEMENTS ARE USED IN DESCIBING TAPE RECORDS. INCLUDES 5000 WORDS OF UPPER STORAGE FOR 1/0 BUFFER.

DATA RECORDS. INCLUDES 5000 WORDS OF UPPER STORAGE FOR 1/0 BUFFER.

THE AUTOMATIC TOOL PATH GENERATION FOR NUMERICAL CONTROL OF MULTIPLE FORMAT STATEMENTS ARE USED IN DESCIBING TAPE RECORDS. INCLUDES 5000 WORDS OF UPPER STORAGE FOR 1/0 BUFFER.

THE AUTOMATIC TOOL PATH GENERATION FOR NUMERICAL CONTROL OF MULTIPLE FORMAT STATEMENTS ARE USED IN DESCIBING TAPE RECORDS. INCLUDES 5000 WORDS OF UPPER STORAGE FOR 1/0 BUFFER.

THE AUTOMATIC TOOL PATH GENERATION FOR NUMERICAL CONTROL OF MULTIPLE FORMAT STATEMENTS ARE USED IN DESCIBING TAPE RECORDS. INCLUDES 5000 WORDS OF UPPER STORAGE FOR 1/0 BUFFER.

THE AUTOMATIC TOOL PATH GENERATION FOR NUMERICAL CONTROL OF MULTIPLE FORMAT STATEMENTS ARE USED IN DESCIBING TAPE RECORDS. INCLUDES 5000 WORDS OF UPPER STORAGE FOR 1/0 BUFFER.

THE AUTOMATIC TOOL PATH GENERATION FOR NUMERICAL CONTROL OF MULTIPLE FORMAT STATEMENTS ARE USED IN DESCIBING TAPE RECORDS. INCLUDES 5000 WORDS OF UPPER STORAGE FOR 1/0 BUFFER.

THE AUTOMATIC TOOL PATH GENERATION FOR NUMERICAL CONTROL OF MULTIPLE FORMAT STATEMENTS ARE USED IN DESCIBING TAPE RECORDS. INCLUDES 5000 WORDS OF UPPER STORAGE FOR 1/0 BUFFER.

THE AUTOMATIC TOOL PATH GENERATION FOR NUMERICAL CONTROL OF MULTIPLE FORMAT STATEMENTS ARE USED IN DESCIBING TAPE RECORDS. INCLUDES 5000 WORDS OF UPPER STORAGE FOR 1/0 BUFFER.

THE AUTOMATIC TOOL PATH GENERATION FOR NUMERICAL CONTROL OF MULTIPLE FORMAT STATEMENTS ARE USED IN DESCIBING TAPE RECORDS. INCLUDES 5000 WORDS OF UPPER STORAGE FOR 1/0 BUFFER.
0704 1674TECKOP
AVAILABLE PRIOR TO JANUARY 1962
FLOATING POINT OPTIMIZED HUNGE KUTTA
PROGRAM WITH OPTIONS GIVEN BELOW FOR
DETERMINING THE INTEGRATION INTERVAL SIZE.
SIMULATES THE INTEGRATION INTERVAL SIZE
DETERMINED FROM THE DIFFERENTIATION EQUATIONS.
DETERMINES AN INTEGRATION STEP SIZE
DEPENDING ON A HANDELED FROM CONTROL.
FOOKED FIXED STEP SIZES MAY BE USED.A MODIFICATION
OF PU.MMFLA 21398 PROGRAM C LAS OF STORAGE.

0704 1616LARNO
AVAILABLE PRIOR TO JANUARY 1962
ROCKET PROGRAM FOR THE PURPOSE OF
DETERMINING CHARACTERISTICS.
A CONVERSION BETWEEN SUBRANGE NOZZLE CONVERSES FOR INVESTED
RIS. WHICH HAS A SPECIFIC IMPULSE FOR SPECIFIED AREA.
RATES AND AMOUNT PER CENT.

0704 1617U9005
AVAILABLE PRIOR TO JANUARY 1962
NUMERICAL INTEGRATION OF UNNEARLY SPACED POINTS
DETERMINES AN INTEGRATION STEP SIZE
DEPENDING ON A HANDELED FROM CONTROL.
FOOKED FIXED STEP SIZES MAY BE USED.A MODIFICATION
OF PU.MMFLA 21398 PROGRAM C LAS OF STORAGE.

0704 1655MS1E13
AVAILABLE PRIOR TO JANUARY 1962
A 1961 PROGRAM TO MAINTAIN THE SHARE LIBRARY ABSTRACTS ON TAPE.
THE PROGRAM RECEIVE A TAPE LOADED, AN UPDATING
PROGRAM, A LISTING PROGRAM AND THE EXISTING ABSTRACTS ON TAPE.
THE TAPE IS THEN SELF-LOADING AND CAPABLE OF UPDATING,
COPYING AND LISTING ITSELF. THE LISTING MAY COVER ALL
PROGRAMS, PROGRAMS ONLY, PROGRAMS ONLY OR 709
AND 709 PROGRAMS TOGETHER. FORTRAN PROGRAMS AND COMMENTS WILL
APPEAR ON ALL LISTINGS. Requires a 4, 1415, with 2 TAPES
SOLUTION ADDRESS M/CASST, HIGH LOW, EQUAL Compare, Sense
SWITCHES AND COLUMN BINARY.

0704 1687VPCPE
AVAILABLE PRIOR TO JANUARY 1962
MULTIPLE COMPONENTS PREDICTION EQUATIONS.
F.P. PROGRAM TO EVALUATE AN EQUATION BY FITTING DATA USING
MULTIPLE CONTROL TECHNIQUE OF COMPONENT ANALYSIS.
METHOD DIFFERS FROM MULTIPLE REGRESSION IN THAT COEFFICIENTS
ARE DERIVED USING ORTHOGONAL CONTRIBUTIONS OF RESPECTIVE
TERMS OF CO, THIS SUPPRESSING EFFECTS OF CORRELATIONS AMONG
INDEPENDENT VARIABLES. AN IDENTITY-CLASSIFICATION ANALYSIS OF
CHARACTERISTIC EQ. OF MATRICES CORRELATIONS EXPRESS
RELATIONSHIP BETWEEN INDEPENDENT VARIABLES AND ORTHOGONAL
COMPONENTS.ADOPTION OF 10 OF 0561 USED AS SUBROUTINE.

0704 1714WNG02
AVAILABLE PRIOR TO JANUARY 1962
A SIMPLE RANDOM NUMBER GENERATOR
GIVEN A NORMALIZED FLOATING POINT NUMBER 2-SHIFT BETWEEN -1
AND 1, THE NUMBER 2-SHIFTS ON THE INTERVAL /+1/.1

0704 1103500C01
AVAILABLE PRIOR TO JANUARY 1962
ECK CARDS U cooperation
LOADS FILE OF STANDARD 709 COLUMN BINARY CARDS
DETERMINES DATA CONNECTION CARDS
PDP PROGRAMS A CARD READER

0704 1104414118
AVAILABLE PRIOR TO JANUARY 1962
PROCESS CONTROL COMPUTER ASSEMBLY FOR IBM 704
1961 PROGRAMS FROM IBM 1600-1701 S, I, ASSEMBLY
WITH LISTING CARDS USING THE IBM 704 FOR RUNNING ON
THE IBM 1600, IBM 702, IBM 704 AND OTHER CONFIGURATIONS OF IBM
PROCESS CONTROL COMPUTER-6.

0704 1106180377
AVAILABLE PRIOR TO JANUARY 1962
MULTIPLE DISTILLATION PROGRAM
SOLVES PL-TO-P PL-TO-P COMPONENT DISTILLATION, BUBBLE,
SURFACE AND FLASH POINT PROBLEMS FOR UP TO 3 COMPONENTS ON
IBM MACHINE.

0704 11075750C2
AVAILABLE PRIOR TO JANUARY 1962
BENEDICT-WEBN-RUSK EQUATIONS OF STATE.
APPLIES THE B-EW-NUK EQUATIONS TO THE SOLUTION OF DISTILLATION
PROBLEMS, F OR USES A SUBROUTINE IN 18027, REQUIRES A
MACHINE.

0704 11080SMCP
AVAILABLE PRIOR TO JANUARY 1962
CRITICAL PATH PROGRAMMING METHOD
THIS PROGRAM IMPLEMENTS THE ALGORITHM OF J. C. KELLEY, THAT
SERVES AS THE BASIS OF THE PROJECT CONTROL TECHNIQUE CALLED
CRITICAL PATH PROGRAMMING BY MACKLIN, ASSOCIATES. THE
ALGORITHM GENERATES A SERIES OF CHARACTERISTIC SCHEDULES
FOR A PROJECT BY ASSIGNING TO EACH ACTIVITY A COST-DURA
TION OPERATING POINT FOR EACH GENERATED SCHEDULE. FOR A
PROJECT, 704 PROGRAMS ARE REQUIRED. THE COST OF EACH
ASSOCIATED PROJECT DURATION USES 10 TAPES IN DM OPER SYS

0704 1109PMPM89
AVAILABLE PRIOR TO JANUARY 1962
INTEGER PROGRAMMING 1, 7090
CONV. OF PKIP02 FOR 7090 USING FORTRAN EN. 1967

0704 1109PMPM89
AVAILABLE PRIOR TO JANUARY 1962
INTEGER PROGRAMMING 1, 7090
CONV. OF PKIP03 FOR 7090 WHICH DOES NOT REQUIRE FORTRAN
MONITOR SYSTEM.

0704 1109PMPM92
AVAILABLE PRIOR TO JANUARY 1962
INTEGER PROGRAMMING 2, 7090
CONV. OF PKIP02 FOR 7090 WHICH DOES NOT REQUIRE FORTRAN
MONITOR SYSTEM.

0704 1109PMPM92
AVAILABLE PRIOR TO JANUARY 1962
INTEGER PROGRAMMING 2, 7090
CONV. OF PKIP03 FOR 7090 WHICH DOES NOT REQUIRE FORTRAN
MONITOR SYSTEM.

1MF 0704 PROGRAM LIBRARY ABSTRACT
******************************************************************************

0704 1108PMPM89
AVAILABLE PRIOR TO JANUARY 1962
INTEGER PROGRAMMING 1, 7090
CONV. OF PKIP01 FOR 7090 USING FORTRAN MONITOR SYSTEM.

0704 1107PMPM89
AVAILABLE PRIOR TO JANUARY 1962
INTEGER PROGRAMMING 1, 7090
CONV. OF PKIP03 FOR 7090 WHICH DOES NOT REQUIRE FORTRAN
MONITOR SYSTEM.

0704 1109PMP59P
AVAILABLE PRIOR TO JANUARY 1962
FAP ASSEMBLY PROGRAM FOR THE IBM 704
THIS PROGRAM IS WRITTEN ON THE FORTRAN SYSTEM TAPI
IT ASSEMBLES THE 709, 7090 AND 709 PROGRAMS WRITTEN IN THE
FAP LANGUAGE. CORR. 12261227.

0704 1206WEX2F
AVAILABLE PRIOR TO JANUARY 1962
FLOATING POINT EXPONENTIAL THE SUBROUTINE IS CALLED
THE EXPONENTIAL FUNCTION ARGUMENT IN THE
ACUMULATOR AND EXIT WITH THE FLOATING POINT EXPONENTIAL
IN THE ACCUMULATOR. SPACE REQUIRED 1663 COMMON.
TYPING IS 23500.

0704 1200NABC
AVAILABLE PRIOR TO JANUARY 1962
AUTOMATIC CODER, COMPATIBLE WITH SPECTRUM
SPECTRAL CODED SYSTEM WHICH SOURCE LANGUAGE INCLUDES SPECTRUM
CODING AS WELL AS STATEMENTS IN MATHEMATICAL LANGUAGE AND
ENGLISH, TRANSLATES AUTOMATIC CODE TO SPECTRUM CODE, WHICH IS THEN
ASSEMBLED, USING SPECTRUM, INCLUDES 7 SUBRoutines IN SYSTEM.
TYPICAL CODE LANGUAGE LIKE FORTRAN, WITH
RESTRICTIONS TO THE STANDARDS IN THE SOURCE LANGUAGE.
CONTAINS DATA PROCESSING PACKAGE WITH GENERAL SUBROUTINE LIG.
OBJECT PROGRAM ON BINARY CARDS WITH SPECTRUM LISTING.

0704 1202NCUSC
AVAILABLE PRIOR TO JANUARY 1962
SHARE CATALOG UPDATES, LISTS, 7041 PROGRAM.
REQUIRES AN LINK WITH ADV, MOLD, MS-704, AND 2 TAPES.

1. UPDATE THE CATALOG FILE ON TAPE WHICH INCLUDES CARD.
2. SEQUENCE CARDING IN INPUT CATALOG CARDS BEFORE UPDATING.
3. LIST THE CATALOG BY THE CLASSIFICATION CODE.
4. LIST THE CATALOG ITEMS FOR ANY INSTALLATION.
5. IF DESIRED, JUST THE TITLES MAY BE LISTED.
0704 1211TPPR

AVAILABLE PRIOR TO JANUARY 1962

PROGRAM TO GENERATE 1401 T/P PROG. ON OUTPUT TAPES.
TO MINIMIZE OPERATOR ATTENTION IN 1401 PRINT OPERATION FROM
OUTPUT TAPE THROUGH PROGRAMMED 1401 INSTRUCTIONS
WRITTEN ON THE TAPE AT THE TIME OF 1401 COMPUTATION. THE 1401
TAPE-TO-PRINT INSTRUCTIONS PRECEDE ANY OUTPUT INFORMATION,
AND THE PRINT OPERATION REQUIRES ONLY THE MOUNTING OF THE
TAPE AND PRESSING THE LOAD TAPE BUTTON.

0704 1221AE46

AVAILABLE PRIOR TO JANUARY 1962

INTEGRATION WITH CONTROLLED ERROR

AA46E IS DESIGNED TO BE USED IN CONJUNCTION WITH AN
INTEGRATION SUBROUTINE AA44E IF DESIRED TO PROVIDE A
NUMERICAL SOLUTION OF AN NTH ORDER SYSTEM OF LINEAR AND/OR
NON-LINEAR DIFFERENTIAL EQUATIONS EXPRESSED AS A SYSTEM OF N
FIRST ORDER EQUATIONS. THE LOCAL ERROR GENERATED BY THE
NUMERICAL PROCESS IS CONTROLLED BY ADJUSTING THE INTEGRATION
STEP SIZE BASED ON THE RELATIVE ERROR AS ESTIMATED BY
EXTRAPOLATION TO ZERO STEP SIZE.

0704 1223AA47H1

AVAILABLE PRIOR TO JANUARY 1962

SECOND, THIRD, AND FOURTH ORDER RUNGE-KUTTA INTEGRATION

AA47H IS A FORTRAN SUBROUTINE DESIGNED TO BE USED IN
CONJUNCTION WITH AA44E TO PROVIDE A SECOND, THIRD, OR FOURTH
ORDER RUNGE-KUTTA SOLUTION OF AN NTH ORDER SYSTEM OF LINEAR
AND/OR NON-LINEAR DIFFERENTIAL EQUATIONS EXPRESSED AS A
SYSTEM OF N FIRST ORDER EQUATIONS.

0704 1234AA48E2

AVAILABLE PRIOR TO JANUARY 1962

WEGSTEIN ITERATION

AA48E IS AN EQUATION OF THE FORM X-FIX/AA AA48E WILL
FIND A VALUE FOR X WHICH WILL PROVIDE A SPECIFIED ACCURACY
IN EITHER A RELATIVE OR ABSOLUTE SENSE.

0704 1244AAC001

AVAILABLE PRIOR TO JANUARY 1962

A GENERAL PROGRAM FOR SYSTEMS EVALUATION

GIVEN A DESCRIPTION OF THE BLOCK DIAGRAM OF A SYSTEM AND THE
TRANSFER FUNCTIONS OF EACH COMPONENT OF THE SYSTEM, THIS
COMPLETE PROGRAM COMPUTER THE TRANSFER FUNCTION OF THE SYSTEM
AND CALCULATES THE ATTENUATION AND PHASE ANGLE FOR GIVEN
VALUES OF FREQUENCY. SIMPLE FEEDBACK LOOPS ARE PERMITTED IN
THE SYSTEM. THE PROGRAM AS SUBMITTED IS DESIGNED FOR A 32K
MEMORY.
ABRAC 01

104 Nuclear Code

(1) Code Originated by:
Westinghouse - Bettis Plant

(2) Computer:
704

(3) Description of Code:
ABRAC 01 is a three-dimensional few-groups neutron diffusion program which treats the effects of water moderator density changes (resulting from flow variations and boiling) on neutron flux distributions and depletion. Thermal and hydraulic calculations performed within the code limit its applicability to water-cooled and moderated cores having one upflow coolant pass. ABRAC 01 is essentially the DRACO 1 program with a thermal and hydraulic calculation added immediately after the power and flux normalization routine and just prior to the depletion routine.

(4) Restrictions or Limitations:
Maximum number of mesh parallelepipeds is 605 or 4570 for machines of 16K or 32K words of core storage, respectively. Ten tape units are required.

(5) Approximate Performance:
For a core represented by a 16x16x26 mesh (two group), the running time might be from 1.5 to 2.0 hr. per iteration. Three to four iterations may be required.

(6) References:
1. W. M. Jacobs, T. J. Lawton, S. H. Mesarov, J. R. Parrish,

ABRAC - An IBM-704 Three Dimensional Nuclear -


APC01

704 Nuclear Code

(1) Code Originated by:
Westinghouse - Bettis Plant

(2) Computer:
704

(3) Description of Code:
The APC01 code processes the flux tapes from a PDQ01 problem and its adjoint. The integrals

\[ \int_{V} \phi^{*} \delta \lambda \]

are obtained in an x-y geometry for all compositions supplied and for all possible combinations of groups i and j.

(4) Restrictions or Limitations:
A 32K memory is required. The flux and adjoint flux calculations must correspond as far as geometry, mesh structure, groups, and number of compositions.

(5) Approximate Performance:
Running time to process the flux tapes from a two-group, 30 x 30 PDQ01 problem and its adjoint is approximately 1.2 minutes with no pointwise product edits, and approximately 4.8 minutes when all pointwise product edits are included.

(6) References:

BINT0

704 Nuclear Code

(1) Code Originated by:
Westinghouse - Bettis Plant

(2) Computer:
704

(3) Description of Code:
BINT0 calculates steady state temperatures in a one- or two-pass cylindrical reactor core. It requires as input the radial and axial power distributions and rules for combining them into three-dimensional power distributions, local peaking factors, hot-channel factors, and geometric data.

(4) Approximate Performance:
5 minutes.

(5) References:


4. NCG Newsletter No. 9, p. 4.
CANDLE 704 Nuclear Code

(1) Code Originated by: Westinghouse - Bettis Plant

(2) Computer: 704

(3) Description of Code: One space dimension and time few-group depletion code for rectangular, cylindrical, and spherical geometry. Fast group constants are computed from effective one-velocity microscopic cross sections. Thermal microscopic cross sections and self-shielding factors are supplied as input data. The WANDA calculation is used to determine the corresponding eigenvalues and flux shape. Criticality may be maintained by varying the transverse buckling, a homogeneous poison, or the location of a boundary between a poisoned and unpoisoned region. The flux is normalized to a specified power and assumed to be constant for a specified length of time. The isotopic densities are recomputed at the end of this time using the normalized flux. A maximum error calculation is optimal at each time step.

(4) Restrictions or Limitations: Max of 25 regions and 350 mesh intervals with either two or four groups. At most 10 time steps may be done automatically. Only the uranium, plutonium, and fissile product chains along with two burnable poisons are considered time dependent with a maximum of 30 elements in all. Code requires 6K core, four tape units, and one drum unit.

(5) Approximate Performance: From 15 min. to 4 hrs. Average of 30 min.

(6) References:

CEPST 704 Nuclear Code

(1) Code Originated by: Combustion Engineering, Inc.

(2) Computer: 704

(3) Description of Code: This program is designed to solve the one-dimensional, monoeenergetic P1 approximation to the transport equation in cylindrical geometry. The cylinder is assumed to be infinitely long and symmetric with respect to rotation about the z-axis. The external boundary condition may be specified as reflecting or vacuum or as a special type of cell condition. Anymaterial region of the problem may be specified as having all zero cross sections, that is, an interval void. An external isotropic source may be specified by region or point wise. The code utilizes the first four spherical harmonics of the scattering cross section.

(4) Restrictions or Limitations: Problems are limited to a maximum of 150 spatial mesh points and 10 material regions. Code performance is most satisfactory for problems with radii of 5 or fewer mean free paths.

(5) Approximate Performance: Maximum problem runs in approximately 1.5 minutes.

(6) References:

COFIT 704 Nuclear Code

(1) Code Originated by: Westinghouse - Bettis Plant

(2) Computer: 704

(3) Description of Code: Fits by least squares the curve \( y = A \cos(B \log(x)) \) to from 4 to 500 points of observed data, computing the parameters \( A, B, C \), and the standard deviations of the estimates of \( A, B, C \). It is also possible to investigate the error in a region about the final values of \( A, B, C \). By computing the sum of the squares of the residuals at a series of points in the neighborhood.

(4) Approximate Performance: 500 point problem \( \times 8 \) min.

(5) References:

COGENT 704 Nuclear Code

(1) Code Originated by: Combustion Engineering, Inc.

(2) Computer: 704

(3) Description of Code: The COGENT Code solves the one-dimensional neutron diffusion equation for 30 coupled energy groups with an external neutron source. The code will handle slab, cylindrical or spherical geometry. COGENT provides for a maximum of ten isotopes and six scattering matrices. The external source may be specified region-wise constant, group-wise constant, region-wise by group, or point-wise by group. As output, in addition to the point-wise fluxes the code provides flux weighted macroscopic constants.

(4) Restrictions or Limitations: Problems are limited to a maximum of 101 spatial mesh points and 4 material regions. IBM 704, 3 tape units, 1 drum unit.

(5) Approximate Performance: Average problem requires approximately 40 minutes.

(6) References:
1. CEND NPC-10.

CURE 704 Nuclear Code


(2) Computer: 704

(3) Description of Code: Solves age-diffusion equations for neutron flux distribution in a reactor for r-x, r-y or x-y geometry. Multiplication of the reactor is computed. Includes calculation of averaged three-group macroscopic cross-sections from physical compositions according to prescriptions of B. W. Dunshee. Cylindrical boundaries, variable mesh spacing, and deletion of points are permitted in the spatial mesh. Several versions are available from KAPL which differ in speed, use of machine, size of problem, and input.

(4) Restrictions or Limitations:
1. The code permits at most 90 compositions and allows about 700 space points for an 8K memory.

(5) Approximate Performance: 3 min./source iteration for 700 pts., 3 groups.

(6) References:

DRACO 704 Nuclear Code

(1) Code Originated by: Westinghouse - Bettis Plant

(2) Computer: 704

(3) Description of Code: Depletion version of TKO

(4) References:

EUPRUS - 3 and DAEDALUS 704 Nuclear Code

(1) Code Originated by: Westinghouse - Bettis Plant

(2) Computer: 704

(3) Description of Code: EUPRUS - 3 calculates the one-dimensional spatial density of neutrons slowing-down past a given energy in an infinite homogeneous medium consisting of hydrogens and one other isotope with arbitrary mass and energy-dependent differential elastic and absorption cross-sections. DAEDALUS determines the corresponding spatial distribution of angular integrals of an

(Continued on next page)
arbitrary function times the vector flux density. Spatial moments of all density functions are furnished directly. The neutron source may be nonmonenergetic with either isotropic or nonisotropically directed distributions, or else the source may be that from deuterium- and tritium-deuterium reactions.

Restrictions or Limitations:

A 32K core memory is required, and 5 tape units are required.

References:


EXFIT

Code Originated by:

Westinghouse - Bettis Plant

Computer:

704

Description of Code:

Fits a set of observed data, \( y_n \), to a curve of the form \( y \sim A e^{ax} \) where each \( y_n \) value may be weighted by a \( w_n \). It is possible to compute the parameters \( A \) and \( B \) and the estimate of the error in each parameter. The maximum allowable number of points in the set is 500.

Restrictions or Limitations:

Requires a 4096 word core. No drums or tapes are used. No account is taken of "wild" points and their inclusion may result in a poor fit.

Approximate Performance:

Two minutes for 30-50 point problem.

References:


FLAK

Code Originated by:

Los Alamos Scientific Laboratory

Computer:

704

Description of Code:

Numerical solution of diffusion equation for slab, cylinder or spherical geometry. Hydrogen, inelastic scattering, continuous slowing down.

Approximate Performance:

1-1/2 minutes

References:

1. LA-2181

FLER

Code Originated by:

GE Nuclear Atomic Power Lab.

Computer:

704

Description of Code:

FLER will solve the three-group, two-dimensional neutron diffusion equation in a triangular coordinate system. Up to 14,000 mesh points are allowed. The outer boundary of the problem must be a parallelogram. A special 120 degree periodic boundary condition is allowed on two of the sides. Available boundary conditions are flux zero, current zero, and a logarithmic boundary condition. Few-group cross sections are calculated within the code. Flux iteration is accomplished by a "best" line relaxation technique.

Restrictions or Limitations:

A 32K memory is required, as well as 7 tapes and 4 drums.

Approximate Performance:

Approximate running time for a problem is about 40 minutes per 1000 points. (Continued on next column)
HECTIC

1. Code Originated by: Westinghouse - Bettis Plant
2. Computer: 704
3. Description of Code: HECTIC is a computer program for calculating heat transfer rates and temperatures in the fuel elements of typical gas-cooled nuclear reactors. Effects of turbulent interchange between flow passages are considered. The computation procedure amounts to a "nodal" or "lumped parameter" type calculation.
4. Limitations or Restrictions: An 8K memory is required.
5. Approximate Performance: A full-size run requires approximately 35 minutes.
6. References:

HERD

1. Code Originated by: Westinghouse - Bettis Plant
2. Computer: 704
3. Description of Code: The HERD codes furnish a numerical approximation to the solution of the one-dimensional, one-velocity neutron transport equation (scattering and sources assumed to be isotropic) in slab geometry using the method of discrete ordinates. Let F(x, θ), F(θ), Fi, F(0, θi), A be the boundaries. The HERD codes differ in the boundary conditions imposed:
   - HERD 1 F(0, θ) = F(0, θ) and A is an axis of symmetry.
   - HERD 2 F(0, θ) is specified for θ = 0 and A is an axis of symmetry.
   - HERD 3 F(0, θ) is specified for θ = 0 and F(0, θ) + 0 for |θ| ≤ Cn.

The primary purpose of HERD 2 and 3 is to compute blackness coefficients.
4. Limitations or Restrictions: Either a 16K or 32K core memory may be used. Limitations on the size problem which may be run depend upon the size of core used, and depend on the number of angles at which the vector flux may be calculated. Details are given on page 2 of Reference 6 (1).
5. Approximate Performance: The average running time for most problems is between 0.5 and 5.0 minutes.
6. References:
   1. A. Hagman, "HERD 1, 2, and 3 - IBM-704 Codes Used to Solve the One-Dimensional, One-Velocity Transport Equation with Isotropic Scattering", WAPD-TM-111, January, 1953.
MUFT 4 704 Nuclear Code

(1) Code Originated by: Westinghouse - Bettis Plant
(2) Computer: 704
(3) Description of Code: Computes the energy distribution of neutrons having a given Fourier code in an infinite medium. MUFT IV is essentially the same as the 650 nuclear code MUFT III. Modifications incorporated into MUFT IV were designed to improve the treatment of non-hydrogenous moderation, and to take into consideration the effect of resonance self-shielding on the production of fission neutrons.
(4) Restrictions or Limitations: 100 or less lethargy groups averaged over 3 few groups; 15 or less isotopes; any value for the total buckling; one approximation per problem.
(5) Approximate Performance: 11 seconds.
(6) References:

PDQ - 5 704 Nuclear Code

(1) Code Originated by: Westinghouse - Bettis Plant
(2) Computer: 704
(3) Description of Code: The program solves the few-group neutron diffusion equations for one to four lethargy groups over a rectangular region of the (x, y) or (r, z) plane. Variable mesh intervals are allowed. The inner iterations are performed by the method of over-relaxation and include a special method of determining the over-relaxation factor for each group.
(4) Restrictions or Limitations: Outer boundary of mesh must be rectangular and material interfaces may occur only on mesh lines. Maximum of 35 different materials, but each may appear in many regions of the mesh. Maximum of 1250 to 6500 mesh points, depending upon core storage available. Requires one drum unit and six tape units.
(5) Approximate Performance: Less than 1 hour for a two-group 2500-point problem.
(6) References:

PECAN 704 Nuclear Code

(1) Code Originated by: Aerojet - General Nuclear
(2) Computer: 704
(3) Description of Code: The PECAN Cycle analysis code calculates various thermodynamic cycle data for gas turbine power plants, based on a given set of design parameters. The calculations enable optimization of a specified power plant design to a major requirement such as weight, economy, or output.
(4) The code is restricted to the use of a gaseous working fluid within a temperature range of 3000°K to 2100°K, but is otherwise general.
(5) References:

P1MC 704 Nuclear Code

(1) Code Originated by: Westinghouse - Bettis Plant
(2) Computer: 704
(3) Description of Code: One-Dimensional P1 multigroup cycle analysis code calculates various thermodynamic cycle data for gas turbine power plants, based on a given set of design parameters. The calculations enable optimization of a specified power plant design to a major requirement such as weight, economy, or output.
(5) References:
Letter, 7/31/58.

POLYPHEMUS 704 Nuclear Code

(1) Code Originated by: Westinghouse - Bettis Plant
(2) Computer: 704
(3) Description of Code: A Monte Carlo study of the penetrations of monoenergetic, mono-directional, isotropic source neutrons from 1 mev to 10 mev through finite water slabs. The program was designed to provide two groups of shielding parameters: the neutron dose rates and dose buildup factors for the several energies. Because it was primarily a production code, emphasis was placed on speed rather than completeness of information.
(5) Approximate Performance: 3 minutes per 1000 histories
(6) References:
1. NCG Newsletter No. 5, page 2.
(1) Code Originated by: Westinghouse - Bettis Plant

(2) Computer: 704

(3) Description of Code:
This program is used to calculate resonance escape probabilities using the procedure described by Adler, Hines, and Nordheim. The code allows three types of reactor composition: homogeneous - metal fuel and xenon used - oxide fuel. The code also calculates the effective resonance integral for each composition using either the narrow resonance (NR), or the narrow resonance, intrinsic mass approximation (NRMA).

(4) Restrictions or Limitations:
- GE-Knolls Atomic Power Laboratory
- Westinghouse - Bettis Plant

(5) References:

QUERY

PROG and JET

704 Nuclear Code

(1) Code Originated by: GE-Knolls Atomic Power Laboratory

(2) Computer: 704

(3) Description of Code:
This program requires either a 16K or 704 computer. The core to be studied may have as many as 63 axial fuel regions, 5 radial fuel regions, and, depending on machine size, up to 3750 or 4000 interior axial mesh points. As many as 62 of these regions and 3200 or 6000 of the radial mesh rectangles may contain fuel.

(4) Restrictions or Limitations:
- IBM-704
- Binary deck.

(5) References:

RANCH

704 Nuclear Code

(1) Code Originated by: GE-Knolls Atomic Power Laboratory

(2) Computer: 704

(3) Description of Code:
The RANCH code numerically solves the one-dimensional, one-velocity neutron transport equation in slab geometry. The source is assumed to be isotropic, but anisotropic scattering is permitted. The method of discrete ordinates is used with the iteration process accelerated by overrelaxation to obtain the solution.

(4) Restrictions or Limitations:
A 32K memory and one tape unit are required. Up to 50 regions are permitted, and the number of mesh points permitted depends upon the number of angles used, and varies from 1,250 points for 4 angles to 833 points for 12 angles.

(5) Approximate Performance:
As an 8 angle, 100-point problem requiring 40 iterations for convergence took 3.1 minutes.

(6) References:
2. Material Available:
- WAPD-TM-268
- Binary deck.

Note: The information above was abstracted from WAPD-TM-268.

BEM

704 Nuclear Code

(1) Code Originated by: GE-Knolls Atomic Power Laboratory

(2) Computer: 704

(3) Description of Code:
This code is a version of CURE which differs from it in that (1) it permits interior (region) and exterior boundaries to run diagonally, as well as horizontally and vertically in the mesh, (2) it does not permit deletion of points, (3) it will presently handle only (x, y) geometry. It is required that an additional index be included for each combination of 2 different compositions along an interior diagonal line.

(4) Approximate Performance:
3 min./source iteration for 700 points, 3 groups.

(5) References:

The SET Codes

704 Nuclear Code

(1) Code Originated by: Westinghouse - Bettis Plant

(2) Computer: 704

(3) Description of Code:
The SET codes (SET 02 and SET 03) obtain a numerical solution to the problem of stresses in a pressure vessel with an ellipsoidal head. The codes are based on a finite-difference approximation to the Love-Weiner equations which are the basis of the bending theory of thin shells. The SET 02 code uses a direct method to solve the system of difference equations while the SET 03 code uses an iterative method.

(4) Restrictions or Limitations:
A typical problem in run on the SET 02 code much faster than on the SET 03 code. On the other hand, the SET 02 code is subject to round off errors when the mesh is sufficiently refined, while the method used in the SET 03 code is inherently "stable." A 32K core memory is required 1 as well as 2 tapes. No drum space is required.

(5) References:
**SIMPL - 2**

704 Nuclear Code

(1) Code Originated by: Westinghouse - Bettis Plant

(2) Computer: 704

(3) Description of Code:

The program is a neutron diffusion code which solves the neutron transport equations in the stationary case, using the $S_3$ method (LA-1991), and assuming isotropic scattering and one-dimensional geometry. The present version of the code has been modified to reduce the number of iterations required in a given problem by better than a factor of two. The code is easily applicable to any $S_n$ approximation of reasonable order (constant for $n = 1, 4, 9$, and $S$ supplied), to any one-dimensional geometry (plane, spherical or infinite cylindrical in symmetry), and to the three eigen-values: reactivity, outer dimension, or exponential rate. The program was written using the Los Alamos Flux Code System (FLUCO).

(4) Restrictions or Limitations:

a. The computer program is written in Fortran 4.

b. The number of mesh intervals may not exceed 98 in the $r$ direction.

c. The diameter of the cylinder is assumed to contain a source of gamma rays which is surrounded by cylindrical shell shielding, and above which are plane-slab shields. The program is capable of computing the uncollided gamma flux at a point outside a right circular cylinder which is surrounded by cylindrical shell shields and above which are plane-slab shields. The cylinder is assumed to contain a source of gamma radiation which varies in the radial and angular directions. Field points may be located in a plane through the axis of the cylinder. The method of integration used is three-dimensional Gaussian quadrature.

d. The code’s primary applications are expected to be in radiation heating problems and in calculating gamma dose rates.

(5) Approximate Performance:

Typical computing and editing time for a 20 point problem, in which there are 10 side and 10 top shields, is four minutes per energy level.

(6) References:


SIMPLO - 1

704 Nuclear Code

(1) Code Originated by: Westinghouse - Bettis Plant

(2) Computer: 704

(3) Description of Code:

Determine $S_n$, $k_x$, or group fluxes due to source in multiplying medium. Solves inhomogeneous $P_2$ or double $P_1$ one-group problem with proper choice of parameters.

(4) Restrictions or Limitations:

a. $P_2$ group, 25 regions, 240 mesh intervals.

b. 1-4 groups, 30 regions, 300 mesh intervals.

c. 1-3 groups, 40 regions, 400 mesh intervals.

d. The number of top shields cannot exceed 60.

e. The number of side shields cannot exceed 30.

(5) Approximate Performance:

1 minute.

(6) References:


SPAN - 2

704 Nuclear Code

(1) Code Originated by: Westinghouse - Bettis Plant

(2) Computer: 704

(3) Description of Code:

The SPAN - 2 code calculates the uncollided gamma flux at a point outside a right circular cylinder which is surrounded by cylindrical shell shields and above which are plane-slab shields. The cylinder is assumed to contain a source of gamma radiation which varies in the radial and angular directions. Field points may be located in a plane through the axis of the cylinder. The method of integration used is three-dimensional Gaussian quadrature.

The code’s primary applications are expected to be in radiation heating problems and in calculating gamma dose rates.

(4) Restrictions or Limitations:

a. The number of mesh intervals may not exceed 98 in the $r$ direction or 53 in the $z$ direction. The total number of mesh intervals may not exceed 600.

b. The number of energy levels cannot exceed 30.

c. The number of side shields cannot exceed 30.

d. The number of top shields cannot exceed 30.

e. There may be 1, 2, or 3 regions inside the core. The sum of thicknesses of these regions must be equal to the core radius.

(5) Approximate Performance:

Typical computing and editing time for a 20 field point problem, in which there are 10 side and 10 top shields, is four minutes per energy level.

(6) References:


SOFOCATE

704 Nuclear Code

(1) Code Originated by: Westinghouse - Bettis Plant

(2) Computer: 704

(3) Description of Code:

By solving the Wigner-Wilkins differential equations, the code determines the neutron spectrum in a homogeneous mixture where the absorption cross sections of the constituents may vary arbitrarily with energy. The code will also compute the macroscopic absorption cross section, $\Sigma_a$, the flux averaged diffusion constant and macroscopic fission cross sections. In addition, any desired function may be averaged over the resultant flux even though it may not be present in the mixture.

(4) Restrictions or Limitations:

Energy limit is 2.0 eV, only two choices of mesh.

(5) Approximate Performance:

30 seconds.

(6) References:


EPIC - 1

704 Nuclear Code

(1) Code Originated by: Westinghouse - Bettis Plant

(2) Computer: 704

(Continued on next page)
(3) Description of Code: The SPGC-1 code calculates the fast-neutron dose rate or the thermal neutron flux at a point outside a right circular cylindrical core which is surrounded by cylindrical shell shields and is capped by plane slab shields. The fast neutron attenuation kernel is empirical and is in the form of a linear combination of single exponentials which has been fitted to the experimental fast-neutron dose rate distribution in pure water. Empirical neutron removal cross-sections are used to represent the attenuation by shells of non-hydrogenous materials located in the water.

(4) Restrictions or Limitations: A 32K core memory is required. Other limitations are those of the SPAN-2 code.

(5) Approximate Performance: Typical computing and editing time for a 20-field-point problem, in which there are 10 side and 10 top shields, is 6.5 minutes.


STDEV-3
704 Nuclear Code

(1) Code Originated by: Westinghouse - Bettis Plant
(2) Computer: 704 (FORTRAN)
(3) Description of Code: STDEV-3 is a computer program designed for the thermal analysis of a pressurized water reactor during steady-state operation. It performs a complete steady-state, parallel channel thermal analysis of a rectangular water channel core with a plate-type fuel element.
(4) Restrictions or Limitations: A 16K memory is required, as well as three tape units and a logical drum.
(5) Approximate Performance: Typical computing time for a two-pass core containing a hot channel in each pass is 6.5 minutes.
2. WAPD-TM-213.

Note: The information given above was abstracted from Reference 1.

TEMP - 1
704 Nuclear Code

(1) Code Originated by: Westinghouse - Bettis Plant
(2) Computer: 704
(3) Description of Code: The TEMP - 1 program solves the difference form of the one-dimensional transient heat-conduction for a body with an arbitrary initial temperature distribution and either the temperature, its normal gradient, or a combination of the two specified on the boundaries. An implicit difference scheme is used. The thermal stresses resulting from the temperature distribution are then obtained by a rigorous application of the analytical stress expressions of Reference 6 (6) below.
(4) Restrictions or Limitations: The size of the core memory required is not given in Reference 6 (6), but it is believed to be 32K. The program provides for minimum of 7 and a maximum of 251 mesh points which may be distributed over a minimum of 3 and a maximum of 25 regions.
(5) Approximate Performance: The solution of a 41-point problem requires about 5 seconds of computer time per time step.

TRIP - 1
704 Nuclear Code

(1) Code Originated by: Westinghouse - Bettis Plant
(2) Computer: 704
(3) Description of Code: TRIP - 1 is a group diffusion code. Its primary mission was to solve the angular flux equations in X-Y geometry. Only one-group cell problems are treated. The cell is assumed to be rectangular, with regionwise constant cross-sections. The source is isotropic and regionwise flat. Anisotropic scattering is dealt with (assuming the same magnitude of cross-section) in a function of the particle flux at a point outside a homogeneous cylinder containing a uniform isotropic source distribution, assuming that the attenuation of the particles is exponential, both within the cylinder as well as in the attenuating shells or slabs.
(4) Approximate Performance: About 2000 cross-sections for a body size of 2800 seconds, where N is number of cases.

WAP M and WU
704 Nuclear Code

(1) Code Originated by: Westinghouse - Bettis Plant
(2) Computer: 704
(3) Description of Code: The code is designed to compute the uncollided particle flux as a function of the distance from a homogenous cylinder containing a uniform isotropic source distribution, assuming that the attenuation of the particles is exponential, both within the cylinder as well as in the attenuating shells or slabs.
TURBO

(1) Code Originated by:
Westinghouse - Bettis Plant

(2) Computer:
704

(3) Description of Code:
Two space dimensions and time version of CANDLE for x-y (TURBO 1) and x and r-z (TURBO 2 and 4) geometry. Otherwise same as CANDLE except that the PDQ spatial calculation is used. Maximum space calculation is TURBO-3 for x-y or TURBO-6 for r-z.

(4) Restrictions or Limitations:
Max of 35 compositions. Number of mesh points limited by size of core according to the number pairs XK-2050, 14K-3782, 52K-1891; with a minimum of 8172 words of core storage. Automatically calculates one time step with provision for continuing later. No automatic criticality search is provided. Also requires ten tape units and one drum unit.

(5) Approximate Performance:
Approximately 1.5 hours per time step.

(6) References:

TURBO III

(1) Code Originated by:
Westinghouse - Bettis Plant

(2) Computer:
704

(3) Description of Code:
Steady state transmission and stresses in anisotropically solid or hollow bodies.

(4) References:
1. Fuller (T. M.). 2. ADD-57-6 and ADD-58-12 describing the program are available with the program from IBM.

TUT - 75

(1) Code Originated by:
Westinghouse - Bettis Plant

(2) Computer:
704

(3) Description of Code:
The TUT - 75 code provides a one-energy model, a means of calculating a regenerative distribution of capture probabilities in a two-dimensional quarter-cell. The method used is the Monte Carlo method, in which neutron histories are simulated by the code and then used to provide estimates for the integrals which define the capture probabilities.

(4) Restrictions or Limitations:
A 32K core memory is required. As many as 32 regions can be treated, all of different material contents; however, the content of each region must be uniform. The number of neutron histories must be less than or equal to 1000.

(5) Approximate Performance:
Running times may be from one to two hours. A method of estimating the time required is given in the reference cited below.

(6) References:

WANDA 2, 3

(1) Code Originated by:
Westinghouse - Bettis Plant

(2) Computer:
704

(3) Description of Code:
Solves the four-group diffusion equation in one space dimension for rectangular, cylindrical, or spherical geometry by setting either the flux or its derivative to zero on the boundaries. The parameters must be continuous within a region, but may have a finite discontinuity at the interfaces between regions. The mesh width must be constant within a region. An initial source guess is required to start the iteration process. Convergence may be defined either by a percentage deviation in the eigen value or by a percentage deviation between successive source vectors.

(4) Restrictions or Limitations:
Requires an 8K core memory, 1 drum unit, and 1 tape unit.

(5) Approximate Performance:
1-1.5 minutes, average 3 minutes.

(6) References:

WANDA 4

(1) Code Originated by:
Westinghouse - Bettis Plant

(2) Computer:
704

(3) Description of Code:
An improved version of WANDA - 3 which eliminates use of the drum unit and provides an automatic extrapolation procedure to accelerate convergence of the iteration process.

(4) Restrictions or Limitations:
An 16K core memory is required as well as one to four tape units.

(5) References:
WB-TSG-L 704 Nuclear Code

(1) Code Originated by:
Westinghouse - Biglerville Plant

(2) Computer:
704

(3) Description of Code:
Compute in one-dimensional form the tangential, axial, and radial thermal stresses for cylinders with internal heat generation.

(4) Approximate Performance:
20 minutes.

(5) References:
3. NCG Newsletter No. 5, p. 5.

2DXY 704 Nuclear Code

(1) Code Originated by:
University of California, Radiation Lab.

(2) Computer:
704

(3) Description of Code:
The 2DXY program solves the homogeneous or inhomogeneous multi-group neutron diffusion equation for slabs, cylinders or spheres. At maximum of 10 materials, 30 regions (or zones) may be used. A higher order differencing is used for the Laplacian and a general transfer matrix is permitted.

(4) Approximate Performance:
10 minutes.

(5) References:

Material Available:
1. Binary Editor Deck (7090).
2. FLOCO D.F. Binary Deck (7090).
3. 2DXY Deck (7090).
4. Sample Problem Input Deck (7090).
5. AGN TM-392.

Notes:
1. The above information was taken from Reference 3.
2. This code was contributed through the Argonne Code Center.

The binary editor program referred to above is essentially a compatibility package for the 7090.

Material Available:
1. Binary Editor Deck (7090).
2. FLOCO D.F. Binary Deck (7090).
3. 2DXY Deck (7090).
4. Sample Problem Input Deck (7090).
5. AGN TM-392.

Notes:
1. The above information was taken from Reference 3.
2. This code was contributed through the Argonne Code Center.

The binary editor program referred to above is essentially a compatibility package for the 7090.
**GUIDE**

**PROGRAM WRITE-UP ABSTRACT**

**INDICATIVE CODE**

AF-003-0

**PROGRAM NAME**

CHANGE CARD LOAD

**PURPOSE**

To load program cards in memory in the same manner as the standard lower load program. Also, to allow special patch cards to be loaded as if they were normal instruction cards.

**MACHINE**

(Continued on next column) of tape input and output. The sub-routines are designed primarily to process tapes using the HQ USAF tape identification system, but tapes lacking headers and trailers may be processed. The major parts of the package are:

a. Input/output macros to read a tape, write a tape, read-while-write a tape, read and block blocked records, and block-up and write blocked records.

b. A sub-routine (IDENT) that provides for TRA operations, output tape labelling and input tape label verification.

c. A sub-routine (IDWCP) that in addition to the IDENT functions includes a check point routine. Check points are taken automatically at EOF but may be taken at any other time desired. Provision is made for program interstage.

d. A restart program for use with IDWCP. This is a separate program that enables you to restart to any check point taken by IDWCP. The routine checks tape labels, today's data, repositions tapes, and restores memory and SBI's 01-13. Since the restart begins with memory cleared it is useful in situations where long runs are interrupted.

**MACHINE**

(Continued on next column)

April 1958, Bulletin 97 - 45

**GUIDE**

**PROGRAM WRITE-UP ABSTRACT**

**INDICATIVE CODE**

AF-012-0

**PROGRAM NAME**

MEMORY PRINT OUT

**PURPOSE**

To punch out program decks incorporating change cards to cut down the size of program decks and serial number cards in the decks. This removes the danger of change cards getting out of sequence. It has an advantage over IBM's Punch Memory 51 utility program in that control cards need not be made to designate memory to be punched. It will also punch out a greater portion of memory than Punch Memory 51.

**MACHINE**

(Continued on next column) of tape input and output. The sub-routines are designed primarily to process tapes using the HQ USAF tape identification system, but tapes lacking headers and trailers may be processed. The major parts of the package are:

a. Input/output macros to read a tape, write a tape, read-while-write a tape, read and block blocked records, and block-up and write blocked records.

b. A sub-routine (IDENT) that provides for TRA operations, output tape labelling and input tape label verification.

c. A sub-routine (IDWCP) that in addition to the IDENT functions includes a check point routine. Check points are taken automatically at EOF but may be taken at any other time desired. Provision is made for program interstage.

d. A restart program for use with IDWCP. This is a separate program that enables you to restart to any check point taken by IDWCP. The routine checks tape labels, today's data, repositions tapes, and restores memory and SBI's 01-13. Since the restart begins with memory cleared it is useful in situations where long runs are interrupted.

**MACHINE**

(Continued on next column)

April 1958, Bulletin 97 - 45

**GUIDE**

**PROGRAM WRITE-UP ABSTRACT**

**INDICATIVE CODE**

AF-011-0

**PROGRAM NAME**

TAPE PRINT OUT

**PURPOSE**

To accomplish a transformation of data from tape to tape in a manner facilitating a more efficient visual interpretation of the data, when listed.

**MACHINE**

(Continued on next column) of tape input and output. The sub-routines are designed primarily to process tapes using the HQ USAF tape identification system, but tapes lacking headers and trailers may be processed. The major parts of the package are:

a. Input/output macros to read a tape, write a tape, read-while-write a tape, read and block blocked records, and block-up and write blocked records.

b. A sub-routine (IDENT) that provides for TRA operations, output tape labelling and input tape label verification.

c. A sub-routine (IDWCP) that in addition to the IDENT functions includes a check point routine. Check points are taken automatically at EOF but may be taken at any other time desired. Provision is made for program interstage.

d. A restart program for use with IDWCP. This is a separate program that enables you to restart to any check point taken by IDWCP. The routine checks tape labels, today's data, repositions tapes, and restores memory and SBI's 01-13. Since the restart begins with memory cleared it is useful in situations where long runs are interrupted.

**MACHINE**

(Continued on next column)

April 1958, Bulletin 97 - 41

**GUIDE**

**PROGRAM WRITE-UP ABSTRACT**

**INDICATIVE CODE**

AF-012-0

**PROGRAM NAME**

CARD TO TAPE LOAD

**PURPOSE**

To create, from card input, blocked or unblocked records of any length on tape.

(Continued on next page)
GUIDE

PROGRAM WRITE-UP ABSTRACT

INDICATIVE CODE
AP-011-0

PROGRAM NAME
Square Table Look-up with Function

PURPOSE:
A set of four macro-instructions is provided to be used for table look-up
operations. Two macros are merely for argument verification and the
other two are for both argument verification and function extraction. Two
macros are for use when the number of entries in the table is a perfect
square. The other two macros will process tables of fluctuating size since
the macro contains a housekeeping portion to calculate the address modification
table.

MACHINE: 702 705 X Model 1 or 2 Other

Forgan ___ X Printer ___ TRC ___ Drum ___ (Specify)

Card Reader ___ 705 ___ Other ___ (Specify)

PROGRAM LANGUAGE: Autocoder ___ Symbolic ___ Actual ___

Other ___ (Specify)

PROGRAM TYPE: Complete Program ___

Macro-Instruction ___ Label ___

Subroutine ___ Label ___

CONTRIBUTED BY:
A. Liet
Headquarters, USAF

Any questions should be addressed to:
George Widding, AFASC-3E
Data Processing Division
Headquarters, USAF, Washington 25, D.C.

April 1958, Bulletin 57 - 43

705 CUSTOMER CONTRIBUTION
Program Write-Up Abstract

INDICATIVE CODE
AL-0001

PROGRAM NAME
705 Assembly Program for
704/705 Symbolic Programs

PURPOSE: To assemble 704 or 705 symbolic cards on an IBM
705, producing an assembly listing and output cards.

* This is strictly a tape-to-tape operation.

RESTRICTIONS: 40,000 character memory capacity
6 tape drives on line

CONTRIBUTED BY:
Robert P. Tiptrott
Allison Division, General Motors Corp.

GUIDE

PROGRAM WRITE-UP ABSTRACT

INDICATIVE CODE
AC-002-0

PROGRAM NAME
PRINT I TRACING ROUTINE

PURPOSE: To function as a debugging aid in cases where debugging by memory print
fails. The routine lists each PRINT I step executed, along with numerical values
of the operands and results, if any.

MACHINE: 702 705 X Model 1 Other ___ (Specify)

Forgan ___ X Printer ___ TRC ___ Drum ___ (Specify)

Card Reader ___ 705 ___ Other ___ (Specify)

PROGRAM LANGUAGE: Autocoder ___ Symbolic ___ Actual ___

Other ___ PRINT ___

PROGRAM TYPE: Complete Program ___

Macro-Instruction ___ Label ___

Subroutine ___ Label ___

CONTRIBUTED BY:
W. R. Brittenham
A. O. Smith Corporation

(August 1957, Bulletin 50 - 117)

GUIDE

PROGRAM WRITE-UP ABSTRACT

INDICATIVE CODE
AC-003-0

PROGRAM NAME
ABBREVIATED PRINT I TRACING

ROUTINE

PURPOSE: To function as a debugging aid in cases where the amount of memory
available for a tracing routine is small. BADD and PDCI are listed for each
PRINT I Program step executed.

MACHINE: 702 705 X Model 1 Other ___ (Specify)

Forgan ___ X Printer ___ TRC ___ Drum ___ (Specify)

Card Reader ___ 705 ___ Other ___ (Specify)

PROGRAM LANGUAGE: Autocoder ___ Symbolic ___ Actual ___

Other ___ PRINT ___

PROGRAM TYPE: Complete Program ___

Macro-Instruction ___ Label ___

Subroutine ___ Label ___

CONTRIBUTED BY:
W. R. Brittenham &
George Knox
A. O. Smith Corporation

(August 1957, Bulletin 50 - 119)

GUIDE

PROGRAM WRITE-UP ABSTRACT

INDICATIVE CODE
AC-004-0

PROGRAM NAME
LEAST SQUARES POLYNOMIAL

CURVE-FITTING ROUTINE

PURPOSE: To produce the coefficients of that polynomial which best gives data in the
least squares sense, and to plot that polynomial and the given points graphically on
the printer. The program makes logarithmic transformations on given data when
required.

(Continued on next page)
### CONTRIBUTED BY:
W. R. Britenham and G. W. Kuss
A. O. Smith Corporation

### PROGRAM
**NAME:** AO-005-0

### PROGRAM WRITE-UP ABSTRACT

**INDICATIVE CODE**
AO-005-0

**PROGRAM NAME**
AO-005-0

**MACHINE:**
705 X Model I Other

**Card Reader:**
700 Other

**Card Reader:**
700 Other

**PROGRAM LANGUAGE:**
Autocoder Symbolic Actual

**PROGRAM TYPE:**
Complete Program X

**Macro-Instruction**
Label

**Subroutine**
Label

**CONTRIBUTED BY:**
W. R. Britenham and G. W. Kuss
A. O. Smith Corporation

---

### PROGRAM WRITE-UP ABSTRACT

**INDICATIVE CODE**
AO-004-0

**PROGRAM NAME**
AO-004-0

**MACHINE:**
705 X Model I Other

**Card Reader:**
700 Other

**PROGRAM LANGUAGE:**
Autocoder Symbolic Actual

**PROGRAM TYPE:**
Complete Program X

**Macro-Instruction**
Label

**Subroutine**
Label

**CONTRIBUTED BY:**
W. R. Britenham and G. W. Kuss
A. O. Smith Corporation

---

### GUIDES

#### PROGRAM WRITE-UP ABSTRACT

**INDICATIVE CODE**
AO-001-0

**PROGRAM NAME**
AO-001-0

**MACHINE:**
705 X Model I Other

**Card Reader:**
700 Other

**PROGRAM LANGUAGE:**
Autocoder Symbolic Actual

**PROGRAM TYPE:**
Complete Program X

**Macro-Instruction**
Label

**Subroutine**
Label

**CONTRIBUTED BY:**
W. R. Britenham and G. W. Kuss
A. O. Smith Corporation

---

### PROGRAM WRITE-UP ABSTRACT

**INDICATIVE CODE**
AO-004-0

**PROGRAM NAME**
AO-004-0

**MACHINE:**
705 X Model I Other

**Card Reader:**
700 Other

**PROGRAM LANGUAGE:**
Autocoder Symbolic Actual

**PROGRAM TYPE:**
Complete Program X

**Macro-Instruction**
Label

**Subroutine**
Label

**CONTRIBUTED BY:**
W. R. Britenham and G. W. Kuss
A. O. Smith Corporation

---

### PROGRAM WRITE-UP ABSTRACT

**INDICATIVE CODE**
AO-005-0

**PROGRAM NAME**
AO-005-0

**MACHINE:**
705 X Model I Other

**Card Reader:**
700 Other

**PROGRAM LANGUAGE:**
Autocoder Symbolic Actual

**PROGRAM TYPE:**
Complete Program X

**Macro-Instruction**
Label

**Subroutine**
Label

**CONTRIBUTED BY:**
W. R. Britenham and G. W. Kuss
A. O. Smith Corporation

---

(August 1957, Bulletin 50 - 121)

---

(August 1957, Bulletin 50 - 121)

---

(August 1957, Bulletin 50 - 121)
GUIDE
PROGRAM WRITE-UP ABSTRACT

INDICATIVE CODE
AO-0116

PROGRAM NAME
Search Master Program Type

PURPOSE:
To search a master program tape on 0201 for a specific PRINT program, re-create any tapes containing portions of the program, bring the program into memory, and transfer control to it.

MACHINE: 702

PROGRAM LANGUAGE: Autocoder

PROGRAM TYPE: Complete Program

CONTRIBUTED BY:
W. B. Brittenham and G. W. Kuss
A. O. Smith Corporation

GUIDE
PROGRAM WRITE-UP ABSTRACT

INDICATIVE CODE
BW - 001 - 1

PROGRAM NAME
Address Modification

PURPOSE:
To provide a common set of address modification macro instructions for 705 Model II and III. This version contains revisions to the macro instruction MOVEA of contribution BW - 001 - 0. The macro instructions included are:

- Macro Name
- Operation Code
  - ADDA
  - SUBA
  - INCR
  - DECR
  - CALCA
  - INITA
  - MOVEA

MACHINE: 702

PROGRAM TYPE: Complete Program

CONTRIBUTED BY:
James O'Malley
Boeing Airplane Company
Wichita Division

GUIDE
PROGRAM WRITE-UP ABSTRACT

INDICATIVE CODE
AIR-0014

PROGRAM NAME
Sort 57 - Blocked Variable

PURPOSE:
Corrections to above-mentioned modification to Sort 57. To transmit a group mark before TRA to dump unreadable records,

Phase 2 @ 38554
Phase 3 @ 38555

MACHINE: 702

PROGRAM TYPE: Complete Program

CONTRIBUTED BY:
The Curtis Publishing Company
Independence Square
Philadelphia, Pennsylvania

Written by: William Anderson
IBM Corporation

Distribution No. 8
GUIDE
PROGRAM WRITE-UP ABSTRACT

INDICATIVE CODE
DE-002-0

PROGRAM NAME
Title: Index and Switch Program

PURPOSE:
This program, using program listing tape from an autocoder or assembler as input, produces cards which, after EAM processing, may be used to make listings to serve as index and halt logs for console operator's manual and a switch log for programmer's use.

MACHINE: 700  705  X  Model II  Other
Tapes:  X  Printer:  TEC  Drum  (Specify)
Card Reader:  (Specify)  Other
PROGRAM LANGUAGE: Autocoder  Symbolic  Actual
Other  (Specify)

PROGRAM TYPE: Complete Program  Patch
Macro-Instruction  Label
Subroutine  Label

CONTRIBUTED BY:
The Detroit Edison Company
1600 Second Avenue
Detroit 26, Michigan

Richard J. Gresy

705 CUSTOMER CONTRIBUTION
Program Write-Up Abstract

INDICATIVE CODE
DP 0001

PROGRAM NAME
Calculation of Seasonal Adjustment Factors

(Continued on next column)
CONTRIBUTED BY,
W. L. Myers, Eastman Kodak
Rochester, New York

GUIDE

PROGRAM WRITE-UP ABSTRACT

INDICATIVE CODE
EQ-003-C

PROGRAM NAME
Eastman Kodak, Consolidated Edison
Transfer Tracing (EKCETT)

PURPOSE:
To print a record of transfers of control within the main program, ten transfers per printer line. Its function is the same as Trac 51; namely, to provide a means of following the actual path used during the run of a program during debugging. This program is relocable.

Restrictions:
The program occupies 643 memory positions. It may be placed in any convenient location in memory, except the 1st 240 digits. Only 324 positions of accumulator 0 are required by the main program.

General Description:
This program is a refinement of a program developed by Mr. Art Brown, Consolidated Edison New York City, customer contribution No. 10.

1. EKCETT may be placed in any convenient location in memory except the 1st 240 digits. The program occupies 643 memory positions.

2. Tracing may be discontinued at any time during a run by turning off 916. This will cause the machine to stop and the typewriter will print two 5 digit numbers.
   a. The address of the next instruction
   b. The operation just performed

   If the operation was a transfer the two numbers are the same.
   To continue without Transfer Tracing make a manual transfer from the console to the address of the next instruction as shown on the typewriter.

3. Tracing can be restarted at any point in the main program by the following:
   a. Manually store 5 digit address of instruction at a position in memory that is 500 - higher than the starting point of transfer tracing routine.

CONTRIBUTED BY:
E. Althoff, Eastman Kodak
Rochester, New York

GUIDE

PROGRAM WRITE-UP ABSTRACT

INDICATIVE CODE
EQ-003-D

PROGRAM NAME
EQ-003-D

PURPOSE: To convert a 705 program written in the symbolic system to a 705 program written in Autocoder language.

MACHINE: 705

PROGRAM TYPE: Complete Program

Macro-Instruction

Subroutine

INDICATIVE CODE
EQ-001-D

PROGRAM NAME
EXACTO - 10 DIGIT CONVERSION

MACHINE: 702

PROGRAM TYPE: Complete Program

Macro-Instruction

Subroutine

CONTRIBUTED BY:
Earl Althoff
Eastman Kodak Company

January 1958, Bulletin 56 - 71

705 CUSTOMER CONTRIBUTION

Program Write-up Abstract

INDICATIVE CODE
EK 0004

PURPOSE:
Enable programmer to write in actual as 10 digits (Ex: ROD 00 25450). The routine processes cards punched in 10 digit form, checks instructions for validity, giving listings and condensed cards as output.

PROGRAM LANGUAGE: Autocoder

PROGRAM TYPE: Complete Program

Macro-Instruction

Subroutine

INDICATIVE CODE
EX-001-D

PROGRAM NAME
EXACTO - 10 DIGIT CONVERSION

MACHINE: 702

PROGRAM TYPE: Complete Program

Macro-Instruction

Subroutine

INDICATIVE CODE
EK 0002

MACHINE: 702

PROGRAM TYPE: Complete Program

Macro-Instruction

Subroutine

CONTRIBUTED BY,
Barry Gordon
Equitable Life Assurance Society of the U.S.
222 Seventh Avenue
New York 1, New York

January 1958, Bulletin 55 - 70

GUIDE

PROGRAM WRITE-UP ABSTRACT

INDICATIVE CODE
EQ-001-C

PROGRAM NAME
CHECKING LOADING ROUTINE

MACHINE: 702

PROGRAM TYPE: Complete Program

Macro-Instruction

Subroutine

CONTRIBUTED BY:
Barry Gordon
Equitable Life Assurance Society of the U.S.
222 Seventh Avenue
New York 1, New York

January 1958, Bulletin 55 - 70
CONTRIBUTED BY:
Esso Standard Oil Company
P.O. Box 422
Linden, N. J.

GUIDE
PROGRAM WRITE-UP ABSTRACT

INDICATIVE CODE
E2-003-0

PROGRAM NAME
Product Inverse Linear Programming

PURPOSE:
To calculate optimum solutions for problems involving up to 99 linear constraints
and 125 variables. The program contains a partitioning feature useful in solving
block-triangular (for instance, Multi-Grade Blending) problems. Multiple
profit functions and/or multiple requirements vectors can be handled.

MACHINE: 702 ______ 705

PROGRAM LANGUAGE:
Autocoder ___ Symbolic ___ Actual ___
Other ___ Autocoder A ___ (Specify)

PROGRAM TYPE:
Complete Program ___

FUNCTIONS AND/OR MULTIPLE REQUIREMENTS

Other ___

CONTRIBUTED BY:
W. G. Hyde
F. R. Pfaff
W. E. Zieman
Esso Standard Oil Company
Linden, New Jersey

GUIDE
PROGRAM WRITE-UP ABSTRACT

INDICATIVE CODE
E2-004-0

PROGRAM NAME
GENERAL TRANSFER ANY ROUTINE
(Also Generalized Edit Note Routine)

PURPOSE:
To avoid need for many specialized TRA routines in a single program.
The current implementation copies (with a little additional subroutine) all
the routines in a single program.

MACHINE: 702 ______ 705

PROGRAM LANGUAGE:
Autocoder ___ Symbolic ___ Actual ___
Other ___ Autocoder A ___ (Specify)

PROGRAM TYPE:
Complete Program ___

OTHER ROUTINES

Other ___

CONTRIBUTED BY:
Esso Standard Oil Company - M. E. Gross
18 West St., N. Y. C.
International Business Machines Corp. - B. F. Dearman
New York City
(August 1957, Bulletin 50 - 129)
GUIDE

PROGRAM WRITE-UP ABSTRACT

INDICATIVE CODE
IBM-COL-0

PROGRAM NAME
LOOPCODER

PURPOSE:
To simplify programming of 705 loop operations. The Loopcoder is a precompiler that expands program loops from a simple form to a detailed form, supplying the initialization, address modification, and counter testing operations. Output from the Loopcoder is in Autocoder input form.

MACHINE: 702

Program Type: Complete Program

Machine precompiler that expands program
HB-DD1-D
supplying the
Output from the Loopcoder is in Autocoder input form.

CARD READER: X
700
Other

PROGRAM LANGUAGE:
Autocoder X Symbolic Actual
Other (Specify)

PROGRAM TYPE: Complete Program X

Macro-instruction__Label
Subroutine__Label

CONTRIBUTED BY:
W. M. Selden, Program Research
Humble Oil and Refining Company
Baytown, Texas
Program written by J. S. Bonner

April 1958, Bulletin 57 - 51

705 CUSTOMER CONTRIBUTION
Program Write-up Abstract

INDICATIVE CODE
IBM 0002

PROGRAM NAME
Card Image

MACHINE SPECIFICATIONS:
20,000 or 40,000 Memory Position 705

FUNCTION:
To establish a card image in memory which may be addressed as CARD, or each column may be addressed as COLXX (i.e., COL 1 or COL 23, etc.).

GENERAL DESCRIPTION:
A card image is established in memory which may be addressed as CARD, or each column may be addressed as COLXX (i.e., COL 1 or COL 23, etc.).

RESTRICTIONS:
The subroutine uses 81 to 85 positions. The programmer must write at least one INCL CARD.

CONTRIBUTED BY:
W. M. Selden, Program Research
IBM, World Headquarters, New York

705 CUSTOMER CONTRIBUTION
Program Write-up Abstract

INDICATIVE CODE
IBM 0003

PROGRAM NAME
Flow Chart Listing From Assembly Program
Print Record Tape

MACHINE SPECIFICATIONS:
40,000 Position 705

PURPOSE:
To produce automatically, a flow chart listing, utilizing the tape which is the listing of the assembled program, as input data. This tape is produced by ASST 72.

(Continued on next column)

RESTRICTIONS:
The program can handle a total of 1700 transfers.

Of these:
1. 800 may connect one location on a page to a higher location on the same page (forward transfers).
2. 240 may connect one location on a page to a lower location on the same page (backward transfers).
3. 799 may connect one page to another (off page transfers).

If the forward or backward transfer table becomes exhausted, transfers of that type are ignored.

The program can handle a maximum of 99 pages of output listing. The program is written to plot the output at eight lines per inch. Five arrows may be plotted at one time in the forward direction and four in the backward direction. Any location for which an arrow position cannot be found is noted on the typewriter.

CONTRIBUTED BY:
A. E. Scott, Diagnostic Engineering,
IBM, Poughkeepsie, New York

705 CUSTOMER CONTRIBUTION
Program Write-up Abstract

INDICATIVE CODE
IBM 0005

PROGRAM NAME
Print I Program for Solution of Simultaneous Equations and Matrix Inversion

MACHINE SPECIFICATIONS:
20,000 or 40,000 Position 705

PURPOSE:
To solve simultaneous equations and matrix inversion.

RESTRICTIONS:
The coding kernel given on page 56 on the PRINT I Intermediate Manual is used with the restriction that only one column vector is allowed.

GENERAL DESCRIPTION:
The program is written for PRINT I system and will handle up to thirty equations with thirty unknowns in core storage. The program will operate using the 10-digit mantissa system.

It is necessary to specify on a control card the number of decimal positions in the data words, \( \text{dum}(i) \) and the number of equations to be solved, \( \text{num} \).

On line print-out of solutions is provided and optional print-out of inverse matrix.

CONTRIBUTED BY:
D. Loposer, IBM, Birmingham

705 CUSTOMER CONTRIBUTION
Program Write-up Abstract

INDICATIVE CODE
IBM 0007

PROGRAM NAME
Tape Duplication

MACHINE SPECIFICATIONS:
20,000 or 40,000 Position 705

764 Tape Control Unit

PURPOSE:
To provide exact duplication of one tape from another.

RESTRICTIONS:
1. Record length may not exceed 19,750 characters for a 20,000 position 764, nor may it exceed 39,750 characters for a 40,000 position 764.
2. Records to be duplicated must not contain the following sequence of five characters: 00000 which is used in determining end of record. If this sequence appears in records, any desired five characters may be substituted for it.

(Continued on next page)
GENERAL DESCRIPTION:
The input tape for this program is mounted on tape unit 0200; output is written on tape 0001. Records to be duplicated may be of fixed or variable length, and may contain group marks. Files separated by tape marks can be reproduced, and the records from several input tapes can be written on the same output tape.

CONTRIBUTED BY:
W. G. Winchester, IBM, Poughkeepsie

705 CUSTOMER CONTRIBUTION
Program Write-up Abstract

INDICATIVE CODE PROGRAM NAME
IB 0009 Calendar Demonstration

MACHINE SPECIFICATIONS:
20,000 or 40,000 Position 705

PURPOSE:
To demonstrate the speed and versatility of a high-speed computing machine.

GENERAL DESCRIPTION:
The Calendar Demonstration Program will compute the day of the week of any given calendar date between March 1, 2001 and December 31, 1999. This program will also compute the given date for the following holidays, both fixed and variable.

Fixed
New Year's Day
St. Valentine's Day
Washington's Birthday
April Fool's Day
Memorial Day
Independence Day
Columbus Day
Halloween
Veterans Day
Christmas Day

Variable
Mothers Day
Fathers Day
Labor Day
Election Day
Thanksgiving Day
Easter Sunday

The participant may, if he likes, try to fool the machine by giving a non-existent date to which the machine will give an appropriate answer.

The program will predict for dates that fall on February 12 or February 22, and years hence they will be born. For dates that precede the adoption of the Gregorian Calendar in 1582, the computation proceeds as if it were in effect; but an explanation is printed for the participant's consideration.

CONTRIBUTED BY:
Mr. Elliot Raiff

705 CUSTOMER CONTRIBUTION
Program Write-up Abstract

INDICATIVE CODE PROGRAM NAME
IB 0010 Generalized Matrix Inversion

MACHINE SPECIFICATIONS:
20,000 or 40,000 Position 705

PURPOSE:
To invert successive matrices printing input and inverse in a convenient format.

RESTRICTIONS:
The largest inversion possible will be found by the following relationship:

\[(a+b) \leq 1000\]
and
\[(a+b) \leq 99\]

where \(n\) = order of matrix
\(k\) = number of column vectors.

GENERAL DESCRIPTION:
This program is designed to perform a matrix inversion on data presented to it in a specified form. The routine is accomplished by using the PRIN T 1 Automatic Coding System. Successive matrices of different order may be inverted; each matrix will have its own control card preceding the elements indicating the order and the number of column vectors. The inversion takes place entirely within memory.

CONTRIBUTED BY:
T. Glans and F. Williams, IBM, WHQ

705 CUSTOMER CONTRIBUTION
Program Write-up Abstract

INDICATIVE CODE PROGRAM NAME
IB 0011 MUSIC

MACHINE SPECIFICATIONS:
20,000 or 40,000 Position 705
Card Reader
Power Amplifier connected to SPR (Store for Print) instruction.

NOTE: See your Customer Engineer

PURPOSE:
This program is designed to permit the use of 705, with an attached amplifier, to play music.

GENERAL DESCRIPTION:
The card deck furnished with this program includes three tunes: "Seems Like Old Times," "Old Piano Roll Blues," and "Entry of the Gladiators." By punching cards according to a specified procedure, other desired tunes may be played on the 705.

CONTRIBUTED BY:
R. W. Berner, W. M. Selden and A. S. Petroulakis, IBM, WHQ

GUIDE
PROGRAM WRITE-UP ABSTRACT

INDICATIVE CODE PROGRAM NAME
20 - 001 - 0 68 Time - Sort 54

SORTING TIME CALCULATION

MACHINE: 702 X Model 11 Other

Printers

Printers

IBM-

IBM-

Autocoder

Symbolic

Actual

Other

Micro-instruction

Label

Subroutine

Label

CONTRIBUTED BY:
Imperial Oil Limited
Toronto, Canada

Distribution No. 8
GUIDE

PROGRAM WRITE-UP ABSTRACT

INDICATIVE CODE
LM-307-0

PROGRAM NAME
End-of-File Search

PURPOSE:

MACHINE: 702 705 X Model I or II Other

Pages: 1 Printer TRC X Drum (Specify)

Card Reader X 760 Other

PROGRAM LANGUAGE: Autocoder X Symbolic Other

Any number of 705 tapes, drum, 717 printer, punch, and card reader. Takes care of RD, WR, WR, and WR, but not WR.

PROGRAM TYPE: Complete Program

CONTRIBUTED BY:

Richard Bulita, IBM
Northwestern Mutual Life Insurance Company
726 East Wisconsin Avenue
Milwaukee 2, Wisconsin

Distribution No. 6

GUIDE

PROGRAM WRITE-UP ABSTRACT

INDICATIVE CODE
NW-001-6, A TRC Modification of AO-005-0

PROGRAM NAME
705 Address Listing

PURPOSE:

To produce an actual address listing following a 705 assembly of programs written in either Autocoder, Print I, or Symbolic language. The program reads the listing tape prepared by the assembly and prepares a sorted table of address-location references - which is written out on the listing tape following the tape mark.

MACHINE: 702 705 X Model I or II Other

Pages: 1 Printer TRC X Drum (Specify)

Card Reader X 760 Other

PROGRAM LANGUAGE: Autocoder X Symbolic Other

Any number of 705 tapes, drum, 717 printer, punch, and card reader. Takes care of RD, WR, WR, but not WR.

PROGRAM TYPE: Complete Program

CONTRIBUTED BY:

The Northwestern Mutual Life Insurance Company
726 East Wisconsin Avenue
Milwaukee 2, Wisconsin

GUIDE

PROGRAM WRITE-UP ABSTRACT

INDICATIVE CODE
NW-003-1

PROGRAM NAME
Tape Compare (ITPME)

PURPOSE:

Compare any two (2) tape files of fixed or variable length records not greater than 300 characters or less than 10 characters in length. Records which are not identical are written out. Record comparison may also be added through preliminary control word comparison at the option of the user. Using this option, all records which are not identical or unmatched are written out.

The Tape Label and Label Routine used in this program is of the same type that is required by IBM's Utility Programs. This program is a revision of Contribution NW-003-0 which contained a specialized Tape Label Routine. (Continued on next column)
January 1958, Bulletin 55 - 65

GUIDE

PROGRAM WRITE-UP ABSTRACT

INDICATIVE CODE
PG-004-0

PROGRAM NAME

PURPOSE:
To check that one and only one tape unit is dialed to the units position of each designated input and output tape. "Check Tape Settings" and half in case of duplicate settings, stops at L/O No Response if no tape is dialed to one of the designated tape addresses.

MACHINE: 702
705 X
Model 1 or II. Other

Files 1 to 10
Printer TRC Drum (Specify)

Card Reader 760
Other

PROGRAM LANGUAGE: Autocoder X Symbolic Actual

Other (Specify)

PROGRAM TYPE: Complete Program

Macro-Instruction X Label

With Linked Subroutine X Label

Contributed By:
Edward B. Berninger
Procter & Gamble

January 1958, Bulletin 55 - 61

GUIDE

PROGRAM WRITE-UP ABSTRACT

INDICATIVE CODE
PG-005-0

PROGRAM NAME

PURPOSE:
To solve the "Transportation Problem", a special case of linear programming. The program can accommodate matrices with M x N x 705, where "M" is number of destinations and "N" is number of sources.

The program was written originally by IBM for the 702, and converted by them to 705 language. Procter & Gamble debugged the converted program and added additional features.

The largest problem run has been 26 x 149, which took up 90 iterations and 50 minutes.

MACHINE: 702
705 X
Model 1 or II. Other

Files Any Number
Printer TRC Drum (Specify)

Card Reader 760
Other Punch (optional)

PROGRAM LANGUAGE: Autocoder Symbolic Actual X

Other (Specify)

PROGRAM TYPE: Complete Program

Macro-Instruction Label

Subroutine Label

Contributed By:
S. Hickenlooper, D. W. Grace, E. B. Berninger
Procter & Gamble

NOTE: Program material includes a "squeeze" deck of approximately 445 cards, complete Operating and card punching instructions, a general description of the method used (the original IBM 702 write-up), typical running times, and a one-page block diagram of the overall program system.

Symbolic instruction cards and listing are not available.

Distribution No. 5

GUIDE

PROGRAM WRITE-UP ABSTRACT

INDICATIVE CODE
PG-006-0

PROGRAM NAME

PURPOSE:
To load an ASU or the accumulator, previously set, compare to a memory field, and make the necessary transfer (E, H, L, M, MM, T, X, XE, XJ, S13 based on the comparison.

MACHINE: 702
705 X
Model 1 or II. Other

Files Any Number
Printer TRC Drum (Specify)

Card Reader 760
Other

PROGRAM LANGUAGE: Autocoder X Symbolic Actual

Other (Specify)

PROGRAM TYPE: Complete Program

Macro-Instruction X Label

Subroutine Label

Contributed By:
Richard B. Thomas, Procter & Gamble
Andrew T. Pogport, IBM, Cincinnati

April 1958, Bulletin 57 - 53

GUIDE

PROGRAM WRITE-UP ABSTRACT

INDICATIVE CODE
PG-007-0

PROGRAM NAME

PURPOSE:
To search a table in memory, using the "binary search" method. To eliminate multiply instructions and other calculation in the subroutine loop, all increments and decrements are calculated once for each BSNCH macro in a program and stored in an in-line record area. Arguments can be up to 79 characters long and functions up to 999, and can be located anywhere in a table item. The number of items in the table can vary during a program. Table size is limited only by memory availability.

MACHINE: 702
705 X
Model 1 or II. Other

Files Any Number
Printer TRC Drum (Specify)

Card Reader 760
Other

PROGRAM LANGUAGE: Autocoder X Symbolic Actual

Other (Specify)

PROGRAM TYPE: Complete Program

Macro-Instruction X Label

Subroutine Label

Contributed By:
Richard B. Thomas
Procter & Gamble

Note: Time for one "binary search loop" in the subroutine is 0.578 + .017 N milliseconds, where N is the number of characters in the argument.
GUIDE
PROGRAM WRITE-UP ABSTRACT

INDICATIVE CODE
PG-208-0

PROGRAM NAME
Group Records

PURPOSE:
To group fixed-length records, using serial or high-speed transmission, and transfer to a designated address after a specified number of records have been grouped.

MACHINE: 702

705 X Model J or II Other (Specify)

Files: Any No. Printer: TRC Drum

Card Reader: 760 Other

PROGRAM LANGUAGE:
Autocoder X Symbolic Actual

Other (Specify)

PROGRAM TYPE:
Complete Program

Macro-Instruction X Label OTHER

Subroutine X Label OTHER

CONTRIBUTED BY:
William F. Reiland
Procter and Gamble

GUIDE
PROGRAM WRITE-UP ABSTRACT

INDICATIVE CODE
PG-209-0

PROGRAM NAME
Sort Internally

PURPOSE:
To sort fixed-length records which are set up for high-speed transmission on a specified single control field. The sort takes place entirely within memory. The control field can be located anywhere in the record and can be up to 355 characters. Maximum record length is 600 characters, but this can easily be changed to any size. The number of records to be sorted can vary within a program.

MACHINE: 702

705 X Model J or II Other (Specify)

Files: Any No. Printer: TRC Drum

Card Reader: 760 Other

PROGRAM LANGUAGE:
Autocoder X Symbolic Actual

Other (Specify)

PROGRAM TYPE:
Complete Program

Macro-Instruction X Label SORT 1

Subroutine X Label SORT 1

CONTRIBUTED BY:
William H. Graver
Procter and Gamble

GUIDE
PROGRAM WRITE-UP ABSTRACT

INDICATIVE CODE
GR-201-0

PROGRAM NAME
SORT 58

PURPOSE:
To sort fixed or variable length records via TCU.

MACHINE: 702

705 X Model J or II Other (Specify)

Files: Any No. Printer: TRC Drum

Card Reader: 760 Other

Program: J Label: OTHER

PROGRAM LANGUAGE:
Autocoder X Symbolic Actual

Other (Specify)

PROGRAM TYPE:
Complete Program

Macro-Instruction X Label

Subroutine X Label

CONTRIBUTED BY:
The Procter & Gamble Company
* patches for existing program

Distribution No. 5

GUIDE
PROGRAM WRITE-UP ABSTRACT

INDICATIVE CODE
GR-012-0

PROGRAM NAME
New Macro Lookup for 705 Autocoder System

PURPOSE:
The method of searching for macros in Phase I of the 705 autocoder system has been revised to reduce assembly time. A conservative estimate of 705 time saved is one minute per 95 macros assembled. The change requires only three patch cards which overlay part of the present routine.

MACHINE: 702

705 X Model J or II Other (Specify)

Files: Any No. Printer: TRC Drum

Card Reader: 760 Other

PROGRAM LANGUAGE:
Autocoder X Symbolic Actual X

Other (Specify)

PROGRAM TYPE:
Complete Program

Macro-Instruction X Label

Subroutine X Label

CONTRIBUTED BY:
The Procter & Gamble Company

* patches for existing program

Distribution No. 5
GUIDE

PROGRAM WRITE-UP ABSTRACT

INDICATIVE CODE: SB-005-0

PROGRAM NAME: Analyzer

PURPOSE: To reduce machine time required for testing, and produce test output shortly after each testing session.

MACHINE: 705 X Model III Other

Card Reader: 760 Other

PROGRAM LANGUAGE: Autocoder X Symbolic Actual

CONTRIBUTED BY: Directorate of Ballistic Missiles
San Bernardino Air Materiel Area
San Bernardino, California

Written by: John R. Clarke

Distribution No. 4

GUIDE

PROGRAM WRITE-UP ABSTRACT

INDICATIVE CODE: SB-004-0

PROGRAM NAME: Tape Test System

PURPOSE: To produce an edited listing in several optional sequences, cross referencing the data available in an Autocoder Assembly Listing Tape, to produce a complete set of operations for all functions involving on-line records, end of tape, checkpoint, and transfer - any analysis. A utility routine provides for restart if the checkpoint options are used.

MACHINE: 705 X Model III Other

Card Reader: 760 Other

PROGRAM LANGUAGE: Autocoder X Symbolic Actual

CONTRIBUTED BY: Directorate of Ballistic Missiles
San Bernardino Air Materiel Area
San Bernardino, California

Written by: K. Lantz

Distribution No. 4
CONTRIBUTED BY:
SPAN Data Processing Center, Inc.

Questions may be addressed to:
Ronald A. Grant
SPAN Data Processing Center, Inc.
95 Woodland Street
Hartford, Conn.

705 CUSTOMER CONTRIBUTION
Program Write-up Abstract

INDICATIVE CODE
PROGRAM NAME
SR-064-0

760 Assembly of 705 programs (20,000 and 40,000)

MACHINE SPECIFICATIONS:
2000 work 650
Alphabetic device on the card reader, no other special devices required.

PURPOSE:
The 705 program assembly as done on the 650 converts symbolic locations and addresses to actual locations and addresses, and converts mnemonic operation codes to actual operation codes.

RESTRICTIONS:
The maximum number of instructions which can be assembled is determined as in Assembly 53 on the 760. Reference should be made to page 7 of Program Brief #12, "Assembly of Programs by 760" as a key to determining the maximum program size. Generally speaking, if a large number of consecutive symbolic locations and few inserts are used, there should be no difficulty in assembling any size program. Programs have been assembled with 2974 and 3779 entries, all classes.

CONTRIBUTED BY:
H. E. Peabody, IBM, Atlanta, Georgia
Assigned to Southern Railway

GUIDE
PROGRAM WRITE-UP ABSTRACT

INDICATIVE CODE
PROGRAM NAME
SR-064-0

Tape Label, TRA, Checkpoint Routine

PURPOSE: A generalized routine to establish a rigid control on all input and output tapes with TRA and check point included.
Input tapes are checked for valid job identification, unit number, and reel order. Output tapes are checked for valid destroy date with new labels written on tape and upper level short.
Routine is set up for program input on card reader but is easily modified for program input on tape.

MACHINE: 702
705 X Model I or II Other

1. Tape X
2. Printer X TRC Drum
3. Card Reader X 760 Other

PROGRAM LANGUAGE: Autocoder X Symbols X Actual X

PROGRAM TYPE: Complete Program

MACRO-INSTRUCTION
Label

SUBROUTINE
Label

CONTRIBUTED BY:
Southern Railway System
Computer Center
125 Spring St., N.W.
Atlanta, Ga.

(August 1967, Bulletin 50 - 130)

GUIDE
PROGRAM WRITE-UP ABSTRACT

INDICATIVE CODE
PROGRAM NAME
SR-064-0

Tape Operation, Tape Label and Trailer Checking

PURPOSE:
1. To provide for the operation of programs from a program tape.
2. To provide for the detection and correction or disposition of errors resulting from the use of the Tape Record Coordinators.
3. To provide for proper tape usage through the use of tape labels and trailers.

MACHINE: 702
705 X Model I or II Other

1. Tape X
2. Printer X TRC Drum
3. Card Reader X 760 Other

PROGRAM LANGUAGE: Autocoder X Symbols X Actual X

PROGRAM TYPE: Complete Program

MACRO-INSTRUCTION
Label

SUBROUTINE
Label

CONTRIBUTED BY:
Southern Railway Company - F. P. Ludlow, Jr.
10th and R-Streets, N. W.
Washington, D. C.

* The generalized routines use three tapes. All other tape requirements depend upon the running program.
This memorandum provides the information needed to incorporate a tabulation program in Phase III of Sort 54, writing no sort output and utilizing the sort's header and trailer routines for the report. Knowledge of the materials in the Modification Section of the Sort 54 Reference Manual, form CRB-6031, is assumed.

**MACHINE:**
- 702
- 703
- X
- Model I
- II
- Other
- (Specify)

**Card Reader:**
- 760
- Other

**PROGRAM LANGUAGE:**
- Autocoder
- X
- Symbolic
- Actual
- Other
- (Specify)

**PROGRAM TYPE:**
- Complete Program

**Macro-Instruction:**
- X
- Label

**Subroutine:**
- X
- Label

**CONTRIBUTED BY:**
- A. F. Rundquist
- Department of the Army
- TAGO, Data Processing Branch
- Washington, D.C.
IBM 709 PROGRAM LIBRARY ABSTRACT

0709 38025T109 AVAILABLE PRIOR TO JANUARY 1962

BASIC 709/10 CONVERSION SUBROUTINES. A SET OF BASIC INPUT AND OUTPUT CONVERSION SUBROUTINES FOR USE WITH THE 709. THE TWO GROUPS OF SUBROUTINES ARE INTER-RELATED AMONG THEMSELVES AND USE A COMMON COMMUNICATION REGION. THE ACTUAL CODING HAS NOT BEEN DISTRIBUTED. SPECIFICATIONS ARE BY THE 709 SYSTEMS COMMITTEE.

0709 4826G89T AVAILABLE PRIOR TO JANUARY 1962

709 PROGRAM FOR CHECKING OPERATIONS NEEDING TRANSLATING SPILLS THOSE INSTRUCTIONS IN A 709 ABSOLUTE BINARY DECK WHICH MUST BE CHANGED BEFORE THE DECK MAY BE RUN ON A 709. LIST THESE INSTRUCTIONS WITH THEIR LOCATIONS.

0709 4853M123 AVAILABLE PRIOR TO JANUARY 1962

SQUARE ROOT. FLOATING POINT 709 ONLY. SUBSTANTIALLY THE SAME PROGRAM AS 2981, DISTRIBUTION 3077 MODIFIED TO CONFORM TO THE STANDARDS OF THE SC6 SYSTEM AND TO TAKE ADVANTAGE OF NEW 709 INSTRUCTIONS. FULL SQUARE-根平方法精度要求。TIME=1.410PL, ERROR RETURN FOR NEGATIVE, NON-ZERO ARGUMENTS. AC INDICATOR USUALLY TURNED ON. SPACE REQUIRED.

-45 LOCATIONS 12 COMMON.

0709 502LTC9 AVAILABLE PRIOR TO JANUARY 1962

TAPE COMPARISON FOR THE 709

0709 5029L09 AVAILABLE PRIOR TO JANUARY 1962

TAPE CUMP FOR THE TWO OCTAL PRINTS RECORDS OR FIELDS. IN LINE OR WRITE TAPE AT OFF LINE PRINT. BINARY CONTROL CARDS. WILL READ MORE THAN ONE CONTROL CARD. WILL PRINT A SELECTED SEQUENCE OF WORDS FROM EACH RECORD.

IBM 709 PROGRAM LIBRARY ABSTRACT

0709 502HLT95 AVAILABLE PRIOR TO JANUARY 1962

TAPE DUPLICATOR FOR THE 709. REPEAT AS, WRITE AS. WILL SKIP FILES ON EITHER AS OR BS BINARY OR DECIMAL TAPE. BINARY CONTROL CARD KEEPS BOTH TAPES MOVING SIMULTANEOUSLY. C0RN. 5646

0709 5071BACS AVAILABLE PRIOR TO JANUARY 1962

FLOATING POINT TRIGONOMETRIC SUBROUTINE MUST BE FOLLOWED BY IX AD. ARMING BINARY KEYS AND KEYS CONTROL REWINDING BEFORE END. LOCATIONS TO BE OUTPUT PAY BE ENERGIZED IF DESIRED, THE SHARE 2 BOARD IS USED FOR ON-LINE OUTPUT.

0709 5071BLOZ AVAILABLE PRIOR TO JANUARY 1962

FLOATING POINT LOGARITHM BASED ON 709 PROGRAM 628, 200 TO 2500 MSEC ERROR ...

0709 510SSC8P1 AVAILABLE PRIOR TO JANUARY 1962

COMPETITIVE PROGRAM, J709 PROGRAM, PRINTS AN EIGHT-BIT CODED WAVE IN ONE LINE. TAKES 63 CELLS PLUS 27 OF CMOP. DELAYS UNTIL PRINTING IS COMPLETED.

0709 514SCE8SH AVAILABLE PRIOR TO JANUARY 1962

TAPE ASSIGNMENT AND CONTROL PROGRAM PROVIDES COMMUNICATION BETWEEN THE OPERATOR, THE PROGRAM AND THE MACHINE FOR CONNECTING, DISCONNECTING, ASSIGNING AND DISASSIGNING MAGNETIC TAPE.

0709 5396S09P AVAILABLE PRIOR TO JANUARY 1962

ASSEMBLER PROGRAM FOR THE IBM 709. THE TRANSFORM ROUTINE THE CONTROL RECORD FOR THE FIRST PASS THE SECOND PASS THE CONTROL RECORD THE ASSEMBLER.
BUFFERED CARD-INPUT SUBROUTINE

NEADS MULTIPLE CARDS AND TRANSLATES TO BCD. CHECKS FOR ILLEGAL PUNCHES.

DOUBLE PREC. FLOATING PT EXPONENTIAL SUBROUTINE A BETWEEN -49 AND CCR, 14.75 MS FOR EXP=15, 14.73 MS FOR EXP=16, 147 LOCATIONS & 10 ERASABLE.

BASE LEVEL,FLOATING POINT POLYNOMIAL EVALUATION ROUTINE FOR 709 EVALUATES A POLYNOMIAL OF DEGREE N WITH REAL COEFFICIENTS. CALCULATION OF FIRST AND SECOND DERIVATIVES IS OPTIONAL.

BASE LEVEL, LEAST SQUARES CURVE-FITTING ROUTINE USING ORTHOGONAL POLYNOMIALS 704-709 FORTRAN FAP STATISTICAL VALUES INDICATING RELIABILITY OF THE DERIVATIVES ARE PROVIDED. WEIGHTS OTHER THAN ONE MAY BE OPTIONALLY PROVIDED. THE MINIMIZATION MAY BE OPTIONALLY CONSTRAINED TO FORCE UP TO SEVEN OF THE LOW-ORDER COEFFICIENTS TO VANISH. 127 CELLS PROGRAM PLUS TEMPORARIES. CORR. 205.

BASE LEVEL, FORTAN TO SQUEEZE CONVERTER PRODUCES AN SOS PERIPHERAL INPUT OR PUNCH TAPE FROM A FORTRAN COMPILED OUTPUT TAPE. IF THE PFD OUTPUT TAPE IS USERS DIRECTED AS SOS COMPILED INPUT TAPE, THE SQ DECK RESULTS. PLUS A FORTRAN PROGRAM MAY BE DEPOSITED USING THE SOS DEBUGGING TOOLS ALTHERNATIVELY, AN SOS SYMLOGIC DECK MAY BE PUNCHED FROM THE SQ OUTPUT TAPE. THE SQ DECK IS THEN SUITABLE FOR INCORPORATION INTO AN EXISTING SQUEEZE DECK FOR AN SQ PACKAGE. ALLOWING FORTRAN SUBROUTINES TO BE USED IN SOS PROGRAMS.

BASE LEVEL, VECTOR TRIPLE CROSS PRODUCT THIS ROUTINE PROCESSES THE PRODUCT V X W X V. RESULTING FROM THE PRODUCT OF V WITH M V AND V. THESE BEING 4-DIMENSIONAL VECTORS. 50 LOCATIONS ARE REQUIRED. 709 TYPING IS 4.05 MS.

BASE LEVEL, FLOW CHART ANALYSIS BY BOOLEAN MATRIX MANIPULATION OUTLINES WHERE IN COMBINATORIAL FLOW CHARTS UP TO 500 BOXES BY TREATING A FLOW CHART AS A BOOLEAN MATRIX. WILL ALSO DECODE SUBPROGRAMS IN THE FLOW CHART TO PRINT OR DISPLAY. FULL DATA FLOW IS GIVEN, PRINTS COMPLETE LIST OF INPUTS AND OUTPUTS OF ANY SPECIFIED RNG. PROGRAM SHOULD ALSO BE USEFUL FOR NETWORK ANALYSIS AND OTHER PROBLEMS INVOLVING BOOLEAN MATRIX MANIPULATION.

BASE LEVEL, SELECTIVE FILE DUPLICATOR ROUTINE A ROUTINE THAT COPIES ANY OR ALL OF THE FILES OF A INPUT REEL ON TO 1 OR 2 OUTPUT REELS. THE RECORDS MAY BE OF VARIABLE LENGTH.
109 0709 PROGRAM LIBRARY ABSTRACT

The text is a collection of abstracts and instructions for various programs and routines.

For example, one abstract is for a program called "Hamming subroutine for computing the Hamming distance between two sequences of characters."

Another abstract is for a program called "Cosine subroutine to compute the cosine of a real argument."

Instructions for using these programs are also provided, such as how to call them and what they are used for.

The text is written in a technical and formal style, typical of computer science and mathematics documentation.
IBM 0709 PROGRAM LIBRARY ABSTRACT

0709 1007ML040 AVAILLE PRIOR TO JANUARY 1962

STUDENT INPUT-OUTPUT

INTERCONNECTION INPUT-OUTPUT COMPATIBLE WITH LARGE IN SOIL.

FIXED POINT EXTERNAL TO MACHING, FLOATING POINT INTERNALLY.

0709 1007MML042 AVAILLE PRIOR TO JANUARY 1962

RESTART PROGRAM FOR THE BINARY EDITOR FNL 0400/

LOADS THE BINARY EDITOR FROM A TAPE.

0709 1033ML044 AVAILLE PRIOR TO JANUARY 1962

FAP ASSEMBLY PROGRAM

THIS DISTRIBUTION INCLUDES A LISTING TAPE, A SYMBOlIC TAPE, A BIP TRACING TAPE, A SHORT WRITE-UP OF THE

PROGRAM'S EFFECTS; AND A LONG WRITE-UP OF THE PROGRAM'S EFFECTS FOR PROGRAMMERS WRITE-UP

SHOULD BE AVAILABLE EARLY IN 1963.

THE PRIMARY SOURCES ON CARD BANKS ARE AFFIRMED BY NO FAP, HOWEVERT INDIVIDUAL INSTALLATIONS WILL WANT TO

REPLACE THE NOTATION SUPPLIED ON ONE MEETING THEIR OWN REQUIREMENTS. SEE WRITE-UP. CORR. 1075.1216

0709 1035ML051 AVAILLE PRIOR TO JANUARY 1962

ROW binary CARD LOADER

MODELED AFTER A5 CAR FOR THE 104.

* 0709 1035ML052 AVAILLE PRIOR TO JANUARY 1962

MATHEMATICAL PROGRAMMING SYSTEM TWO

A REVISION OF RS MCA A SINGLE PRECISION 7090 CODE USING THE

REVISED SIMPLEX METHOD WITH PRODUCT FORM ISOLATE. CAN HANDLE

PROBLEMS HAVING UP TO 200 ROWS, 399 COLUMNS, AND 1800 NON-

ZERO PATRIX ELEMENTS. INCLUDES COMPOSITE, MULTIPLE OBJECTIVES, INTERMEDIATE AND PUNCH-OUT ABILITY, USE OF SYSTEM TAPE, AND

BATCH RUNNING, CORR. 0 7

IBM 0709 PROGRAM LIBRARY ABSTRACT

0709 1007ML040 AVAILLE PRIOR TO JANUARY 1962

R1 EDITOR FOR PROGRAMMED 704/709/90 COMPATIBILITY

PROVIDES THE NECESSARY SIMULATION, MONITORING AND UTILITY ROUTINES TO ALLOW THE EXECUTION OF 704 A9TULATE binary

PROGRAMS ON THE 709 OR 7090, OPERATES EITHER IN CONJUNCTION WITH OR INDEPENDENT OF THE SHARE OPERATING SYSTEM 704S.

GROUPS CAN BE SIMULATED.

THIS PROGRAM REQUIRES CELLS 0-2790 AND A PORTION OF UPPER

MEMORY EQUAL IN LENGTH TO THE LONGEST RECORD TO BE PROCESSED PLUS APPROXIMATELY 900 CELLS. Voids fill 4-349 50A 687

0709 1035ML051 AVAILLE PRIOR TO JANUARY 1962

ROW binary CARD LOADER

MODELED AFTER A5 CAR FOR THE 104.

* 0709 1035ML052 AVAILLE PRIOR TO JANUARY 1962

MATHEMATICAL PROGRAMMING SYSTEM TWO

A REVISION OF RS MCA A SINGLE PRECISION 7090 CODE USING THE

REVISED SIMPLEX METHOD WITH PRODUCT FORM ISOLATE. CAN HANDLE

PROBLEMS HAVING UP TO 200 ROWS, 399 COLUMNS, AND 1800 NON-

ZERO PATRIX ELEMENTS. INCLUDES COMPOSITE, MULTIPLE OBJECTIVES, INTERMEDIATE AND PUNCH-OUT ABILITY, USE OF SYSTEM TAPE, AND

BATCH RUNNING, CORR. 0 7

IBM 0709 PROGRAM LIBRARY ABSTRACT

0709 1007ML040 AVAILLE PRIOR TO JANUARY 1962

R1 EDITOR FOR PROGRAMMED 704/709/90 COMPATIBILITY

PROVIDES THE NECESSARY SIMULATION, MONITORING AND UTILITY ROUTINES TO ALLOW THE EXECUTION OF 704 A9TULATE binary

PROGRAMS ON THE 709 OR 7090, OPERATES EITHER IN CONJUNCTION WITH OR INDEPENDENT OF THE SHARE OPERATING SYSTEM 704S.

GROUPS CAN BE SIMULATED.

THIS PROGRAM REQUIRES CELLS 0-2790 AND A PORTION OF UPPER

MEMORY EQUAL IN LENGTH TO THE LONGEST RECORD TO BE PROCESSED PLUS APPROXIMATELY 900 CELLS. Voids fill 4-349 50A 687

0709 1035ML051 AVAILLE PRIOR TO JANUARY 1962

ROW binary CARD LOADER

MODELED AFTER A5 CAR FOR THE 104.

* 0709 1035ML052 AVAILLE PRIOR TO JANUARY 1962

MATHEMATICAL PROGRAMMING SYSTEM TWO

A REVISION OF RS MCA A SINGLE PRECISION 7090 CODE USING THE

REVISED SIMPLEX METHOD WITH PRODUCT FORM ISOLATE. CAN HANDLE

PROBLEMS HAVING UP TO 200 ROWS, 399 COLUMNS, AND 1800 NON-

ZERO PATRIX ELEMENTS. INCLUDES COMPOSITE, MULTIPLE OBJECTIVES, INTERMEDIATE AND PUNCH-OUT ABILITY, USE OF SYSTEM TAPE, AND

BATCH RUNNING, CORR. 0 7
0709 1120NLICG
ADDRESS LOCATION SUBROUTING.

0709 1121NMRC
FIND THE LOCATION OF ANY CONSTANT OR VARIABLE IN THE PROGRAM
VARIABLES MAY BE FIXED OR FLOATING, SUBSCRIPTED OR NOT.
SUBSCRIPTS MAY BE EXPRESSIONS OF STANDARD FORTRAN FORM.

0709 1121MNMC
FORTRAN MULTIPLE CORRELATION ANALYSIS PROGRAM
THIS PROGRAM IS FOR THE STATISTICAL ANALYSIS OF A SET OF
POINTS P1, P2,...,PN WHERE P1 = (X1,Y1),...,PN = (XN,YN). THE
PROGRAM WILL PERFORM MULTIPLE CORRELATIONS OF THE FORM
K1 = X1/Y1, K2 = X2/Y2, ..., KN = XN/YN WHERE Ki IS THE
DEPENDENT VARIABLES X1, X2,...,XN ARE INDEPENDENT
VARIABLES AND THE N VALUES ARE TO BE STATISTI-
CALLY ESTIMATED FROM THE DATA.

0709 1133ELIUP
AVAILABLE PRIOR TO JANUARY 1962

0709 1130WVIFP
VARIABLE INFORMATION PROCESSING PACKAGE
TWO-LOG, VIFF, LIKE TONEVIFP, IS A COLLECTION OF SUBROUTINES
DESIGNED TO SERVE AS AN OFFICIAL GENERAL PURPOSE
DATA PROCESSING PACKAGE CORR. 1170

0709 1136WVIFPM
AVAILABLE PRIOR TO JANUARY 1962

0709 1130WVIFPS
AVAILABLE PRIOR TO JANUARY 1962

0709 1137WPRUG
VIPP DIG TRAP,
DETECTED TO ASSIST IN CHECKOUT OF PROGRAMS USING SUBROUTINES
FROM VIFF. A DIGITAL CALL WILL CAUSE ON-The INDICA-
TION OF THE CALL AND DIG TRAPS.

0709 1137WPSVIN
AVAILABLE PRIOR TO JANUARY 1962

0709 1140NPDSAT
DOUBLE PRECISION FLOATING POINT ARGUMENT SUBROUTINE
RATIONAL APPROXIMATION METHOD, INPUT IN AC-HD OR FROM CORE.
OUTPUT IN BINARY OR REAL VALUE. A SIMPLE INSTRUCTION, THE
USER DESIGNING THE CALLING SEQUENCE IS ADVISED TO CONSULT THE
MISSING DOCUMENTATION.

0709 1159MDSAT
AVAILABLE PRIOR TO JANUARY 1962

0709 1159MDSGT
GENERALIZED VARIABLE LENGTH RECORD SORT.
THIS GENERALIZED SORT PROGRAM PROVIDES A 2-WAY MERGE-BCH OR
BINARY INPUT OF N, FIXED OR VARIABLE LENGTH FIXED LENGTH
RECORDS, 2-W SCATTERED CONTROL, FIELD, INTERRUPT FEATURES
OPTIONAL INPUT AND OUTPUT LABELING, MINIMUM MACHINE REQUIRE-
MENTS, 80 CHARACTERS, A TCP.S, A CD. READER OR 8 TAPES, PRINTER,
FEATURES ALLOW TWO OR MORE SUBROUTINES TO SPECIFY ALL SORT PARAMETERS.
SPECIFIED LEVELS MAY BE DELETED FROM THE FILE. DUPLICATE
RECORDS ARE SUMMARIZED OUT.

0709 1160MSAT
AVAILABLE PRIOR TO JANUARY 1962

0709 1160MDAT
RESTART PROGRAM FOR MD SORT.
USED TO RESTART A SORT AT THE BEGINNING OF ANY PHASE OR MERGE
PASS, NEEDED CHECKPOINT TAPE INTO CORE AND CHECKS THE TAPE
TRANSMISSION.

0709 1163MPATC
AVAILABLE PRIOR TO JANUARY 1962

0709 1159MPATC
FORTRAN CARD OR TAPE FROM AND/OR COLUMN BINARY LOADER.
LODS FORTRAN PROGRAMS FROM TAPE, FROM CARDS, OR FROM FIRST
CARDS THEN TAPES, BASICALLY AN EXTENSION OF THE F2 BSS LOADER,
THE PROGRAM ALLOWS LOCAL CORRECTION AND COMMENT CARDS AT
OBJECT ITEMS, AND ADDITIALLY THESE ON- OR OFF-LINE A
MAP OF MEMORY ALLOCATION IS ALSO OPTIONAL. CARDS
MAY BE IN ROW OR COLUMN BINARY FORM OR A MIXTURE OF
BOTH.

0709 1154WFOO
AVAILABLE PRIOR TO JANUARY 1962

0709 1160WAAD
INTERCEPTS, IF THE CURRENT TYPE OF RECORD IS THE
APPLET RECORD, DISREGARD ANY TAPES ON TAPE.
WRITE COPY OF MEMORY, AS IT IS IF IT IS ENCOUNTERED
DURING READING BY F3, PRECEDED BY A SELF-LOADING TAPE
READING PROGRAM, SO THAT THE TAPE MAY BE FREELY
RENEWED AND BAD RECORDS ARE IGNORED, IF IT IS TAPED.
COMMENTS, IF ANY, ARE WRITTEN ON THE TAPE.

IDM 0709 PROGRAM LIBRARY ABSTRACT

0709 1159MPATC
AVAILABLE PRIOR TO JANUARY 1962

0709 1158MPATC
CPMFT = GENERAL PURPOSE LANGUAGE FOR SYMBOL MANIPULATION
USING FOR READ-NON-MODIFIED PROGRAMS: TRANSLATION
INFORMATION RETRIEVAL, DICTIONARY WORK, FILE MAINTENANCE AND
SEARCH, FORMATTING, PROGRAM TESTING, SIMULATION, GAMES,
PLAYING, TEXT PROCESSING, DATA REDUCTION, ARTIFICIAL INTELLI-
IGENCE, ETC. A CONVENIENT, HIGH-LEVEL LANGUAGE EASY TO USE
AND QUICK TO CHECK OUT. FEATURES DIRECTNESS OF EXPRESSION,
EASY USE OF MESSAGES, BUILT-IN PUSH DOWN LISTS AND ADDRESS-
ABLE STORAGE, PRECISION FROM FIXED FORMAT AND WORD-LENGTH
SPECIFICATIONS, AUTO. INTERNAL STORE ALLOCATION 1272

0709 1201NMDV
SINGLE PRECISION TO DOUBBLE PRECISION FORTRAN INPUT
ALLOWS A FORTRAN PROGRAMMER TO READ A SINGLE PRECISION
NUMBERS - WITH K DECIMAL DIGITS FAMING K IS EQUAL TO OR LESS
THAN 25 WITH EXPONENT E FAMING K IS EQUAL OR LESS THAN 17
ACCORDING TO A SPECIFIED CARD FORMAT - AND TO CONVERT
YESE DECIMAL NUMBERS TO DOUBLE PRECISION NUMBERS.
SHOULD BE USED ONLY WITH THE ROCKETDYNE JSHARE CODE N/A
DOUBLE PRECISION PACKAGE MPP.

0709 1202NMDV
DOUBLE PRECISION OUTPUT FOR FORTRAN
ALLOWS A FORTRAN PROGRAMMER TO CONVERT A DOUBLE PRECISION
NUMBER TO N/K EQUAL TO OR LESS THAN 22 DECIMAL DIGITS WITH
EXPONENT AND E IN THE 4TH HOUR ACCORDING TO A SPECIFIED FORMAT.
SHOULD BE USED ONLY WITH THE ROCKETDYNE JSHARE CODE N/A
DOUBLE PRECISION PACKAGE MPP.

0709 1215AAO073
DOUBLE PRECISION POLYNOMIAL ROOT EXTRACTION PROGRAM
EXTRACTS THE ROOTS OF AN N ORDER POLYNOMIAL WITH REAL
COEFFICIENTS. IT CANNOT EXCEED 17 DIGITS IN THE DOUBLE
PRECISION MODE.

0709 1222WHDUL
HOLLERITH WORD GENERATOR
SUBROUTINE HOLRITH FACILITATES THE HANLING OF HOLLERITH
CHARACTERS IN A FORTRAN PROGRAM, IT PLACES A STRING
OF HOLLERITH CHARACTERS INTO A ONE-DIMENSIONAL ARRAY SO THAT
THE USER CAN REFER TO THE STRING BY REFERRING TO THE NAME
OF THE ARRAY, INDEXES 15 LOCATIONS IN CORE STORAGE.
LISTING INCLUDED IN SHORT WRITE-UP.
APWRC-SYNFAR

(1) Code Originated by:
The Martin Co. (Baltimore)

(2) Computer:
709 (FORTRAN II and FAP)

(3) Description of Code:
This code does a synthesis computation of the static flux and reactivity, or of the stable period and corresponding flux shape, in XY or RZ geometry. A direct computation of the same quantities is made in one-dimensional spherical geometry. It is assumed, in two-dimensional problems, that the flux is separable in the two perpendicular directions. One-dimensional calculations are carried out alternately in each direction, and are coupled through lithargy dependent bucklings.

(4) Restrictions or Limitations:
A 32K memory with ten tape units. For transport calculations, two or three groups may be used, and $F_1$, $S_2$, $S_3$, and $S_4$ calculations may be made. The $S_{54}$ calculation may not be done in spherical geometry. Up to 199 space intervals in each direction.

(5) Approximate Performance:
12 minutes on the 709 for 3 passes on a right-circular cylinder with homogeneous core and reflector.

(6) References:
2. D. H. Frederick, "APWRC-SYNFAR, A FORTRAN II Program for Two-Dimensional Static or Dynamic Synthesis Using $F_1$ or $S_{54}$ Flux or Adjunct in Slab, Cylinder, or Spherical Geometry", MND-C-2460, 1961.

(7) Material Available:
1. SYNFAR-01 Binary Deck.
2. SYNFAR-01 Tape (5 files).
   File 1 Nuclear Data Tape (Binary).
   File 2 Source Listing (BCD).
3. Sample Problem Input Decks.
4. Sample Problem Output Listings.
5. MND-C-2460.

Notes: 1. The above information was taken from Reference 1.
   2. This code was contributed through the Argonne Code Center.
IBM 1401 PROGRAM LIBRARY ABSTRACT

**MASCOT (Modified Assembly System Converted to Tape)**

Aaron C. Williams
IBM Corporation
340 Market Street
San Francisco 11, California

**Purpose:** This program is a variation of the MSL SPS I system that uses magnetic tape to store intermediate results rather than punched cards.

**Method:** Source Language 1401 Symbolic Programming System.

**Restrictions / Range:** Reiteration is possible with 200 labels.

**Storage Requirements:** Not Given.

**Equipment Specifications:** 4K Model C 1401 with High - Low - Equal Compare, six sense switches and advanced programming.

IBM 1401 PROGRAM LIBRARY ABSTRACT

**CARAT I**

Aaron C. Williams & Jacob McElmell

**Direct Inquiries to:** Mr. Aaron C. Williams
IBM Corporation
340 Market Street
San Francisco 11, California

**Purpose / Description:** CARAT I automates the 1401 SPS Assembly process. It allows the user to assemble a number of source programs sequentially as they are "stacked" in the 1402 Card Reader, without subsequent card handling or operator intervention. The output "object program" can be prepared in the form of punched cards, magnetic tape, or both.

**Method:** N/A

**Restrictions / Range:**
1. A maximum of 260 labels per program assembled.
2. Each program to be assembled must have a CTL and END card.
3. The CTL card should not specify a 1.4K processor.

**Storage Requirements:** N/A

**Equipment Specifications:** 4K Model C Tape System with Store B-Address Register feature, and High-Low-Equal compare. 3 Model 729 or 7330 Tape Drives.

IBM 1401 PROGRAM LIBRARY ABSTRACT

**CARAT II**

Aaron C. Williams & Margery C. Redahl

**Direct Inquiries to:** Mr. Aaron C. Williams
IBM Corporation
340 Market Street
San Francisco 11, California

**Purpose / Description:** CARAT II automates the 1401 SPS Assembly process. It allows the user to assemble a number of source programs sequentially as they are stacked in the 1402 Card Reader, without subsequent card handling or operator intervention. The output, object programs, can be prepared in the form of punched cards, magnetic tape, or both.

**Method:** N/A

**Restrictions / Range:** Assembly time is reduced by at least 40%. An even greater saving accrues when assembling small decks. Post Listing from tape allows the printer to run at maximum speed during the listing operation.

**Storage Requirements:** A Clear Storage and Post Listing Punch routine have been added to the systems tape.

**Equipment Specifications:** 4K model C tape system, with Store B Address Register feature, and High-Low-Equal compare. Three model 729 or 7330 tape drives.

(Continued on next column)
IBM 1401 PROGRAM LIBRARY ABSTRACT  File Number  1.2.001

SORT 1401

Mr. Hal Durette
IBM
140 Market Street
San Francisco, Calif.

Purpose:  To perform a two- or three-way sort on 4K to 16K utilizing the advantages of the advanced programming feature.

Method:  Source Language 1401 SP5.

Restrictions, Range:

a) counts the number of blocks written in Phase 1 and checks this during all merge passes.

b) a given number of records may be sorted in 25-50% less time than if sorted by Sort 1.

c) analyst must scale blocking to equal blocking by considering number of character/record.  No variable output blocking.  A maximum of two records is required, however, there is room in Phase 1 to modify so that single records may be read and blocked for the internal sort.

d) padding the last block with records with blanks or nines in the control field has to be done before the sort.

e) maximum block length

<table>
<thead>
<tr>
<th>4K</th>
<th>8K</th>
<th>12K</th>
<th>16K</th>
</tr>
</thead>
<tbody>
<tr>
<td>3-way</td>
<td>1884</td>
<td>2520</td>
<td>2520</td>
</tr>
<tr>
<td>2-way</td>
<td>1884</td>
<td>2520</td>
<td>2625</td>
</tr>
</tbody>
</table>

f) there is a provision in Phase 2 to collate a sorted reel with same specifications (record length, blocking length, control field) with the records that are presently being sorted.

g) a fixed control field of any number of characters is possible.

Storage Requirements:  There are approximately 1291 positions of memory used for the Phase 1 program.

Equipment Specifications:  Minimum 4K 1401 with H-L-E Compare Feature, Advanced Programming Feature and 4 or 6 729 II or IV.

IBM 1401 PROGRAM LIBRARY ABSTRACT  File Number  1.3.002

1401 Generalized Merge Program for Unblocked Records  J. E. Carreka & P. MacGregor

Direct Inquiries to:  J. E. Carreka & P. MacGregor
IBM Corporation
590 Madison Avenue
New York, N.Y.

Purpose/Description:  This merge program is specifically designed to merge files of any type of unblocked record on a 1401 tape system.

Method:  The merge consists of two phases:  the assignment phase, and the merge program.

The assignment phase initializes and optimizes the merge program on the basis of information supplied by the user on a control card.

The merge program tests, by means of a comparison loop, for the low record of those currently contained in storage.  When the low record is found, it is written on the output tape, the file from which it came is read up, and the program returns to the comparison loop.  Records are checked for sequence, redundancies, correct length, etc.

Restrictions/Range:  

<table>
<thead>
<tr>
<th>Number of files</th>
<th>Maximum</th>
<th>Minimum</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Number of cards per file</td>
<td>9</td>
<td>1</td>
</tr>
<tr>
<td>Record length (number characters)</td>
<td>997</td>
<td>10</td>
</tr>
</tbody>
</table>

(Continued on next column)

IBM 1401 PROGRAM LIBRARY ABSTRACT  File Number  1.3.001

CARD REPORT PROGRAM GENERATOR AND AUTOCORDER ASSEMBLY  J. L. Dorsey

Direct Inquiries to:  Mr. J. L. Dorsey
IBM Corporation
Time-Life Building
1271 Avenue of the Americas
New York, New York

Purpose/Description:  The purposes of this program is to lessen machine time required for generation and assembly of a program generated by the standard CRPG deck.  Autocoder is automatically read in and assembly takes place with no card handling by the operator, (the generated symbolics are written on tape and not punched).

Mathematical Method:  Does not apply

Restrictions/Range:  Does not apply

Storage Requirements:  Does not apply

Equipment Specifications:  For generation and assembly, same requirements as for Autocoder.  For execution of the generated program, any 1401 card system whose storage capacity will accommodate the program.

IBM 1401 PROGRAM LIBRARY ABSTRACT  File Number  1.3.002

1401 TAPE REPORT PROGRAM GENERATOR AND AUTOCORDER ASSEMBLY  J. L. Dorsey

Direct Inquiries to:  Mr. J. L. Dorsey
IBM Corporation
Time-Life Building
1271 Avenue of the Americas
New York, New York

Purpose/Description:  The purpose of this program is to lessen machine time required for generation and assembly of a program generated by the standard CRPG deck.  Autocoder is automatically read in and assembly takes place with no card handling by the operator, (the generated symbolics are written on tape and not punched).

Mathematical Method:  Does not apply

Restrictions/Range:  Does not apply

Storage Requirements:  Does not apply

Equipment Specifications:  For generation and assembly, same requirements as for Autocoder.  For execution of the generated program, any 1401 card system whose storage capacity will accommodate the program.

IBM 1401 PROGRAM LIBRARY ABSTRACT  File Number  1.3.003

GENERAL PURPOSE TAB-BACK PROGRAM  Bernard T. Smith

Direct Inquiries to:  Bernard T. Smith
The Warner Brothers Company
225 Lafayette Street
Bridgeport 1, Connecticut

Purpose/Description:  To provide tabulations or listings of summary cards or initial data cards for control and verification purposes.

Method:  This method of instructing the machine as to the various card formats was chosen because of its simplicity and flexibility.

Restrictions/Range:  This program may have the following:

(Continued on next page)
CORRECTION CARD LOADER

F. E. Johnston
IBM
3500 Central Avenue, S.E.
Albuquerque, New Mexico

Purpose: To alter a 1401 program after it is loaded. Corrections will be punched with one instrument or up to 31 character of data per card. The instruction cards will contain the length of the instruction, location to be loaded and the instruction. The location as well as the A and B address of the instruction may be actual machine language or 4 digit addresses.

Method: Source Language SPS.

Storage Requirements: Not given.

Equipment Specifications: Standard 1401 with 1600 positions of core storage.
No special features needed.

IBM 1401 PROGRAM LIBRARY ABSTRACT
File Number 1, 4, 002

CALL (Carat Assembled Logical Loader)

Robert W. Heald
IBM
1401 Market Street
San Francisco, California

Purpose: The CALL program loads the CARAT (1,1,002) assembled programs directly from tapes into the 1401. Thus object program decks need not be punched until the programs are completely "debugged".


Restrictions, Range: a) When used with CARAT, as much as 75% of the machine time required to assemble and test a program can be saved.
b) The CALL program provides for patching.

Storage Requirements: Not given.

Equipment Specifications: 4K Model C 1401 with High - Low - Equal Compare and six sense switches.

IBM 1401 PROGRAM LIBRARY ABSTRACT
File Number 1, 4, 002

CARD REPRODUCING AND/OR LISTING PROGRAM FOR THE IBM 1401

(Continued on next page)
IBM 1401 PROGRAM

Purpose:

256

IBM 1401 PROGRAM

Purpose:

1401 TCS (Tape Control System)

Catherine Bellbeck

IBM Corporation

3424 Wilshire Blvd.

Los Angeles, California

Equipment Specifications:

IBM 1401 PROGRAM LIBRARY ABSTRACT

File Number 1.4.007

IBM 1401 PROGRAM LIBRARY ABSTRACT

File Number 1.4.006

IBM 1401 PROGRAM LIBRARY ABSTRACT

File Number 1.4.008

IBM 1401 PROGRAM LIBRARY ABSTRACT

File Number 1.4.005

IBM 1401 PROGRAM LIBRARY ABSTRACT

File Number 1.4.010

IBM 1401 PROGRAM LIBRARY ABSTRACT

File Number 1.4.015

IBM 1401 PROGRAM LIBRARY ABSTRACT

File Number 1.4.020

IBM 1401 PROGRAM LIBRARY ABSTRACT

File Number 1.4.025

IBM 1401 PROGRAM LIBRARY ABSTRACT

File Number 1.4.030

IBM 1401 PROGRAM LIBRARY ABSTRACT

File Number 1.4.035

IBM 1401 PROGRAM LIBRARY ABSTRACT

File Number 1.4.040

IBM 1401 PROGRAM LIBRARY ABSTRACT

File Number 1.4.045

IBM 1401 PROGRAM LIBRARY ABSTRACT

File Number 1.4.050

IBM 1401 PROGRAM LIBRARY ABSTRACT

File Number 1.4.055

IBM 1401 PROGRAM LIBRARY ABSTRACT

File Number 1.4.060

IBM 1401 PROGRAM LIBRARY ABSTRACT

File Number 1.4.065

IBM 1401 PROGRAM LIBRARY ABSTRACT

File Number 1.4.070

IBM 1401 PROGRAM LIBRARY ABSTRACT

File Number 1.4.075

IBM 1401 PROGRAM LIBRARY ABSTRACT

File Number 1.4.080

IBM 1401 PROGRAM LIBRARY ABSTRACT

File Number 1.4.085

IBM 1401 PROGRAM LIBRARY ABSTRACT

File Number 1.4.090

IBM 1401 PROGRAM LIBRARY ABSTRACT

File Number 1.4.095

IBM 1401 PROGRAM LIBRARY ABSTRACT

File Number 1.4.100

IBM 1401 PROGRAM LIBRARY ABSTRACT

File Number 1.4.105

IBM 1401 PROGRAM LIBRARY ABSTRACT

File Number 1.4.110
IBM 1401 PROGRAM LIBRARY ABSTRACT

**FTTS** (Fourteen-One Input-output Tape-control System)

E. J. MacArthur

**Purpose/Description:** This program supplies Open, Close, Get, and Put closed subroutine to users wanting the full 10065 package for 1401 Autocoders. In addition, it supplies the advantages of an IOCS compatible package to users who are unable to assemble Autocoder due to their system's configuration (less than 4 tape drives).

**Method:** FTTS has been written in two source languages, aimed at the two groups mentioned in the "Purpose" paragraph. FTTS is written in 1401 Autocoder. FTTS II is written in SPS II.

**Restrictions/Range:**
1. Since the header labels are processed in the punch area, the use of Punch Feed Read requires patching.
2. Writing is in the Move Mode only.
3. Header and trailer labels are always written on the output files.
4. Input files by column may be acceptable with an ending header label.
5. The FTTS subroutines provide the following:
   - A. Open
     1. Input File: Checks file ID name and reel number.
     2. Output File: Checks creation date and retention cycle.
   - B. Get
     Places the next record in a work area for use by the program. All tape reading, de-blocking, error routines and end of real conditions are taken care of by the subroutine.
   - C. Put
     Moves each record sequentially from a work area to a blocking area, automatically writing in tape when the blocking area is full. All error routines are taken care of by the subroutine. A trailer label is written, a status card is punched, and a new reel is opened when an end of real condition occurs.
   - D. Close
     Processes the end of file trailer label and removes the tape from use.

**Storage Requirements:** Approximately 1700 positions.

**Equipment Specifications:** 1401 Model C, D, E, F 13-18, or F 23-28, Advanced Programming Package, High-Low-Equal Compare

**IBM 1401 PROGRAM LIBRARY ABSTRACT**

**SCOOT I and II**

Robert E. Engelson & Louis P. Paulin

**Purpose/Description:** To provide a simple method of converting 90 column (or more) cards to 80 column cards (or magnetic tape) in ascending sequence.

**Method:** The use of SCOOT specifies in Column Control Cards each column to be translated FROM and TO. A Translation Table control card permits complex control over character translation. The user must program his own output routine and assemble it with SCOOT. Program Exit and Entry points have been provided for this purpose.

**Restrictions/Range:** Field tests and actual customer conversion usage have proven that inverted 90 column cards and blank cards can be accurately read in a 1402 read feed when it is properly adjusted for normal 80 column card reading. Vertied 90 column cards have an elongated hole. To prove accuracy of conversion, control totals should be taken prior to translation and a control total routine should include as part of the output routine.

**Storage Requirements:** 4,000 positions of storage (Continued on next page)
IBM 1401 PROGRAM LIBRARY ABSTRACT

File Number: 1.4.010

TRAP (Tape Record Analyzer Print)

W. J. Wilson & C. L. Craig

Direct Inquiries to: W. J. Wilson & C. L. Craig
Computation Division
Huntsville Computer Center
Huntsville, Alabama

Purpose/Description: To automatically analyze and print at 600 lpm in optimum readable form the contents of a magnetic tape written in BCD mode.

Method: This program reads, analyzes, and prints tape records maintaining vertical alignment of equivalent fields from record to record and blocks to block which avoids the staggered print pattern associated with most tape print programs. This program handles both variable and constant length, single and blocked records which may be intermixed on tape. No parameters are required as the program is completely generic. A count representing the actual position of the last character of each line printed is maintained on the right margin - print positions 175-132. To indicate the last portion of each tape record printed the notation RAPREC is appended to the left of the count. The following options are included: The ability to interrupt, to print multile record, and to simulate end-of-file at any time.

Restrictions/Range: Tape records of length greater than 2500 characters will have only the first 2500 characters printed.

Storage Requirements: 4K

Equipment Specifications: Advanced programming features, High, Low Equal Compare.

IBM 1401 PROGRAM LIBRARY ABSTRACT

File Number: 1.4.010

SD 1202 (Search Program-Cmd Version)

Fred G. Stockton

Direct Inquiries to: Fred G. Stockton
Shell Development Company
4000 Horton Street
Emeryville, California

Purpose/Description: This program searches a deck of IBM cards (library deck) for cards which meet any of optimally all of a number of criteria.

The criteria are specified in a simple code on set-up cards placed in the library deck. Matched cards are counted for the criteria which they satisfy. Optionally they may be printed, a replica may be punched, or the machine may be stopped for examination of the original card. At the end of the run a summary of the card count for each criterion is printed.

The program is used for information retrieval, especially in improperly situations, and for descriptive statistical purposes. It can effectively simulate the searching and counting functions of the IBM 51.

Method: A "finder" card identifies those punches of the 99 possible punches on an IBM card which are referred to by any of the criteria. "Finder" cards carry the codes for the criteria. Each coded criterion refers to all the punches on the "finder" card and may demand that a punch be present or absent, or ignore its presence or absence, or demand the presence or absence of some digit of a group of punches. The program constructs a coded "signature" for each library card, and compares it with the "name" to see if there is a match. Output and other options are controlled by input indicators, or by sense switches.

Restrictions/Range: No more than 120 punches on the "finder" card, and therefore no more than 100 characters in any "name". No more than 1000 characters for all "names" together. Cards are counted separately for the first 40 criteria; card counts for higher numbered criteria are lumped together.

Storage Requirements: 276 positions

Equipment Specifications: 4000 core-storage positions
1403 Printer
1403 Card Read-Punch
Advanced Programming Features
High,Low Equal Compare
Column Binary Feature
Sense Outlining

Additional Hardware: The speed is 400 cards per minute for squashed cards, for the simplest cards. At least 1100 cards per minute for the clearest cases.
IBM 1401 PROGRAM LIBRARY ABSTRACT

File Number 2.0.031

IBM 1401 TAPE LIBRARY CONTROL SYSTEM
Robert W. Heald

Direct Inquiries to: Mr. Robert W. Heald
IBM Corporation
1415 15th Street
Sacramento 16, California

Purpose/Description: To insure the proper mounting of magnetic tapes for each
machine run and to facilitate the maintenance of the tape library. To eliminate
the necessity for coding tape error routines. To provide end of reel and end of
file logic in a routine manner.

Method: N/A

Restrictions/Range: N/A

Additional Remarks: The program

IBM 1401 PROGRAM LIBRARY ABSTRACT

File Number 2.0.025

ASC SYSTEM (Aeronutronic Simplified Coding System)
S. Schlesinger & L. Sashkin

Direct Inquiries to: S. Schlesinger & L. Sashkin
Aeronutronic, A Division of Ford Motor Company
Ford Road
Newport Beach, California

Purpose/Description: To eliminate the requirement for hard computation using a
desk calculator and sets of tables by a method which is more reliable and less
costly.

Method: Does not apply

Restrictions/Range: Does not apply

Storage Requirements: 4000 positions of storage. Models C3 or E3 equipped with
multiply and divide, Advanced Programming Feature, and two magnetic tape units.

Equipment Specifications: Model C3 or E3

Additional Remarks: If a program is less than 350 ASC instructions and no
instruction blocks are stored on magnetic tape, then only one tape unit is needed.

IBM 1401 PROGRAM LIBRARY ABSTRACT

File Number 3.0.025

9 x 9 TEN MILLISECOND MULTIPLY SUBROUTINE

Mr. Richard B. Fauster & Mr. William H. Post
140 Market Street
San Francisco 2, California

Purpose: This program will multiply two nine position fields together, with
sign control, in significantly less time than previous programs.

Method: Source Language SPS.

Restrictions/Range: Timing 10 m.s. per multiplication.

Storage Requirements: 324 Positions.

Equipment Specifications: 1401 - any model, no special features required.

IBM 1401 PROGRAM LIBRARY ABSTRACT

File Number 3.0.025

SCION (Scientific 1401 Programming with Floating Point)

John Discola
IBM
9250 Wilshire Blvd.
Beverly Hills, California

Purpose: SCION provides the programmer with closed floating point subroutines.
The subroutines include the normal arithmetic operations in addition to mode-
conversion type operations. The programmer is also afforded the option of
utilizing one of three sizes of floating point mantissa - namely, 4, 8, and 16
digits. This gives what normally would be termed 6, 30, and 14 digit floating
point. The subroutines are mapped so that modular utilization is possible in
those cases where some additional memory space is needed.

Method: Source Language SPS-I

For those who prefer to code with pseudo-hardware instructions, a pre-assembly
program is provided that edits a source program at the SPS level and creates
the required linkage for the floating point operations written in macro form.

Restrictions: Range:

Two digit characteristic (excess-50) gives the following ranges for
floating point operations.

10 digit: 
1000 x 10^-50 to 999999999 x 10^-49

4 digit: 
1000000000 x 10^-49 to 9999999999 x 10^-49

Accuracy: Subroutines truncate significant digits of result after normalizing.

Storage Requirements:

Total package
6 digit: positions 0333 thru 140
10 digit: positions 0333 thru 1172
14 digit: positions 0333 thru 1204.

Scion packages are not restricted to memories larger than 4K since the Modify-
Address (MA) instruction peculiar to the larger memory configurations is not
used in any of the subject routines.

Index registers 1 and 2 are used by the subroutines. This should not concern
the programmer because they are restored to the entry condition at exit time.

One proton is made however, mainly - that word marks are not left in their
locations at entry time.

Equipment Specifications: IBM 1401 B, C, D, or E with the following special
features:
1) Multiply-Divide
2) Advanced Programming Package
3) Hi-Lo-Equal Compare.

IBM 1401 PROGRAM LIBRARY ABSTRACT

File Number 3.0.003

SQUARE ROOT SUBROUTINE

Kenneth Johnson

Direct Inquiries to: Kenneth Johnson
Bureau of Public Roads
Department of Commerce
Washington 25, D. C.

Purpose/Description: Computes the Square Root of a single-precision fixed
point 19 digit number.

Mathematical Method: Accuracy = 

Restrictions/Range: 999999999999999999999999

Storage Requirements: 344 positions of core storage.

Equipment Specifications: Maximum 1401 with automatic multiply-divide and
high, low, equal compare features.

Additional Remarks: This routine was converted directly from a modification
of the routine in the original 650 manual. It can be incorporated with other
programs without modification.

IBM 1401 PROGRAM LIBRARY ABSTRACT

File Number 2.0.004

1401 FLOATING POINT SUBROUTINES (Normalized)
St. H. Horci

Direct Inquiries to: Horace P. Hori
U. S. Department of Commerce
Bureau of Public Roads
Washington 25, D. C.

Purpose/Description: Computes floating point add, add absolute, subtract,
subtract absolute, multiply, and divide.

Mathematical Method: N/A

Restrictions/Range: 00 00 00 00 00 to 99 99 99 99 99

Storage Requirements: 806 cores of memory

Equipment Specifications: Any size 1401 with index registers, multiply-divide,
High-Low-Equal Compare.

(Continued on next page)


**IBM 1401 PROGRAM LIBRARY ABSTRACT**

**1401 SIN-COS SUBROUTINE**

Kenneth Johnson

Direct Inquiries to: Kenneth Johnson

U. S. Department of Commerce

Bureau of Public Roads

Washington 25, D. C.

Purpose/Description: Computes SIN and/or COS converting degrees to radians producing a nine decimal place result.

Method: Hastings Approx. Result in location 1689 with sign in units position.

Restrictions/Range: N/A to 359.9 degrees

Storage Requirements: Approximately 708 positions of core storage.

Equipment Specifications: Minimum 1401 with automatic multiply-divide and high, low, equal compare features.

Additional Remarks: This routine was converted directly from a modification of the subroutine No. 9 by G. R. Tryble. It can be incorporated in other programs with only modification of sample exit instructions.

**IBM 1401 PROGRAM LIBRARY ABSTRACT**

**DIVERSITY STUDY**

Henry L. Schmitz, Jr.

Direct Inquiries to: Mr. Henry L. Schmitz, Jr.

Systems Engineer-Scientific

IBM Corporation

273 State Street

Springfield, Massachusetts

Purpose/Description: Analysis of customer demand to determine the following:

1. Maximum demand for each customer
2. Maximum Coincident Demand for 1, 2, 3, ... N customers where N is the number of customers in the sample.
3. Coincidence Factors for 1, 2, 3, ... N customers

Mathematical Method: Not pertinent

Restrictions/Range: N/A

Storage Requirements: 4000 positions of storage

Equipment Specifications:

1. 3 tapes
2. Advanced programming
3. Multiply-Divide
4. High-Low-Equal Compar
5. Card input-output
6. Expanded print storage

Additional Remarks: Program handles 2 digit demand for up to 39 customers. Coincident demand cannot exceed 4 digits.

**IBM 1401 PROGRAM LIBRARY ABSTRACT**

**1401 LINEAR PROGRAM**

Harm K. Schreur

IBM

284 Cedar Springs Road

Dallas 19, Texas

Purpose: This program attempts to obtain a maximum functional of A unknowns in B equations.

Method: The Simplex method, such as described by Charnes, Cooper and Henderson (Wiley and Sons - An Introduction to Linear Programming) is used to obtain the Maximal.

Restrictions/Range: A 1401 Model B3 or C3 system with 4000 core storage positions. Direct multiply, divide and the high-low-equal compare features (Continued on next column)

will accommodate a matrix, subject to the following restrictions:

\[ B(WL)(A+1)(B+1) \leq 259.9 \]

where

- \( B \) is the number of rows in the matrix,
- \( A \) is the number of columns in the matrix, and
- \( WL \) is the number of digits in the elements.

Storage Requirements: Not given.

Equipment Specifications:

- A 1401 Model B3 or C3 system with 4000 core storage positions.

**IBM 1401 PROGRAM LIBRARY ABSTRACT**

**717/720 SIMULATION ON 1401**

W. Stokes

IBM

425 Park Avenue

New York, New York

Purpose: To achieve maximum 1403 print speed while printing tapes originally prepared for "off line" use on IBM Tape 717 and 720 printers.

Method: Not given.

Restrictions/Range: Tape records must be 1000 characters or less in length. Blocked data records must be separated by a record mark. (Last data record may or may not end in a record mark).

1. Accepts single fixed or variable length records with or without a record mark in terminal position.
2. Accepts blocked fixed or variable length records, each data record must be separated by a record mark, however last data record may or may not have a record mark in terminal position.
3. Number of data records per block is unlimited, however total length may not exceed 1000 characters.
4. Files may be: unlabeled, labeled followed by tape records, labeled followed by T/M followed by tape records.
5. Multifle reels may be printed.
6. No control cards required.

Storage Requirements: 4000 positions of memory - approximately 700 positions available for patching.

**IBM 1401 PROGRAM LIBRARY ABSTRACT**

**1401 PROGRAM LIBRARY**

File Number 10.1.005
specifications for solution of unit card algebraic equations. The object program generated by BANG requires no manual insertions, modifications or patching. This new package includes all the functions of BANG 1, 2, 3 plus the subroutine option.

Method: An optional code has been added to the specifications cards of BANG. This code is the means of requesting BANG to include, within the generated object program, closed multiplication and division subroutines with all required entry and return linkage. If the user W1 O is equipped with the high-speed trigger feature, he can so specify and BANG will not generate the subroutine.

Restrictions/Ranges: The subroutines incorporated by BANG to the object program are non-dimensional in that there is no limit to the size of the product or quotient developed. Each subroutine is completely self-initializing based on the parameters of the factors involved. At the completion of multiplication and/or division, the B-field contains the product, or quotient, and remainder positioned with associated signs exactly as though the Multiply-Divide feature had been used.

Storage Requirements: 4,000 positions of core are required to generate object program with BANG. The generated and then assembled program will require core capacity directly related to the complexity of the problem.

Equipment Specifications: Card 1401 with 5K memory; Hi-Lo-Equal compare; read/punch feed are required for BANG operation.

IBM 1401 PROGRAM LIBRARY ABSTRACT

File Number 10.3.001

1401 LESS (Least-cost Estimating and Scheduling) 16K
Lou Granato, Jim Borden, and Joe Rose

Direct Inquiries to: Lou Granato, Jim Borden, and Joe Rose
IBM Corporation
631 Cooper Street
Camden 2, New Jersey

Purpose/Description: This program is a high speed method of determining the critical path and related information (float time etc.) for problems where scheduling is important.

Method: Not available.

Restrictions/Ranges: This program will handle 575 events (node points), any number of jobs (arrows). The length of the critical path cannot exceed 6 digits (999999).

Storage Requirements: 4,000 positions of core required. Will handle 575 events in approximately ten minutes including card handling time. This is a (3) Phase, three (3) pass program.

Equipment Specifications: Basic 1401 Card System
4,000 positions of storage
No special features required

IBM 1401 PROGRAM LIBRARY ABSTRACT

File Number 10.3.002

1401 LESS (Least-cost Estimating and Scheduling) 32K, 12K, 16K
Lou Granato, Jim Borden, and Joe Rose

Direct Inquiries to: Lou Granato, Jim Borden, and Joe Rose
IBM Corporation
631 Cooper Street
Camden 2, New Jersey

Purpose/Description: This program is a high speed method of determining the critical path and related information (float time etc.) for problems where scheduling is important.

Method: Not available.

Restrictions/Ranges: The program will handle:

8K Memory - 595 Events
12K Memory - 1555 Events
16K Memory - 2155 Events

*Any number of jobs (arrows) can be handled. Length of the critical path cannot exceed 7 digits (9999999).

Storage Requirements: 8, 12, or 16 thousand positions of core required. Will handle 1000 arrows in approximately 12 minutes including card handling time. This is a three (3) Phase, two (2) Pass program.

Equipment Specifications:
1401 Card System with 8, 12 or 16K memory
Multiply Divide Feature
Hi-Lo-Equal Compare

IBM 1401 PROGRAM LIBRARY ABSTRACT

File Number 11.3.001

Solution of an Equation with Newton-Raphson's Method on the IBM 1401
Hans Johansson

Direct Inquiries to: Hans Johansson
IBM Sweden
Fack
Stockholm 36, Sweden

Purpose/Description: A demonstration program which solves the non-linear equation, \( \frac{d}{dx} \left( x^2 - 2 \right) = 0 \) with regard to \( x \) by use of floating point arithmetic.

Mathematical Method: The Newton-Raphson's iterative method is used. All arithmetic calculations are executed in floating point arithmetic with six significant digits. The logarithm function is approximated with a formula taken from booklets "Approximations for Digital Computers."

Restrictions/Ranges: N/A

Storage Requirements: 6000 Storage Positions

Equipment Specifications:
IBM 1401 Model A3, B3, C3 or E3
equipped with the Expanded Print Edit feature
IBM 1402 Card Read Punch
IBM 1403 Printer Model 1

IBM 1401 PROGRAM LIBRARY ABSTRACT

File Number 11.5.002

NUMERICAL SOLUTION OF LEGENDRE'S DIFFERENTIAL EQUATION ON THE IBM 1401
Curt Kamlin

Direct Inquiries to: Curt Kamlin
IBM Sweden
Fack
Stockholm 36, Sweden

Purpose/Description: A demonstration program which computes and tabulates the Legendre functions \( P_n^m \).

Mathematical Method: Numerical integration of Legendre's differential equation

\[ \int \frac{d}{dx} \left( x^2 - 1 \right) P_n^m = x P_n^m + \left( 1 - \frac{m^2}{n^2} \right) P_{n+1}^m = 0 \]

in the interval

\[ 0 \leq x \leq 1 \]

and for

\[ m \leq n \]

by the Range-Kutta 2nd order method according to the scheme in figure 1. Integration step: 0.01.

Restrictions/Ranges: N/A

Storage Requirements: 2,800 positions

Equipment Specifications:
IBM 1401 with 6000 positions of core storage, sense switches and expanded print edit features, IBM 1402 Card Read Punch and IBM 1403 Printer, Model 1.

Additional Remarks: This program uses 2,800 storage positions compute and tabulates the Legendre functions \( P_n^m \) in 6.4 minutes by numerical solution of Legendre's differential equation.

IBM 1401 PROGRAM LIBRARY ABSTRACT

File Number 11.5.003

A PROGRAM FOR SOLVING SYSTEMS OF LINEAR EQUATIONS ON THE IBM 1401
Soren Nordin

Direct Inquiries to: Soren Nordin
IBM Sweden
Fack
Stockholm 36, Sweden

Purpose/Description: A program for solving linear equation systems. It is also well suited as a demonstration program.

Mathematical Method: The system of equation is solved using the elimination method. All arithmetic operations are performed in floating point numbers.
### IBM 1401 PROGRAM LIBRARY ABSTRACT

**File Number:** 11.0.004

**Title:** PRINTING THE CONSTANT e TO 10,096 DECIMALS AND TESTING THE RANDOMNESS OF THE DECIMALS

**Author:** Knut Angstrom

**Purpose/Description:** A demonstration program which uses the results from the famous calculation of $e^7$ on the IBM 704 in Paris to print all decimals, thereby showing the high speed printing. As an optional feature the randomness of the decimals can be tested.

**Mathematical Method:** The randomness is tested by using a common $X^2$ test.

**Restrictions/Range:** N/A

**Storage Requirements:** 1100 positions

**Equipment Specifications:**
- IBM 1401 Model C1
- IBM 1402 Card Read Punch
- IBM 1403 Printer
- One IBM 729 Tape Unit

---

### IBM 1401 PROGRAM LIBRARY ABSTRACT

**File Number:** 13.1.001

**Title:** 1401 Tape Duplication or Compare

**Author:** Dick Nichols

**Purpose:** This program permits multi-file duplication or Compare of Binary and BCD information. The information may be in mixed or single mode.

**Method:** The Tape Duplication reads a physical record BCD and/or Binary and writes it out on another tape. With the settings of sense switches and/or control cards it can duplicate single or multiple files.

**Restrictions:** The following restrictions are applicable for this Duplicate and Compare Program:

1. When duplicating, input tape cannot exceed 3200 characters.
2. When comparing, block size input tape cannot exceed 1600 characters.
3. With a Control Card up to 999 files may be duplicated.
4. Tape drive I must be used for input.
5. Tape drive 2 must be used for output.
6. Control Card must follow last card of program deck.
7. If one file is to be duped or compared and sense switch "E" is used instead of control card, user cannot select file. Only the first file will be duped or compared.

**Storage Requirements:** Program occupies 800 positions in core. Storage requirements are any size system with the larger the system available the larger blocks can be duplicated (with little modification to program).

**Equipment Specifications:** Model C 3, 2 tape drives, optional column binary.

---

### IBM 1401 PROGRAM LIBRARY ABSTRACT

**File Number:** 13.1.002

**Title:** IBM 1401 PROGRAM LIBRARY ABSTRACT

**Purpose:** The program was written specifically to replace the SHARE 66 x 64 boards of the IBM 704 card reader. It will read cards (column binary or BCD) at full speed (800 cpm) and place them on tape with "look ahead" bits as described in the SHARE 709 Reference Manual. An "END OF FILE" Card is provided.

**Method:** Each card is read as a column binary card. If it has a 7-9 punch in column 1, it is treated as such; otherwise it is a Hollerith card and the normal read area is used. So that "look ahead" may be added, two cards are kept in core.

**Restrictions/Range:** This program has been written for a 4K machine with the read release feature, column binary read, and high-low-equal compare. One tape drive is required. Because each card is read as a binary card, validity checking is not in effect.

**Storage Requirements:** Not given.

**Equipment Specifications:** IBM 1401 4K with read release feature, column binary read, and high-low-equal compare.

---

### IBM 1401 PROGRAM LIBRARY ABSTRACT

**File Number:** 13.1.003

**Title:** 1401 Tape-to-Card Program

**Author:** R. F. Vorwald

**Purpose:** The program was written to punch, in the first 80 columns of a card, the corresponding positions of any tape (binary or BCD). Tape are provided at an end of file and at persistent tape read errors. In either mode, cards are punched at 250 cpm.

**Method:** Each record in read and tested for error. If in error, the mode is switched. This process is repeated until either a correct read or 10 errors occur in both modes. If the read is correct, reading continues in the same mode until another error occurs.

**Restrictions:** The program has been written for a 4K machine with advanced programming and the punch column binary feature. It will read a record of any length and punch only the first 80 columns. One tape drive is required.

**Storage Requirements:** Not given.

**Equipment Specifications:** IBM 1401 4K with advanced programming and the punch column binary feature.

---

### IBM 1401 PROGRAM LIBRARY ABSTRACT

**File Number:** 13.1.004

**Title:** IBM 1401 PROGRAM LIBRARY ABSTRACT

**Author:** Robert Kasemuri

**Direct Inquiries to:** Lloyd W. Green

**Purpose/Description:** This is a system where a minimum amount of operator intervention is required, which also obviates the need for the programmer to be present at the computer for his run. Tape data or master tapes will be created from cards as specified by the programmer thus eliminating the need to reserve or mount special input tapes for each run. The system will notify the user where the input tapes were created, give him a core dump of the object program, tape prints on whichever tapes he desires along with any printing his object program.

(Continued on next page)
IBM 1401 PROGRAM LIBRARY ABSTRACT

IBM 1401 PROGRAM TAPE WRITER

Direct Inquiries to: C. A. Irvine
Space Technology Laboratories Inc.
P.O. Box 95001
Los Angeles 45, California
O. S-5-4577

Purpose/Description: To write either an SPS or PEST produced 1401 absolute program on tape in a self-loading, self-starting format.

Method: The program to be written on tape is permitted to load in the normal fashion except that the transfer is not executed, but is simply read into the read area. A group-mark is inserted into and memory from 001 to the first group-mark word-mark is written with word-marks on logical tape 1. If sense switch B is on, the tape is not rewound before writing, and if sense switch C is on, it is not rewound after writing.

Restrictions/Range: This program will operate on any model C 1401 which has sufficient storage for the object program. The object program may contain at most one group-mark word-mark which must be in the highest addressed cell of the program. However, this group-mark word-mark is lost when the program is loaded from tape and replaced by a group-mark without word-mark. Thus if group-mark word-marks are required they should be constructed during execution. The program may not occupy cells 11-153 inclusive. Any word separate characters (1-7-8 punches, B-8-4-2-1 bits) will be lost during the process.

Storage Requirements: 40K.

Equipment Specifications: 1401 machine

IBM 1401 PROGRAM LIBRARY ABSTRACT

IBM 1401 PROGRAM DATA FILE SYSTEM FOR THE IBM 1401

Fred Barry

Purpose/Description: This provides a means for the generation of input tape for an IBM 1401 using master tape files on the 1401. It also provides for the generation and updating of these files and for the maintenance of usage statistics.

Method: The input tape is read initially ignoring all records until "Page 1" occurs in the proper locations. To avoid confusion of an assembly listing with another type "job" which might have "Page 1" in the same print positions, a search is then made for the alter number 1, 2 or 3 occurring in the first lookup record. Punching of the symbolic deck then commences with the first alter number encountered. At any time that the present alter number is not exactly one more than the immediately previous alter number, a "SPACE" card is punched. Usually, the punching of a card corresponding to the previous record occurs shortly after reading the present record. This is done so that a symbol attached to the first generated instruction of a MACRO may be correctly punched in the symbolic macro-generating card. The only special considerations for a given record are whether it was generated from a Remarks ("rem") card or has a "BCD" operation code. Punching of a deck will cease upon encountering an "EOD" card.

Restrictions/Range: The variable field of a source card must not have exceeded 101 characters. The output listing includes file count, block count, number of characters in each block, mode of the block and contents of the block.

Storage Requirements: Memory 4K. All programs are written in SPS.

Equipment Specifications: 1401 Model C-3

IBM 1401 PROGRAM LIBRARY ABSTRACT

IBM 1401 PUNCH A SCAT DECK

Chuck Holmes

Direct Inquiries to: J. J. Watson
McDonald Automation Center
P.O. Box 518
E. Levis St., Missouri

Purpose/Description: To punch a SCAT symbolic deck from a magnetic tape containing an SPS assembly listing.

Method: The input tape is read initially ignoring all records until "Page 1" occurs in the proper locations. To avoid confusion of an assembly listing with another type "job" which might have "Page 1" in the same print positions, a search is then made for the alter number 1, 2 or 3 occurring in the first lookup record. Punching of the symbolic deck then commences with the first alter number encountered. At any time that the present alter number is not exactly one more than the immediately previous alter number, a "SPACE" card is punched. Usually, the punching of a card corresponding to the previous record occurs shortly after reading the present record. This is done so that a symbol attached to the first generated instruction of a MACRO may be correctly punched in the symbolic macro-generating card. The only special considerations for a given record are whether it was generated from a Remarks ("rem") card or has a "BCD" operation code. Punching of a deck will cease upon encountering an "EOD" card.

Restrictions/Range: The variable field of a source card must not have exceeded 57 characters, e.g., it must have originally fitted into columns 16 to 72 of the symbolic source card.

Storage Requirements: 1-99, 101-180, 401-1445

Equipment Specifications: The following special features are needed:
1. Indicating
2. Core storage greater than 1,4K
3. Punch release
4. High-Low-Equal Compare

IBM 1401 PROGRAM LIBRARY ABSTRACT

IBM 1401 PROGRAM LIBRARY ABSTRACT

Direct Inquiries to: B. J. Manring
North American Aviation, Inc.
Dept. 97, Building 6
4500 East 5th Avenue
Columbus 15, Ohio

Purpose/Description: The purpose of this 1401 Utility is to have the facility of "dumping" the contents of magnetic tape; whether in BCD, or Decimal equivalent if in Binary. Output listing includes file count, block count, number of characters in each block, mode of the block and contents of the block.

Method: When initiating "DUMP 01" the tape may be moved forward or backward from the original position before printing begins. The first record read from a file is read in Binary Mode.

Restrictions/Range: The following restrictions are applicable to this program.
1. Maximum block size is 65536 characters in Binary 62240 cards. Records longer than the maximum will be truncated and treated as though they were exactly 65536 characters. No indication of the truncation will be given.
2. When sense switches D through G are open, switches E and F are not active.

Storage Requirements: Memory 4K. All programs are written in SPS.

Equipment Specifications: 1401 Model C-3

IBM 1401 PROGRAM LIBRARY ABSTRACT

IBM 1401 PROGRAM LIBRARY ABSTRACT

File Number: 13.1.006

IBM 1401 PROGRAM LIBRARY ABSTRACT

File Number: 13.1.007

IBM 1401 PROGRAM LIBRARY ABSTRACT

File Number: 13.1.008

IBM 1401 PROGRAM LIBRARY ABSTRACT

File Number: 13.1.009

IBM 1401 PROGRAM LIBRARY ABSTRACT

File Number: 13.1.010

IBM 1401 PROGRAM LIBRARY ABSTRACT

File Number: 13.1.011

IBM 1401 PROGRAM LIBRARY ABSTRACT

File Number: 13.1.012

IBM 1401 PROGRAM LIBRARY ABSTRACT

File Number: 13.1.013

IBM 1401 PROGRAM LIBRARY ABSTRACT

File Number: 13.1.014

IBM 1401 PROGRAM LIBRARY ABSTRACT

File Number: 13.1.015

IBM 1401 PROGRAM LIBRARY ABSTRACT

File Number: 13.1.016

IBM 1401 PROGRAM LIBRARY ABSTRACT

File Number: 13.1.017

IBM 1401 PROGRAM LIBRARY ABSTRACT

File Number: 13.1.018

IBM 1401 PROGRAM LIBRARY ABSTRACT

File Number: 13.1.019

IBM 1401 PROGRAM LIBRARY ABSTRACT

File Number: 13.1.020

IBM 1401 PROGRAM LIBRARY ABSTRACT

File Number: 13.1.021

IBM 1401 PROGRAM LIBRARY ABSTRACT

File Number: 13.1.022

IBM 1401 PROGRAM LIBRARY ABSTRACT

File Number: 13.1.023

IBM 1401 PROGRAM LIBRARY ABSTRACT

File Number: 13.1.024

IBM 1401 PROGRAM LIBRARY ABSTRACT

File Number: 13.1.025
Method: The program reads a series of control cards, which set up the operations to take place. If there are errors in the control card set up, or sense switch settings, a message will print and the machine will stop at this point.

Restrictions/Range: A Punch-release instruction may be deleted by the switch settings. If a number of punch-release instructions were deleted, a message will print and the machine will stop at this point.

Storage Requirements: Not given.

Equipment Specifications: 4K machine.

IBM 1401 SIMULTANEOUS CARD-TO-TAPE AND/OR TAPE-TO-PRINTER
J. Oldenburg

Purpose/Description:

IBM 1401 SIMULTANEOUS CARD-TO-TAPE AND/OR TAPE-TO-PRINTER

Source: 4K machine

Storage Requirements: Not given.

Equipment Specifications: 4K machine.

IBM 1401 PROGRAM LIBRARY ABSTRACT

File Number: 13.1.06

1401 SIMULTANEOUS CARD-TO-TAPE AND/OR TAPE-TO-PRINTER
J. Oldenburg

Direct inquiries to:
J. Oldenburg
Republic Aviation Corporation
Farmland
Long Island, New York

Purpose/Description: KFX506 was written to take advantage of the overlap in load release and print DCO output tapes and load mixed model input decks onto tape as does other peripheral equipment. (Cards containing a 7-8 punch in column 1 are considered to be binary cards.

Method:

Card-To-Tape:

Each record is written and tested for tape error. If an error is detected, the tape is backspaced and rewritten five times. If the error persists, the tape is rewound and the above sequence repeated. After three front edges, the machine stalls at location 1382. The program will not continue. (See "USAGE")

Tape-To-Printer:

Each record on tape is tested for error when reading. If an error is detected, the tape is backspaced and read. This process is repeated, ten times after which a halt occurs at Location 1382. The program will continue after printing the record if the start button is pressed. (See "USAGE")

Restrictions/Range: The program has been written for a 4K machine with Read Release and Print Storage. It will read a record of any length (only 132 characters are stored) and print 131 characters with print control and 132 characters on single space control. At the same time it will read 60 Columns from cards and put their contents plus four blanks on tape.

Storage Requirements: Read Release and Print Storage. Only 132 characters are stored.

Equipment Specifications: 4K machine.

IBM 1401 PROGRAM LIBRARY ABSTRACT

File Number: 13.2.00

1401 ASSEMBLY OF 1401 SPS PROGRAMS

R. Nelson
IBM
2500 Central Avenue S. E.
Albuquerque, New Mexico

Purpose: To use the 1401 to assemble 1401 SPS programs which include special features and revised mnemonic operating codes.

Method: Source Language. SAP.

Restrictions/Range:

a) Timing - processes approximately 750 cards per minute.

b) Load and process program occupies approximately 0.56.38.

c) No limit to the number of cards per program. There is a maximum of 200 symbols per program.

d) Input-Output to 704 is via tape only.

Storage Requirements: Not given.

Equipment Specifications:

a) 704, either 6K or 32K. (Continued on next column)
IBM 16Z0 PROGRAM LIBRARY ABSTRACT

LAMP (Less Arithmetic More Programming) (CARD)

E. Mathys

Direct Inquiries to: E. Mathys
IBM Corporation
Green Bay, Wisconsin

Purpose: LAMP is a revised version of SPS II for card I/O. It was designed specifically for commercial applications requiring more than the 132 symbols allowed by SPS II for assemblers on a 20K 16Z0. LAMP allows 450 symbols and will reduce assembly time by up to 35%.

Mathematical Method: Does not apply

Restrictions, Range: LAMP will accept any SPS II statement with the following exceptions:

1. DAS, DSB, DNS, DNTY, and DNCD
2. BN and BA (User must specify BNCI, BNTY, RACD, RATT)
3. BN, BE, BI, BNP, BNS, and BNZ (User must use instead.
   BI, BE, BE, BNS, BNP, BNS)
4. DV, DTV, DBC, BCI, DCG, and BNC1, BNC2, BCNI, BNC4
   (User must use BI and BNI)
5. The TDM instruction will be assembled with a flag in position 7.
6. The input for both pass 1 and pass 2 must be from card.
7. All references to subroutines have been eliminated.
8. Error 1 and Error 7 have been eliminated.

Checking for record marks in lnter and op., code fields has been eliminated.

Storage Requirements: Processor occupies all of memory.

Equipment Specifications: 20K, 1620 and 1622.

IBM 16Z0 PROGRAM LIBRARY ABSTRACT

PROGRAM LOADERS (CARD)

R. E. Boos & W. W. Marks

Direct Inquiries to: R. E. Boos
W. W. Marks
Systems Engineering
3424 Wilshire Boulevard
Los Angeles 5, California

Purpose/Description: Program Loader for the IBM 16Z0 with card input.

Mathematical Method: N/A

Restrictions/Range: N/A

Storage Requirements: Not given

Equipment Specifications: 20K, 16Z0 with I/A for one of the two loaders listed.

IBM 16Z0 PROGRAM LIBRARY ABSTRACT

RELOCATING LOADER (Card)

W. J. Richards

Direct Inquiries to: W. J. Richards
Pettyjohn Engineering Co. Inc.
6145 N.E. Cully Boulevard
Portland, Oregon

Purpose/Description: To load SPS programs of a specified type into arbitrary locations in memory.

Method: N/A

Restrictions/Range: Programs must not include SPS subroutines, have flags in the middle of CNT or GTS fields, nor have constants exactly 12 digits in length. One change is required in the SPS processor.

Storage Requirements: Locations 99800 - 00399

Equipment Specifications: Paper tape, Memory 20K, and no other special features required.

IBM 16Z0 PROGRAM LIBRARY ABSTRACT

SELECTIVE TRACE (CARD)

W. H. Jefferys

Direct Inquiries to: W. H. Jefferys
Van Vleck Observatory
Wesleyan University
Middletown, Conn.

Purpose/Description: This program provides a detailed listing of the operations executed during the running of a program which is being debugged. Inflexible addresses are completely traced. The mnemonics for the commands are printed. The programmer specifies, by two numbers input to the routine, which instructions he wants traced. Outside of the specified range the instructions are executed, but not printed. In this manner already debugged portions of the program and routines such as the floating point subroutines can be run through at high speed. Several options as to the mode of tracing are provided.

Mathematical Method: Not Applicable.

Restrictions/Range: Console Switch 40 cannot be interpreted by the traced program without special (but trivial) modification of the program.

Storage Requirements: 2366 locations.

Equipment Specifications: Any 16Z0 with indirect addressing.

Additional Remarks: Of the 2366 locations, all but one are completely relocatable. The digi with label DIGIT must be at the end of a memory module. The routine is written in SPS except for the symbol table, which cannot be compiled with the SPS processor. Provision is provided for relocation in the form of a program which will punch standard SPS constant cards for the symbol table. These cards may be inserted in the object deck produced by SPS, which may be compressed, if desired. It is possible to include optional instructions such as IF-MF etc., without difficulty. The program has been written for card I/O only.

IBM 16Z0 PROGRAM LIBRARY ABSTRACT

TRACE PROGRAM FOR THE IBM 16Z0 WITH CARD INPUT/OUTPUT (CARD)

Ralph L. Miller

Direct Inquiries to: Ralph L. Miller
IBM Corporation
415 S. Michigan Avenue
Chicago 5, Illinois

Purpose/Description: Output of one card per instruction executed showing instruction, its address, and P. Q. and general products field (where applicable).

Method: Not available

Restrictions/Range: Not available

Storage Requirements: 1139 core locations -- relocatable SPS

Equipment Specifications: Memory 20K, and no other special features required.

IBM 16Z0 PROGRAM LIBRARY ABSTRACT

SELECTIVE TRACE (CARD)

W. H. Jefferys

Direct Inquiries to: W. H. Jefferys
Van Vleck Observatory
Wesleyan University
Middletown, Conn.

Purpose/Description: This program provides a detailed listing of the operations executed during the running of a program which is being debugged. Inflexible addresses are completely traced. The mnemonics for the commands are printed. The programmer specifies, by two numbers input to the routine, which instructions he wants traced. Outside of the specified range the instructions are executed, but not printed. In this manner already debugged portions of the program and routines such as the floating point subroutines can be run through at high speed. Several options as to the mode of tracing are provided.

Mathematical Method: Not Applicable.

Restrictions/Range: Console Switch 40 cannot be interpreted by the traced program without special (but trivial) modification of the program.

Storage Requirements: 2366 locations.

Equipment Specifications: Any 16Z0 with indirect addressing.

Additional Remarks: Of the 2366 locations, all but one are completely relocatable. The digi with label DIGIT must be at the end of a memory module. The routine is written in SPS except for the symbol table, which cannot be compiled with the SPS processor. Provision is provided for relocation in the form of a program which will punch standard SPS constant cards for the symbol table. These cards may be inserted in the object deck produced by SPS, which may be compressed, if desired. It is possible to include optional instructions such as IF-MF etc., without difficulty. The program has been written for card I/O only.

IBM 16Z0 PROGRAM LIBRARY ABSTRACT

TRACE PROGRAM FOR THE IBM 16Z0 WITH CARD INPUT/OUTPUT (CARD)

Ralph L. Miller

Direct Inquiries to: Ralph L. Miller
IBM Corporation
415 S. Michigan Avenue
Chicago 5, Illinois

Purpose/Description: Output of one card per instruction executed showing instruction, its address, and P. Q. and general products field (where applicable).

Method: Not available

Restrictions/Range: Not available

Storage Requirements: 1139 core locations -- relocatable SPS

Equipment Specifications: Memory 20K, and no other special features required.

IBM 16Z0 PROGRAM LIBRARY ABSTRACT

16Z0 MULTI-TRACE (Card)

Jim Moore

Direct Inquiries to: Jim Moore
IBM
2145 Highland
Birmingham, Alabama

Purpose/Description: Virtually eliminates tedious debugging. A mere scan of MULTI-TRACE output will turn up a majority of user errors. Complete tracing versatility in one program. Card or typed output yields before and after snapshots of data as well as effective addresses if indirect. Sense switch controls address stop, full or branch trace, elimination of BT subroutines, and typed or card output.

Mathematical Method: Each traced instruction selects its own output format.

(Continued on next page)
IBM 1620 PROGRAM LIBRARY ABSTRACT

File Number 1.4.004

STROBIC - Shelly Trace Routine with Option on Branch and transmit and Indirect address Converter (Card)

O. R. Boyer & R. R. Tieman

Direct Inquiries to: O. R. Boyer

Shelly Oil Company

Accounting Department - Computer Programming Unit

P. O. Box 1650

Tulsa, Oklahoma

Luther 4-2311, Extension 654

Purpose/Description: STROBIC is a full trace routine for the 1620 computer equipped with a 1629 card read/punch unit and the indirect addressing special feature. STROBIC will trace the automatic divide, the indirect address feature, and the transfer numeric string/move flag/transfer numeric full instruction package.

Mathematical Method: N/A

Restrictions/Range: N/A

Storage Requirements: 2,434 positions

Equipment Specifications: Computer: IBM 1620, card input/output. Special Features: Must have indirect addressing special feature.


IBM 1620 PROGRAM LIBRARY ABSTRACT

File Number 1.4.005

TRACE AND IA SIMULATOR (Tape)

Charles E. Berry

Direct Inquiries to: Charles E. Berry

IBM Corporation

1112 S. W. 4th Avenue

Portland, Oregon

CA 97213

Purpose/Description: To simulate a 1620 program written with or without indirect addressing and type out instructions and data fields at user's option. Traces all instructions, types address chains. Output format selected by operation code - may be digit, field, or record. User may execute portions of program at full speed with return to trace at a predetermined instruction.

Method: Not applicable

Restrictions/Range: Cannot re-enter trace made from automatic mode internal to a BT-EE pair.

Storage Requirements: 2513 plus 20 at the end of program

Equipment Specifications: Memory 20K, 60K, 60K, Automatic Divide and Paper Tape. No other special features required.

Additional Remarks: Reusable. Immediate fields may be 12 digits long. Record marks unlisted in fields or to instructions are acceptable. Typewriter control commands are not executed while in type mode. In non-type mode all typewriter commands are executed normally.

IBM 1620 PROGRAM LIBRARY ABSTRACT

File Number 1.4.006

1620 MULTI-TRACE (Tape)

Jim Moore

Direct Inquiries to: Jim Moore

IBM Corporation

2145 Highland

Birmingham, Alabama

Purpose/Description: Virtually eliminates tedious debugging. A mere scan of MULTI-TRACE output will turn up a majority of user errors. Complete tracing versatility is one program. Card or typed output yields before and after snapshots of data as well as effective addresses if indirect. Sense switch control of address stop, full or branch trace, elimination of BY subroutines, and typed or card output.

Mathematical Method: Each traced instruction selects its own output format.

Restrictions/Range: Will not properly handle more than 5 digits in an immediate command. Record mark encountered in instruction or data will result in short line if typed. No such restriction in card mode.

Storage Requirements: 3720 positions.

Equipment Specifications: 200 card 1620 with IDA

Additional Remarks: Program largely made up of subroutines. Easily expanded to any size memory. One digit change for adaptation to paper tape. The speed is full punch with output, otherwise about 7 instructions per second. The source language--SPL--completely relocatable.

IBM 1620 PROGRAM LIBRARY ABSTRACT

File Number 1.5.001

FORTRAN SOURCE TAPE CONNECTOR (Tape)

D. R. Gardner

Direct Inquiries to: D. R. Gardner

General Foods Research Center

Tarrytown, New York

Purpose/Description: To correct a FORTRAN source tape to produce a new FORTRAN source tape.

Mathematical Method: N/A

Restrictions/Range: The maximum number of changes is 105.

Storage Requirements: 1980 + I/O area

Equipment Specifications: Minimum 1620

IBM 1620 PROGRAM LIBRARY ABSTRACT

File Number 1.5.002

FORTRAN BUTLER (Tape)

Jack Burgeson

Direct Inquiries to: Jack Burgeson

IBM

340 S. 4th Street

Akron, Ohio

Purpose/Description: Under sense switch control, this program accepts either typewriter or tape input and prepares either typewriter or tape output (or both). Input to 1620 FORTRAN program, unaltered with respect to "card column". Output is a typed up statement, C (if present) in position 1; statement number (if present) in positions 2-5; statement itself in positions 6-12. Excessively long statements are edited by elimination of blank to fit in positions 7-12 when this is possible. The program is most useful when preparing to convert a 1620 Fortran program to some other machine by going tape to card through an ODT.

Method: N/A

Restrictions/Range: N/A

Equipment Specifications: Basic paper tape 1620

Additional Remarks: The language is SPS.
IBM 1620 PROGRAM LIBRARY ABSTRACT

File Number 1.5.003

TAP EDIT (Tape)

Jack Burgerson

Direct Inquiries to: Jack Burgerson
IBM Corporation
340 S. Broadway
Akron, Ohio

Purpose/Description: Procedures are in this program to edit source tapes such as Fortran or 360 tapes. The operator can make changes in part or in whole, insert before or after, delete or skip over sections of the tape by choosing among several edit codes. Maximum record length checking is also done.

Method: N/A

Restrictions/Range: N/A

Storage Requirements: Uses most of storage

Equipment Specifications: Basic paper tape 1620

Additional Remarks: The language is SPS.

IBM 1620 PROGRAM LIBRARY ABSTRACT

File Number 1.5.004

POST MORTEM DUMP FOR CARD 1620 (Card)

W. T. Gault

Direct Inquiries to: W. T. Gault
IBM Corporation
609 S. State Street
Salt Lake City, Utah

Purpose/Description: To dump portions of memory in data or instruction format for debugging at either a programmed or error halt.

Method: Does not apply

Restrictions/Range: The program destroys the multiply tables, loads its own add tables, and loads into either 602-1432 (lower memory) or 18798-19998 (upper memory). It requires either a 483 or 487 for listing the output with a 80 by 80 board.

Storage Requirements: 1020 locations

Equipment Specifications: Memory 20 K and 1620 Card Reader. No other special features required.

Additional Remarks: It operates at punch speed and is loaded after the running of a main program.

IBM 1620 PROGRAM LIBRARY ABSTRACT

File Number 1.6.001

Regression Analysis Data Preparation Program for the 1620 (Tape)

T. N. Korolits

Direct Inquiries to: T. N. Korolits
Badger Manufacturing Company
363 Third Street
Cambridge, Massachusetts

Purpose/Description: This program prepares data in a form required by the RAP program written by D. N. Luearsen

Method: N/A

Restrictions/Range: N/A

Storage Requirements: N/A

Equipment Specifications: Memory 20K. No other special features required.

Additional Remarks: 1. SPS language used
2. Fixed point notation
3. Running time depends on amount of data to be prepared
4. Has been run successfully about 25 times.
5. The program occupies positions 2178-01853. Symbols and data input area are in locations 07854-12231

IBM 1620 PROGRAM LIBRARY ABSTRACT

File Number 1.6.003

1620 AUTOPLOTTER (Tape)

Neil Lewis

Direct Inquiries to: Neil Lewis
Systems Engineer-Scientific (TS444)
IBM Corporation
Hollywood, Hawaii

Purpose/Description: This routine is an indirect addressing version of the 1620 Edit Subroutine 1.6.010.

Restrictions/Range: There are no restrictions as to the length of a record to be edited. Floating dollar signs are not handled.

Storage Requirements: 305 positions

Equipment Specifications: Tape 1620, memory 20K, 40K, 60K with indirect addressing, no other features required.

Additional Remarks: Language-Symbolic (Relativized) Symbolic

IBM 1620 PROGRAM LIBRARY ABSTRACT

File Number 1.6.004

1620 AUTOPLOTTER (Card)

Bob Louden
IBM Detroit North
7700 Second Boulevard
Detroit 2, Michigan

Purpose: To provide two-color graph plotting for a tape 1620 system. The graphs are plotted off-line on an 870 system. See preliminary Autoplotter manual.

Restrictions/Range: Graph paper sizes up to 20 inches high and 100 inches wide. Accuracy plus or minus .100 inches on all points plotted. Graphs include automatic generation of all scales and labels.

Speed: Main Frame time 5 to 10 minutes; off line typing time 5 to 10 minutes.

Method: An original scanning and curve-fitting technique is used.

Storage Requirements: All 20,000 digits.

Remarks: This is an independent program and is not relocatable. The Language used is SPS.

Equipment Specifications: 20K tape, no special features. Modified 870 system used as plotter. See preliminary manual.

IBM 1620 PROGRAM LIBRARY ABSTRACT

File Number 1.6.005

1620 AUTOPLOTTER (Card)

Bob Louden
IBM Detroit North
7700 Second Boulevard
Detroit 2, Michigan

Purpose: To provide two-color graph plotting for a card 1620 system. The graphs are plotted off-line on an 870 system. See preliminary Autoplotter manual.

Restrictions/Range: Graph paper sizes up to 20 inches high and 100 inches wide. Accuracy plus or minus .100 inches on all points plotted. Graphs include automatic generation of all scales and labels.

Speed: Main Frame time 30 seconds to one minute; plotting time 5 to 10 minutes.

Method: An original scanning and curve-fitting technique is used.

Storage Requirements: All 20,000 digits.

Remarks: This is an independent program and is not relocatable. The Language used is SPS.

Equipment Specifications: 20K card, no special features. Modified 870 system used as plotter. See preliminary manual.
1620 PROGRAM LIBRARY ABSTRACT

File Number 1.4.005

IBM 1620 EDIT Subroutine (Card)

Neil Lewis

Direct Inquiries to: Neil Lewis
System Engineer - Scientific (756644)
IBM Corporation
Honolulu, Hawaii

Purpose/Description: This routine is an indirect addressing version of the 1620 Edit Subroutine 1.6.10.

Restrictions, Range: There are no restrictions as to the length of a record to be edited. Floating dollar signs are not handled.

Storage Requirements: 356 positions.

Equipment Specifications: Card 1620, memory 20K, 40K, 60K with indirect addressing, or other features required.


IBM 1620 PROGRAM LIBRARY ABSTRACT

File Number 1.4.106

1620 FORCOM (Card)

Bob Louden
IBM Detroit North
1720 Second Boulevard
Detroit 2, Michigan

Purpose: To provide alphabetic comments and column headings for 1620 FORTRAN, and to control tabs and carriage returns.

Restrictions, Range: A maximum of nine 40-character records may be stored in core at one time.

Speed: Essentially that of I/O instructions.

Method: None.

Storage Requirements: 490 digits.

Equipment Specifications: IBM 1620 card, any core size. No special features required.

Additional Remarks: 1620 FOR is a FORTRAN Subroutine or Independent. It is relocatable. Machine Language (24 instructions).

IBM 1620 PROGRAM LIBRARY ABSTRACT

File Number 1.4.097

SPS - To - FORTRAN SUBROUTINE EDIT (Tape)

G. L. Johnson
IBM Corporation
1730 Cambridge Street
Cambridge, Mass.

Purpose: To convert an SPS object program to the format required to include it in the subroutine library tape for FORTRAN. It allows distribution of a program in SPS source language for use as an SPS program or as a FORTRAN subroutine.

Restrictions, Range: Does not apply.

Speed: Approximately limited by tape read and punch speed.

Method: Does not apply.

Storage Requirements: Program is always loaded between 00402 and 03569.

Equipment Specifications: Basic Tape 1620.

Additional Remarks: Programs to be edited must be written in SPS and must follow a few additional rules detailed in the write-up of the edit program.

The edit routine converts the SPS object program automatically. An optional feature is the ability to insert up to 100 digits of remarks on the listing in addition to a heading including the name, date and number of the subroutine.

This program is non-relocatable.

IBM 1620 PROGRAM LIBRARY ABSTRACT

File Number 1.4.010

1620 EDIT SUBROUTINE (Tape)

Neil Lewis

Direct Inquiries to: Neil Lewis
System Engineer - Scientific (756644)
IBM Corporation
Honolulu, Hawaii

Purpose/Description: This routine inserts a continuous series of numeric data held in an alphabetic record as specified by the programmer, leaving it ready for printing or punching. Automatic zero suppression and the ability to handle all alphabetic characters are standard features. All data following a decimal point is printed. When room is provided ahead of a decimal point, the routine ensures that at least one figure or zero precedes the decimal point.

(Continued on next page)
IBM 1620 PROGRAM LIBRARY ABSTRACT

1620 EDIT SUBROUTINE (Card)  Nell Lewis

Direct Inquiries to: Nell Lewis

Purpose/Description: This routine inserts a continuous series of numeric data fields into an alphabetic record as specified by the programmer, leaving it ready for printing or punching. Automatic zero suppression and the ability to handle all alphabetic characters are standard features. All data following a decimal point is printed. When room is provided ahead of a decimal point, the routine ensures that at least one figure or zero precedes the the decimal point.

Mathematical Method: None

Restrictions, Range: There are no restrictions as to the length of a record to be edited. Floating dollar signs are not handled.

Storage Requirements: 390 positions

Equipment Specifications: Tape 1620, memory 20K, 40K, 60K. No other special features required.

Additional Remarks: Language-Relocatable (Relativized) Symbols.

IBM 1620 PROGRAM LIBRARY ABSTRACT

1620 EDIT SUBROUTINE (Card)  Nell Lewis

Direct Inquiries to: Nell Lewis

Purpose/Description: This routine inserts a continuous series of numeric data fields into an alphabetic record as specified by the programmer, leaving it ready for printing or punching. Automatic zero suppression and the ability to handle all alphabetic characters are standard features. All data following a decimal point is printed. When room is provided ahead of a decimal point, the routine ensures that at least one figure or zero precedes the the decimal point.

Mathematical Method: None

Restrictions, Range: There are no restrictions as to the length of a record to be edited. Floating dollar signs are not handled.

Storage Requirements: 390 positions

Equipment Specifications: Tape 1620, memory 20K, 40K, 60K. No other special features required.

Additional Remarks: Language-Relocatable (Relativized) Symbols.

IBM 1620 PROGRAM LIBRARY ABSTRACT

1620 EDIT SUBROUTINE (Card)  Nell Lewis

Direct Inquiries to: Nell Lewis

Purpose/Description: This routine inserts a continuous series of numeric data fields into an alphabetic record as specified by the programmer, leaving it ready for printing or punching. Automatic zero suppression and the ability to handle all alphabetic characters are standard features. All data following a decimal point is printed. When room is provided ahead of a decimal point, the routine ensures that at least one figure or zero precedes the the decimal point.

Mathematical Method: None

Restrictions, Range: There are no restrictions as to the length of a record to be edited. Floating dollar signs are not handled.

Storage Requirements: 390 positions

Equipment Specifications: Tape 1620, memory 20K, 40K, 60K. No other special features required.

Additional Remarks: Language-Relocatable (Relativized) Symbols.

IBM 1620 PROGRAM LIBRARY ABSTRACT

1620 EDIT SUBROUTINE (Card)  Nell Lewis

Direct Inquiries to: Nell Lewis

Purpose/Description: This routine inserts a continuous series of numeric data fields into an alphabetic record as specified by the programmer, leaving it ready for printing or punching. Automatic zero suppression and the ability to handle all alphabetic characters are standard features. All data following a decimal point is printed. When room is provided ahead of a decimal point, the routine ensures that at least one figure or zero precedes the the decimal point.

Mathematical Method: None

Restrictions, Range: There are no restrictions as to the length of a record to be edited. Floating dollar signs are not handled.

Storage Requirements: 390 positions

Equipment Specifications: Tape 1620, memory 20K, 40K, 60K. No other special features required.

Additional Remarks: Language-Relocatable (Relativized) Symbols.

IBM 1620 PROGRAM LIBRARY ABSTRACT

1620 EDIT SUBROUTINE (Card)  Nell Lewis

Direct Inquiries to: Nell Lewis

Purpose/Description: This routine inserts a continuous series of numeric data fields into an alphabetic record as specified by the programmer, leaving it ready for printing or punching. Automatic zero suppression and the ability to handle all alphabetic characters are standard features. All data following a decimal point is printed. When room is provided ahead of a decimal point, the routine ensures that at least one figure or zero precedes the the decimal point.

Mathematical Method: None

Restrictions, Range: There are no restrictions as to the length of a record to be edited. Floating dollar signs are not handled.

Storage Requirements: 390 positions

Equipment Specifications: Tape 1620, memory 20K, 40K, 60K. No other special features required.

Additional Remarks: Language-Relocatable (Relativized) Symbols.

IBM 1620 PROGRAM LIBRARY ABSTRACT

1620 EDIT SUBROUTINE (Card)  Nell Lewis

Direct Inquiries to: Nell Lewis

Purpose/Description: This routine inserts a continuous series of numeric data fields into an alphabetic record as specified by the programmer, leaving it ready for printing or punching. Automatic zero suppression and the ability to handle all alphabetic characters are standard features. All data following a decimal point is printed. When room is provided ahead of a decimal point, the routine ensures that at least one figure or zero precedes the the decimal point.

Mathematical Method: None

Restrictions, Range: There are no restrictions as to the length of a record to be edited. Floating dollar signs are not handled.

Storage Requirements: 390 positions

Equipment Specifications: Tape 1620, memory 20K, 40K, 60K. No other special features required.

Additional Remarks: Language-Relocatable (Relativized) Symbols.

IBM 1620 PROGRAM LIBRARY ABSTRACT

1620 EDIT SUBROUTINE (Card)  Nell Lewis

Direct Inquiries to: Nell Lewis

Purpose/Description: This routine inserts a continuous series of numeric data fields into an alphabetic record as specified by the programmer, leaving it ready for printing or punching. Automatic zero suppression and the ability to handle all alphabetic characters are standard features. All data following a decimal point is printed. When room is provided ahead of a decimal point, the routine ensures that at least one figure or zero precedes the the decimal point.

Mathematical Method: None

Restrictions, Range: There are no restrictions as to the length of a record to be edited. Floating dollar signs are not handled.

Storage Requirements: 390 positions

Equipment Specifications: Tape 1620, memory 20K, 40K, 60K. No other special features required.

Additional Remarks: Language-Relocatable (Relativized) Symbols.

IBM 1620 PROGRAM LIBRARY ABSTRACT

1620 EDIT SUBROUTINE (Card)  Nell Lewis

Direct Inquiries to: Nell Lewis

Purpose/Description: This routine inserts a continuous series of numeric data fields into an alphabetic record as specified by the programmer, leaving it ready for printing or punching. Automatic zero suppression and the ability to handle all alphabetic characters are standard features. All data following a decimal point is printed. When room is provided ahead of a decimal point, the routine ensures that at least one figure or zero precedes the the decimal point.

Mathematical Method: None

Restrictions, Range: There are no restrictions as to the length of a record to be edited. Floating dollar signs are not handled.

Storage Requirements: 390 positions

Equipment Specifications: Tape 1620, memory 20K, 40K, 60K. No other special features required.

Additional Remarks: Language-Relocatable (Relativized) Symbols.

IBM 1620 PROGRAM LIBRARY ABSTRACT

1620 EDIT SUBROUTINE (Card)  Nell Lewis

Direct Inquiries to: Nell Lewis

Purpose/Description: This routine inserts a continuous series of numeric data fields into an alphabetic record as specified by the programmer, leaving it ready for printing or punching. Automatic zero suppression and the ability to handle all alphabetic characters are standard features. All data following a decimal point is printed. When room is provided ahead of a decimal point, the routine ensures that at least one figure or zero precedes the the decimal point.

Mathematical Method: None

Restrictions, Range: There are no restrictions as to the length of a record to be edited. Floating dollar signs are not handled.

Storage Requirements: 390 positions

Equipment Specifications: Tape 1620, memory 20K, 40K, 60K. No other special features required.

Additional Remarks: Language-Relocatable (Relativized) Symbols.

IBM 1620 PROGRAM LIBRARY ABSTRACT

1620 EDIT SUBROUTINE (Card)  Nell Lewis

Direct Inquiries to: Nell Lewis

Purpose/Description: This routine inserts a continuous series of numeric data fields into an alphabetic record as specified by the programmer, leaving it ready for printing or punching. Automatic zero suppression and the ability to handle all alphabetic characters are standard features. All data following a decimal point is printed. When room is provided ahead of a decimal point, the routine ensures that at least one figure or zero precedes the the decimal point.

Mathematical Method: None

Restrictions, Range: There are no restrictions as to the length of a record to be edited. Floating dollar signs are not handled.

Storage Requirements: 390 positions

Equipment Specifications: Tape 1620, memory 20K, 40K, 60K. No other special features required.

Additional Remarks: Language-Relocatable (Relativized) Symbols.

IBM 1620 PROGRAM LIBRARY ABSTRACT

1620 EDIT SUBROUTINE (Card)  Nell Lewis

Direct Inquiries to: Nell Lewis

Purpose/Description: This routine inserts a continuous series of numeric data fields into an alphabetic record as specified by the programmer, leaving it ready for printing or punching. Automatic zero suppression and the ability to handle all alphabetic characters are standard features. All data following a decimal point is printed. When room is provided ahead of a decimal point, the routine ensures that at least one figure or zero precedes the the decimal point.

Mathematical Method: None

Restrictions, Range: There are no restrictions as to the length of a record to be edited. Floating dollar signs are not handled.

Storage Requirements: 390 positions

Equipment Specifications: Tape 1620, memory 20K, 40K, 60K. No other special features required.

Additional Remarks: Language-Relocatable (Relativized) Symbols.

IBM 1620 PROGRAM LIBRARY ABSTRACT

1620 EDIT SUBROUTINE (Card)  Nell Lewis

Direct Inquiries to: Nell Lewis

Purpose/Description: This routine inserts a continuous series of numeric data fields into an alphabetic record as specified by the programmer, leaving it ready for printing or punching. Automatic zero suppression and the ability to handle all alphabetic characters are standard features. All data following a decimal point is printed. When room is provided ahead of a decimal point, the routine ensures that at least one figure or zero precedes the the decimal point.

Mathematical Method: None

Restrictions, Range: There are no restrictions as to the length of a record to be edited. Floating dollar signs are not handled.

Storage Requirements: 390 positions

Equipment Specifications: Tape 1620, memory 20K, 40K, 60K. No other special features required.

Additional Remarks: Language-Relocatable (Relativized) Symbols.

IBM 1620 PROGRAM LIBRARY ABSTRACT

1620 EDIT SUBROUTINE (Card)  Nell Lewis

Direct Inquiries to: Nell Lewis

Purpose/Description: This routine inserts a continuous series of numeric data fields into an alphabetic record as specified by the programmer, leaving it ready for printing or punching. Automatic zero suppression and the ability to handle all alphabetic characters are standard features. All data following a decimal point is printed. When room is provided ahead of a decimal point, the routine ensures that at least one figure or zero precedes the the decimal point.

Mathematical Method: None

Restrictions, Range: There are no restrictions as to the length of a record to be edited. Floating dollar signs are not handled.

Storage Requirements: 390 positions

Equipment Specifications: Tape 1620, memory 20K, 40K, 60K. No other special features required.

Additional Remarks: Language-Relocatable (Relativized) Symbols.
IBM 1620 PROGRAM LIBRARY ABSTRACT

Format Control Subroutines for IBM 1620 Card Fortran (Fat & Clear) (Card)

William M. Fleischman

Direct Inquiries to: William M. Fleischman

Worthington Corporation
410 Worthington Avenue
Harrison, New Jersey

Purpose/Description: These subroutines permit the Fortran programmer the use of both fixed length, variable point format, the standard Fortran print routine, and variable length, fixed point format - FAT & CLEAR subroutine provides full interchangeability of both three modes within a single program.

Method: N/A

Restrictions/Range: FAT subroutine allows the programmer to specify the number of places to be printed before the point, the number to be printed after the point, and the number of trailing spaces to be allowed. He is limited to a maximum of nine of each. He must specify at least one place before the point. There are no other restrictions placed on the use of this subroutine.

Storage Requirements: FAT and CLEAR are relocatable subroutines for IBM 1620 card Fortran and occupy 816 and 56 digits of core storage respectively.

Equipment Specifications: Memory 20K, Indirect Addressing.

Additional Remarks: These subroutines were written for IBM 1620 card Fortran but may be easily accommodated to IBM 1620 Fortran for tape I/O. Example Fortran Statements: FORM = FAT (421) Notes (1) Form would be any unused symbol (4) 4 of (421) specifies digits before decimal (2) 2 of (421) specifies digits after decimal (6) 1 of (421) specifies spaces between words (5) Decimal in (421) is essential to make a floating point number.

CLEAR = CLEAR (666).

Notes: (1) This statement restores normal Fortran format. (2) Any blank point number in parentheses will achieve same result.

IBM 1620 PROGRAM LIBRARY ABSTRACT

GOHOT Generator Of Hermaphroditic Object Types) (Tape)

Dick Cooner

Direct Inquiries to: Frank Mosina

IBM Corporation
421 Seventh Avenue
Pittsburgh, Pennsylvania

Purpose/Description: Gohot produces a program in self-loading, self-reproducing form. This tape, and any of its descendants, loads itself or reproduces itself, depending on the initial instruction entered at the typewriter. The program tape produced by Gohot is 20-40% shorter and 20-40% faster than the same program in SPS output form.

Method: N/A

Restrictions/Range: The program to be processed by Gohot must lies entirely within cells 00040-19999 and must use decimal arithmetic. Record marks throughout the program do not constitute an obstacle to Gohot.

Storage Requirements: 00000-00049 (tables are restored at end)

Equipment Specifications: Memory 20K, 40K, 60K, and no other special features required.

Additional Remarks: Gohot was written in actual and is not relocatable. Running time depends on the length of the program to be processed. Programming hours, 25

IBM 1620 PROGRAM LIBRARY ABSTRACT

FORTAN II DIAGNOSTICIAN (CARD)

James Snellken, Charles Snyder, & Jack Burgeson

Direct Inquiries to: Jack Burgeson

IBM Akron

(Continued on next column)

Ed Schaefer

S. F. Goodrich, Akron

Purpose/Description: To diagnose (error check) Fortran I, Fortran II, or any subset thereof, source decks prior to compilation. Will diagnose source decks destined for:

650 (Fortran III only) 1620
704
705
709
1401
1410
7090

Method: N/A

Restrictions/Range: N/A

Storage Requirements: 20K

Equipment Specifications: 20K Card 1620 with indirect addressing

Additional Remarks: The language is SPS with patches. Most coding errors, such as mixed mode expressions, improperly written statements, undefined labels, missing statement numbers, improper subscripting, open DO loops, unmatched parentheses, improper modification of DO indices within a DO loop, duplicate statement numbers, and others are picked up by this program. Provision is made for batch diagnosing.

IBM 1620 PROGRAM LIBRARY ABSTRACT

INTERPRETIVE PROGRAMMING SYSTEM (IPS) (Tape)

Lawrence C. Brown
Midwestern Regional Office
IBM Corporation
618 South Michigan Avenue
Chicago 5, Illinois

Purpose: IPS is an interpretive programming system for the 1620. The one-address interpretive language includes the commands of the Intercom System -- widely used on the Bendix G-15:

Restrictions/Range: The only subroutines supplied are sine, cosine, logarithm, exponential, square root, arctangent, and fraction selection. The single precision system carries five significant digit floating point numbers. The double precision system carries twelve significant figures.

Method: Floating arithmetic is rounded, the transcendental subroutines are truncated. Single precision subroutines are calculated by Hastings Approximations, except for square root which is done by the "odd-number subtraction" method. The double precision subroutines are done by Taylor series after suitable argument reduction. The double precision square root is done by "odd-number subtraction".

Storage Requirements: 20,000-digit storage.

Source Language: Written in 1620 absolute, revised version created in SPS language.

Remarks: This is an independent system, which includes relocatable subroutines but, the program is non-relocatable.

Equipment Specifications: 1620 with 20K storage, paper tape I/O. Hardware divide required for the double precision system, optional for single precision. No use is made of indirect addressing, but it will not cause any conflict if installed.

IBM 1620 PROGRAM LIBRARY ABSTRACT

INTERPRETIVE PROGRAMMING SYSTEM (IPS) (Card)

Lawrence C. Brown
Midwestern Regional Office
IBM Corporation
618 South Michigan Avenue
Chicago 5, Illinois

Abstract data for this program is identical to data for program number 2,0005 except that this program is for the IBM 1620 card system.

IBM 1620 PROGRAM LIBRARY ABSTRACT

INTERPRETIVE PROGRAMMING SYSTEM (IPS) (Tape)

W. R. Glass and J. G. Hasenek
School of Aeronautical & Engineering Sciences
Purdue University

(Continued on next page)
IBM 1620 PROGRAM LIBRARY ABSTRACT

IBM 650 Simulator Program (Card)
F. C. Toscano

Direct Inquiries to: W. H. Jefferys
Van Vleck Observatory
Wesleyan University
Middletown, Conn.

Purpose/Description: To take the square root of any number, given an arbitrary number of digits. The resulting square root has as many digits as the number input to the subroutine.

Mathematical Method: Odd-Number Subtraction Method.

Restrictions/Range: X, the number whose square root is to be taken, must be greater than or equal to zero. If it is negative, the routine will halt after printing "SQRTE NEG NO," and then take the square root of |X|.

Storage Requirements: If N is the number of digits in the longest number whose square root is to be taken, the routine requires 422 + 2N digits for the indirect Addressing version, and 519 + 3N digits for the version which does not require indirect addressing.

Equipment Specifications: There is a version for machines with indirect addressing, and another for machines without indirect addressing.

Additional Remarks: The routine is written in SPS, 2-Pass. It is completely relocatable. The numbers involved are fixed point.

IBM 1620 PROGRAM LIBRARY ABSTRACT

IBM 650 Simulator Program (Card)
F. C. Toscano

Direct Inquiries to: W. H. Jefferys
Van Vleck Observatory
Wesleyan University
Middletown, Conn.

Purpose/Description: Simulation of the IBM 650 on the IBM 1620. It allows execution of 650 language programs in a 1620 without reprogramming.

Mathematical Method: Odd-Number Subtraction Method.

Restrictions/Range: This program is the tape system of the program No. 2. 0. 008.

Storage Requirements: N/A

IBM 1620 PROGRAM LIBRARY ABSTRACT

IBM 650 Simulator Program (Card)
F. C. Toscano

Direct Inquiries to: W. H. Jefferys
Van Vleck Observatory
Wesleyan University
Middletown, Conn.

Purpose/Description: To take the square root of any number, given an arbitrary number of digits. The resulting square root has as many digits as the number input to the subroutine.

Mathematical Method: Odd-Number Subtraction Method.

Restrictions/Range: X, the number whose square root is to be taken, must be greater than or equal to zero. If it is negative, the routine will halt after printing "SQRTE NEG NO," and then take the square root of |X|.

Storage Requirements: If N is the number of digits in the longest number whose square root is to be taken, the routine requires 422 + 2N digits for the indirect Addressing version, and 519 + 3N digits for the version which does not require indirect addressing.

Equipment Specifications: There is a version for machines with indirect addressing, and another for machines without indirect addressing.

Additional Remarks: The routine is written in SPS, 2-Pass. It is completely relocatable. The numbers involved are fixed point.
A matrix loading program allowing ease of data preparation and solves for the \( N \) eigenvectors associated with the phase for assembly with the user's program.

Additional Remarks: The subroutine `storage` is supplied in symbolic form, on cards, for assembly with the user's program. It is completely relocatable. It has successfully calculated results of numbers to as many as 2000 places. The general timing formula is the following: 
\[
T = 0.5T_0 + 4.5T_1 + 0.580T_2 \text{ milliseconds,}
\]
where \( T \) is the number of digits in the result.

**IBM 1620 PROGRAM LIBRARY ABSTRACT**

File Number 5.0.000

**SIMULTANEOUS EQUATION PROGRAM (Tape)**

D. N. Leeson
IBM
Eastern Regional Office
425 Park Avenue
New York, New York

Purpose: This program generates the solutions to a linear system of maximum size, 39 x 39.

Restrictions, Range: All arithmetic is done in 10 digit excess floating point.

Method: Variation on the Gauss elimination technique, known as the product matrix method, is employed.

Storage Requirements: For the maximum program (39), all of core is required.

Remarks: The program will yield the solution to the linear system for up to 99 constant vectors without matrix inversion.

Equipment Specifications: 1620, paper tape, 20K core. No other devices are necessary.

**IBM 1620 PROGRAM LIBRARY ABSTRACT**

File Number 5.0.002

**SIMULTANEOUS EQUATION SOLUTION (Card)**

D. N. Leeson
IBM
Eastern Regional Office
425 Park Avenue
New York, New York

Purpose: This program generates the solutions to a linear system of maximum rank 39 x 39. One may have 99 constant vectors per matrix of coefficients.

Restrictions, Range: 39 x 39

Accuracy: Rounding error for very large systems noticeable.

Speed: Variable dependent upon problem size.


Storage Requirements: All of core is required for the maximum problem.

Equipment Specifications: 1620 with 1621 attachment. Division feature not required.

Additional Remarks: This program uses SPS Language, and is non-relocatable.

**IBM 1620 PROGRAM LIBRARY ABSTRACT**

File Number 5.0.003

**EIGENVALUES OF REAL SYMMETRIC MATRICES ON THE 1620 DATA PROCESSING SYSTEM (Card)**

Neil Lewis
IBM
818 Kapioani Blvd.
Honolulu, Hawaii

Purpose: Will solve for the eigenvalues and associated eigenvectors of a real, symmetric matrix to order 50.

(Continued on next column)

Restrictions, Range: The program consists of 3 basic parts.

A) Phase 1 -- a matrix loading program allowing ease of data preparation and including certain error detection features. Corrections are facilitated by direct keyboard entry of corrected records.

B) Phase 2 -- eigenvalue solution phase. Solves by a modification of the bisection method. Eigenvalues are typed out at the conclusion of phase 2. Rate of convergence is also indicated on the typewriter. Sense switch control allows the selection of punched card output of the rotation angles to be used in phase 3.

C) Phase 3 -- solves for the \( N \) eigenvectors associated with the phase 2 eigenvalues. Vectors are printed out on the typewriter together with identifying information.

Method: Floating point arithmetic is used for all calculations in phase 2 and 3. No other subroutines are used in any of the three phases.

Source Language: Programming language is SPS.

Remarks: Eigenvalue for a 20 x 20 well behaved matrix was 40 minutes. Precision for a 20 x 20 well behaved matrix was 6 significant digits.

Equipment Specifications: IBM 1620 card system with core system and indirect addressing.

**IBM 1620 PROGRAM LIBRARY ABSTRACT**

File Number 5.0.004

**EIGENVALUES OF REAL SYMMETRIC MATRICES ON THE 1620 DATA PROCESSING SYSTEM (Card)**

D. N. Leeson
IBM
Eastern Regional Office
425 Park Avenue
New York, New York

Purpose/Description: To evaluate determinants

Mathematical Method: Crout Reduction

Restrictions, Range: The determinant may not have a rank exceeding 40 or less than two.

Storage Requirements: All of core

Equipment Specifications: Card 1620 20K core. No other devices necessary.

**IBM 1620 PROGRAM LIBRARY ABSTRACT**

File Number 5.0.005

**MATRIX INVERSION (Card)**

Dale Anderson

Direct Inquiries to: Dale Anderson
IBM Corporation
340 S. Broadway
Akron, Ohio

Purpose/Description: This program will invert any non-singular square matrix of size 22 x 22 or less. Provision is made for re-inversion to check accuracy. Input is from tape or typewriter, output is on typewriter. Since this program is written in Fortran, it may be applied with equal facility to a card 1620, with minor I/O changes to any hardware accepting the Fortran language.

Method: N/A

(Continued on next page)
REGRESSION ANALYSIS PROGRAM

IBM 1620 PROGRAM LIBRARY ABSTRACT

File Number: 5.0.007

SOLUTION OF SIMULTANEOUS LINEAR EQUATIONS (Cards)

Purpose: This program performs a complete regression analysis on a maximum of 14 variables.

Restrictions, Range: All arithmetic is done in 10 digit excess 50 floating point.

Method: All mathematical models are linearized, using a special technique. The Gaussian least squares technique is applied.

Storage Requirements: The program with a maximum number of variables (24) occupies all of core for a 20,000 position 1620. Speed cannot be determined due to the many configurations of the program. The program is not relocatable.

Remarks: This program will fit nonlinear functions and surfaces. Data may be pretransformed by any one of 21 available transformations. The system is in 2 passes. Pass 1 prepares data as input to Pass 2.

Equipment Specifications: Card 1620. 20K. Core-Divide not required.

IBM 1620 PROGRAM LIBRARY ABSTRACT

File Number: 6.0.003

SCRAP (Sixteen-twenty Card Regression Analysis Program) (Card)

Purpose/Description: This program performs a complete linear or non linear regression analysis for the card 1620 system. A plotback program is also included. Output of all phases is on cards for subsequent listing. A typewritten output is also available.

Mathematical Method: Gaussian Least Square Technique

Restrictions/Range: No more than 23 variables total may be processed. The linearity check if not performed.

Storage Requirements: 20K for maximum program.

Equipment Specifications: Memory 20K. No other special features required.

IBM 1620 PROGRAM LIBRARY ABSTRACT

File Number: 6.0.004

STRAP (Stepwise Regression Analysis Program) (Tape)

Purpose/Description: The computer program containing program for transforming input variables. It is useful in determining the relationships between the independent and dependent variables of a set of observations by an equation of the form:

\[ Y = \sum_{i=1}^{k} a_i x_i \]

Where \( Y \) is the dependent variable, \( a_i \) are the independent variables, and \( x_i \) are the coefficients to be determined.

Mathematical Method: N/A

Restrictions/Range: N/A

Storage Requirements: 10,000 positions

Equipment Specifications: Basic 1620, paper tape input & output.

IBM 1620 PROGRAM LIBRARY ABSTRACT

File Number: 6.0.005

FREQUALIZER (Tape)

Purpose: This program performs a complete regression analysis on a maximum of 14 variables.

Restrictions, Range: All arithmetic is done in 10 digit excess 50 floating point.

Method: All mathematical models are linearized, using a special technique. The Gaussian least squares technique is applied.

Storage Requirements: The program with a maximum number of variables (24) occupies all of core for a 20,000 position 1620. Speed cannot be determined due to the many configurations of the problems. The program is not relocatable.

Remarks: This program will fit nonlinear functions and surfaces. Data may be pretransformed by any one of 21 available transformations. The system is in 2 passes. Pass 1 prepares data as input to Pass 2.

Equipment Specifications: Card 1620. 20K. Core-Divide not required.

IBM 1620 PROGRAM LIBRARY ABSTRACT

File Number: 6.0.006

REGRESSION ANALYSIS PROGRAM (Tape)

Purpose: This program performs a complete regression analysis on a maximum of 14 variables.

Restrictions, Range: All arithmetic is done in 10 digit excess 50 floating point.

Method: All mathematical models are linearized, using a special technique. The Gaussian least squares technique is applied.

Storage Requirements: The program with a maximum number of variables (24) occupies all of core for a 20,000 position 1620. Speed cannot be determined due to the many configurations of the problems. The program is not relocatable.

Remarks: This program will fit nonlinear functions and surfaces. Data may be pretransformed by any one of 21 available transformations. The system is in 2 passes. Pass 1 prepares data as input to Pass 2.

Equipment Specifications: Tape 1620. 20K Core-Divide not required.

IBM 1620 PROGRAM LIBRARY ABSTRACT

File Number: 6.0.007

REGRESSION ANALYSIS PROGRAM (Card)

Purpose: This program performs a complete regression analysis on a maximum of 14 variables.

Restrictions, Range: All arithmetic is done in 10 digit excess 50 floating point.

Method: All mathematical models are linearized, using a special technique. The Gaussian least squares technique is applied.

Storage Requirements: The program with a maximum number of variables (24) occupies all of core for a 20,000 position 1620. Speed cannot be determined due to the many configurations of the problems. The program is not relocatable.

Remarks: This program will fit nonlinear functions and surfaces. Data may be pretransformed by any one of 21 available transformations. The system is in 2 passes. Pass 1 prepares data as input to Pass 2.

Equipment Specifications: Tape 1620. 20K Core-Divide not required.

IBM 1620 PROGRAM LIBRARY ABSTRACT

File Number: 6.0.008

REGRESSION ANALYSIS PROGRAM (continued on next column)

Purpose: This program performs a complete regression analysis on a maximum of 14 variables.

Restrictions, Range: All arithmetic is done in 10 digit excess 50 floating point.

Method: All mathematical models are linearized, using a special technique. The Gaussian least squares technique is applied.

Storage Requirements: The program with a maximum number of variables (24) occupies all of core for a 20,000 position 1620. Speed cannot be determined due to the many configurations of the problems. The program is not relocatable.

Remarks: This program will fit nonlinear functions and surfaces. Data may be pretransformed by any one of 21 available transformations. The system is in 2 passes. Pass 1 prepares data as input to Pass 2.

Equipment Specifications: Tape 1620. 20K Core-Divide not required.

IBM 1620 PROGRAM LIBRARY ABSTRACT

File Number: 6.0.009

REGRESSION ANALYSIS PROGRAM (continued on next column)

Purpose: This program performs a complete regression analysis on a maximum of 14 variables.

Restrictions, Range: All arithmetic is done in 10 digit excess 50 floating point.

Method: All mathematical models are linearized, using a special technique. The Gaussian least squares technique is applied.

Storage Requirements: The program with a maximum number of variables (24) occupies all of core for a 20,000 position 1620. Speed cannot be determined due to the many configurations of the problems. The program is not relocatable.

Remarks: This program will fit nonlinear functions and surfaces. Data may be pretransformed by any one of 21 available transformations. The system is in 2 passes. Pass 1 prepares data as input to Pass 2.

Equipment Specifications: Tape 1620. 20K Core-Divide not required.
Purpose/Description: This program analyzes the frequencies present in a time series by means of power spectra.
Method: Fourier transform of auto-covariance function.
Restrictions/Range: Maximum of 200 lags, any number of data points.
Storage Requirements: 20,000 digits
Equipment Specifications: Memory 20K, and no other special features required.
Additional Remarks: Running time: (MIN 1252) / 2000 minutes
IBM 1620 PROGRAM LIBRARY ABSTRACT
File Number 6.0.066

STEPWISE MULTIPLE LINEAR REGRESSION (Tape)
R. Bukacek & W. Galie

Direct Inquiries to: W. J. Galie
Armour & Company
Operations Research
60 N. Wabash
Chicago, Illinois

Purpose/Description: Accepts sets of observations and forms linear regressions in a stepwise fashion subject to statistical criterion (F-Test).
Method: Stepwise linear regression
Restrictions/Range: Maximum number of independent variables = 18, $5 \leq N_{\text{Ind}} \leq 25$ - Ind., $2 \leq |N_{\text{Ind}}| \leq 15$, Maximum number of independent variables which can be Apriori suppressed from consideration in stepwise process $\leq 15$ $2 \leq N_{\text{Ind}} \leq 15$
Storage Requirements: 20K
Equipment Specifications: Memory 20K, and paper tape. No other special features required.
Additional Remarks: Restrictions above apply to 20K basic tape machine. See attached writeup for complete description and notes.
IBM 1620 PROGRAM LIBRARY ABSTRACT
File Number 6.0.007

Stepwise Multiple Linear Regression Analysis for the IBM 1620 (Card)
D. G. Wyman

Direct Inquiries to: D. G. Wyman
IBM Corporation
401 Grand Avenue
Oakland, California

Purpose/Description: The 1620 Stepwise Regression Analysis Program has been coded in SPS as a series of independent subroutines. Each can be assembled inde­pendently as long as the data areas are constant. This should allow easy modification. With efficient utilization of storage, a problem of 35 variables can be run on a basic 1620. Analysis of variance is combined with Multiple Regression Analysis to control the selection of terms for an equation.
Restrictions/Range: Single precision floating point has been used throughout. 40 variables is maximum for Phase I, i.e., single correlation matrix. 35 variables can be run in Phase II, the Stepwise solution. Any of 13 transformations can be used up to 70 per observation. Data input format must be defined by a header card.
Storage Requirements: 20,000 positions
Equipment Specifications: Memory 25K. No other special features required.
IBM 1620 PROGRAM LIBRARY ABSTRACT
File Number 6.0.010

ANALYSIS OF VARIANCE (Card)
Louis J. Granato

Direct Inquiries to: Louis J. Granato
IBM Corporation
631 Cooper Street
Camden, N. J.

Purpose/Description: Reduce the total variation in a set of data to components associated with possible sources of variability whose relative importance we wish to assess.
Mathematical Method: Sums of Squares
Restrictions/Range: Maximum of eight (8) factors, with not more than eight levels per factor. Total data cannot exceed 12,935 digits.
Storage Requirements: N/A
Equipment Specifications: Basic 1620 with card I/O
POLYNOMIAL CURVE FITTING (Tape)

Direct Inquiries to: W. R. Graves
IBM
5640 Canal Street
New Orleans 19, Louisiana

Purpose: This program generates an approximating polynomial by the least squares technique. The equation so derived contains as many terms as necessary to bring the standard error of the dependent variable within a range specified by the user, or to fit a 15th order polynomial.

Printing of intermediate coefficients and the printing of a tabulation of observed vs calculated values of the dependent variable are under the control of program switches as is the inclusion of weighting factors.

The calculations utilize floating arithmetic with an 8 digit mantissa.

Restrictions, Range: Not given.

Method: A modified Gaussian elimination technique is used to solve the resulting set of simultaneous equations. Experimental data are recorded in standard IBM FORTRAN format.

Storage Requirements: Not given.

WS: This program uses FORTRAN language.

Equipment Specifications: IBM 1620, 20K core, paper tape reader, paper tape punch. Will run on any IBM for which FORTRAN is written.

IBM 1620 PROGRAM LIBRARY ABSTRACT

POLYNOMIAL CURVE FIT (Card)

Direct Inquiries to: W. R. Graves
IBM
5640 Canal Street
New Orleans 19, Louisiana

Purpose/Description: This program generates an approximating polynomial by the least squares technique. The equation so derived contains as many terms as necessary to bring the standard error of the dependent variable within a range specified by the user, or to fit a 15th order polynomial.

Printing of intermediate coefficients and the printing of a tabulation of observed vs calculated values of the dependent variable are under the control of program switches as is the inclusion of weighting factors.

The calculations utilize floating arithmetic with an 8 digit mantissa.

Restrictions, Range: Not given.

Method: A modified Gaussian elimination technique is used to solve the resulting set of simultaneous equations. Experimental data are recorded in standard IBM FORTRAN format.

Storage Requirements: Not given.

WS: This program uses FORTRAN language.

Equipment Specifications: IBM 1620, 20K core, 1622 card read-punch. Will run on any IBM for which FORTRAN is written.

IBM 1620 SUBDIVISION PROGRAM (Tape)

H. W. Van Ness
C. E. Berry
IBM
1202 S. W. 6th Avenue
Portland 4, Oregon

Purpose: Compute necessary data for the subdivision of land into smaller parcels. The program starts with a closed boundary traverse and proceeds to compute all necessary curves and tangents. The design engineer then submits data for lot computations and receives complete information for staking and plotting the subdivision. Lot characteristics are checked against zoning requirements. Output includes co-ordinates of points; length and bearing of lines; length and radius of arcs; and area, depth, and width of lots.

Restrictions, Range: Up to 250 points and 25 curves may be processed at one time.

Method: Does not apply.

Storage Requirements: Four program passes are required - utilizing all of the 20,000 positions except in Pass 1 and the co-ordinate type out.

Equipment Specifications: Minimum 1620, 20,000 positions of core and paper-tape input-output.

POLYNOMIAL CURVE FITTING (Card)

Direct Inquiries to: W. R. Graves
IBM
5640 Canal Street
New Orleans 19, Louisiana

Purpose/Description: This program generates an approximating polynomial by the least squares technique. The equation so derived contains as many terms as necessary to bring the standard error of the dependent variable within a range specified by the user, or to fit a 15th order polynomial.

Printing of intermediate coefficients and the printing of a tabulation of observed vs calculated values of the dependent variable are under the control of program switches as is the inclusion of weighting factors.

The calculations utilize floating arithmetic with an 8 digit mantissa.

Restrictions, Range: Not given.

Method: A modified Gaussian elimination technique is used to solve the resulting set of simultaneous equations. Experimental data are recorded in standard IBM FORTRAN format.

Storage Requirements: Not given.

WS: This program uses FORTRAN language.

Equipment Specifications: IBM 1620, 20K core, paper tape reader, paper tape punch. Will run on any IBM for which FORTRAN is written.

IBM 1620 PROGRAM LIBRARY ABSTRACT

POLYNOMIAL CURVE FITTING (Card)

Direct Inquiries to: W. R. Graves
IBM
5640 Canal Street
New Orleans 19, Louisiana

Purpose/Description: This program generates an approximating polynomial by the least squares technique. The equation so derived contains as many terms as necessary to bring the standard error of the dependent variable within a range specified by the user, or to fit a 15th order polynomial.

Printing of intermediate coefficients and the printing of a tabulation of observed vs calculated values of the dependent variable are under the control of program switches as is the inclusion of weighting factors.

The calculations utilize floating arithmetic with an 8 digit mantissa.

Restrictions, Range: Not given.

Method: A modified Gaussian elimination technique is used to solve the resulting set of simultaneous equations. Experimental data are recorded in standard IBM FORTRAN format.

Storage Requirements: Not given.

WS: This program uses FORTRAN language.

Equipment Specifications: IBM 1620, 20K core, paper tape reader, paper tape punch. Will run on any IBM for which FORTRAN is written.

IBM 1620 PROGRAM LIBRARY ABSTRACT

POLYNOMIAL CURVE FITTING (Card)

Direct Inquiries to: W. R. Graves
IBM
5640 Canal Street
New Orleans 19, Louisiana

Purpose/Description: This program generates an approximating polynomial by the least squares technique. The equation so derived contains as many terms as necessary to bring the standard error of the dependent variable within a range specified by the user, or to fit a 15th order polynomial.

Printing of intermediate coefficients and the printing of a tabulation of observed vs calculated values of the dependent variable are under the control of program switches as is the inclusion of weighting factors.

The calculations utilize floating arithmetic with an 8 digit mantissa.

Restrictions, Range: Not given.

Method: A modified Gaussian elimination technique is used to solve the resulting set of simultaneous equations. Experimental data are recorded in standard IBM FORTRAN format.

Storage Requirements: Not given.

WS: This program uses FORTRAN language.

Equipment Specifications: IBM 1620, 20K core, paper tape reader, paper tape punch. Will run on any IBM for which FORTRAN is written.

IBM 1620 PROGRAM LIBRARY ABSTRACT

POLYNOMIAL CURVE FITTING (Card)

Direct Inquiries to: W. R. Graves
IBM
5640 Canal Street
New Orleans 19, Louisiana

Purpose/Description: This program generates an approximating polynomial by the least squares technique. The equation so derived contains as many terms as necessary to bring the standard error of the dependent variable within a range specified by the user, or to fit a 15th order polynomial.

Printing of intermediate coefficients and the printing of a tabulation of observed vs calculated values of the dependent variable are under the control of program switches as is the inclusion of weighting factors.

The calculations utilize floating arithmetic with an 8 digit mantissa.

Restrictions, Range: Not given.

Method: A modified Gaussian elimination technique is used to solve the resulting set of simultaneous equations. Experimental data are recorded in standard IBM FORTRAN format.

Storage Requirements: Not given.

WS: This program uses FORTRAN language.

Equipment Specifications: IBM 1620, 20K core, paper tape reader, paper tape punch. Will run on any IBM for which FORTRAN is written.

IBM 1620 PROGRAM LIBRARY ABSTRACT

POLYNOMIAL CURVE FITTING (Card)

Direct Inquiries to: W. R. Graves
IBM
5640 Canal Street
New Orleans 19, Louisiana

Purpose/Description: This program generates an approximating polynomial by the least squares technique. The equation so derived contains as many terms as necessary to bring the standard error of the dependent variable within a range specified by the user, or to fit a 15th order polynomial.

Printing of intermediate coefficients and the printing of a tabulation of observed vs calculated values of the dependent variable are under the control of program switches as is the inclusion of weighting factors.

The calculations utilize floating arithmetic with an 8 digit mantissa.

Restrictions, Range: Not given.

Method: A modified Gaussian elimination technique is used to solve the resulting set of simultaneous equations. Experimental data are recorded in standard IBM FORTRAN format.

Storage Requirements: Not given.

WS: This program uses FORTRAN language.

Equipment Specifications: IBM 1620, 20K core, paper tape reader, paper tape punch. Will run on any IBM for which FORTRAN is written.

IBM 1620 PROGRAM LIBRARY ABSTRACT

CUT AND FILL (Tape)

Ben A. Shaw
IBM
1620
470 N. Robert Street
St. Paul 1, Minnesota

Purpose: Compute grades, apply typical sections, compute slope interest areas, and volumes when given P. V. I. Stations, Elevations, and Lengths of Vertical Curves, Typical Sections and where they are to be used, Shoreline Factors, and Preliminary Terrain Cross Sections.

Restrictions, Range: This program does not compute horizontal curve transitions. It is limited to 30 Terrain Points per Cross Section, Ten Typical Sections, and ten P. V. I.'s. The horizontal distances are to even feet, and the elevations are to tenths of a foot.

Method: Does not apply.

[Continued on next column]
IBM 1620 PROGRAM LIBRARY ABSTRACT

CUT AND FILL (Card) Raymond Peck

Ray Peck
IBM-San Francisco
340 Market Street
San Francisco, California

Abstract data for this program is identical to data for program number 5.2.002 except that this program is for the IBM 1620 card system.

IBM 1620 PROGRAM LIBRARY ABSTRACT

WATERWAY COMPUTATIONS (Tape) C. G. Carlson and J. M. Peck

Direct Inquiries to: Charles E. Carlson
Bridge Section
Wis. Highway Commission
Madison, Wisconsin
Alpine Ave. 6-4411, Ext. 471

Purpose/Description: The purpose of this program is to compute the velocity, area, and flow for an individual channel in a flow system and the average velocity, area, and flow for the entire network.

Mathematical Method: Manning's formulas.

Restrictions/Range: A maximum of 25 water elevations.

Storage Requirements: See sheet.

Equipment Specifications: IBM 1620 Tape System: Memory 308; No Special Features Required.

Additional Remarks: Easily converted to Card System.

IBM 1620 PROGRAM LIBRARY ABSTRACT

SKEWED BRIDGE ELEVATIONS (Tape) J. F. Gilbon and C. E. Carlson

Direct Inquiries to: C. E. Carlson
Bridge Section
Wis. Highway Commission
Madison, Wisconsin
Alpine Ave. 6-4411, Ext. 471

Purpose/Description: The program computes slab elevations and geometry for bridge superstructures with skewed substructures on a vertical curve with straight horizontal alignment. Horizontal and vertical geometry is found at the intersection of a chosen series of offset lines with a skewed line. These skewed lines may be at specific stations, at quarter points of spans, or at constant increment stations.

Mathematical Method: Not applicable.

Restrictions/Range: A maximum of fifty beams or offsets.

Storage Requirements: 250K

Equipment Specifications: IBM 1620 Tape System: Memory 208; No Other Special Features Required.

Additional Remarks: Input to the computer may be either paper tape or typewriter. Geometry for skewed bridges may be obtained by manipulation of input data. Provisions are made for up to 50 offsets divided into 1 to 1 groups. Program language - FORTRAN

Run successfully about 100 times to date - August 22, 1961. The program is easily converted to a Card System.

IBM 1620 PROGRAM LIBRARY ABSTRACT

1620 TRAVERSE ANALYSIS PROGRAM (Tape) D. T. Mitchell

Direct Inquiries to: D. T. Mitchell
IBM Corporation
Midwestern Regional Office
618 South Michigan
Chicago, Illinois

Purpose/Description: This program will solve traverse problems requiring balancing of misclosure or solution for unknown azimuths and/or distances. No provision is made to handle other than straight-line courses. Areas of traverses can be calculated (user's option). All possible solutions for problems are presented in the output.

Method: Standard methods outlined in writeup. All output is via the typewriter.

Restrictions/Range: All angles and cosines are calculated within 2x10^-8 insuring 3-decimal place accuracy in latitudes and departures.

Storage Requirements: 20K Core is required.

Equipment Specifications: Basic 1620 without any features.

Additional Remarks: The source language is machine.

IBM 1620 PROGRAM LIBRARY ABSTRACT

GAS NETWORK ANALYSIS (Tape)

R. E. Eadell
IBM
5930 Holmes Avenue
Hammond, Indiana

Purpose: The analysis of a gas distribution network is necessary when a gas utility is considering the modification and/or expansion of a gas system or when an increased load on the system is contemplated. With the use of this program, such an analysis can be made for as many as 750 pipes in a low and/or medium pressure system.

Restrictions/Range: The program will handle a gas network of approximately 750 pipe sections and 250 loops. The program requires an assumed flow rate and friction factor for each pipe section as input. The flow and friction can be in any units provided the units chosen remain constant for a given network. Rather than friction, a user may specify a diameter and length of pipe section. The accuracy depends upon the tolerance factor within the program which may be changed by the user.

Speed: 0.3 sec/loop/iteration exclusive of input and output.

Method: Modified Hardy Cross Method.

(Continued on next page)
IBM 1620 PROGRAM LIBRARY ABSTRACT  File Number 9.3.005

M-100 MOMENT OF INERTIA AND CENTROID CALCULATIONS  (Tape)
G. J. Reed

Direct Inquiries to:
R. C. Weswick
AFC Industries, Inc.
P. O. Box 1666
Albuquerque, New Mexico

Purpose/Description: This program is used to compute the Moments of Inertia, area, and Centroid of a complicated two dimensional body. The system is divided into a grid system with grid spacing and formula number for each rectangle entered as input.

Method: N/A

Restrictions/Range: The code will handle up to a maximum of 65x and 65y spaces.

Storage Requirements: 19,534 core locations.

Equipment Specifications: Memory 20K, and no other special features required.

Additional Remarks: Language is SPS. The running time depends on the number of grid spaces required to define the body. The time may be approximated by T = 4.19 x SDC 38 seconds. SDC is the number of divisions in the grid system.

IBM 1620 PROGRAM LIBRARY ABSTRACT  File Number 9.4.001

ELECTRIC LOAD FLOW PROGRAM  (Tape)

Frank Minona
Systems Engineer
IBM Corp.
472 7th Avenue
Pittsburgh 29, Pa.

Purpose: The program is designed to calculate voltages and power flows in a system of a maximum size of 150 buses and 240 lines, and allows changes to be made to the base system and be rerun.

Restrictions/Range: All calculations are done in a fixed point.

a) Net load or generation at any bus must be less than 10,000 Megawatts and Megavars.

b) The self impedance of any bus must have both R and X components of less than 1.000000 per unit.

c) The sum of squares of R and X components of self admittance of any bus must be less than 1,000,000,000 per unit.

d) The accuracy of the results may be predetermined by the operator by specifying higher tolerances in the iterative solution.

Speed: Average time per iteration:

Time in milliseconds = 460.7 x No. of buses + 112.9 x No. of lines + 512.2 x No. of Generator Buses

Method: Solution is obtained by the Gauss-Seidel iteration method.

Storage Requirements: Full 20K memory is required, with the program broken down into 5 passes.

Source Language: SPS 1 PASS

Remarks: This is an independent program and is assembled into fixed locations but is not relocatable unless reassembled.

Equipment Specifications: Basic 1620, 10K paper tape system.

IBM 1620 PROGRAM LIBRARY ABSTRACT  File Number 9.4.002

LOCATION OF SHUNT CAPACITORS ON RADIAL LINES  (Tape)
L. S. Rankine, R. F. Steinhart
IBM
455 Park Avenue
New York, New York

Purpose: This program may be used by electric utilities engineers to compute optimum locations for shunt capacitor banks in radial distribution systems so as to minimize losses and to improve voltage. It may also be used to demonstrate one of the many ways in which digital computers may be used by utilities engineers.

(Continued on next page)
ELECTRIC LOAD FLOW PROGRAM (Card)

Frank Moonin
Systems Engineer
IBM Corporation
421 7th Avenue
Pittsburgh 19, Pennsylvania

Abstract data for this program is identical to data for program number 9.4.001 except that this program is for the IBM 1620 card system.

IBM 1620 PROGRAM LIBRARY ABSTRACT File Number 9.4.004

Selection of Economic Conductor Size - Specific Case
New England Electric System Program F18 (Card)
R. H. Snow

Direct Inquiries to: R. H. Snow
New England Electric System
245 South Main Street
Northampton, Massachusetts

Purpose/Description: This program is based upon the methods presented in the following Electrical World Articles, by J. J. Rankine.

Method: This program is based upon the methods presented in the following Electrical World Articles, by J. J. Rankine.

Date: October 3, 1955

Place: New England Electric

Two-thirds Rule Used for Capacitors KVAR

Method of Locating Shunt Capacitors Suitable for Computer Solutions.

Restrictions, Range: Four standard capacitor bank sizes are considered.

Storage Requirements: 12,000 locations are used.

Equipment Specifications: Basic 1620 - Tape input/output.

IBM 1620 PROGRAM LIBRARY ABSTRACT File Number 9.4.005

ECONOMIC CONDUCTOR SIZE SELECTION BY KELVIN'S LAW (File)

R. F. Slade

Direct Inquiries to: R. F. Slade
IBM Corporation
New York City, New York

Purpose/Description: To choose the conductor size that minimizes the overall cost of material and line losses.

Method: The following equations are the basis of the program

Mathematical Method: Kelvin's Law

Restrictions, Range: Does not apply

Storage Requirements: 20 K

Equipment Specifications: Any 1620 System

Additional Remarks: FORTRAN with machine language. The speed is about 20 seconds/case.

IBU 1620 PROGRAM LIBRARY ABSTRACT File Number 9.4.006

SHORT CIRCUIT ANALYSIS (Card)
G. S. Haralampu

Direct Inquiries to: G. S. Haralampu
New England Electric System
441 Stuart Street
Boston 16, Massachusetts

Purpose/Description: This program is to be used for the determination of current distribution constants, line voltages, X/R ratios, and impedances to the point of fault, under faulted conditions.

Mathematical Method: Gauss-Seidel Iterative method

Restrictions, Range: 33 buses and 56 lines

Storage Requirements: 28,000 digits

Equipment Specifications: Computer, IBM 1620, 20 K core, 1620 Card Reader and Punch.

Additional Remarks: The speed is approximately 1.5 seconds per bus per iteration. Negative impedances, such as those obtained in mutual equalizations, should be avoided.

IBM 1620 PROGRAM LIBRARY ABSTRACT File Number 9.4.007

TRANSMISSION LOSSES AND PENALTY FACTORS (card)
D. Hayward

Direct Inquiries to: D. Hayward
New England Electric System
441 Stuart Street
Boston 16, Massachusetts

Purpose/Description: This program will figure generated power, losses and received power and the penalty factor at each entry point of the system represented by the B-constant matrix. It does not figure the B-constants. They must be available to use the program.

Method: The following equations are the basis of the program

Loss = Penalty Factor

Restrictions/Range: The program is limited to a 28 by 28 B-constant matrix

Storage Requirements: The program uses essentially the entire 20 K core. The speed depends on the matrix size. Once the B-constants have been read an average case might take about 2 minutes.
**IBM 1620 PROGRAM LIBRARY ABSTRACT**

**CURVE FITTING - SIMULATED PLANT RECORD METHOD (Card)**

William D. Garland

**Direct Inquiries to:**
William D. Garland
New England Electric System
441 Stuart Street
Boston, MA, Massachusetts

**Purpose/Description:**
This program is designed to find the best fitting average life with a generalized empirical curve tried for a plant account (cf. Methods of Estimating Utility Plant Life, Edison Electric Institute, 1952).

The best of all fits derived for a series of curves (such as the Iowa curves) is selected by visually examining the output data for the least sum of squared differences between the balances and the balances simulated for the best fit locate.

**Method:**

**A. Formula Terms:**
- \( LC \) = longer life assumed
- \( LL \) = shorter life assumed
- \( LC \) = best fit life as calculated
- \( BO \) = book balances
- \( BL \) = balances simulated for \( LL \)
- \( BL \) = balances simulated for \( LC \)

**B. Formula:**

\[
\frac{BO - BL}{BO - BL} = \frac{BO - BL}{BO - BL} \text{ for } 0.55 \text{ and } 0.55.
\]

**Restrictions/Range:**
N/A

**Storage Requirements:**
9,950 - program and fixed point divide routines.

**Equipment Specifications:**
- IBM 1620 Computer with a 20K memory card and a 1622 Card Reader-Punch

**Additional Remarks:**
The speed depends on accuracy of starting assumption given program. The best fit for one curve is usually produced within a few seconds at most.

---

**IBM 1620 PROGRAM LIBRARY ABSTRACT**

**STRAIN GAGE DATA REDUCTION ON THE IBM 1620 (Tape)**

R. C. Wenrick

**Direct Inquiries to:**
R. C. Wenrick
ACF Industries
P.O. Box 1646
Albuquerque, New Mexico
C11 1-0341, Ext. 511

**Purpose/Description:**
To reduce data as recorded for rectangular strain gage rosettes by the Gilmore, B and K or similar recorders.

**Method:**
N/A

**Restrictions/Range:**
- 100 Channels of data may be reduced with one pass through the system.

**Storage Requirements:**
About 18,000

**Equipment Specifications:**
- Memory 20K, Automatic Divide, and no other special features required.

**Additional Remarks:**
- The language is SPMS. Although indirect addressing and automatic divide features are used, very few corrections are required to enable a basic machine to process the data. The program has been used for reduction of more than 10,000 rosettes. The input has been prepared to a great extent by the tape punching facilities of the Gilmore.
- Program based on IBM 650 Program 9.7.002 entitled "Hydraulic Network Analysis." The speed is approximately one second per pipe per iteration.

---

**IBM 1620 PROGRAM LIBRARY ABSTRACT**

**DISTRIBUTION OF WATER FLOW IN A PIPE NETWORK (Tape)**

C. Bartholomew

**Direct Inquiries to:**
C. Bartholomew
IBM Corporation
Boston, Massachusetts

**Purpose/Description:**
This program balances the flow of water in a pipe network starting with estimated flows and produces the corrected system flows.

**Mathematical Method:**
Hydro-Cross

**Restrictions/Range:**
- Maximum of 150 pipes and 67 loops

**Storage Requirements:**
- Entire 20K memory

**Equipment Specifications:**
- As submitted to the program library, the basic paper tape 1620 is required. The FORTRAN source program in the documentation may be compiled for any configuration.

**Additional Remarks:**
Program based on IBM 650 Program 9.7.002 entitled "Hydraulic Network Analysis." The speed is approximately one second per pipe per iteration.

---

**IBM 1620 PROGRAM LIBRARY ABSTRACT**

**GENERALIZED PLOTTING II (Card)**

Jack Burgessan

**Direct Inquiries to:**
Jack Burgessan - IBM
340 S. Broadway
Akron, OH, Ohio

**Purpose/Description:**
Given up to 180 pairs of Y values equally spaced along the X axis, this program scales them to the range 0-50 and plots them on the 1620 typewriter. Baseline indication is plotted also.

**Method:**
Not applicable

**Restrictions/Range:**
Up to 180 pairs of Y values

**Storage Requirements:**
All of storage is used

**Equipment Specifications:**
Basic card 1620

**Additional Remarks:**
The language is Gotran
IBM 1620 PROGRAM LIBRARY ABSTRACT  
File Number 5.7.003

GENERALIZED PLOTTOR (Card)
Jack Burgeseon

Direct Inquiries to: Jack Burgeson - IBM
340 S. Broadway
Akron, Ohio

Purpose/Description: Given up to 600 Y values, equally spaced along the X axis, this program scales these to a range 0-50 and plots them on the 1620 typewriter. Baseline indication is plotted also.

Method: Not applicable
Restrictions/Range: Up to 600 points
Storage Requirements: Use all storage
Equipment Specifications: Use basic card 1620
Additional Remarks: The language is Copram

IBM 1620 PROGRAM LIBRARY ABSTRACT  
File Number 5.7.004

5-109 STRESS ANALYSIS OF A FLANGE WITH A TAPERED HUB (Card)
D. A. Oliver

Direct Inquiries to: R. C. Wenrick
ACF Industries Inc.
P.O. Box 1666
Albuquerque, New Mexico

Purpose/Description: This program can be used to size tapered sections used for damping the discontinuities produced at flange-shell junctures or can provide stress and discontinuity levels of existing designs.


Storage Requirements: 18,500 core positions
Equipment Specifications: Memory 28K, Automatic Divide, and no other special features required.

IBM 1620 PROGRAM LIBRARY ABSTRACT  
File Number 5.7.005

2-109 STRESS ANALYSIS OF A FLANGED TAPERED HUB (Card)
R. C. Wenrick

Direct Inquiries to: R. C. Wenrick
ACF Industries Inc.
P.O. Box 1666
Albuquerque, New Mexico

Purpose/Description: This program can be used to size tapered sections used for damping the discontinuities produced at Flange-shell junctures or can provide stress and discontinuity levels of existing designs.

Method: Approximations as described in ASME "Design Data and Methods.

Restrictions/Range: N/A
Storage Requirements: 30,000 core locations
Equipment Specifications: 40K, Automatic Divide and no other special features required.

Additional Remarks: The program is written in Spectra and utilizes three library subroutines which are the following:
1) L-109 Computation of O and 1st Order Bessel Functions
2) L-105 Floating Point Output Routine
3) L-105 Solution of Simultaneous Equations.

The running time varies between 4 and 6 minutes depending on the hub dimensions. It has been run 96 times successfully. All subroutines are included in the card deck.

IBM 1620 PROGRAM LIBRARY ABSTRACT  
File Number 10.1.001

LINEAR PROGRAMMING FOR THE 1620 (Tape)
C. R. Nichols
IBM Corporation
9250 Wilshire Blvd.
Beverly Hills, California

Purpose: A generalized code for the solution of linear programming problems. Allows variable format input/output gives complete details of results. Optional routines allow previously solved problems to accept changed cost and/or requirement coefficients with subsequent re-solution.

Restrictions/Range: The basic 1620 with paper-tape reader is required. Program runs on any available core size, with the matrix size being limited according to the expression:

(N = 2) (N + 1) (Memory - 3761)

where: N = number of restricting equations, N = number of non-basis variables, Memory = core size in digits.

All computations are done in 2- and 8 floating point.

Storage Requirements: Storage locations 00012 through 03150 are occupied by sub-programs and floating point routines. The rest of memory is available for matrix storage.

Source Language: SPS.

Remarks: The program is a self-contained series of subroutines.

Equipment Specifications: Basic 1620 with 1621 paper tape reader.

IBM 1620 PROGRAM LIBRARY ABSTRACT  
File Number 10.1.002

LINEAR PROGRAMMING CODE FOR THE IBM 1620 WITH CARD INPUT AND OUTPUT (Card)
Katherine Krieger
Ray Data
IBM
31-05 Woodside Blvd.
Woodside, N. Y.

Purpose: Solves linear programming problems with output of detailed results.

Method: The two main routines of the program are the simplex algorithm and the "dual algorithm." All computations are in 2- and 8 floating point.

Restrictions/Range: N/A
Storage Requirements: Any size storage can be used. The larger the storage, the larger the problem that can be solved.

Source Language: The program is written in actual machine language.

Equipment Specifications: Basic 1620 with card input and output.
TRANSPORTATION PROGRAM FOR THE IBM 1620 (Tape)

D. E. Madden
IBM
9510 Wilshire Blvd.
Beverly Hills, California
G. Smith
IBM
3424 Wilshire Blvd.
Los Angeles, California

Purpose: The program provides an optimal solution to transportation problems (special type linear programming problems) and is based on the maximal flow in networks. The cost is minimized for shipping a product from a set of sources to a set of destinations. The cost function is a distance matrix. The program also includes solution to the Assign. Problem. Other applications include vehicle distribution, production scheduling, transshipment, and personnel assignment.

Restrictions/Ranges: Input consists of sources (M), destinations (N), and costs for shipping form sources to destinations. These values must be non-negative and of fixed-point arithmetic. The range of 0 to 999,999 includes all possible values.

Storage Requirements: IBM 1620, 20K storage, paper tape reader, paper tape punch.

LIBRARY ABSTRACT File Number 10.1.003

IBM 1620 PROGRAM LIBRARY ABSTRACT

IBM 1620 PROGRAM LIBRARY ABSTRACT

TRANSPORTATION PROGRAM FOR THE IBM 1620 (Card)

J. N. Bole

Direct Inquiries to: James N. Bole
University of California
2062 Gillis Hall
Berkeley, California

Purpose/Description: This program is a simple adaptation of the Transportation Program for the IBM 1620 (Tape) by Madden and Smith, File No. 10.1.003. It provides an optimal solution to the linear programming transportation problem.

Mathematical Method: The method used is that of Ford and Fulkerson (Management Science 3 (3): pp. 24-32, October, 1956).

Restrictions/Range: Input data are the number of rows (sources), M, the number of columns (destinations), N, the product from the sales and input data. The program requires that the input data be non-negative.

Storage Requirements: IBM 1620, 20K storage, card reader/punch.

LIBRARY ABSTRACT File Number 10.1.005

IBM 1620 PROGRAM LIBRARY ABSTRACT

IBM 1620 PROGRAM LIBRARY ABSTRACT

MKV Program for Linear Program Matrix Preparation (Card)

E. L. Maline

Direct Inquiries to: E. L. Maline
Union Oil Company of California
8600 Eastlake
Beverly Hills, California

Purpose/Description: The purpose of this program is to prepare a linear program matrix for the Nichols, Nickel, Davis Card Linear Program. The program prepares a machine program which can be used to solve a linear program problem. The program is designed for linear program problems with up to 20 variables and 20 equations. The program requires that the input data be non-negative and of fixed-point arithmetic.

Mathematical Method: N/A

Restrictions/Range: The range of both equations and vectors can be specified for each MKV calculation. Zero elements in output vectors are not punched out.

Storage Requirements: Program stored in locations 00000 to 02890.

Equipment Specifications: IBM 1620, 20K, indirect addressing card input/output.

Additional Remarks: The approximate running time is 7 minutes to produce 30 vectors which have about 40 equations. The run program approximately 30 times successfully to 9/2/61. The run program approximately 30 times successfully. The run program approximately 30 times successfully.

LIBRARY ABSTRACT File Number 10.1.004

IBM 1620 PROGRAM LIBRARY ABSTRACT

IBM 1620 PROGRAM LIBRARY ABSTRACT
b. Data must be prepared in the format specified in Appendix B.

c. Output may either be on the typewriter or on cards. The optional final matrix punchout is on cards. (See Addendum No. 1 to program written).

**Storage Requirements:** Any size memory - see Restrictions.

**Equipment Specifications:** Basic 1620 with Card input and output.

**IBM 1620 PROGRAM LIBRARY ABSTRACT**

**File Number:** 10.2.001

**Title:** An Inventory Management Simulator (Card)

C. J. Walker & G. M. Goodfriend

IBM Corporation

618 S. Michigan Avenue

Chicago 5, Illinois

**Purpose/Description:** This simulator will allow various inventory control policies to be studied as they are applied independently to each item. Jointly replenished items, such as a group of items whose individual order quantities summed must not exceed a carload, cannot be accumulated. However, a group of items which have the same review period or method of order point/order quantity determination may be conveniently batched.

Mathematical Method: N/A

Restrictions/Range: N/A

Storage Requirements: N/A

**Program Specifications:** This program was written in the FORTRAN language and has been compiled for the IBM 1620. With minor modification of the input/output statements, it can readily be compiled for any computer which accepts FORTRAN.

**Additional Remarks:** Flexibility is available in the following respects. Both the order point and order quantity may be fixed or variable as specified. Review may be periodic or occur every transaction. A forecast through the lead time is available by means of exponential smoothing with trend correction and an option of adjusting for seasonality. Lead time may either be fixed or be generated by Monte Carlo techniques. At any time, as in a good real world system, modification may be made of the order point, order quantity, safety stock level and the exponential smoothing factor.

The output will present a running account of all significant happenings. In summary, for each item the average inventory level, service percentage, number of out of stocks, number of replenishment orders and approximate standard deviation of forecast error are reported.

**IBM 1620 PROGRAM LIBRARY ABSTRACT**

**File Number:** 10.2.002

**Title:** The Inventory Management Simulator (Tape)

C. J. Walker & G. M. Goodfriend

IBM Corporation

618 S. Michigan Avenue

Chicago 5, Illinois

**Abstract Data for this program is identical to data for program number 10, 2, 001 except that program is for the IBM 1620 tape system.**

**IBM 1620 PROGRAM LIBRARY ABSTRACT**

**File Number:** 10.2.003

**Title:** A N INVENTORY MANAGEMENT SIMULATOR (Card)

J. L. Spivack & Cliff Smith

Direct Inquiries to: John L. Spivack

IBM Corporation

1955 The Alameda

San Jose, California

**Purpose/Description:** This simulator allows the user to test various decision rules concerning the management of inventory levels, ordering quantities, and forecasting techniques. It gives costs for each set of decision rules.

Method: N/A

Restrictions/Range: N/A

Storage Requirements: N/A

(Continued on next column)
IBM 1620 PROGRAM LIBRARY ABSTRACT

LESS

(Least-Cost Estimating and Scheduling)(Scheduling Portion)-(Card)

Ray N. Souer
IBM
2401 South Main
Houston, Texas
CA-3-4321

Purpose/Description: For a project that may be described in terms of an arrow diagram of its component jobs, this program finds the minimum project completion time. The earliest and latest start and finish times for each job and the total and free float times are calculated.

Mathematical Methods: Standard

Restrictions, Range: The sum of nodes and job arrows may be as high as 1612

Storage Requirements: Program - 3275 digits

Equipment Specifications: 162K 1622 Card Read Punch. No other special features required

Additional Remarks: Programmed in SPS. The usual restriction on numbering of jobs and order of input have been removed.

---

IBM 1620 PROGRAM LIBRARY ABSTRACT

LESS II (Least - Cost Estimating and Scheduling)(Scheduling only) (Card)

R. Poland

Direct Inquiries to: R. Poland
IBM Corporation
South Bend, Indiana
CE-2-851

Purpose/Description: Critical path scheduling routine in which time (Duration) units are expressed in terms of hours or days. The output is listed in the same units of time.

Method: N/A

Restrictions/Range: Will handle 2100 events.

Storage Requirements: None.

Equipment Specifications: Memory 20K; Automatic Divide; No other Special Features Required.

Additional Remarks: Demonstration tape runs approximately 15 minutes for three forms of output - underlined time interval, time in terms of shop days and in terms of hours.

---

IBM 1620 PROGRAM LIBRARY ABSTRACT

CRITICAL PATH SCHEDULING (Card)

Chuck Snyder & Jim Sondik

Direct Inquiries to: Jack Burgeson
IBM Corporation
340 S. Broadway
Akron, Ohio

Purpose/Description: The purpose of this brief program is, primarily, to illustrate how simple the Critical Path Scheduling Algorithm (a type of Dynamic Programming) really is. This is accomplished by coding the entire critical path finding portion in the Fortran language for up to 180 jobs in less than one page of statements.

Method: Dynamic programming algorithm

Restrictions/Range: 180 jobs. Finds total project time and indicates critical jobs.

Storage Requirements: N/A

Equipment Specifications: Basic card 1620. Program available on cards in Fortran form. Could easily be translated to any machine configuration accepting Fortran language.

---

IBM 1620 PROGRAM LIBRARY ABSTRACT

THE CHINESE BAR & RING PUZZLE  (Card)

D. N. Leeson
IBM Corporation
425 Park Avenue
New York City, N. Y.

Purpose: This program generates an optimal solution to the Chinese Bar & Ring Puzzle. The program has only intellectual interests and serves no useful function unless one is interested in the problems of generating a reflective gray code.

Method: Not given, SPS Language.

Restrictions, Range: Does not apply.

Speed: Variable depending upon initial game conditions.

Storage Requirements: 2,500 core positions.

Equipment Specifications: 1620 with attached 1622.

---

IBM 1620 PROGRAM LIBRARY ABSTRACT

1620 SIMULATION OF A ONE-ARMED BANDIT (Tape)

Direct Inquiries to: Dick Conner
IBM Corporation
425 Seventh Avenue
Pittsburgh 19, Pennsylvania

Purpose: The program uses a pseudo-random number generator to select and print a combination of three characters from a six character set (S, V, R, G, T, R). The payoff, if any, is calculated and printed in edited format. Each depression of the "start" key initiates another play. The pseudo-random number generator determines how much the wheel spins, by varying the interval between printing of the characters; but there is no significant correlation between this delay and the character selected.

Stakes, which may be changed between plays, are determined by the sense switch settings. The wheel may be stopped by the sense switch, with maximum payoffs or losses to be printed in edited format, and the program to reinitialize for another player. If "bust" occurs, the "house" may at any time cease printing of gross totals of bets, payoffs and net profit for the day.

Restrictions, Range: Not given.

Method: Runs the pseudo-random number generator, partially initialized by playing player's selection output each time the program is loaded.

Source Language: 1620 SPS.

---

IBM 1620 PROGRAM LIBRARY ABSTRACT

Chinese Bar and Ring Puzzle  (Card)

D. N. Leeson

Direct Inquiries: D. N. Leeson
IBM Corporation
425 Park Avenue
Pl 1-6500

Purpose/Description: This program generates an optimal solution to the Chinese Bar and Ring Puzzle. The program has only intellectual interest.

Mathematical Method: Not Given.

(Continued on next page)
IBM 1620 PROGRAM LIBRARY ABSTRACT

THE EXECUTIVE GAME (Tape)
E. Jerry & J. A. N. Lee

Direct Inquiries to: Dr. J. A. N. Lee
Queen's University Computing Center
Ontario, Canada

Purpose/Description: To familiarize business students with the processes of business decision and the resulting effects on the market. This program is a translation of the U.C.L.A. game for the IBM 650.

Method: N/A

Restrictions/Range: Eight teams

Storage Requirements: Total memory 20K and no other special features required.

Additional Remarks: This program is written in L.P.S. The need for an automatic divide feature will be a function of which L.P.S. tape is available. The 1620 User's Group has permission to publish this program and preliminary writeup, but its use should be restricted to members of the Group only. A more complete writeup will be available later. This has been put in this form following many requests from users.

IBM 1620 PROGRAM LIBRARY ABSTRACT

BLACKJACK GAME (Tape)
A. J. Lang

Direct Inquiries to: A. J. Lang
Fairchild Camera and Instrument Corporation
Da Mont Military Electronics Department
Defense Products Division
750 Bloomfield Avenue
Clifton, New Jersey

Purpose: The program to play the game of blackjack (commonly known as "21") was designed for demonstration purposes for the 1620 Data Processing System.


Restrictions/Range: Does not apply.

Storage Requirements: 6607 core positions.

Equipment Specifications: 1620 with attached 1621. No other special features are required.

IBM 1620 PROGRAM LIBRARY ABSTRACT

BLACKJACK DEMONSTRATION (Card)
Earl E. Hitt

Direct Inquiries to: Earl E. Hitt
IBM Corporation
3800 Lindell Boulevard
St. Louis, Missouri

Purpose/Description: Demonstration Game of Black Jack between the 1620 as dealer and two players. 1620 deals two cards to each of two players and itself. Players may take additional cards as they desire. 1620 makes these decisions for itself. Progress of game is clearly pictured on typewriter and choice comments are typed out at end of each hand giving almost human image to 1620.

Method: N/A

(Continued on next column)
Restrictions/Range: N/A
Storage Requirements: From approximately 00000 to 13000
Equipment Specifications: Tape system, memory 20K, automatic divide, indirect addressing. No other special features required.

Additional Remarks: The original program was written in the Symbolic Programming System, with fixed point input. No subroutines are required, and the program is not relocatable.

An average run takes approximately 30 seconds running time and from four to five minutes for the entire output.

The random number generator used is admittedly not the best, but has been found to be quite adequate.

IBM 1620 PROGRAM LIBRARY ABSTRACT
File Number 11, 010

The 1620 Self-Demonstrator (Tape)
Jack Mises

Direct Inquiries to: Jack Mises
IBM Corporation
340 W. Washington Ave.
Madison, Wisconsin

Purpose/Description: This program demonstrates the 1620 Tape System by giving pertinent facts, punching and reading tape, typing and demonstrating arithmetic speed. It is a real attention-getter in showing the IBM 1620 Tape System.

Mathematical Method: N/A

Restrictions Range: None

Storage Requirements: N/A

Equipment Specifications: Memory 20K; no other special features required.

Additional Remarks: The second and last records on the program tape can be changed to suit individual needs. The first record on tape is program. The second record can be changed for specific organization. The last record can also be changed for specific organization.

IBM 1620 PROGRAM LIBRARY ABSTRACT
File Number 11, 011

1620 SIMULATION OF A ONE-ARMED BANDIT (Card)
Dick Connor

Direct Inquiries to: Dick Connor
IBM Corporation
421 Seventh Avenue
Pittsburgh 19, Pennsylvania

Purpose/Description: The program uses a pseudo-random number generator to select and print a combination of three characters from a six character set ($, *, @, =, , ; ). The payoff, if any, is calculated and printed in edited format. Each depression of the "start" key initiates another play. The pseudo-random number generator also determines how long each wheel spins, by varying the interval between printing of the characters; but there is no significant correlation between this delay and the character selected.

Stakes, which may be changed between plays, are determined by the sense switch settings, thus affording the actor a choice of fifteen different amounts to bet, from five cents to ninety cents. The sixteen combination of switch settings causes the player's net winnings or losses to be printed in edited format, and the program to relaunch for another player. The "house must" can at any time cause printing of grand totals of bets, payoffs and net profit for the day.

Restrictions, Range: Not given.

Method: Runcible pseudo-random number generator, partially initialized by player to prevent identical output each time the program is loaded.

The mode of arithmetic is fixed point, with maximum grand total permitted equal to $999,999,999,99, which permits several months of continuous play.

Storage Requirements: Locations 00000 through 05455, not relocatable.

Source Language: 1620 SPS.

Remarks: Running Time: Due to random times the wheels spin, running time per play varies from about nine seconds to about 13.5 seconds.

Equipment Specifications: Standard 1620 Card. The I/O equipment is used only for loading. The end-of-job memory clearing routine works only on a 20K machine.
IBM 7070 Library Program Abstracts
Available prior to January 1962

IBM 7070 - Adding to Basic Fortran
Russell Ranshaw
Computer and Data Processing Center
University of Pittsburgh
Pittsburgh, Pa., Pennsylvania

a. Purpose: The additions to Basic Fortran were made to bring the Basic Fortran System up to date. The additions are:
1. IF SENSE SWITCH 0 a,
2. IF SENSE LIGHT 0 a, b,
3. SENSE LIGHT 0 ON,
4. ASSIGN 0 TO V,
5. GO TO V,
6. ASSIGN a, 70 V,
7. ASSIGN a, 70 V,
8. ASSIGN a, 70 V. 

b. Machine Requirements:
Processor: The additions occupy 120 locations; at present they are assembled into 0000-0199. There is no room, however, in a 6k machine to make the same additions.
Object: Electronic switches 1-9 may be used as SENSE LIGHT instructions for "lights" 1-9 are used.

General Descriptions: The Machine language Realizations of the above statements are:
1. STMT 9 (1, STAINT 9)
2. STMT 9 (1, STAINT 9)
3. STMT 9 (1, STAINT 9)
4. STMT 9 (1, STAINT 9)
5. STMT 9 (1, STAINT 9)
6. STMT 9 (1, STAINT 9)
7. STMT 9 (1, STAINT 9)
8. STMT 9 (1, STAINT 9)

Capabilities and Limitations: Does not apply.

IBM 7070 Library Program Abstracts
Available prior to January 1962

IBM 7070 Library Program Abstracts
Available prior to January 1962

7070 - BSTRP - Functions Subroutines for Basic Fortran
Russell Ranshaw
Computer and Data Processing Center
University of Pittsburgh
Pittsburgh, Pa., Pennsylvania

a. Purpose: The Format statement for Basic Fortran does not include printer control options. BSTRP has been written to restore the 7400 printer paper when desired. Fortran use:

b. Machine Requirements: IBM for 7070, 5000 printer on type: IBM 7400 printer panel.

c. General Description: The routine is supplied in 5/0 relocatable form, suitable for use with the Basic Fortran Package dock. Upon entry, the Routine prints a report consisting of one word, having control information to cause the 7400 to restore to channel 1. Control is then returned to the main Program.

d. Capabilities and Limitations: Does not apply.

IBM 7070 Library Program Abstracts
Available prior to January 1962

IBM 7070 Library Program Abstracts
Available prior to January 1962

7070 Generation of 1461 Optimized Programs (GOOP)

Contributed By
Author: Elmer D. Stonehill
Organization: The Ohio Oil Company

a. Purpose: To generate efficient 1461 card-to-tape, tape-to-printer, and tape-to-card programs which reduce 7070 programming effort and eliminate the need for 1461 programmers and 1461 program maintenance.

b. Machine Requirements:

1961 (1) 10K Memory, and (2) five Model 7201 or 7210 Tape Units.
1961: (1) Model C2 Processing Unit with a minimum of 10 Memory, (2) 1461 Card-Read Punch, (3) 1461 Model 2 Printer, (4) One Model 7201 or 7210V Tape Unit, (5) High-Low-Equal Compare, and (6) the Advanced Programming Package.

c. General Description: Parameters describing the input and output of the 1461 programs desired are input to the 7070 generator with the generated output being a 1461 program loaded and program rating. Although the generator program has 42 program phases consisting of 35,000 instructions, only 5-3 minutes of 7070 time is required per generation. The resulting 1461 programs process approximately 400 cards per minute (card-to-tape with a one-tooth clutch), 400 lines per minute (tape-to-printer, single-spaced with and print buffer), and 250 cards per minute (tape-to-card).

d. Capabilities and Limitations: Does not apply.

IBM 7070 Library Program Abstracts
Available prior to January 1962

IBM 7070 Library Program Abstracts
Available prior to January 1962

IBM 7070 Library Program Abstracts
Available prior to January 1962

IBM 7070 Library Program Abstracts
Available prior to January 1962

IBM 7070 Library Program Abstracts
Available prior to January 1962

IBM 7070 Library Program Abstracts
Available prior to January 1962

IBM 7070 Library Program Abstracts
Available prior to January 1962

IBM 7070 Library Program Abstracts
Available prior to January 1962

IBM 7070 Library Program Abstracts
Available prior to January 1962

IBM 7070 Library Program Abstracts
Available prior to January 1962

IBM 7070 Library Program Abstracts
Available prior to January 1962

Continued on next page
IBM 7070 Library Program Abstracts

SEUS PROGRAM ANALYSIS (ZPA) COMPUTER SYSTEM

Contributed By:

Author: Operations Engineering Department
Organization: Western Electric Company, Inc.
Address: 9215
304 Graham-Hoytale Road
Burlington, North Carolina

a. Purpose: The ZPA Computer System is a series of four programs designed to process PERT type networks on an IBM 1401/7070 computer system.

b. Machine Requirement: The programs in the system are written for an IBM 1401. 11, machine and an IBM 7070, 2 channels, 11,0,0, tape oriented machine.

c. General Description: The four programs in the ZPA System are as follows:

1. ZPA Card to Tape (Program 61000 - 1401) One reel of magnetic
2. ZPA Calculation (Program 61050 - 7070) tape required for 7070
3. ZPA Sort and Merge (Program 61020 - 7070) Program deck and listing.
4. ZPA Print and Edit (Program 61100 - 1401)

The 1401 computer is used primarily as an input and output device. The 7070 is used to calculate network data, to merge activity descriptions with calculated data, and to sort the critical path and negative slack activities. Input to the system is on cards and the output is a series of printed reports. Any number of networks may be processed during the same computer run and each program of the system will process all networks without interruption. Each network is separated by segment marks on tape. The existence of input errors in a network will not restrict the successful processing of other valid networks.

d. Capabilities and Limitations: There are certain requirements that must be considered in processing networks with the ZPA System. First, the programs were designed to process 'activity oriented' networks. Although 'event oriented' networks can be processed, some confusion could result in the interpretation of the program outputs. Second, the programs have been written to analyze networks with a maximum of 1,500 activities. Third, random numbering of network activities is not permissible. Events must be numbered sequentially in ascending order. The successor event number of an activity must be higher than its predecessor. Consideration of these requirements is important when preparing the basic network drawings.

IBM 7070 Library Program Abstracts

288 to 7070 Tape Record Conversion (XXA15)

R. T. Seidler, Jr.
Texas Instruments Incorporated
August 10, 1960

a. Purpose: To convert 650 tape records, written either alpha or numeric, to 7070 tape records.

b. Machine Requirement: One (1) 7500 card reader, two (2) 729 II or 729 IV tape drives, 10K words of core storage

c. General Description: The parameters of this routine are established from control card information. The information in these cards defines the 650 record, the format of the desired 7070 record, output blocking, individual record lengths (input and output), alpha/numeric words, field changes, and other information necessary to create a required 7070 file from an existing 650 file.

d. Capabilities and Limitations: The routine is capable of converting any 650 record of from 1 to 45 words in length to a 7070 record; these are certain limitations as to output records and field changes which are covered in detail under the section headed "Complete Description". The routine utilizes the IBM Input-Output Control System (OCS).

IBM 7070 Library Program Abstracts

288 to 7070 Tape Record Conversion (XXA15)
a. General Description: With the program in storage, a priority branch to the 705 will occur when channel A is switched on. The routine reads the control card, sets up the tape operation, alters 9004 to enter the second phase of the routine, and returns control to the main program. Processing interrup tends to read a data card using a 16 word READ routine and reorganize the tape if the first word is not a status word and a type record where the status word is written. The card is then processed as a tape record is written; if the card is blank, a segment mark is written. In all cases, a priority return occurs after the tape is written.

d. Capabilities and Limitations: Does not apply.

IBM 7070 Library Program Abstracts

7070 - Load Subroutine
R. Haertle
AO Data Div GMC
Milwaukee, Wisconsin

a. Purpose: To load data at object time into specified locations. This may be fixed, floating, or alphabetic data.


c. General Description: Input data of the following form will be converted:

\[-12.945, -123.665-7, +1, 0, -1.045, \]

to the following internal form:

\[-501040000, -5010400000, +0000000000, +0000001054, \]

d. Capabilities and Limitations: Input format must conform to detailed operational description.

IBM 7070 Library Program Abstracts

7070 Module 11 Self-Checking Digit Calculator
Contributed by: Alex Serboff
IBM Datacenter
2425 Euclid Avenue
Cleveland, Ohio

a. Purpose: To allow Module 11 self-checking digits to numbers over a predetermined range or series of ranges.

b. Machine Requirements: 5 K by 60 tape 7070 with program to be brought in from additional tape, card reader, or console card reader.

c. General Description: The program is designed to compile check digits for numbers of from one to nine digits. A count and hash total of valid numbers is included for output purposes. The program calculates check digits at a rate of 900 per second.

IBM 7070 Library Program Abstracts

7070 - IBM 7070 Program Modification Routine
R. B. Reinhart and G. F. Crane
IBM Poughkeepsie
Rensselaer, New York

a. Purpose: The IBM 7070 Program Modification Routine is a subroutine which processes program modifications, prepared as outlined in the General Description. It should be mentioned that a program about to be loaded is charged while it is being loaded into core storage. It offers the unique advantage of easy reassemblage of the corrected program at any stage in its development.

b. Machine Requirements: The Modification Routine utilizes all available memory below word 0000. All memory assignments, with the exception of the tape error routine, may be changed through reassembly of the program. Overlay with the subject program is possible and often desirable.

For card input the following devices are required:

- 7090 Card Reader with Utility Control Panel
- 7000 Unit Record Synchronizer

For tape input the following devices are required:

- Tape Units - one or two channels with associated tape units as required to load the subject program.

IBM 7070 Library Program Abstracts

7070 - Tape Copy Routine
Russell Frensham
Computational and Data Processing Center
University of Pittsburgh
Pittsburgh, Pennsylvania

a. Purpose: This routine will read input tape records any reasonable length, in either high or low density, and write output tape records of the same size, in either high, low, or the same density. Input or output tapes may be rewound or backspaced before copying.

b. Machine Requirements: The entire 7070 is assumed to be available for use. The routine is written in present 7070. Any number of tape channels may be used, according to the copying pattern to be followed.

c. General Description: Pseudo-instructions, punched up to 8 per card, are interrogated. The "Instruction" provides information as follows:

1. Input tape
2. Output tape
3. Output density
4. Input backspacing
5. Output backspacing
6. Tape rewind - yes or no
7. Output rewind - yes or no
8. Input rewind - yes or not
9. Output rewind - yes or not

The routine is tape-limited in operating speed.

IBM 7070 Library Program Abstracts

7070 - 7070 Library Program Abstracts

General Description: The routine will copy up to 8,880 word records, any density, any combination of segment marks, tape marks, record marks, and alphanumeric records, an uncorrectable read error on the input file stops the current "Instruction".

Chain of Thought:

IBM 7070 Library Program Abstracts

7070 - Tool Library Program Abstracts

General Description: The routine will copy up to 8,880 word records, any density, any combination of segment marks, tape marks, record marks, and alphanumeric records, an uncorrectable read error on the input file stops the current "Instruction".
IBM 7070 Library Program Abstracts

7070 SIMPLE IOCS

Contributed By: Robert Judson
The B. F. Goodrich Company
Akron 38, Ohio

A. Purpose: To provide a simple method for handling tapes which uses priority routines to handle possible errors but not to save time. For small input-output scientific problems.

B. Object Routine Machine Requirements: Tape Units

C. Object Routine Produced: Routines to handle all priority possible tape commands. Operations which have no priority mode do not need and do not use this package.

D. Source Language Entry: T JXL, TCX, P 3
P (Tape Command) Any channel-unit, and RDW (if applicable)

E. Capabilities and Limitations: In case of an uncorrectable error, the priority mask register and initial and final status words (as required by tape units used).

IBM 7070 Library Program Abstracts

7070 MATES (Master Tape Executive Program)

Author: Vincent J. Battaglia

Organisation: INTERNATIONAL BUSINESS MACHINES
Chicago Downtown
618 S. Michigan Avenue
Chicago, Illinois

A. Purpose: To Generate data files from card input for use in testing 7090 programs.

b. Machine Requirements: (Include machine components, special features, storage requirements, control panels - standard or special).

i. 10,000 words of memory.

ii. Card-to-Tape equipment to create an input tape to the BFG.

iii. One 727 II, IV or 729 II, IV Tape drive (in addition to drives for files being created).

The BFG program can be patched for use with certain other machine configurations. See BFG writeup for details.

c. General Description: (Mathematical method, accuracy, speed, if appropriate).

d. Capabilities and Limitations: Only the tape operations (PTTR, DPTTR, DITW, DPTW2 and DPTWC) will be performed.

IBM 7070 Library Program Abstracts

7070 PAT COMPILER

Contributed By: W. J. Walker
IBM Corporation
N. Y. Financial
2 Broadway
New York 4, N. Y.

A. Purpose: The Pat Compiler Program compiles a PAT (Procedure for Automatic Testing) System Tape supplying the desired utility programs used in testing.

b. MACHINE REQUIREMENTS:

5 K memory
1 Output Tape unit
1 Input Tape unit or 7500 card reader
Standard IBM utility panel SW'S 1 & 2 on A

c. GENERAL DESCRIPTION: The Pat Compiler program will create a 1093 Pat System Tape of program packets from either the card reader or a tape created off line in alpha card image form. As the Pat tape is being created each program packet number will be typed. The tape reads from and writes into each program packet number as the program packet number is created. The Pat compiler is designed to create the utility program, namely, the Pat tape. The typed list will be used as the sequence and programs on tape and serve as a reference sheet during use. The PAT Compiler Call card defines the beginning of the program and contains the necessary information for compiling and finding the utility program.

d. CAPABILITIES AND LIMITATIONS: Utility Programs can be compiled only in the normal logical sequence as specified by the control card.
IBM 7070 Library Program Abstracts

7070 PAT COMPILER SYSTEM

Contributed By: Joseph G. Cappe, Jr.
IBM Corporation
Los Angeles Datacenter
1424 Wilshire Blvd.
Los Angeles 5, California

A. Purpose: This system, consisting of several programs, is designed to assist in the debugging of multiple object programs by facilitating the preparation and use of a PAT system tape. This PAT Compiler System allows multiple programs and data to be incorporated into individual test packets on a single PAT tape, with the insertion of all utility routines needed by the PAT Compiler program.

B. Machine Requirements: The PAT Compiler System requires, as a minimum, a 5K core, four-tape IBM 7070 with either a 7000 or a 1401 Card Reader. The PAT Compiler program is available in two versions, one using the IBM 7070 DCG system and requiring a 10K core 7070; the other not using DCG and not requiring the 10K core 7070. Either PAT Compiler may be modified to run on any given logical configuration by the insertion of a Configuration Control card, containing the desired machine configuration.

The object programs being tested must make use of the standard IBM 7070 Load Program. During testing, the PAT Compiler System places no restrictions on the use of the computer by the object program.

C. General Description: For each program to be debugged, one control card must be punched. Its purpose is to separate the programs and to supply to the PAT Compiler pertinent information. Multiple sets, consisting of a control card, test data, and object program, may then be processed by the PAT Compiler program to produce a self loading PAT tape. The resulting PAT tape may then be used as many times as desired to test the programs.

The utility programs used by the PAT Compiler System are modified versions of the standard utility programs.

IBM 7070 Library Program Abstracts

7070 LORELLI (Location Reference Listing)

Author: Mike Clark
Organization: Zurich Insurance

A. Purpose: LORELLI is a program used in conjunction with a modified Sort 70 Compiler, designed to create a cross-reference listing of programs assembled by Autocoder 74.

B. Machine Requirements: STORAGE TAPES

- LORELLI: 5000 words
- SORT 70: 5000 words

C. General Description: The cross-reference of the object program is in several major areas:

1. Listing by address
2. Listing by field word usage
3. Listing by Electronic switch usage
4. Listing by Accumulator usage

D. Capabilities and Limitations: The listing may or may not cross-reference the following based on alternative switches:

1. Listing by Accumulator usage
2. Comments statements (in column A)
3. Steps generated by DCG or other macros or subroutines on the 74A assembly tape.

IBM 7070 Library Program Abstracts

7070 SCAN

Contributed by: Ronald J. Reggie
IBM Corporation
Charleston, West Virginia

A. Purpose: To edit basic Fortran programs prior to doing a Fortran assembly.

B. Machine Requirements: Basic 7070. Program is set up to accept information from a card reader or a tape unit.

C. General Description: This program will find many common errors in Fortran programs. Over fifty errors are caught by this routine. For example:

1. Mixed arithmetic mode
2. Dimensioned variable written without subscript
3. Intersecting D O loops
4. Misplaced commas in control statements
5. Undefined branches and D O's
6. Names that are used but never defined

D. Capabilities and Limitations: This routine was written to be inserted into a Fortran compiler system that will make batch assemblies using five tape drives without any card equipment, but it can be run separately. The tables have been set up to Basic Fortran specifications, i.e., D or O's 150 variables, etc. Subscripts are not checked.

IBM 7070 Library Program Abstracts

7070 - 650 PANEL SIMULATOR

Contributed by: C. W. Kasper & J. W. Lake
Texas Instruments Incorporated

A. Purpose: This program is designed for use in conjunction with the IBM 7070 Program which simulates the IBM 650. This program simulates the 650 panel, thus eliminating the need for wiring 7070 read and punch panels to replace the 532 panels used by the 650 programs.

B. Machine Requirements: Index words 70 through 81, electronic switches 22 through 29, and 1500 instructions and locations that may be assembled anywhere outside of the area required by the IBM 7070 Simulation Program.

(Continued on next page)
The IBM 7070 Simulation Program with the Panel Simulator included can usually be run on a 5K core machine by removing unused portions of the program. The entire system is retested, and it must have a 10K core machine. Some of the sections which can be easily removed are: ram, segments, (-) OP codes, floating point, index registers, or any of the other routines which your particular installation does not use.

c. General Description: For each 650 program a set of read- and/or punch-format cards must be prepared. From these format cards, the program will set up the card image in memory just as the Type 533 panel would have read the card in, or will punch the card image just as the Type 533 panel would have punched it.

d. Capabilities and Limitations: The running time is increased only slightly above that of the usual procedure of using a board for each program.

IBM 7070 Library Program Abstracts

V707 - Simulation of Basic 650 on Basic 7070

Contributed By: L. J. Berg, R. Nunn, H. Monroe

Purpose: To simulate a basic 650 program on a basic 7070. The 650 control panel is also simulated.

Machine Requirements: Includes machine components, special features, storage requirements, control panels - standard or special.

1. 5K - 7070
2. 7500 Card Reader
3. 6240 Card Punch
4. 80-90 Alpha panels for reader and punch

General Description: Mathematical method, accuracy, speed, if appropriate

Most 650 programs run 3-1/2 to 5 times as fast on the 7070.

IBM 7070 Library Program Abstracts

V707 - Gronk - a W70 Simulator for the 650

Contributed By: L. J. Berg, R. Nunn, H. Monroe

Purpose: Gronk is a program for the IBM 650 to simulate an IBM 7070.

Machine Requirements:
1. IBM 650
2. 6240 Card Punch
3. 80-90 Alpha panels for reader and punch

General Description: Gronk is primary function is to provide potential users with currently have a 650 with a means of testing small 7070 programs and subroutines without the expense of 7070 time elsewhere.

Capabilities and Limitations: Gronk is able to simulate most of the 7070 machine, including editing commands, priority processing, electronic switch, 90 index words, all three table-look-up, and tape. It will not, however, simulate the following:
1) Edit commands (EHA, EAN, etc.)
2) Double precision floating commands
3) Some tape commands:
   a) TEIL
   b) TDL
   c) TDLD
   d) TRA
4) Diagnostic interrogation (C00)
5) Assembler steps
6) Tables
7) Stacking latch commands

Gronk simulates the first 600 words of 7070 storage; if no tapes are used, an additional 200 words become available.

IBM 7070 Library Program Abstracts

V707 SIMULATING THE CARD 650 ON A TAPE ORIENTED 7070

Contributed By: John D. Fehd

Purpose: - - This program is designed to simulate card 650 programs at speeds ranging from 2 to 3 times faster than the present IBM 650 Simulator for the 7070.

Machine Requirements: - - A 5K 7070 with one tape channel and two 729 tape drives. No control panels and no special features are required.

General Description: - - This program is designed to handle multiple 650 programs on one or more tapes. A segment mark is to be placed just prior to each 650 program and the first record must give the section setting and program number. The 7070 can be halted just prior to each 650 program if desired (alt. SW). If a 650 program cannot be completed, it can be by-passed and the 7070 will start the next 650 program on the input tape.

Capabilities and Limitations: - - Three types of 650 programs have been tested and timed on both the 650 and the 7070 with the following results:

Limit

<table>
<thead>
<tr>
<th>Operation</th>
<th>Speed</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Read Bound</td>
<td>500 Words</td>
</tr>
<tr>
<td>2. Punch Bound</td>
<td>1800 Words</td>
</tr>
<tr>
<td>3. Compute Bound</td>
<td>1900 Words</td>
</tr>
</tbody>
</table>

This program uses five cards per tape record and the tapes are controlled by the IBM 7070 Input/Output Control System. It will not simulate any of the internal operation code instructions and it is restricted to one type 533.

Each 650 program that is to be simulated will require 4001 programs for input and output.

An operators manual and technical description will be supplied with the program.

IBM 7070 Library Program Abstracts

SIMULATION OF CARD OR TAPE 650 ON THE 7070

Contributed By: L. J. Berg, R. Nunn, H. Monroe

Organization: Curtiss-Wright Corporation, Wood-Ridge, New Jersey

Purpose: To simulate a card or tape 650 on a tape oriented 4010-7070 system.

Machine Requirements:

Minimum of 729 II or 729 IV tape drives for simulating unit record input and output. Additional tape drives are required for tape input and output.

This system is designed for a 10K machine but can be reduced to a 5K machine.

General Description:

This operating technique combines the use of a portion of the PAT System (Procedure for Automatic Testing developed by IBM's New York Data Center), IBM's 650 Simulator Program, modifications to the Simulator Program and a 1401 Program developed at the Wright Aeronautical Division.

Capabilities:

A card deck containing the PAT System, the Simulator Program, and the 650 Program is developed for each 650 Program to be simulated. A series of these decks can be written on a reel of tape using a Type 1401C System. The card decks are made up so that:

1. The information which the Simulator Program normally calls for through the use of control cards is built into the package.
2. Instructions for initializing the succeeding package are included.
3. A routine to write a tape mark on the tape unit which simulates the card output is included.
4. Multiple data files may be processed using the same 650 Program without the need to prepare a separate input tape for each input file.
5. A dump (both core and tape) may be taken on any channel and tape drive.
6. 650 load cards are recognized by an alpha sign in word 10 rather than by a plus sign.
7. The output tape simulating card output may be written in either compressed or normal mode.
IBM 7070 Library Program Abstracts

ABFLOATSIM — ABBreviated FLOATing point hardware Simulator

Contributed By:
Author: H. Hyman, Applied Science
Organization: IBM Svenska AB

Gitegatan 20
Stockholm 6, SWEDEN

b. Capabilities and Limitations: The normal restrictions on the
floating decimal hardware.

b. Purpose: An interpretive subroutine which essentially simulates
floating decimal hardware.

b. Machine Requirements: 2 index words and 126 ordinary storage
locations.

c. General Description: When the subroutine is entered, ABFLOATSIM
will perform instructions sequentially starting with the instruction
immediately following the linkage instruction. These instructions may
be floating decimal or ordinary 7070 instructions. Floating decimal
instructions are written as for a machine equipped with floating
decimal hardware. An unconditional branch instruction or a conditional
branch instruction, where the branch condition is met, will, when it
appears in the sequence, cause an exit from the subroutine.

Machine Requirements: Basic

Purpose: Instrument the original simplex algorithm with variations
for the IBM 7070.

General Description: Utilizes the original simplex algorithm with variations
for the IBM 7070.

a. Purpose: Instrument the original simplex algorithm with variations
and options for the IBM 7070.

IB 7070 Library Program Abstracts

IBM 7070 Linear Programming Code S1

Contributed By:
Author: A. E. Speckhard
Organization: Western Region

a. PURPOSE: To calculate the structure factors of crystals of Triclinic,
Monoclinic or Orthorhombic classes (and also of Hexagonal,
Tetragonal or Cubic with redundant atoms).

b. MACHINE REQUIREMENTS:

10,000 curves (or 5,000 curves)

1 750 (Synchronizer 1) with IBM utility board
1 7550 (Synchronizer 2) with IBM utility board
2 channels (1 and 2), 1 unit each

2. Accommodates large problems. A realistic limit is approximately 200–250 equations although larger problems
may be run depending on availability of floating point
hardware and program options desired by the user.

3. Describes the solution completely including cost ranges
with upper and lower limiting variables, and activity
ranges with upper and lower limiting variables.

4. Operates in single or double precision floating point as
option of the user. Input data is in single precision
fitted point form.

d. Capabilities and Limitations: The code is written in a special
symbolic assembly language using subroutine structure
and includes a highly flexible operating system. Maximum
problem size is approximately 400 equations and 10,000
variables.

IBM 7070 Library Program Abstracts

IBM 7070 Linear Programming Code S2

Contributed By:
Author: D. C. Potter & A. E. Speckhard
Organization: Western Region

a. PURPOSE: Instrument the revised simplex product form algorithm with
variations and options for the IBM 7070.

(Continued on next column)
IBM 7070 Library Program Abstracts

7070 SINE COSINE SUBROUTINE

Contributed By: 2S Applied Programming Dept., IBM Corporation
1271 Avenue of Americas
New York, New York

A. Purpose: This program computes SINE X or COSINE X for $x \leq 1$.

B. Machine Requirements: The program uses only fixed point operation codes and can be used on all 7070 configurations.

C. General Description: The method consists of a separation into integral and decimal parts, an evaluation of \( \sin x \times L_c \) and an adjustment of sign for quadrant correction. The maximum error is \( \pm 10^{-8} \). Average execute time is 16.8 milliseconds.

D. Capabilities and Limitations: Input must be normalized floating decimal of form \( 1.00000000 \times 10^n \) and an evaluation of square root will cause an error halt.

IBM 7070 Library Program Abstracts

ARCSINE N

Applied Programming Dept., IBM

a. Purpose: This program computes the Arcsin N for \( N = 1 \) in floating decimal form.

b. Machine Requirements: The program uses only fixed point operation codes, and can be used on all 7070 configurations.

c. General Description: The Arcsin is approximated by means of the expression

\[
\frac{x}{\sqrt{1 - x^2}} < \text{Arcsin } x < \frac{x}{\sqrt{1 - x^2}} + \frac{x^3}{6}
\]

The maximum error is not greater than \( 5 \times 10^{-9} \). Average execute time is 9.7 milliseconds.

d. Capabilities and Limitations: Input must be normalized floating decimal numbers. The program requires 61 locations and will alter the three accumulators, index word 98, and the high-law-equal indicator. There must be a floating decimal square root subroutine available.

IBM 7070 Library Program Abstracts

Subroutine for IBM 7070

Rolls Royce Ltd., P. O. Box 31
Derby, England

Range: \[ |x| < 1 \]

Entry: X in accumulator 2 to 9 decimal places.

Method: Hastings, p. 140, with the coefficients \( C_1 \) multiplied by \( 10^2 \)

\[ C_1 = 0.00 0086 0541 \quad C_7 = -0.00 0156 0740 \]

Error: Max. error is 1 in 8th decimal place.

IBM 7070 Library Program Abstracts

Rolls Royce Ltd., P. O. Box 31
Derby, England

Range: \[ |x| < 10 \]

Entry: X in accumulator 2 to 9 decimal places.

Method: Hastings, p. 143.

\[ y^2 = 1 - x^2 \]

Error: Max. error is 1 in 8th decimal place.

IBM 7070 Library Program Abstracts

Rolls Royce Ltd., P. O. Box 31
Derby, England

Range: \[ |x| < 10 \]

Entry: X in accumulator 2 to 9 decimal places.

Method: Hastings, p. 143.

\[ y^2 = 1 - x^2 \]

Error: Max. error is 1 in 8th decimal place.

IBM 7070 Library Program Abstracts

Rolls Royce Ltd., P. O. Box 31
Derby, England

Range: \[ |x| < 10 \]

Entry: X in accumulator 2 to 9 decimal places.

Method: Hastings, p. 143.

\[ y^2 = 1 - x^2 \]

Error: Max. error is 1 in 8th decimal place.


IBM 7070 Library Program Abstracts

Available prior to January 1962

Sine-Cosine Subroutine

R. Hyman, Applied Science

Organisation: IBM Svenska AB

Address: Götgatan 20

Stockholm 6, SWEDEN

a. Purpose: To find sine and cosine of an angle.

b. Machine Requirements: Floating point hardware, 45 core storage words.

c. General Description: The subroutine will compute sine and cosine of an angle.

d. Capabilities and Limitations: Angles must be normalised floating point numbers. Accuracy: The maximum error is 0.001 radians. The result is stored in a 45-word word table. Entry is permitted in radians or degrees.

IBM 7070 Library Program Abstracts

Available prior to January 1962

Sine-Cosine Subroutine

M. Roberts

AC Spark Plug Div GMC

Milwaukee, Wisconsin

a. Purpose: To find sine and cosine of an angle.

b. Machine Requirements: Floating point hardware, 45 core storage words.

c. General Description: The subroutine will compute sine and cosine of an angle.

d. Capabilities and Limitations: Angles must be normalised floating point numbers. Accuracy: The maximum error is 0.001 radians. The result is stored in a 45-word word table. Entry is permitted in radians or degrees.

IBM 7070 Library Program Abstracts

Available prior to January 1962

Sine-Cosine Subroutine

M. Roberts

AC Spark Plug Div GMC

Milwaukee, Wisconsin

a. Purpose: To find sine and cosine of an angle.

b. Machine Requirements: Floating point hardware, 45 core storage words.

c. General Description: The subroutine will compute sine and cosine of an angle.

d. Capabilities and Limitations: Angles must be normalised floating point numbers. Accuracy: The maximum error is 0.001 radians. The result is stored in a 45-word word table. Entry is permitted in radians or degrees.

IBM 7070 Library Program Abstracts

Available prior to January 1962

Sine-Cosine Subroutine

M. Roberts

AC Spark Plug Div GMC

Milwaukee, Wisconsin

a. Purpose: To find sine and cosine of an angle.

b. Machine Requirements: Floating point hardware, 45 core storage words.

c. General Description: The subroutine will compute sine and cosine of an angle.

d. Capabilities and Limitations: Angles must be normalised floating point numbers. Accuracy: The maximum error is 0.001 radians. The result is stored in a 45-word word table. Entry is permitted in radians or degrees.
a. **Purpose:** A full precision, fixed point subroutine to compute the hyperbolic tangent.

b. **Machine Requirements:** All accumulators, the compare indicators, 1 electronic switch, 2 index words and 57 ordinary storage locations.

c. **General Description:** The program computes the hyperbolic tangent or cotangent of an angle given in radians. The argument X must satisfy: 

\[-10 < X < 10\]

IBM 7070 Library Program Abstracts

**MODULO 2** CONVERSION SUBROUTINE

**Contributed By:**

**Author:** S. Nordin, Applied Science

**Organization:** IBM Svenska AB

**Gavlegatan 20**

Stockholm 6, SWEDEN

---

**IBM 7070 Library Program Abstracts**

**Available prior to January 1962**

---

**SINE AND COSINE SUBROUTINE**

**Contributed By:**

**Author:** H. Hyman, Applied Science

**Organization:** IBM Svenska AB

**Gavlegatan 20**

Stockholm 6, SWEDEN

---

**IBM 7070 Library Program Abstracts**

**Available prior to January 1962**

---

**TANGENT-COTANGENT SUBROUTINE**

**Contributed By:**

**Author:** S. Nordin, Applied Science

**Organization:** IBM Svenska AB

---

**IBM 7070 Library Program Abstracts**

**Available prior to January 1962**

---
a. **Purpose:** A full precision, fixed point subroutine to compute the arcsine or arccosine function.

b. **Machine Requirements:** All accumulators, the compare indicators, 3 index words and 101 ordinary storage locations.

c. **General Description:** For arguments in the interval [-0.1, 0.1], repeated applications of the formula arcsine X = 2, 25 π X^2 + 0.1 arcsine (2X^2 - 1) will bring the argument to the interval [-0.9, 0.9]. The latter interval is subdivided into five intervals. In each interval the arcsine function is approximated by a polynomial of the fifth degree. In the interval [-0.9, 0.9], the function is approximated by arcsine X = 0.27 p.v. [1 - (X^2)^1/2 + (X^2)]^1/2. Accuracy: The magnitude of the maximum error is 2 x 10^-7. Average execution time: 6.8 milliseconds.

d. **Capabilities and Limitations:** The routine will give the principal values expressed in radians. The argument X must satisfy -1 ≤ X ≤ 1.

**IBM 7070 Library Program Abstracts**  
Available prior to January 1962

**SINE-COSINE SUBROUTINE**

**Contributed By:**

**Author:** G. J. Elliott, Applied Science

**Organisation:** IBM Svenska AB

Gevelsgatan 29
Stockholm 6, SWEDEN

**Purpose:** A full precision, fixed point subroutine to compute the sine, cosine, or tangent of a number.

**Machine Requirements:** All accumulators, the compare indicators, 2 electronic switches, 2 index words and 101 ordinary storage locations.

**General Description:** This subroutine uses an Exponential Subroutine by T. Ekav. Sine X and cosh are computed according to the definition formula. Both X are also computed in this way for X > 0.1 but other- wise both X are approximated by a polynomial. Accuracy: The maximum error is 8 in the last digit. Average execution time: 14.5 milliseconds.

**Capabilities and Limitations:** The magnitude of the argument must be less than 10^15.

**IBM 7070 Library Program Abstracts**  
Available prior to January 1962

**LOG BASE 10 OR BASE e**

**Contributed By:**

**Author:** S. Nordis, Applied Science

**Organisation:** IBM Svenska AB

Gevelsgatan 29
Stockholm 6, SWEDEN

**Purpose:** A full precision, fixed point subroutine to compute in floating decimal form.

**Machine Requirements:** All accumulators, the compare indicators, 2 index words and 55 ordinary storage locations.

**General Description:** By the use of well-known trigonometrical identities, the problem may be reduced to that of calculating the functions with arguments in the interval [-0.9740, π/4]. Then the functions are approximated by the polynomials:

\[
\begin{align*}
\sin (X) & 
= a_0 + a_1 X + a_2 X^2 + a_3 X^3 + a_4 X^4 + a_5 X^5 \\
\cos (X) & 
= b_0 + b_1 X + b_2 X^2 + b_3 X^3 + b_4 X^4 + b_5 X^5
\end{align*}
\]

Accuracy: The magnitude of the maximum error is 10^-7. Average execution time: 6.4 milliseconds.

**Capabilities and Limitations:** The argument X must be expressed in radians and satisfy -15 < X < 10.

**IBM 7070 Library Program Abstracts**  
Available prior to January 1962

**COTANGENT**

**Contributed By:**

**Author:** G. J. Elliott, Applied Science

**Organisation:** IBM Svenska AB

Gävlegatan 20
Stockholm 6, SWEDEN

**Purpose:** A full precision, fixed point subroutine to compute the cotangent.

**Machine Requirements:** All accumulators, the compare indicators, 2 electronic switches, 2 index words and 101 ordinary storage locations.

**General Description:** By the use of well-known trigonometrical identities, the problem may be reduced to that of calculating the cotangent function using the above polynomials. Accuracy: The maximum error is 8 in the last digit. Average execution time: 14.5 milliseconds.

**Capabilities and Limitations:** The magnitude of the argument must be less than 10^15.
IBM 7070 Library Program Abstracts

**Log x for IBM 7070**

M. Roberts

**AC Spark Plug Div GMC**

**Milwaukee, Wisconsin**

- **Purpose:** To find the logarithm of argument x in x or log x
- **Machine Requirements:** Floating point hardware, 63 words core storage

**General Description:** Evaluation of the following series:

\[
\log x = \sum_{j=1}^{\infty} \frac{x - 1}{j10^j} \quad \text{for} \quad 1 < x < 10 \quad \text{and} \quad 2 < x < 10
\]

**Capabilities and Limitations:** Input must be normalized floating point notation. Accumulators and X, L, E indicators are not saved.

---

IBM 7070 Library Program Abstracts

**EXPONENTIAL SUBROUTINE**

**Contributed By:**

T. Rabe, Applied Science

**Organisation:** IBM Svenska AB

**Göteborg 20 Stockholm 6, SWEDEN**

- **Purpose:** A full precision, fixed point subroutine to compute the exponential function.
- **Machine Requirements:** All accumulators, the overflow indicator, and 182 ordinary storage locations.

**General Description:** The exponential function is approximated by a polynomial of the third degree. The constants of this polynomial depend on the argument and are stored in a 64 word table. Accuracy: The maximum error is 10^-10. Average execution time is 8.4 milliseconds.

**Capabilities and Limitations:** The argument X of e^X must satisfy:

\[
10^{-14} < X < 10^{14}
\]

---

IBM 7070 Library Program Abstracts

**SQUARE ROOT X**

**Contributed By:**

H. Hyman, Applied Science

**Organisation:** IBM Svenska AB

**Göteborg 20 Stockholm 6, SWEDEN**

- **Purpose:** This program computes the square root of a 2-0 64 bit floating point number.
- **Machine Requirements:** This program uses only fixed point operation codes, and can be used on all 7070 configurations.

**General Description:** The method consists of a linear approximation followed by two iterations of Newton's formula (modified). The maximum error is 10^-9 in the eighth place of the digitals. Average execution time is 10.9 milliseconds.

**Capabilities and Limitations:** Input must be normalized floating point numbers of the form MMMMMMMMMMMM.M (MM = exponent = 20). An attempt to take the square root of a negative number will produce an error halt. The program requires 62 locations for instructions and constants, and temporary storage. It also requires 13 locations for instructions, constants, and temporary storage. It also requires 13 locations for instructions, constants, and temporary storage.

---

IBM 7070 Library Program Abstracts

**NATURAL LOGARITHM SUBROUTINE**

**Contributed By:**

T. Rabe, Applied Science

**Organisation:** IBM Svenska AB

**Göteborg 20 Stockholm 6, SWEDEN**

- **Purpose:** A full precision, fixed point subroutine to compute the logarithmic function.
- **Machine Requirements:** All accumulators, the compare indicators, and 115 ordinary storage locations.

**General Description:** The logarithmic function is approximated by a polynomial of the third degree. The constants of this polynomial depend on the argument and are stored in a 64 word table. Accuracy: The maximum error is 10^-10. Average execution time is 7.1 milliseconds.

**Capabilities and Limitations:** The argument X of log x must satisfy:

\[
10^{-10} < X < 10^{10}
\]
IBM 7070 Library Program Abstracts

Square Root, Taylor Method
Rolled Royce Ltd.
P.O. Box 31
Derby, England

Purpose: This subroutine computes square root x to a controlled accuracy for a single precision fixed point argument.

Range: 0 ≤ x ≤ 1

Usage: Input: x to 10 decimal places in 9992.

(A) If maximum accuracy is required-
Calling sequence: a BLX SI, R3095
n+1 error return, x ≤ 0
n+2 normal return.

(B) If less accuracy is required, enter 0001 in LW, 32 (B, 3), where n is the number of decimal places of accuracy required, n ≤ 2.
Calling sequence: a BLX SI, R3342
n+1 error return, x ≤ 0
n+2 normal return.

Output: √x to 10 decimal places in 9992
# 0 in 9991.

Space: 32 locations, Index words S1 (2, 5), 52 (4, 9).

Method: The Trapezoid method of successive subtraction of odd numbers. This is based on the fact that:

\[ x \in (B - 1) \to x = \pm 1 \]

and the method normally used in desk machine computation.

Accuracy: When used with maximum accuracy, the maximum error is 2 in the 9th decimal place.

Timing: Average execution time is approx. .7±1.3 ms.
For maximum accuracy (n+2), the time is approx. 1.1 ms.

---

IBM 7070 Library Program Abstracts

7070 - Cube Root Subroutine
Rolled Royce Ltd.
P.O. Box 31
Derby, England

Purpose: This subroutine computes a\(^{\frac{1}{3}}\) root x for a single precision fixed point argument.

Range: 0 ≤ x ≤ 1

Usage: Input: x to 10 decimal places in 9992.

Output: ∛x to 10 decimal places in 9992.

Space: 32 locations excluding index words 31 (2, 5), 52 (4, 9).

Method: Newton’s iteration process:

\[ f(n+1) = f(n) - \frac{x}{f(n)} \]

with \( f(0) = x \)

Accuracy: As accurate as oscillations allow. With reasonable combinations of n and m, a small and a near normal, maximum error is about 5 in the 10th decimal place.

Timing: Average execution time is approx. 3.8 ± (1.2±1.3) ms
where m is the number of iterations.

---

IBM 7070 Library Program Abstracts

Subroutine for IBM 7070
Rolled Royce Ltd.
P.O. Box 31
Derby, England

Purpose: This subroutine computes square root x for a single precision fixed point argument.

Range: 0 ≤ x ≤ 1

Usage: Input: x to 10 decimal places in 9992.

Output: √x to 10 decimal places in 9992.

Space: 32 locations, Index words 31 (2, 5), 52 (4, 9).

Method: A prediction of √x correct to .001 is given by:

\[ x \in 10\% \]

\[ x \in 10\% \]

\[ x \in 10\% \]

Accuracy: Maximum error is 2 in the 10th decimal place.

Timing: Average execution time is approx. 12.7 ms.

---

IBM 7070 Library Program Abstracts

7070 - Cube Root Subroutine
R. Culp
AC Spark Plug Div GMC
Milwaukee, Wisconsin

Purpose: To compute the cube root of a real number in floating point form.

Machine Requirements: Floating hardware, 60 core storage words.

General Description: Bailey iteration

\[ x_{n+1} = \frac{2}{3} \left( \frac{x_n + 2}{3} \right) \]

d. Capabilities and Limitations: Input must be in normalized floating point form.

---

Double Precision Square Root Subroutine
A. Dickerman
AC Spark Plug Div GMC
Milwaukee, Wisconsin

Purpose: To extract the square root of a 16 digit floating point number.

Machine Requirements: Floating hardware, 171 core locations.

General Description: Iterates:

\[ \sqrt{x} \approx 3.1623 \]

Accuracy: Input must be in normalized floating point, Maximum error is 1 in the 16th place.

---
IBM 7070 Library Program Abstracts

Square Root Subroutine

M. Roberts
AC Spark Plug Div GMC
Milwaukee, Wisconsin

a. Purpose: To find square root of argument A
b. Machine Requirements: Floating hardware, 45 words storage
c. General Description: Iterate:

\[ \sqrt{2 + \frac{a}{T^2 + A}} \]

where initial approximation is

\[ \frac{A}{T} \]

d. Capabilities and Limitations: Input must be normalized floating point. Maximum error is 1 in eighth place.

Contributed By:

IBM 7070

Purpose: Divide a 16 digit floating point number by a 16 digit floating point number to obtain a 16 digit floating point quotient.

a. Purpose: To find square root of argument A
b. Machine Requirements: Floating hardware, 35 core storage words

c. General Description: The “odd-integer method” is used. Accuracy: Eight digits. Average execution time: 11 milliseconds. This space-saving but fairly time-consuming routine is included in the Arcsine-Arc tangent Subroutine by the same author.

d. Capabilities and Limitations: Does not apply.

Available prior to January 1962

IBM 7070 Library Program Abstracts

Square Root Subroutine

Contributed By:

Author: T. Rabe, Applied Science
Organization: IBM Svenska AB
Gavlegatan 20
Stockholm 6, SWEDEN

a. Purpose: A half-precision, fixed point subroutine to compute the square root.

b. Machine Requirements: All accumulators, the compare indicators, 2 index words and 115 ordinary storage locations.

c. General Description: The square root is approximated by a polynomial of the second degree. The choice of constants is made from the roots of the second polynomial of one application of the Newtonian formula gives ten digits accuracy. The program will also accept the arguments +0, and -0.

d. Capabilities and Limitations: Negative arguments will cause a programed stop.

Contributed By:

G. J. Elliott, Applied Science
IBM Svenska AB
Gavlegatan 20
Stockholm 6, SWEDEN

a. Purpose: A full precision, fixed point subroutine to compute the square root.

b. Machine Requirements: All accumulators, the compare indicators, 2 index words and 45 ordinary storage locations.

c. General Description: The subroutine obtains a first approximation using the half-precision square root subroutine by T. Rabe. Then, one application of the Newtonian formula gives ten digits accuracy. Average execution time: 5.9 milliseconds.

d. Capabilities and Limitations: Negative arguments will cause a programed stop.

Available prior to January 1962

IBM 7070 Library Program Abstracts
a. Purpose: Add two 16-digit floating numbers
b. Machine Requirements: Floating hardware, 22 core storage words
c. General Description: The subroutine utilizes the double precision add code with logic necessary to accomplish the algebraic summation of two double precision numbers.
d. Capabilities and Limitations: Input must be in normalized floating point form. (The low order word of the double precision number must have a characteristic of eight less the high order word of that double precision number).

**IBM 7070 Library Program Abstracts**

Interpolation Subroutine
Rolls Royce Ltd.,
P.O. Box 3111
Derby, England

**Purpose:** To find an interpolated value using 2, 3 or 4 points.

**Method:** 2, 3 or 4 point Aitken.

**Entry:** x in 9992 with the same alignment as x_i's.

The number of points to be used, n, in the non-indexing portion of index word 52,

y_i's in symbolic locations $G16 + 1$ to $G16 + n$

z_i's in symbolic locations $G16 + n + 1$ to $G16 + 2n$

a BLM 51, RIVUS

* return

y will be placed in 9992 with the same alignment at the y_i's.

**Space:** 23 locations and G8H to G8H + 10 index words,

51, (5), 28, 53.

**Timing:** 0.58 milliseconds approx.

**IBM 7070 Library Program Abstracts**

**Table Interpolation**

M. Roberts
AC Spark Plug Div GMC
Milwaukee, Wisconsin

**a. Purpose:** Given x and a table of y_i and a associated dependent functions, $y_i = f(x_i)$, to interpolate to the desired order for the y_i specified in the subroutine linkage.

**b. Machine Requirements:** Floating hardware, 88 words of storage plus table area.

**c. General Description:** A search is performed with the argument x to locate the best available $K_n + 1$ coordinates. Interpolation of order $K_n$ is then performed by passing a polynomial of degree $K_n$ through $K_n + 1$ points. The Aitken form of the polynomial is used. When $x$ lies outside the range of the table, extrapolation is performed.

**d. Capabilities and Limitations:** Input must be in normalized floating form.

**IBM 7070 Library Program Abstracts**

**FLOATER,** a subroutine to convert numbers from fixed to floating decimal form.

**Contributed By:**

**Author:** B. Hyman, Applied Science

**Organisation:** IBM Svenska AB

**Address:** Stockholm 6, SWEDEN

**a. Purpose:** To make a conversion from fixed to floating decimal form.

**b. Machine Requirements:** All accumulators, the compare indicators, index word $F10$, 7 other index words, 28 ordinary storage locations and a storage area for the block to be converted.

**c. General Description:** A sequential block of fixed decimal numbers will be replaced by their corresponding floating decimal numbers. Average execution time: 0.8 milliseconds per word to be converted.

**d. Capabilities and Limitations:** Alphabetic words will not be converted, but ignored. If a fixed decimal number, greater in magnitude than 9999999999, is tried to be converted, it will be considered to be a 9999999999.

**IBM 7070 Library Program Abstracts**

**7070 POLYNOMIAL ROOT EXTRACTION (TIESK)**

**Contributed by:**

George E. Priest
Texas Instruments Technical Computations
P. O. Box 5474
Dallas 21, Texas

**a. Purpose:** This routine is designed to solve for all zeros (roots) of a polynomial in one unknown with real coefficients.

**b. Machine Requirements:**

As the source deck stands it calls for one card reader (alpha) and one magnetic tape on unit 14. This may be easily altered in the source program. The routine requires 299 storage locations when assembled plus package deck and square root subroutine.

**c. General Description:** The program employs a variation of Baur's method as the solution technique. This method is not subject to breakdown when there are multiple roots.

**d. Capabilities and Limitations:**

The routine is designed for polynomial with only real coefficients, however it solves for both real and complex roots.

**IBM 7070 Library Program Abstracts**

**Double Precision Matrix Multiplication**

**Contributed By:**

Dickerman
AC Spark Plug Div GMC
Milwaukee, Wisconsin

**a. Purpose:** To multiply two matrices with any number of rows and columns within the limitations of core storage.

**b. Machine Requirements:** Floating hardware, 97 storage words plus AC Spark Plug double precision add and multiply subroutines. The user must also reserve the area of the two matrices as well as the product matrix.

**c. General Description:** Standard matrix multiplication

**d. Capabilities and Limitations:** Input in normalized floating form. Indicators and accumulators are not saved.
IBM 7070 Library Program Abstracts

7070 MATRIX INVERSION AND SIMULTANEOUS EQUATIONS

CONTRIBUTED BY: W. W. Marks and Gordon Smith
IBM Corporation
Los Angeles, California

PURPOSE: To invert a given matrix and/or to solve a system of simultaneous linear equations.

MACHINE REQUIREMENTS: A 5K or 10K 7070 with floating point hardware.

GENERAL DESCRIPTION: An elimination method with interchange of columns to bring the largest element in the row into the diagonal.

CAPABILITIES AND LIMITATIONS: A matrix of approximately 97 x 97 can be inverted on a 10K machine and a 67 x 67 on a 5K machine. The matrix package occupies 691 locations.

IBM 7070 Library Program Abstracts

SINGLE PRECISION MATRIX INVERSION

Contributed By:

Author: H. Hynas, Applied Science
Organization: IBM Sveala AB

GENERAL DESCRIPTION: The program uses the pivotal elimination method of Jordan, and will automatically select a non-zero pivot element. The elementary operations are performed on the augmented matrix and one or two tape units.

CAPABILITIES AND LIMITATIONS: Let n be the order of the matrix and n be the number of systems of equations. The restrictions are then as follows:

a. 5,000 word machine: n < 67
b. 10,000 word machine: n < 97

IBM 7070 Library Program Abstracts

Solution of Simultaneous Linear Equations

J. Roberts
AG Spark Plug Div. GMIC
Milwaukee, Wisconsin

a. PURPOSE: To find x1, x2, ..., xn of the following equation set:

\[ a_11 x_1 + a_12 x_2 + \cdots + a_1n x_n = b_1 \]
\[ a_21 x_1 + a_22 x_2 + \cdots + a_2n x_n = b_2 \]
\[ \vdots \]
\[ a_n1 x_1 + a_n2 x_2 + \cdots + a_nn x_n = b_n \]

b. MACHINE REQUIREMENTS: Floating hardware, approximately 200 words plus the matrix area are the storage requirements.

c. GENERAL DESCRIPTION: Compute a Solution.

d. CAPABILITIES AND LIMITATIONS: Input must be in normalized floating form. Accumulators and indicators are not saved.

IBM 7070 Library Program Abstracts

7070 SLERP, SOLVE SIMULTANEOUS LINEAR EQUATIONS WITH PIVOTING

Contributed By: Robert Judson
The B. F. Goodrich Co.
Akron, Ohio

A. PURPOSE: Solve N simultaneous linear equations with one right hand column vector (one set of constant terms). Includes pivoting so that equations may be arranged in any order and may have zeros on diagonal.

B. OBJECT ROUTINE MACHINE REQUIREMENTS: Floating point hardware.

C. NOTE: Can be furnished for non-floating point hardware if desired. Working storage is (N*M) locations for N equations. Location PV must not be disturbed.

D. SOURCE LANGUAGE: Autocoder

E. SOURCE LANGUAGE ENTRY: BXL, LINK, SOLVE with equations stored sequentially by rows, a [n] in location PV0 and n [j] accumulator No. 1, right justified. Solution will be in same locations as original right hand vectors, inclusive.

IBM 7070 Library Program Abstracts

7070 STEPWISE MULTIPLE REGRESSION ANALYSIS, MR 1

Contributed By: Gary Lotto

Author: R. E. Boss

Organization: University of Pittsburgh

Solve simultaneous linear equations. To perform matrix operations as subroutines. To invert a matrix to be inverted. Average execution time is approximately 3.7, 0.8 (in millisecond), where n is the order of the matrix and m is the number of systems.

d. CAPABILITIES AND LIMITATIONS: Input must be in normalized floating form. Accumulators and indicators are not saved.
The program also provides the standard error of the estimate of the dependent variable, and a multiple regression equation. Each linear regression equation expresses a single "dependent" variable as a function of up to 39 "independent" variables. The standard error of each regression coefficient is computed.

Variables may be transformed if so desired.

The transformed observed data values are listed in the output with the type of transformation specified. The following transformations are available:

- Log X
- (X^p)
- Square Root X
- Natural log X

Any weight can be applied to any observation if so desired. If no specific weight is given, the observation is assumed to have unit weight.

**IBM 7070 Library Program Abstracts**

**IBM 7070 - Principal Axis Factor Analysis**

**Contributed By**

**Author:** Donald G. Wyman

**Organization:** IBM Corporation

**Machine Requirements:** The program is written for a 10K machine with floating point hardware. It may easily be modified to use a 5K machine, and/or no floating point hardware (by subroutine simulation) with a subsequent reduction in the maximum number of variables that may be handled and with a possible reduction in the speed of a part of the program. The amount of storage used is a function of the number of variables involved. Input is on cards. Output is on the printer or on cards.

**General Description:** Cumulation of sums, sums of squares, and sums of cross products proceeds in fixed point arithmetic at a speed relative to the number of variables specified, and to the number of digits in the average observation of input data. For 4 digits, 130 variables are processed at approximately 7 1/2 seconds per case. The time is approximately proportional to V^2 (V the number of variables), and about 10 per cent is saved per digit fewer than 4.

The transfer routine occurs once per run, and is approximately 1 1/2 minutes for 130 variables.

The program occurs at maximum print speed, and prints 23 columns of the matrix at a time. The columns of means and standard deviations is also printed. All output is to 3 decimal places.

**Capabilities and Limitations:** The program will handle up to 130 variables (approx. 85 variables on a 5K machine) with the restriction that the maximum sum of squares (treating the data as whole numbers) must be less than 10^{12}. The matrix is in storage for further analysis, if desired.

**IBM 7070 Library Program Abstracts**

**IBM 7070 - Stepwise Multiple Linear Regression Analysis on the IBM 7070**

**Contributed By**

**Author:** Donald G. Wyman

**Organization:** IBM Corporation

**Machine Requirements:** (Include machine components special features

5000 words of storage, 3 tapes and card reader or 4 tapes. (less tape if residuals are not calculated).

**General Description:** (Mathematical method, accuracy, speed, if appropriate)


**Capabilities and Limitations:** The program has been written as two independent phases. Phase 1 reads and transforms input and forms simple correlations for up to 72 variables. Phase 2 solves for the coefficients, either directly or iteratively, for any set of equations formed as a subset of the 72 variables to a maximum of 55 independent and one dependent.
IBM 7070 Library Program Abstracts

Multiple Correlation and Regression Analysis by the Stepwise Method. 1
Contributed By:
Outline:

a. Purpose: To rotate the factor loadings of the original variables through an iterative process until the number of statistically significant factors is reduced to the quantity specified by the user. The program provides means, standard deviations, and simple correlation coefficients for all variables.

b. Equipment Specifications: (a) 5,000 or 10,000 word 7070 (b) On-line card reader (c) Minimum of three tapes

c. Source Language: FORTRAN

d. Timing: (n/2) additions and multiplications and (n/2)! divisions

e. Accuracy: Single precision floating point.

IBM 7070 Library Program Abstracts

7070 - Normalized Various Factor Rotation
Contributed By:

a. Purpose: To rotate the factor loadings of Y variables on F factors to orthogonal simple structure.

b. Range: 10^6 1.0 to 16.000


d. General Description: Pairs of factors are rotated by an iterative process until all pairs are stabilized within a tolerance value. The normalized various criterion value, the rotated factor loadings, and the transformation matrix is the output.

IBM 7070 Library Program Abstracts

Random Numbers and Random Normal Deviates Generator

Contributed By:

A. Doherty

AC Spark Plug Div. GMC
Milwaukee, Wisconsin

(Continued on next column)
7070 - Transportation Problem (Dennis Technique)

Robert Judson
The B. F. Goodrich Company
Deps. 6973 + Bidg. 34-C
Akron 18, Ohio

a. **Purpose:** To solve fairly large transportation problems in reasonably short times using magnetic tape to store Supply, Demand and Cost Data. Also to permit suppression of any desired shipping path, even to the extent of suppressing an entire row (which essentially becomes an artificial vector).

b. **Machine Requirements:** 3 tape units and 5K memory. To solve any problem between 50 x 50 and 275 x 275. Program will be furnished in Symbolic Assembler form so that it can be readily modified for a 10K or larger memory.

c. **Timing:** 115 x 12. Approx. 70 seconds with 1/2 costs excluded
   12 x 115 Approx. 90

b. **General Description:** Reference: Jack B. Dennis "A High Speed Computer Technique for the Transportation Problem"

   Program is in two parts. Cost tape to Matrix Tape (BFC No. 79102) and Main Program (BFC No. 79101) so as to facilitate adaptation by users with card oriented equipment.

---

7070 Management Decision-Making Exercises

Contributed By:

**Author:** John A. Tilot
**Organization:** IBM Corporation

**H. James Farver**
**Peoria, Illinois**

a. **Purpose:** Using the 7070, the operation of five firms manufacturing similar low profit products in a highly competitive industry is simulated. Management "teams" are given an opportunity to make decisions and to see the results of these decisions almost immediately.

b. **Machine Requirements:**
   1 7500 Card Reader
   1-4 720-B or 1440-Tapes (Channel 1 only)
   10 K Storage
   Peripheral printer (720 or 1440)

c. **General Description:** The exercise has been modeled after the business strategy game constructed by Richard Bellman, Franco Ricciardi, and others for the American Management Association in 1957. While the general form of this exercise resembles the AMA game, there are a number of innovations which have been introduced to add realism and difficulties to the strategy problems encountered.

   The basic decision problem involved in the exercise is that of deciding on courses of action with only a vague knowledge of the outcome of such actions. The results of decisions made by each management team depend not only on their own decisions, but also on the decisions made by the competitive teams.

   The result is a realistic simulation of every-day business operation with the flavor and incentive necessary for an interesting "Management Decision" exercise.

d. **Capabilities and Limitations:** Not applicable.
IBM 7090 PROGRAM LIBRARY ABSTRACT

1094HSY53
AVAILABLE PRIOR TO JANUARY 1962

ENTHALPY AND ENTROPY OF SATURATED LIQUID
COMPUTES ENTHALPY AND ENTROPY OF SATURATED LIQUID AS
FUNCTIONS OF PRESSURE AND TEMPERATURE

1095MMHSL
AVAILABLE PRIOR TO JANUARY 1962

ENTROPY OF SATURATED LIQUID
COMPUTES ENTROPY OF SATURATED LIQUID AS FUNCTION OF
TEMPERATURE

1095MMHSS
AVAILABLE PRIOR TO JANUARY 1962

ENTROPY SPECIFIC VOLUME OF SUPERHEATED STEAM
COMPUTES ENTROPY, SPECIFIC VOLUME, ENTROPY, AND SPECIFIC
VOLUME OF SATURATED STEAM AS FUNCTIONS OF PRESSURE AND
TEMPERATURE

1095MHWSV
AVAILABLE PRIOR TO JANUARY 1962

ENTROPY SPECIFIC VOLUME OF SATURATED VAPOR
COMPUTES ENTROPY AND SPECIFIC VOLUME, SPECIFIC VOLUME
AND TEMPERATURE OF SATURATED VAPOR AS FUNCTIONS OF
PRESSURE

1095WHSDL
AVAILABLE PRIOR TO JANUARY 1962

SPECIFIC VOLUME OF SATURATED LIQUID
COMPUTES SPECIFIC VOLUME OF SATURATED LIQUID AS FUNCTION
OF TEMPERATURE

IBM 7090 PROGRAM LIBRARY ABSTRACT

1095WHSDL
AVAILABLE PRIOR TO JANUARY 1962

COMPUTES ENTROPY OF SATURATED LIQUID
COMPUTES SPECIFIC VOLUME OF SATURATED LIQUID AS FUNCTION
OF PRESSURE AND TEMPERATURE

1095WHSDL
AVAILABLE PRIOR TO JANUARY 1962

SPECIFIC VOLUME OF SATURATED LIQUID
COMPUTES SPECIFIC VOLUME OF SATURATED LIQUID AS FUNCTION
OF TEMPERATURE

1095WHSDL
AVAILABLE PRIOR TO JANUARY 1962

SPECIFIC VOLUME OF SATURATED VAPOR
COMPUTES SPECIFIC VOLUME OF SATURATED VAPOR AS FUNCTION
OF TEMPERATURE

1095WHSDL
AVAILABLE PRIOR TO JANUARY 1962

SPECIFIC VOLUME OF SATURATED VAPOR
COMPUTES SPECIFIC VOLUME OF SATURATED VAPOR AS FUNCTION
OF TEMPERATURE

IBM 7090 PROGRAM LIBRARY ABSTRACT

1095WHSDL
AVAILABLE PRIOR TO JANUARY 1962

SPECIFIC VOLUME OF SATURATED VAPOR
COMPUTES SPECIFIC VOLUME OF SATURATED VAPOR AS FUNCTION
OF TEMPERATURE

1095WHSDL
AVAILABLE PRIOR TO JANUARY 1962

SPECIFIC VOLUME OF SATURATED VAPOR
COMPUTES SPECIFIC VOLUME OF SATURATED VAPOR AS FUNCTION
OF TEMPERATURE

1095WHSDL
AVAILABLE PRIOR TO JANUARY 1962

SPECIFIC VOLUME OF SATURATED VAPOR
COMPUTES SPECIFIC VOLUME OF SATURATED VAPOR AS FUNCTION
OF TEMPERATURE

1095WHSDL
AVAILABLE PRIOR TO JANUARY 1962

SPECIFIC VOLUME OF SATURATED VAPOR
COMPUTES SPECIFIC VOLUME OF SATURATED VAPOR AS FUNCTION
OF TEMPERATURE

1095WHSDL
AVAILABLE PRIOR TO JANUARY 1962

SPECIFIC VOLUME OF SATURATED VAPOR
COMPUTES SPECIFIC VOLUME OF SATURATED VAPOR AS FUNCTION
OF TEMPERATURE

1095WHSDL
AVAILABLE PRIOR TO JANUARY 1962

SPECIFIC VOLUME OF SATURATED VAPOR
COMPUTES SPECIFIC VOLUME OF SATURATED VAPOR AS FUNCTION
OF TEMPERATURE

1095WHSDL
AVAILABLE PRIOR TO JANUARY 1962

SPECIFIC VOLUME OF SATURATED VAPOR
COMPUTES SPECIFIC VOLUME OF SATURATED VAPOR AS FUNCTION
OF TEMPERATURE

1095WHSDL
AVAILABLE PRIOR TO JANUARY 1962

SPECIFIC VOLUME OF SATURATED VAPOR
COMPUTES SPECIFIC VOLUME OF SATURATED VAPOR AS FUNCTION
OF TEMPERATURE

1095WHSDL
AVAILABLE PRIOR TO JANUARY 1962

SPECIFIC VOLUME OF SATURATED VAPOR
COMPUTES SPECIFIC VOLUME OF SATURATED VAPOR AS FUNCTION
OF TEMPERATURE

1095WHSDL
AVAILABLE PRIOR TO JANUARY 1962

SPECIFIC VOLUME OF SATURATED VAPOR
COMPUTES SPECIFIC VOLUME OF SATURATED VAPOR AS FUNCTION
OF TEMPERATURE

1095WHSDL
AVAILABLE PRIOR TO JANUARY 1962

SPECIFIC VOLUME OF SATURATED VAPOR
COMPUTES SPECIFIC VOLUME OF SATURATED VAPOR AS FUNCTION
OF TEMPERATURE

1095WHSDL
AVAILABLE PRIOR TO JANUARY 1962

SPECIFIC VOLUME OF SATURATED VAPOR
COMPUTES SPECIFIC VOLUME OF SATURATED VAPOR AS FUNCTION
OF TEMPERATURE

1095WHSDL
AVAILABLE PRIOR TO JANUARY 1962

SPECIFIC VOLUME OF SATURATED VAPOR
COMPUTES SPECIFIC VOLUME OF SATURATED VAPOR AS FUNCTION
OF TEMPERATURE

1095WHSDL
AVAILABLE PRIOR TO JANUARY 1962

SPECIFIC VOLUME OF SATURATED VAPOR
COMPUTES SPECIFIC VOLUME OF SATURATED VAPOR AS FUNCTION
OF TEMPERATURE

1095WHSDL
AVAILABLE PRIOR TO JANUARY 1962

SPECIFIC VOLUME OF SATURATED VAPOR
COMPUTES SPECIFIC VOLUME OF SATURATED VAPOR AS FUNCTION
OF TEMPERATURE
IBM 2050 PROGRAM LIBRARY ABSTRACT

**T070** 1151540124 AVAILABLE PRIOR TO JANUARY 1962
ADMIT 13 OR 99 DIFFERENTIAL EQUATIONS TO A MICROFORUM OF N SIMULTANEOUS FIRST ORDER EQUATIONS. SUBROUTINE GIVES CONVERGENCE AND NUMBER OF ITTERATIONS REQUIRED. 379 CALLS.

**T070** 11525402 AVAILABLE PRIOR TO JANUARY 1962
GENERALIZED INTEGRATION SUBROUTINE A SET OF SIMULTANEOUS ORDINARY DIFFERENTIAL EQUATIONS IS SOLVED USING EITHER HUNGE-Kutta OR ONE OF SEVERAL SETS OF ROCKS SCHEME DESCRIPTORS. EQUATIONS ARE STARTED WITH HUNG-Kutta POINTS. A VARIABLE INTEGRATION INTERVAL WITH ERROR CONTROL CAN BE USED OPTIMALLY WITH PRECITION-CORRECTOR FORMULAS USES 473 LOCATIONS.

**T070** 115984W5P AVAILABLE PRIOR TO JANUARY 1962
DECIMAL, OCTAL, BCD LOADER ALLOWS SELECTIVE INPUT WITH A SINGLE CALL STATEMENT, AND ALLOWS FOR CHANGES IN VALUES WHICH WERE NOT ORIGINAL, DESIGNATED AS INPUT. REQUIRES 670 WORDS OF STORAGE WITH ALL TEMPORARIES SELF-CONTAINED.

**T070** 11645480 AVAILABLE PRIOR TO JANUARY 1962
TIME SERIES DECOMPOSITION AND ADJUSTMENT PORTRAIT PROGRAM TO ADJUST SEASONAL AND IRRGULAR TIME SERIES TO A FORM THAT SHOWS PRIMARILY THE TREND-CYCLICAL MOVEMENTS. SEASONAL FACTORS, IRRGULAR FLUCTUATIONS AND MANY SUMMARY MEASURES USED IN TIME SERIES ANALYSIS ARE COMPUTED IN THE PROCESS. BASICALLY ADAPTED FROM TENNESSEE VALLEY AUTHORITY PROGRAMS WRITTEN TO BE K. 5% EFFICIENT. PROGRAM ALSO PERMITS FITS ADJUSING FOR DELIVERY DAYS AND Fitting SQHTEST SQUARES TREND LINE AS FORECASTING AID. CORrl. 1159

**T070** 116644MPLOT AVAILABLE PRIOR TO JANUARY 1962
GENERALIZED PLOT ROUTINE THIS ROUTINE IS USED TO GENERATE AND LABEL GRAPHS FOR THE SC 4020 MICROFILM RECOROER. CARDS ARE WRITTEN ON TAPE. THE ROUTINE WILL PERFORM THE SCALING REQUIRED AND PLOT SETS OF POINTS WHERE COORDINATES ARE GIVEN IN FLOATING-POINT FORM. GRID LINES MAY BE SPECIFIED TOGETHER WITH A FORMAT TO CONTROL THEIR LABELING. IT IS POSSIBLE TO PRINT BOTH THE HORIZONTAL AND VERTICAL TITLES. USES 1800 STORAGE.

IBM 2050 PROGRAM LIBRARY ABSTRACT

**T070** 1169240123 AVAILABLE PRIOR TO JANUARY 1962
LARGE DOUBLE PRECISION SIMULTANEOUS EQUATION SOLVER AND DETERMINANT EVALUATION, DISCUSSION AND ELIMINATION USED TO SOLVE THE SIM. G-DIAGONAL INPUT AND OUTPUT ARE SINGLE PRECISION. SUBROUTINE GIVES TIME BTN ENTRIES. CORrl. 1159

**T070** 11595048 AVAILABLE PRIOR TO JANUARY 1962
TAYLOR SERIES RATIONAL FUNCTION CURVE FITTING FINDS THE COEFFICIENT OF A RATIONAL FUNCTION BY THE TAYLOR SERIES METHOD. CORrl. 1121

**T070** 11550045S AVAILABLE PRIOR TO JANUARY 1962
CRITICAL PATH AND RESOURCE SUMMARY CALCULATION, CALCULATES CRITICAL PATH PARAMETERS FOR EACH JOB AND THE SUM OF EACH RESOURCE IN USE AT ANY TIME, DURING THE SPAN OF A GIVEN PROJECT OF N JOBS. N TAPES REQUIRED.

**T070** 11569945C AVAILABLE PRIOR TO JANUARY 1962
ROOT TRACING ENABES ONE TO LOCATE THE Zeros OF NON-LINEAR FUNCTIONS, THE LOCUS OF COMPLEX ROOTS OF A CHARACTERISTIC EQUATION WITH A REAL PARAMETER, AND TO FIND THE LOCS OF AN N-DIMENSIONAL VECTOR, USING SUBROUTINES DIF AND DGE.

**T070** 11794045D AVAILABLE PRIOR TO JANUARY 1962
UNLOAD ALL TAPE ONE-CARD SELF-LOADING PROGRAM CONTAINS WHICH TAPE UNITS ARE IN READY STATUS, THEN ISSUES HEADING-AND-UPLOAD INSTRUCTIONS FOR THOSE TAPE UNITS.

**T070** 11774046G AVAILABLE PRIOR TO JANUARY 1962
NORMALIZED INCOMPLETE GAMMA FUNCTION WITH POISSON TERM GAVE AND A POSITIVE-REAL ON ZERO. SUMMATION SUBROUTINE WILL COMPUTE THE NORMALIZED INCOMPLETE GAMMA FUNCTION.

**T070** 11774045G AVAILABLE PRIOR TO JANUARY 1962
NORMALIZED INCOMPLETE GAMMA FUNCTION WITH POISSON TERM GAVE AND A POSITIVE-REAL ON ZERO. SUMMATION SUBROUTINE WILL COMPUTE THE NORMALIZED INCOMPLETE GAMMA FUNCTION.

IBM 2050 PROGRAM LIBRARY ABSTRACT

**T070** 11945046P AVAILABLE PRIOR TO JANUARY 1962
STEPWISE MULT. RECOSSION WITH VARIABLE TRANSFORMATIONS TRANSFORMS RAN INPUT DATA AND PERFORMS A STEPWISE MULTIPLE REGRESSION UPON THE TRANSFORMED DATA. THE TRANSFORMED DATA CONSISTS OF N SETS CONTAINING N INDEPENDENT VARIABLES AND ONE DEPENDENT VARIABLE. EACH SET IS TREATED AS A SEPARATE RUN. THE TRANSFORMED DATA IS THEN ANALYZED WITH EACH SET OF DATA, A SUBSET OF REGRESSION COEFFICIENTS FOR K VARIABLES, A LESS THAN OR EQUAL TO K, WILL BE OBTAINED WHICH ARE SIGNIFICANT AT A GIVEN LEVEL OF SIGNIFICANCE. SIMILAR TO OR PMCL, DIST. 36, ALSO ALLOWS MAP OF 130 REGRESSION VARIABLES, REQUIRES 52 CORE AND 3 TAPES.

**T070** 1191600P AVAILABLE PRIOR TO JANUARY 1962
LINCOLN IPLV INTERPRETIVE SYSTEM - 709, 7040 TO EXECUTE PROGRAMS WRITTEN IN IPLV AS DESCRIBED IN HAND. THE PROGRAM SYSTEM IS BASED ON AN ASSEMBLY SYSTEM THAT CONTAINS AN ASSEMBLER, INTERPRETER, FRAMES, AND A DATA LONG DESCRIPTION OF HOW TO RUN 스스로, TAPE READING PROGRAMS, AND SPECIFICATIONS. IT CAN BE USED SPECIFICALLY ON THE 7040, ASSEMBLY OF SAP DECK PRODUCES SYMBOL TABLE, MINIMAL DGA, 2 WHITE TAPE CARDS, CALL A FIX RESUME, TO START CARD, BINARY DECK MUST FOLLOW UPPER BINARY DETAIL LOADER. 1396 11225

IBM 2050 PROGRAM LIBRARY ABSTRACT

**T070** 11919640M AVAILABLE PRIOR TO JANUARY 1962
BOOLEAN ALGEBRA MINIMIZER FINDS THE TWO-LEVEL MAXIMUM SUM OF PRODUCTS OR PRODUCT OF SUMS FORM FOR SETS OF SIMULTANEOUS BOOLEAN EQUATIONS, HAS THE ABILITY TO MINIMIZE UP TO 36 SIMULTANEOUS BOOLEAN EQUATIONS, EACH OF WHICH CONTAINS UP TO 30 INDEPENDENT VARIABLES.

**T070** 1199400E FIELD AVAILABLE PRIOR TO JANUARY 1962
TO ASSIGN TAPE UNIT USAGE OTHER THAN THAT WHICH IS STANDARD IN 18 105

**T070** 1200404M AVAILABLE PRIOR TO JANUARY 1962
N DIMENSIONAL TABLE LOOK UP GIVEN THE ARGUMENTS X/1, X/2, , X/N UNIT USING A LOOK-UP INTERPOLATION FROM A TABLE OF XS. IF DESIRED, THIS PROGRAM WILL ALSO EXTRAPOLATE ON THE UPPER AND LOWER LIMIT.

**T070** 12050044D AVAILABLE PRIOR TO JANUARY 1962
ORDINARY DIFF. EQUATION SOLUTION FRANCE-KUTTA TO INTEGRATE STEEPWAL. SET OF N SIMULTANEOUS FIRST ORDER DIFFERENTIAL EQUATIONS USING GILL'S VARIATION OF THE RUNGE-KUTTA METHOD.

**T070** 12050040G AVAILABLE PRIOR TO JANUARY 1762

307
DIFFERENTIATION OF THE DEGREE DIFFERENTIATION, SUM OF TERMS CONSISTING OF OUTPUT IS

IBM 7090 PROGRAM LIBRARY ABSTRACT

7090 123992F3P AVAILABLE PRIOR TO JANUARY 1962
BELL LABS PERMUTATION INDEX PROGRAM
PRODUCES FROM INPUT BIBLIOGRAPHIC DATA A FOUR-PART DOCUMENT INDEX. THE PRINCIPAL PART IS A PERMUTED TITLE INDEX WITH A 300-CHARACTER LINE, ALSO OUTPUT ON THE SAME TAPE AS THE PERMUTED INDEX IS A COMPLETE BIBLIOGRAPHY OF THE INPUT DATA. THE OTHER TWO INDEXES ARE OUTPUT AS A MIXED CARD FILE OF /F AND /J PROJECT NUMBERS, EXCEPT FOR THE RE SYS INPUT, OUTPUT AND TAP CONTROL ROUTINES, THIS IS AN INDEPENDENT PROGRAM.

7090 124144D01 AVAILABLE PRIOR TO JANUARY 1962
MODERN CURVE SMOOTHING ROUTINE
THIS CURVE SMOOTHING ROUTINE USES A METHOD OF AVERAGING THREE PARABOLAS. FOR EACH SMOOTHED POINT, THE THREE CLOSEST GIVEN POINTS ARE OBTAINED. EACH PARABOLA THEN IS CONSTRUCTED THROUGH THREE OF THESE POINTS.

IBM 7090 PROGRAM LIBRARY ABSTRACT

7090 123992M3J AVAILABLE PRIOR TO JANUARY 1962
CRISTALLOGRAPHIC PROGRAM
THIS USES THE DIAGNOSIS OF THE REGRSSION MATRIX ONLY, IT IS BASED ON NUR, WHICH IS USED ON THE 701. THE PROGRAM ALLOWS SPACE FOR ABOUT 120 ATOMS IN TWO ASYMMETRIC UNIT AND AN UNLIMITED NUMBER OF REFLECTIONS, IT IS SUITABLE FOR USE WITH ANY OF THE 230 SPACEGROUPS, AND HANKELS X-RAY AS WELL AS NEUTRON DIFFRACTION DATA. IT IS INTENDED FOR USE WITH THE FORTRAN monitor.

IBM 7090 PROGRAM LIBRARY ABSTRACT

7090 124144D01 AVAILABLE PRIOR TO JANUARY 1962
MODERN CURVE SMOOTHING ROUTINE
THIS CURVE SMOOTHING ROUTINE USES A METHOD OF AVERAGING THREE PARABOLAS. FOR EACH SMOOTHED POINT, THE THREE CLOSEST GIVEN POINTS ARE OBTAINED. EACH PARABOLA THEN IS CONSTRUCTED THROUGH THREE OF THESE POINTS.

IBM 7090 PROGRAM LIBRARY ABSTRACT

7090 123992M3J AVAILABLE PRIOR TO JANUARY 1962
CRISTALLOGRAPHIC PROGRAM
THIS USES THE DIAGNOSIS OF THE REGRSSION MATRIX ONLY, IT IS BASED ON NUR, WHICH IS USED ON THE 701. THE PROGRAM ALLOWS SPACE FOR ABOUT 120 ATOMS IN TWO ASYMMETRIC UNIT AND AN UNLIMITED NUMBER OF REFLECTIONS, IT IS SUITABLE FOR USE WITH ANY OF THE 230 SPACEGROUPS, AND HANKELS X-RAY AS WELL AS NEUTRON DIFFRACTION DATA. IT IS INTENDED FOR USE WITH THE FORTRAN monitor.

IBM 7090 PROGRAM LIBRARY ABSTRACT

7090 123992M3J AVAILABLE PRIOR TO JANUARY 1962
CRISTALLOGRAPHIC PROGRAM
THIS USES THE DIAGNOSIS OF THE REGRSSION MATRIX ONLY, IT IS BASED ON NUR, WHICH IS USED ON THE 701. THE PROGRAM ALLOWS SPACE FOR ABOUT 120 ATOMS IN TWO ASYMMETRIC UNIT AND AN UNLIMITED NUMBER OF REFLECTIONS, IT IS SUITABLE FOR USE WITH ANY OF THE 230 SPACEGROUPS, AND HANKELS X-RAY AS WELL AS NEUTRON DIFFRACTION DATA. IT IS INTENDED FOR USE WITH THE FORTRAN monitor.
### AETRA

<table>
<thead>
<tr>
<th>Code Originated by:</th>
<th>Atomics International</th>
</tr>
</thead>
<tbody>
<tr>
<td>Computer:</td>
<td>7090 (FORTRAN)</td>
</tr>
<tr>
<td>Description of Code:</td>
<td>(Indicated status, if known)</td>
</tr>
<tr>
<td>Approximate Performance:</td>
<td>About 2 seconds per cycle, each cycle divided into three parts.</td>
</tr>
</tbody>
</table>

**MATERIAL AVAILABLE:**

- Computer: (Language) FORTRAN
- Limitations:

**REFERENCES:**


### AIMFIRE

<table>
<thead>
<tr>
<th>Code Originated by:</th>
<th>Atomics International</th>
</tr>
</thead>
<tbody>
<tr>
<td>Computer:</td>
<td>7090 (FORTRAN)</td>
</tr>
<tr>
<td>Description of Code:</td>
<td>A computer program designed to solve reactor kinetics equations with respect to time. The mathematical method used is that developed by E. R. Cohen.</td>
</tr>
<tr>
<td>Approximate Performance:</td>
<td>About 2 seconds per cycle, each cycle divided into three parts.</td>
</tr>
</tbody>
</table>

**MATERIAL AVAILABLE:**

- Computer: (Language) FORTRAN
- Limitations:

**REFERENCES:**


### AIM-6

<table>
<thead>
<tr>
<th>Code Originated by:</th>
<th>Atomics International</th>
</tr>
</thead>
<tbody>
<tr>
<td>Computer:</td>
<td>7090 (FORTRAN, FAC)</td>
</tr>
<tr>
<td>Description of Code:</td>
<td>AIM-6 is a one-dimensional diffusion theory code with options similar to those of FOG, except for the buckling iteration program. A library of microscopic cross section data is utilized to form the macroscopic cross sections. In addition to the searches available to FOG, a concentration search on one or two elements is permitted. An extensive data edit is available.</td>
</tr>
<tr>
<td>Restrictions or Limitations:</td>
<td>A maximum of 101 spaces or more than 18 energy groups. Only downscattering is permitted, but can be from a given group to any lower group.</td>
</tr>
<tr>
<td>Approximate Performance:</td>
<td>For a 16-group, 100 mesh point problem, 3 minutes would be a typical time for a single problem, although times may be as low as 30 seconds.</td>
</tr>
</tbody>
</table>

**MATERIAL AVAILABLE:**

- Computer: (Language) FORTRAN
- Limitations:

**REFERENCES:**


### AIREK-II

<table>
<thead>
<tr>
<th>Code Originated by:</th>
<th>Atomics International</th>
</tr>
</thead>
<tbody>
<tr>
<td>Computer:</td>
<td>7090 (FORTRAN)</td>
</tr>
<tr>
<td>Description of Code:</td>
<td>The AIREK code is designed to solve the reactor kinetics equations with respect to time. The mathematical method used is that developed by E. R. Cohen.</td>
</tr>
<tr>
<td>Restrictions or Limitations:</td>
<td>The maximum number of differential equations that can be solved simultaneously is 50. Within this limitation, there may be 1 delayed neutron groups, a 9° &lt; 25, and other linear feedback equations, a 9° &lt; 49.</td>
</tr>
</tbody>
</table>

**MATERIAL AVAILABLE:**

- Computer: (Language) FORTRAN
- Limitations:

**REFERENCES:**


### CLOUD

<table>
<thead>
<tr>
<th>Code Originated by:</th>
<th>Atomics International</th>
</tr>
</thead>
<tbody>
<tr>
<td>Computer:</td>
<td>7090 (FORTRAN)</td>
</tr>
<tr>
<td>Description of Code:</td>
<td>The CLOUD code calculates the external gamma-ray dose rate and total integrated dose resulting from the continuous release of radioactive materials to the atmosphere. Meteorological parameters such as wind velocity, lateral and vertical diffusion parameters, stability parameters, and the presence of physical boundaries such as a ground surface and a temperature inversion layer, are considered. Decay of the source material is described either by the use of a simple parent-daughter decay scheme or by a Way-Wigner type relationship.</td>
</tr>
<tr>
<td>Restrictions or Limitations:</td>
<td>A 32K memory is required.</td>
</tr>
</tbody>
</table>

**MATERIAL AVAILABLE:**

- Computer: (Language) FORTRAN
- Limitations:

**REFERENCES:**


### EQUIPOISE

<table>
<thead>
<tr>
<th>Code Originated by:</th>
<th>Atomics International</th>
</tr>
</thead>
<tbody>
<tr>
<td>Computer:</td>
<td>7090 (FORTRAN)</td>
</tr>
<tr>
<td>Description of Code:</td>
<td>EQUIPOISE is a two-dimensional, two-group, neutron diffusion code for the IBM 7090 Computer.</td>
</tr>
<tr>
<td>Restrictions or Limitations:</td>
<td>EQUIPOISE - 2 is an IBM-7090 FORTRAN programmed code for the solution of two-group, two-dimensional, neutron diffusion equations. A maximum of 2100 mesh points may be used, and the code will solve problems in either rectangular or cylindrical geometry. Logarithmic derivative boundary conditions are allowed, and removal of neutrons from both groups is permitted.</td>
</tr>
<tr>
<td></td>
<td>2. Computer: IBM 7090</td>
</tr>
<tr>
<td></td>
<td>3. ABSTRACT: EQUIPOISE - 3 is an IBM-7090 FORTRAN programmed code for the solution of two-group, two-dimensional, neutron diffusion equations. A maximum of 2100 mesh points may be used, and the code will solve problems in either rectangular or cylindrical geometry. Logarithmic derivative boundary conditions are allowed, and removal of neutrons from both groups is permitted.</td>
</tr>
</tbody>
</table>

**Nuclear Code:**

- Code: 7090 Nuclear Code
FOG

7090 Nuclear Code

(1) Code Originated by:
Atoms International

(2) Computer:
7090 (FORTRAN)

(3) Description of Code:
The FOG codes are one-dimensional neutron diffusion theory codes. The difference equations used are designed to conserve neutrons in cylindrical and spherical geometries. The principal options available include calculation of the adjoint flux, four different criticality searches, and choice of one of nine possible sets of boundary conditions (including energy-dependent extrapolation lengths). In addition, an automatic calculation of extrapolation parameters is permitted, and there is available a heatup iteration program for a fully-reflected, right circular cylinder.

(4) Restrictions or Limitations:
Only macroscopic input data is permitted. From one to four energy groups are permitted, and up to 25 mesh points and 40 regions. Scattering is permitted only in the next lower group.

(5) Approximate Performance:
Varies widely, but execution time generally may be expected to be less than 30 seconds.

(6) Reference:

(7) Material Available:
1. NAA-SR-0114
2. FORTRAN source deck.

FORM

7090 Nuclear Code

(1) Code Originated by:
Atoms International

(2) Computer:
7090 (FORTRAN)

(3) Description of Code:
The FORM, or FORTRAN-MUFT, code is a fourier transform solution of a code quite similar to the MUFT-4 code, but containing some additional options, including the option of changing cross sections in the 54 group library at execution time. Library editing routines are included as auxiliary codes.

(4) Restrictions:
A 32K memory and 2 tape units are required.

(5) Approximate Performance:
About 3-4 seconds.

(6) Reference:

(7) Material Available:
1. NAA-SR-MEMO 5764
2. FORTRAN source deck.

Note: The information given above was abstracted from NAA-SR-MEMO 5764.

FORTRAN SNGL

7090 Nuclear Code

(1) Code Originated by:
Atoms International

(2) Computer:
7090 (FORTRAN)

(3) Description of Code:
This code is a revision of an earlier code written by Argonne National Laboratory (Ref. 468/MEMO107 by J. E. Dees). The principal changes that were made were to eliminate use of drum and any on-line printing, as well as to increase the size of the dimension statements. In addition, in the regular flux calculations in plane, spherical, and cylindrical geometry, various criticality searches are permitted.

(4) Restrictions or Limitations:
A 32K memory is required. Up to 100 space intervals and 20 energy groups may be used.

(5) Approximate Performance:
About 5-6 seconds.

(6) Reference:

(7) Material Available:
1. NAA Program Description.
2. FORTRAN source deck.

GAM-J

7090 Nuclear Code

(1) Code Originated by:
General Electric Company

(2) Computer:
7090 (FORTRAN)

(3) Description of Code:
The GAM-J is an extension of the GAM code for the calculation of fast neutron spectra and multigroup constants.

(4) Restrictions or Limitations:
No memory is required. Each gamma-ray group must be calculated individually.

(5) Approximate Performance:
About 1 minute.

(6) Reference:

CRACK-I

7090 Nuclear Code

(1) Code Originated by:
Atoms International

(2) Computer:
7090 (FORTRAN)

(3) Description of Code:
CRACK-I is a multigroup, multiregion, gamma-ray attenuation code designed primarily for computing gamma-ray heating and gamma-ray dose rates in multiregion finite or semi-infinite slab shields. A different buildup factor may be specified for each source region.

(4) Restrictions or Limitations:
If a 704 is used, at least an 8K memory is required. As many as 30 regions, 10 mesh points per region, 20 gamma-ray energy groups, 10 shield materials, and 5 material buildup factors may be included in a single calculation.
**7090 Nuclear Code**

(5) **Approximate Performance:**
A sample problem involving 1 source region, 9 mesh points and 1 energy group required 65 minutes on the 799.

(6) **Reference:**

(7) **Material Available:**
1. NAA-SR-3719 (A listing of the FORTRAN source program is in this document).
2. FORTRAN source deck.

**GRACE-II**

7090 Nuclear Code

(1) **Code Originated by:**
Atomics International

(2) **Computer:**
7090 (FORTRAN)

(3) **Description of Code:**
GRACE-II is a multigroup, multiregion, gamma-ray attenuation code which computes the total dose rate or heat generation rate from a cylindrical or a spherical source. The source, which may be located in either the central region of the system or at a concentric shell region surrounding it, may be uniform, exponential, or have a polynomial variation in the radial direction. In the case of cylindrical geometry, it may also have a polynomial variation in the axial direction.

(4) **Restrictions or Limitations:**
If used on the 799, at least 16K memory is required. As many as 22 regions, 50 mesh points per region, 20 gamma-ray energy groups, 20 shielcl materials, and 20 material buildup factors may be included in a single calculation.

(5) **Approximate Performance:**
A sample problem required 3,64 minutes on the 799.

(6) **Reference:**

(7) **Material Available:**
1. NAA-SR-MEMO 4649.
2. FORTRAN source deck.

**PDQ-2**

7090 Nuclear Code

(1) **Code Originated by:**
Internationa] Rollins-NucIonic Cooperation

(2) **Computer:**
7090 (FORTRAN)

(3) **Description of Code:**
PDQ-2 is a program which eliminates need for use of compatibility package. Handles up to 5000-500 mesh points.

**PERT**

7090 Nuclear Code

(1) **Code Originated by:**
Atomics International

(2) **Computer:**
7090 (FORTRAN)

(3) **Description of Code:**
The PERT code is a perturbation theory code designed for use with the AIM-5, AIM-6, and FDO codes. Punched card output from these codes is used as input to the PERT code. Using cross section data, fluxes, and adjusted fluxes, the relation change in k_eff may be calculated. Cross sections may be weighted with the adjusted flux and/or flux. The neutron lifetime for the delay groups may also be evaluated.

(4) **Restrictions or Limitations:**
A linear perturbation theory is used for the calculations of the relative change in k_eff. (Continued on next column)

**SAIL**

7090 Nuclear Code

(1) **Code Originated by:**
Westinghouse-Beitl Plant

(2) **Computer:**
NORC

(3) **Description of Code:**
Elastic scattering transfer cross-sections are calculated using mass m., lethargy spectrum, and Legendre expansion coefficients for differential elastic scattering cross-sections. The computed cross-sections for a given element are placed on a library tape upon which as many as 30 elements may be accumulated.

(4) **Restrictions or Limitations:**
A maximum of 99 groups and 30 elements are allowed.

(5) **Approximate Performance:**
1 hour.

(6) **Reference:**
Summary, September, 1958.

(7) **Material Available:**
1. NAA Program Description.
2. FORTRAN source deck.

**SIZZLE**

7090 Nuclear Code

(1) **Code Originated by:**
Atomics International

(2) **Computer:**
7090 (FORTRAN)

(3) **Description of Code:**
The PERT code is a perturbation theory code designed for use with the AIM-5, AIM-6, and FDO codes. Punched card output from these codes is used as input to the PERT code. Using cross section data, fluxes, and adjusted fluxes, the relation change in k_eff may be calculated. Cross sections may be weighted with the adjusted flux and/or flux. The neutron lifetime for the delay groups may also be evaluated.

(4) **Restrictions or Limitations:**
A linear perturbation theory is used for the calculations of the relative change in k_eff. (Continued on next column)
TEMPEST II

(1) Code Originated by:
Atomic Energy of Canada Limited

(2) Computer:
312 (FORTRAN)

(3) Description of Code:
This code solves the one-dimensional monoenergetic Boltzmann
equation in cylindrical geometry, using the S4 approximation. In
dependence of the flux distribution, self-consistent parameters are
computed. No input data to the flux may be used if a diffusion
they may be used to provide an initial guess. In
addition, when running multiple cases, the converged flux from
the previous case may be used.

(4) Restrictions or Limitations:
The present restrictions are

(5) Approximate Performance:
About 15 seconds for a 50 mesh point problem.

(6) References:
1. J. A. Temple, "S4 CYLINDRICAL GEOMETRY CELL CODE",
AEC-842, 1961.

7090 Nuclear Code

TEMPEST

(1) Code Originated by:
Atomic Energy of Canada Limited

(2) Computer:
7090 (FORTRAN)

(3) Description of Code:
(Indicated status, if known)
Thermal cross-section, Wigner-Wilkins or Wigner equations. In
use, available.

(4) References:
"FORTRAN Nuclear Code"

7090 Nuclear Code

TEMPEST II

(1) Code Originated by:
Atomic Energy of Canada Limited

(2) Computer:
7090 (FORTRAN)

(3) Description of Code:
TEMPEST II is a neutron thermalization code based upon the
Wigner-Wilkins approximation for light moderators and the Wilkins
approximation for heavy moderators. A Maxwellian distribution
may also be used. The model used may be selected as a function
of energy. The second-order differential equations are integrated
directly rather than transforming to the Ricianctt equation. The code
provides microscopic and macroscopic cross-sections averaged
over the thermal neutron spectrum.

(6) Restrictions or Limitations:
A 512 memory is required.

(5) Approximate Performance:
About 15-25 seconds.

(6) References:
1. R. R. Budde, "TEMPEST II", NAA Program Description.

7090 Nuclear Code

7090 Nuclear Code

(1) Code Originated by:
Aeroflot-General Aviation

(2) Computer:
7090 (FORTRAN)

(3) Description of Code:
S4 CYLINDRICAL GEOMETRY CELL CODE

(1) Code Originated by:
Atomic Energy of Canada Limited

(2) Computer:
7090 (FORTRAN)

(3) Description of Code:
This code solves the one-dimensional monoenergetic Boltzmann
equation in cylindrical geometry, using the S4 approximation. In
addition, when running multiple cases, the converged flux from
the previous case may be used.

(4) Restrictions or Limitations:
The present restrictions are

(5) Approximate Performance:
About 15 seconds for a 50 mesh point problem.

(6) References:
1. J. A. Temple, "S4 CYLINDRICAL GEOMETRY CELL CODE",
AEC-842, 1961.

7090 Nuclear Code

TEMPEST

(1) Code Originated by:
Atomic Energy of Canada Limited

(2) Computer:
7090 (FORTRAN)

(3) Description of Code:
(Indicated status, if known)
Thermal cross-section, Wigner-Wilkins or Wigner equations. In
use, available.

(4) References:
"FORTRAN Nuclear Code"

7090 Nuclear Code

TEMPEST II

(1) Code Originated by:
Atomic Energy of Canada Limited

(2) Computer:
7090 (FORTRAN)

(3) Description of Code:
TEMPEST II is a neutron thermalization code based upon the
Wigner-Wilkins approximation for light moderators and the Wilkins
approximation for heavy moderators. A Maxwellian distribution
may also be used. The model used may be selected as a function
of energy. The second-order differential equations are integrated
directly rather than transforming to the Ricianctt equation. The code
provides microscopic and macroscopic cross-sections averaged
over the thermal neutron spectrum.

(6) Restrictions or Limitations:
A 512 memory is required.

(5) Approximate Performance:
About 15-25 seconds.

(6) References:
1. R. R. Budde, "TEMPEST II", NAA Program Description.

7090 Nuclear Code

7090 Nuclear Code

(1) Code Originated by:
Aeroflot-General Aviation

(2) Computer:
7090 (FORTRAN)

(3) Description of Code:
S4 CYLINDRICAL GEOMETRY CELL CODE

(1) Code Originated by:
Atomic Energy of Canada Limited

(2) Computer:
7090 (FORTRAN)

(3) Description of Code:
This code solves the one-dimensional monoenergetic Boltzmann
equation in cylindrical geometry, using the S4 approximation. In
addition, when running multiple cases, the converged flux from
the previous case may be used.

(4) Restrictions or Limitations:
The present restrictions are

(5) Approximate Performance:
About 15 seconds for a 50 mesh point problem.

(6) References:
1. J. A. Temple, "S4 CYLINDRICAL GEOMETRY CELL CODE",
AEC-842, 1961.

7090 Nuclear Code

TEMPEST

(1) Code Originated by:
Atomic Energy of Canada Limited

(2) Computer:
7090 (FORTRAN)

(3) Description of Code:
(Indicated status, if known)
Thermal cross-section, Wigner-Wilkins or Wigner equations. In
use, available.

(4) References:
"FORTRAN Nuclear Code"

7090 Nuclear Code

TEMPEST II

(1) Code Originated by:
Atomic Energy of Canada Limited

(2) Computer:
7090 (FORTRAN)

(3) Description of Code:
TEMPEST II is a neutron thermalization code based upon the
Wigner-Wilkins approximation for light moderators and the Wilkins
approximation for heavy moderators. A Maxwellian distribution
may also be used. The model used may be selected as a function
of energy. The second-order differential equations are integrated
directly rather than transforming to the Ricianctt equation. The code
provides microscopic and macroscopic cross-sections averaged
over the thermal neutron spectrum.

(6) Restrictions or Limitations:
A 512 memory is required.

(5) Approximate Performance:
About 15-25 seconds.

(6) References:
1. R. R. Budde, "TEMPEST II", NAA Program Description.

7090 Nuclear Code

7090 Nuclear Code

(1) Code Originated by:
Aeroflot-General Aviation

(2) Computer:
7090 (FORTRAN)

(3) Description of Code:
S4 CYLINDRICAL GEOMETRY CELL CODE

(1) Code Originated by:
Atomic Energy of Canada Limited

(2) Computer:
7090 (FORTRAN)

(3) Description of Code:
This code solves the one-dimensional monoenergetic Boltzmann
equation in cylindrical geometry, using the S4 approximation. In
addition, when running multiple cases, the converged flux from
the previous case may be used.

(4) Restrictions or Limitations:
The present restrictions are

(5) Approximate Performance:
About 15 seconds for a 50 mesh point problem.

(6) References:
1. J. A. Temple, "S4 CYLINDRICAL GEOMETRY CELL CODE",
AEC-842, 1961.